

Stellaris® LM3S9B92 RevC3/C5 Errata

This document contains known errata at the time of publication for the Stellaris LM3S9B92 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
March 2011	1.7	<ul style="list-style-type: none"> Changed title of issue "GPTM 2A and 2B are not usable with μDMA" to "The μDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B" on page 13 and reworded description. Deleted issue "PWM6 and PWM7 do not function" because it was erroneously included. Removed Appendix A since information is now in the data sheet.
February 2011	1.6	<ul style="list-style-type: none"> Clarified issue "The PIOSC cannot be calibrated by the user" on page 3. Added issue "The MINOR field in Device Identification 0 (DID0) register is incorrect" on page 4. Added issue "Debug interface is reset by any type of reset" on page 4. Added issue "The BOOTCFG register cannot be reliably written until after a special mass erase" on page 9. Added issue "Flash memory endurance cycle specification is 100 cycles" on page 13. Added issue "The μDMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B" on page 13. Added issue "PB0 and PB1 have permanent internal pull-down resistance" on page 14. Added issue "Differential pair encodings are incorrect" on page 21. Added issue "USB paired JK jitter compliance test requires automatic waiver" on page 23. Added issue "USB low speed crossover voltage compliance test requires automatic waiver" on page 24. Added issue "PWM fault latch does not operate correctly" on page 24. Added issue "PWM6 and PWM7 do not function." Added Appendix A.
January 2011	1.5	<ul style="list-style-type: none"> Added issue "The PIOSC cannot be calibrated by the user" on page 3. Added issue "Brown-out interrupt is never triggered" on page 4. Added diagram to issue "Flash corruption or device failure may occur at power on" on page 9 for industrial temperature circuits. Combined "ROM_USBHostMode function is incorrect" and "ROM_CANBitRateSet function is incorrect" into a single item, renamed it to "Some ROM functions are incorrect," and added an additional function. Added issue "PB0 and PB1 have permanent internal pull-down resistance" on page 14.

Date	Revision	Description
December 2010	1.4	<ul style="list-style-type: none"> Updated diagram for issue "Flash corruption or device failure may occur at power on" on page 9 and clarified the power-on constraints.
November 2010	1.3	<ul style="list-style-type: none"> Updated diagram for issue "Flash corruption or device failure may occur at power on" on page 9 and added clarification.
November 2010	1.2	<ul style="list-style-type: none"> Updated diagram for issue "Flash corruption or device failure may occur at power on" on page 9.
October 2010	1.1	<ul style="list-style-type: none"> Added issue "ROM_USBHostMode function is incorrect". Added issue "ROM_CANBitRateSet function is incorrect". Added additional information about the effect of "Deep-Sleep mode must not be used" on page 8 issue on USB operation. Added issue "Flash corruption or device failure may occur at power on" on page 9. Added issue "USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable" on page 22. Added issue "USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail" on page 23.
September 2010	1.0	Started tracking revision history.

Table 2. List of Errata

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
1.1	The PIOSC cannot be calibrated by the user	System Control	B1, C3
1.2	Brown-out interrupt is never triggered	System Control	C3
1.3	The MINOR field in Device Identification 0 (DID0) register is incorrect	System Control	C5
1.4	Debug interface is reset by any type of reset	System Control	C3, C5
2.1	Some ROM functions are incorrect	ROM	C3
3.1	Deep-Sleep mode must not be used	Flash Memory	C3, C5
3.2	Mass erase must not be used if Flash protection bits are used	Flash Memory	C3, C5
3.3	Page erase or program must not be performed on a protected Flash page	Flash Memory	C3, C5
3.4	The BOOTCFG register cannot be reliably written until after a special mass erase	Flash Memory	C3
3.5	Flash corruption or device failure may occur at power on	Flash Memory	C3
3.6	Flash memory endurance cycle specification is 100 cycles	Flash Memory	C3, C5
4.1	The μ DMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules	μ DMA	C3, C5
4.2	The μ DMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B	μ DMA	C3, C5
5.1	PB0 and PB1 have permanent internal pull-down resistance	GPIO	C3, C5
6.1	The General-Purpose Timer match register does not function correctly in 32-bit mode	General-Purpose Timers	C3, C5

Erratum Number	Erratum Title	Module Affected	Revision(s) Affected
6.2	A spurious DMA request is generated when the timer rolls over in Input-Edge Time mode	General-Purpose Timers	C3, C5
6.3	A spurious DMA request is generated when the timer rolls over the 16-bit boundary	General-Purpose Timers	C3, C5
6.4	The value of the prescaler register is not readable in Edge-Count mode	General-Purpose Timers	C3, C5
6.5	ADC trigger and Wait-on-Trigger may assert when the timer is disabled	General-Purpose Timers	C3, C5
6.6	Wait-on-Trigger does not assert unless the TnOTE bit is set	General-Purpose Timers	C3, C5
6.7	Do not enable match and timeout interrupts in 16-bit PWM mode	General-Purpose Timers	C3, C5
6.8	Do not use μ DMA with 16-bit PWM mode	General-Purpose Timers	C3, C5
6.9	Writing the GPTMTnV register does not change the timer value when counting up	General-Purpose Timers	C3, C5
6.10	The prescaler does not work correctly when counting up in periodic or one-shot mode	General-Purpose Timers	C3, C5
6.11	Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode	General-Purpose Timers	C3, C5
7.1	Writes to Watchdog Timer 1 module WDTLOAD register sometimes fail	Watchdog Timers	C3, C5
8.1	ADC hardware averaging produces erroneous results in differential mode	ADC	C3, C5
8.2	Differential pair encodings are incorrect	ADC	C3, C5
9.1	The RTRIS bit in the UARTRIS register is only set when the interrupt is enabled	UART	C3, C5
10.1	Encoding error in the Ethernet MAC LED Encoding (MACLED) register	Ethernet Controller	C3, C5
11.1	USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable	USB	C3
11.2	USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail	USB	C3
11.3	USB paired JK jitter compliance test requires automatic waiver	USB	C5
11.4	USB low speed crossover voltage compliance test requires automatic waiver	USB	C5
12.1	PWM fault latch does not operate correctly	PWM	C3, C5

1 System Control

1.1 The PIOSC cannot be calibrated by the user

Description:

The PIOSC is trimmed by the factory, but cannot be user calibrated using the UPDATE bit in the **Precision Internal Oscillator Calibration (PIOSCCAL)** register.

Workaround:

None.

Silicon Revision Affected:

B1, C3

Fixed:

Fixed on devices with date codes of 0x13 (March, 2011) or later.

1.2 Brown-out interrupt is never triggered

Description:

The brown out circuitry always resets the microcontroller when V_{DD} drops to the brown-out threshold voltage (V_{BTH}), regardless of the state of the `BORIOR` bit in the **PBORCTL** register.

Workaround:

None.

Silicon Revision Affected:

C3

Fixed:

Fixed on parts with a date code of 0x0C (December, 2010) or later.

1.3 The MINOR field in Device Identification 0 (DID0) register is incorrect

Description:

The `MINOR` field, bits[7:0], in the **Device Identification 0 (DID0)** register is incorrect. The `MINOR` field should be 0x05 indicating the fifth metal layer change. Instead, the field reads as 0x04.

Workaround:

None.

Silicon Revision Affected:

C5

Fixed:

Not yet fixed.

1.4 Debug interface is reset by any type of reset

Description:

The Serial Wire JTAG Debug Port (SWJ-DP) is reset by any reset condition. These reset types include:

- Watchdog reset
- Brown-out reset

- Software reset
- Reset pin assertion
- Main oscillator fail

Normal operation of the device is not affected by the reset of the SWJ-DP, however, users should bear this functionality in mind during development and debugging of applications. If a debugger does a `SYSRESREQ`, or if the debugger is being used in a session and a system reset occurs due to one of the reset sources above, then the debugger loses its state, including breakpoints, watchpoints, vector catch, and trace. Most debuggers attempt a recovery, usually after reporting the error to the user. If the debugger is able to recover control, the state of the application at that time reflects that the code has been running from reset and has not stopped on any breakpoints. If the application has breakpoint instructions physically in the code, such as for system calls that run through the debugger, then the code will have entered the fault handlers.

Another consequence of this erratum is that the USB may fail to enumerate when using a debugger. When debugging a board with USB and a crystal with a frequency greater than 8 MHz, the debugger writes the **RCC** register with the `XTAL` value for 8 MHz. As a result, the USB PLL cannot lock onto the required frequency and requires a hard reset.

Workaround:

Because some ARM debuggers expect to maintain connectivity when a system reset is requested, the `SYSRESREQ` bit in the **Application Interrupt and Reset Control (APINT)** register should not be used when using these debuggers; instead the `VECTRESET` bit, which only resets the core, should be used so that debug connectivity is uninterrupted. `VECTRESET` does not reset on-chip peripherals, which must be reset with specific reset operations.

When debugging code that requires a software reset, the `SYSRESREQ` software reset mechanism in the NVIC (which is used by the Stellaris Peripheral Driver Library `SysCtlReset()` and `ROM_SysCtlReset()` APIs) should not be used; instead, use the sequence of register writes with a `VECTRESET` in the NVIC as shown in the code below.

In addition, the ROM is mapped into address 0x0 during reset. The ROM code determines if boot loading is needed, and if not, transfers control to the normal application in Flash memory. As a result, the ROM is visible to the debugger on the reset entry. Debugging can be affected during Flash memory verification because the debugger compares the expected image with the ROM contents and not the Flash memory as intended. The disassembly shown to the user is also affected. To avoid these issues, debuggers must switch off the ROM mapping. However, if the debugger in use does not switch off the ROM, the user can either step through the first assembly instructions until the ROM gets remapped or write a 1 to the `BA` bit in the **ROM Control (ROMCTL)** register at location 0x400F.E0F0 using the debugger GUI, debugger command line, or debugger startup script.

Use of any reset source listed above other than software reset causes the debugger to lose connectivity.

In addition, when debugging with USB and a crystal greater than 8 MHz, always disable the USB PLL before writing the **RCC** `XTAL` value to ensure that the USB PLL starts up correctly.

```
//
// Disable processor interrupts.
//
IntMasterDisable();

//
// Disable the PLL and the system clock divider (this is a NOP if they are
// already disabled).
```

```
//
HWREG(SYSCTL_RCC) = ((HWREG(SYSCTL_RCC) & ~(SYSCTL_RCC_USESYSDIV)) |
                     SYSCTL_RCC_BYPASS);
HWREG(SYSCTL_RCC2) |= SYSCTL_RCC2_BYPASS2;

//
// Now, write RCC and RCC2 to their reset values.
//
HWREG(SYSCTL_RCC) = 0x078e3ad0 | (HWREG(SYSCTL_RCC) & SYSCTL_RCC_MOSCDIS);
HWREG(SYSCTL_RCC2) = 0x07806810;
HWREG(SYSCTL_RCC) = 0x078e3ad1;

//
// Reset the deep sleep clock configuration register.
//
HWREG(SYSCTL_DSLPCLKCFG) = 0x07800000;

//
// Reset the clock gating registers.
//
HWREG(SYSCTL_RCGC0) = 0x00000040;
HWREG(SYSCTL_RCGC1) = 0;
HWREG(SYSCTL_RCGC2) = 0;
HWREG(SYSCTL_SCGC0) = 0x00000040;
HWREG(SYSCTL_SCGC1) = 0;
HWREG(SYSCTL_SCGC2) = 0;
HWREG(SYSCTL_DCGC0) = 0x00000040;
HWREG(SYSCTL_DCGC1) = 0;
HWREG(SYSCTL_DCGC2) = 0;

//
// Reset the remaining SysCtl registers.
//
HWREG(SYSCTL_PBORCTL) = 0;
HWREG(SYSCTL_IMC) = 0;
HWREG(SYSCTL_GPIOHBCTL) = 0;
HWREG(SYSCTL_MOSCCTL) = 0;
HWREG(SYSCTL_PIOSCCAL) = 0;
HWREG(SYSCTL_I2SMCLKCFG) = 0;

//
// Reset the peripherals.
//
HWREG(SYSCTL_SRCR0) = 0xffffffff;
HWREG(SYSCTL_SRCR1) = 0xffffffff;
HWREG(SYSCTL_SRCR2) = 0xffffffff;
HWREG(SYSCTL_SRCR0) = 0;
HWREG(SYSCTL_SRCR1) = 0;
HWREG(SYSCTL_SRCR2) = 0;

//
// Clear any pending SysCtl interrupts.
//
HWREG(SYSCTL_MISC) = 0xffffffff;
```

```

//
// Wait for any pending flash operations to complete.
//
while((HWREG(FLASH_FMC) & 0xffff) != 0)
{
}
while((HWREG(FLASH_FMC2) & 0xffff) != 0)
{
}

//
// Reset the flash controller registers.
//
HWREG(FLASH_FMA) = 0;
HWREG(FLASH_FCIM) = 0;
HWREG(FLASH_FCMISC) = 0xffffffff;
HWREG(FLASH_FWBVAL) = 0;

//
// Issue the core reset.
//
HWREG(NVIC_APINT) = NVIC_APINT_VECTKEY | NVIC_APINT_VECT_RESET;

```

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

2 ROM

2.1 Some ROM functions are incorrect

Description:

The following ROM functions do not work and should not be used.

- ROM_USBHostMode
- ROM_CANBitRateSet
- ROM_uDMAChannelTransferSet

Workaround:

Use the StellarisWare functions in Flash memory.

Silicon Revision Affected:

C3

Fixed:

Fixed in Rev C5.

3 Flash Memory

3.1 Deep-Sleep mode must not be used

Description:

Deep-sleep mode must not be used.

Due to this erratum, the use of this device in USB bus-powered applications is prohibited because sleep mode current consumption exceeds the USB specification.

Workaround:

Use Sleep or Hibernation mode.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

3.2 Mass erase must not be used if Flash protection bits are used

Description:

The mass erase function using the `MERASE` bit in the **Flash Memory Control (FMC)** register must not be used in systems that clear any of the **Flash Memory Protection Program Enable n (FMPPE_n)** bits. For Rev C1 devices, this means that mass erase must not be used because bits in the **FMPPE0** registers are cleared to protect the reset patch that is stored in the first block of Flash memory. For Rev C3 and C5 devices, mass erase can be used as long as none of the **FMPPE_n** bits are cleared.

Workaround:

Erase Flash memory with the page erase function using the `ERASE` bit in the **FMC** register instead of the mass erase function.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

3.3 Page erase or program must not be performed on a protected Flash page

Description:

The erase function using the `ERASE` bit in the **Flash Memory Control (FMC)** register and the program function using the `WRITE` bit in the **FMC** register or the `WRBUF` bit in the **FMC2** register must not be used in systems that clear the bit in **FMPPE_n** that corresponds to that page of Flash. For C1 devices, this means that erase and program of locations 0x0 through 0xFFF must not be used because bits in the **FMPPE0** registers are cleared to protect the reset patch that is stored in the first block of Flash memory. For Rev C3 and C5 devices, erase and program can be used as long as neither of the corresponding **FMPPE_n** bits are cleared.

Workaround:

Only erase and program memory that is not protected by the corresponding **FMPPEn** bits.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

3.4 The BOOTCFG register cannot be reliably written until after a special mass erase

Description:

The **Boot Configuration (BOOTCFG)** register cannot be reliably written until a special type of mass erase is executed.

Workaround:

Execute the special mass erase sequence described in "Recovering a 'Locked' Microcontroller" section of the JTAG chapter prior to a write access to the **BOOTCFG** register. The debug port unlock sequence in LMFlash Programmer can be used to perform this special type of mass erase.

Silicon Revision Affected:

C3

Fixed:

Fixed on devices with date codes of 0x0B (November, 2010) or later.

3.5 Flash corruption or device failure may occur at power on

Description:

There is a small risk of flash corruption or device failure on power up. The issue can occur with certain V_{DD} and V_{DDC} power sequences. The failure is not in the flash memory itself but in the control logic to the flash.

Workaround:

To eliminate the risk of flash corruption, two power-on requirements must be met:

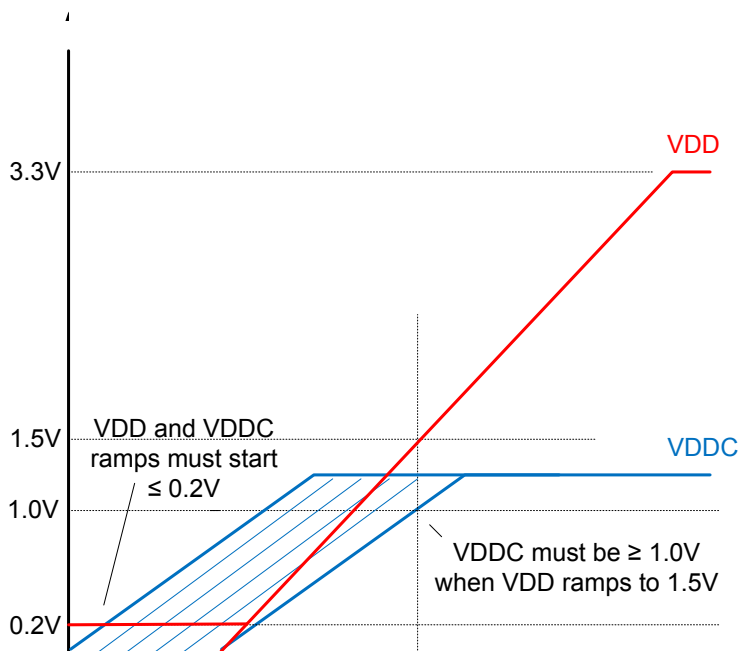
- The ramp of both V_{DD} and V_{DDC} must begin below 0.2 V.
- V_{DDC} must reach at least 1.0 V before V_{DD} rises above 1.5 V.

Figure 1 on page 10 details these requirements. Three workaround circuits have been identified that meet these requirements and are described below.

Normally V_{DDC} is supplied by the device's internal voltage regulator from the LDO output pin, however in some circuits the internal regulator may not meet this V_{DDC} timing requirement. A circuit combining an external 1.2 V regulator, a voltage supervisor and a power switch can be used to ensure that this timing requirement are met. The 1.2 V regulator has an integrated Power-OK (POK) circuit that is used to enable V_{DD} when V_{DDC} reaches 1.08 V. During power-down or transient conditions, the

POK circuit disables the load switch if V_{DDC} drops below 1.02 V or V_{DD} drops below 1.5 V. The load switch has an internal clamp to accelerate V_{DD} decay.

Figure 1. V_{DDC} and V_{DD} Rise Time Relationships



When implementing this workaround, it is important to consider all possible power conditions for the system, including:

- Brown-out (momentary sags in the power source)
- Switch and contact bounce
- Other EMI susceptibility tests
- Various battery and power source disturbances

Three recommended circuits that eliminate the occurrence of this issue are shown below. Although the LDO regulator output is unused in the workaround circuit, a capacitor (1-3 μF) must remain connected for regulator stability. In addition, the LDO pin of the Stellaris device must be disconnected from the external 1.2 V LDO to prevent electronic over stress of the pin. All of these circuits include two jumpers which provide the option to bypass the workaround circuit for future silicon revisions.

Figure 2 shows a small chip-scale load switch to control V_{DD} . This circuit is suitable for V_{DD} current up to 2 A peak.

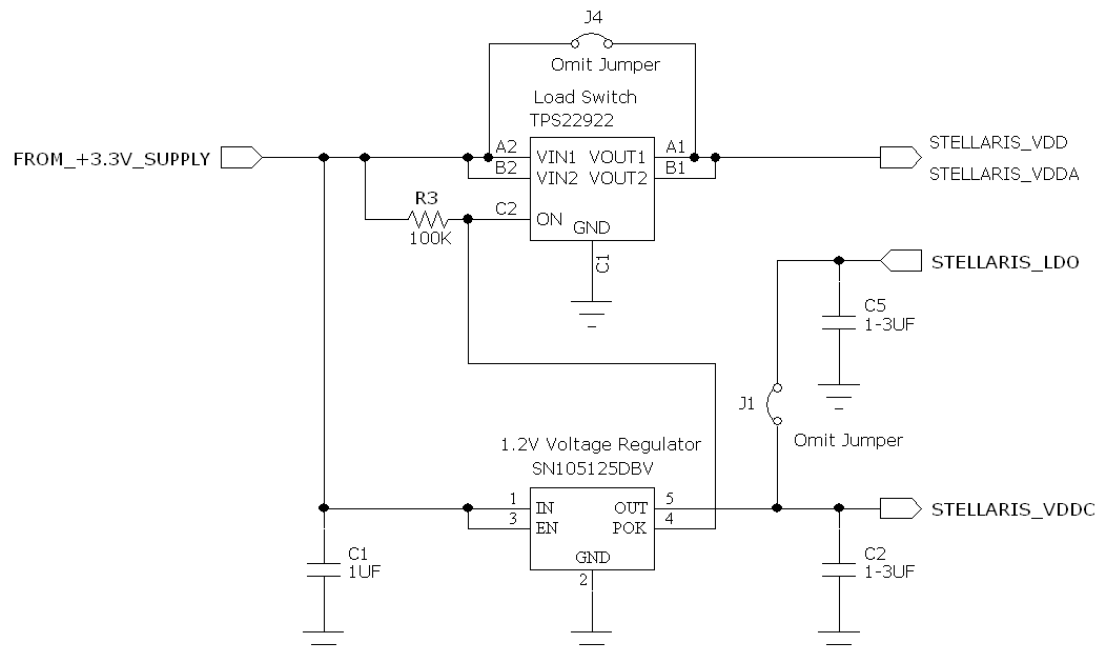
Figure 2. Recommended Voltage Supply Circuit 1

Figure 3 on page 11 shows a larger SOT-packaged load switch with V_{DD} current capabilities up to 400 mA peak (both channels in parallel).

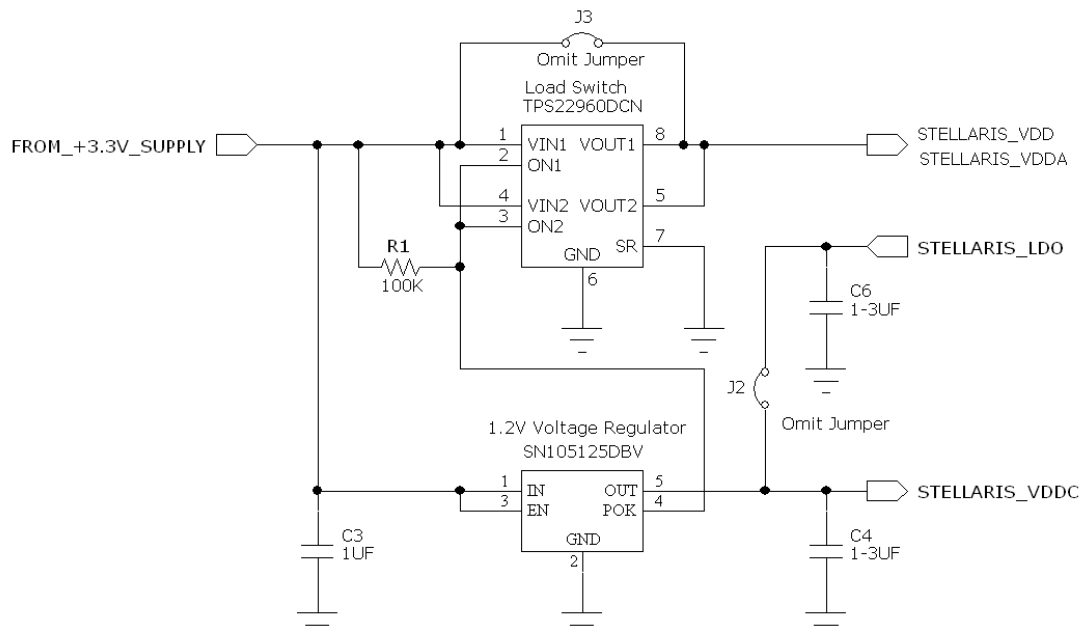
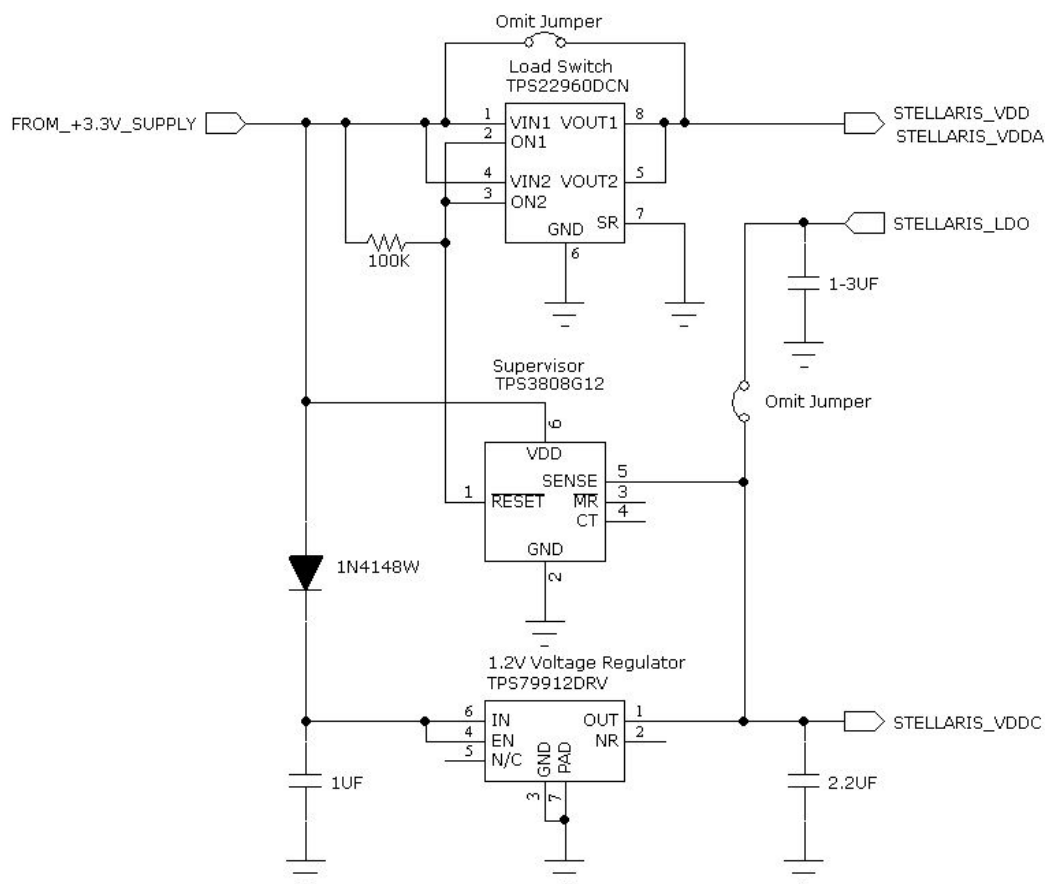
Figure 3. Recommended Voltage Supply Circuit 2

Figure 4 on page 12 is recommended for designs that require an industrial-temperature operating range. The TPS3808G12 is a dedicated 1.2 V voltage supervisor that ensures V_{DD} is only applied once V_{DDC} is valid. An important consideration is the power dissipation in the 1.2 V LDO voltage regulator. The regulator should remain within its thermal limits while accommodating a worst-case V_{DDC} current of 125 mA. The TPS79912DRV regulator has a θ_{ja} of 74.2 °C/W. With a 2.4 V (maximum) voltage drop, the power dissipation is 300 mW. The junction temperature will be approximately 108°C (23°C + 85°C) at 85°C ambient temperature which is well within the rating of the part. For more information about thermal metrics, see the IC Package Thermal Metrics application report, SPRA953.

Figure 4. Recommended Voltage Supply Circuit 3



Silicon Revision Affected:

C3

Fixed:

Fixed on Rev C5.

3.6 Flash memory endurance cycle specification is 100 cycles

Description:

The Flash memory endurance cycle specification (maximum program/erase cycles) is 100 cycles. Failure to adhere to the maximum number of program/erase cycles could result in corruption of the Flash memory contents and/or permanent damage to the device.

Workaround:

None. Because the failure mechanism is a function of the third-party Flash memory technology used in this device, there is no workaround. This third-party Flash memory technology is used only in the affected 130-nm Stellaris products and will not be used in any future devices. All other Stellaris products use Flash memory technology that exceeds industry quality and endurance cycle standards.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

4 μ DMA

4.1 The μ DMA controller fails to generate capture mode DMA requests from Timer A in the Timer modules

Description:

The μ DMA 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:

C3, C5

Fixed:

Not yet fixed.

4.2 The μ DMA does not generate a completion interrupt when transferring to and from GPTM 2A and 2B

Description:

The μ DMA module does not generate a completion interrupt on the Timer 2 interrupt vector when transferring data to and from Timers 2A and 2B. The μ DMA can successfully transfer data to and from Timers 2A and 2B; however, there is no interrupt to indicate that the transfer is complete.

Workaround:

If a completion interrupt is required, use an alternate GPTM.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

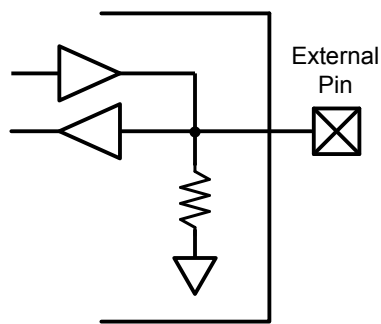
5 GPIO

5.1 PB0 and PB1 have permanent internal pull-down resistance

Description:

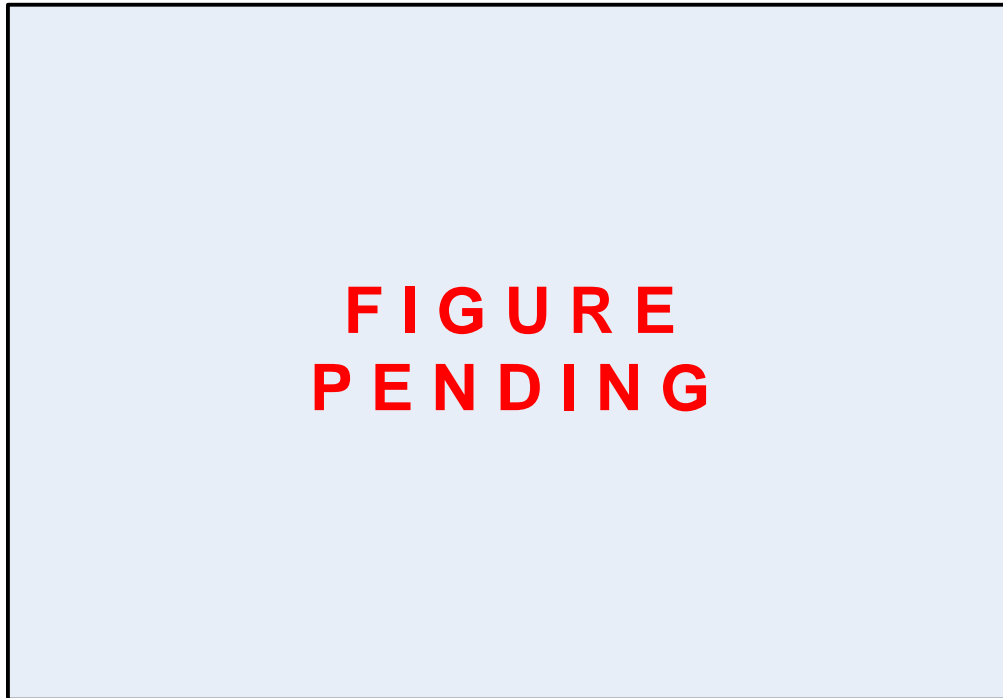
Regardless of their configuration, PB0 and PB1 have an internal pull-down resistance. The internal structure of these pins is shown in Figure 5 on page 14.

Figure 5. Internal Structure of PB0 and PB1



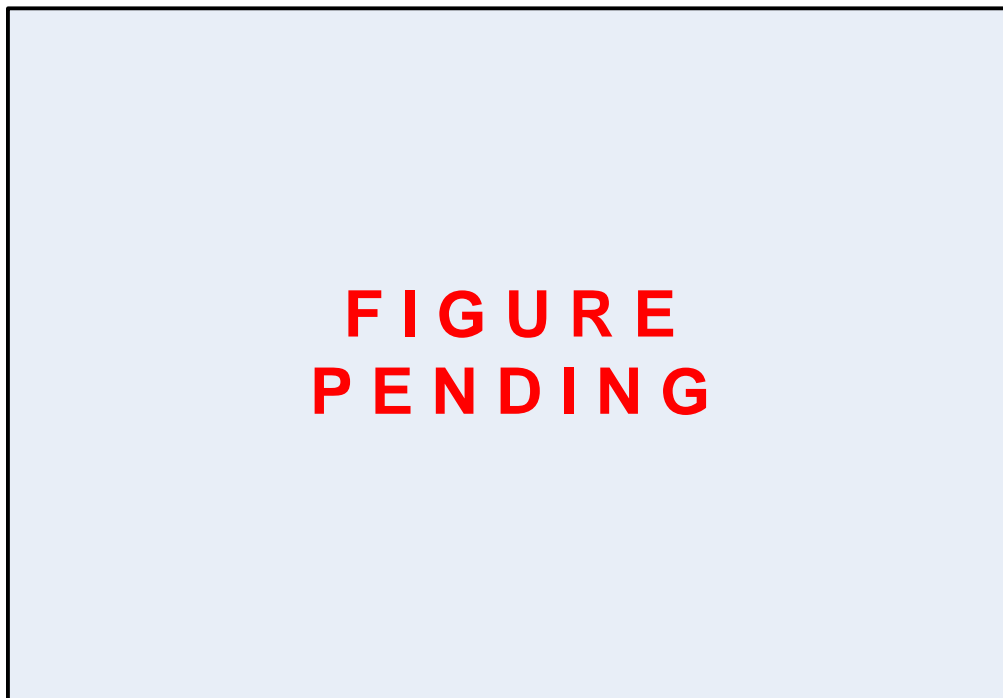
The characteristic of the pull-down for PB0 is shown in Figure 6 on page 15. The data labels for each data point show the external pull-up resistance in Ohms.

Figure 6. Voltage vs. Current for Various External Pull-up Resistors on PB0



The characteristic of the pull-down for PB1 is shown in Figure 7 on page 15. The data labels for each data point show the equivalent resistance in Ohms.

Figure 7. 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. Analysis is ongoing to determine what value external resistors should have.

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:

C3, C5

Fixed:

Not yet fixed.

6 General-Purpose Timers

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 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:

C3, C5

Fixed:

Not yet fixed.

6.2 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 capture the edge time into a buffer via DMA, then the spurious interrupt can be detected by noting that the captured value is the same as the previous capture value.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.3 A spurious DMA request is generated when the timer rolls over the 16-bit boundary

Description:

When the timer is in 32-bit periodic or one-shot mode and is enabled to generate periodic DMA requests, a spurious DMA request is generated when the timer rolls past 0x0000FFFF.

Workaround:

Only use DMA with a 16-bit periodic timer.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.4 The value of the prescaler register is not readable in Edge-Count mode

Description:

In Edge-Count mode, the prescaler is used as an 8-bit high order extension to the 16-bit counter. When reading the **GPTM Timer n (GPTMTnR)** register as a 32-bit value, the bits [23:16] always contain the initial value of the **GPTM Timer n Prescale (GPTMTnPR)** register, that is, the "load" value of the 8-bit extension.

Workaround:

None.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.5 ADC trigger and Wait-on-Trigger may assert when the timer is disabled

Description:

If the value in the **GPTM Timer n Match (GPTMTnMATCHR)** register is equal to the value of the timer counter and the **TnOTE** bit in the **GPTM Control (GPTMCTL)** register is set, enabling the ADC trigger, the trigger fires even when the timer is disabled (the **TnEN** bit in the **GPTMCTL** register is clear). Similarly, if the value in the **GPTMTnMATCHR** register is equal to the value of the timer counter and the **TnWOT** bit in the **GPTM Timer n Mode (GPTMTnMR)** register is set, enabling the Wait-on-Trigger mode, the trigger fires even when the timer is disabled.

Workaround:

Enable the timer before setting the `TnOTE` bit. Also, for the Wait-on-Trigger mode, ensure that the timers are configured in the order in which they will be triggered.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.6 Wait-on-Trigger does not assert unless the `TnOTE` bit is set

Description:

Wait-on-Trigger does not assert unless the `TnOTE` bit is set in the **GPTMCTL** register.

Workaround:

If the `TnWOT` bit in the **GPTM Timer n Mode (GPTMTnMR)** register is set, enabling the Wait-on-Trigger mode, the `TnOTE` bit must also be set in the **GPTMCTL** register in order for the Wait-on-Trigger to fire. Note that when the `TnOTE` bit is set, the ADC trigger is also enabled.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.7 Do not enable match and timeout interrupts in 16-bit PWM mode

Description:

16-bit PWM mode generates match and timeout interrupts in the same manner as periodic mode.

Workaround:

Ensure that any unwanted interrupts are masked in the **GPTMTnMR** and **GPTMIMR** registers.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.8 Do not use μ DMA with 16-bit PWM mode

Description:

16-bit PWM mode generates match and timeout μ DMA triggers in the same manner as periodic mode.

Workaround:

Do not use μ DMA to transfer data when the timer is in 16-bit PWM mode.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.9 Writing the GPTMTnV register does not change the timer value when counting up

Description:

When counting up, writes to the **GPTM Timer n Value (GPTMTnV)** register do not change the timer value.

Workaround:

None.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.10 The prescaler does not work correctly when counting up in periodic or one-shot mode

Description:

When counting up, the prescaler does not work correctly in 16-bit periodic or snap-shot mode.

Workaround:

Do not use the prescaler when counting up in 16-bit periodic or snap-shot mode.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

6.11 Snapshot must be enabled in both Timer A and B when in 32-bit snapshot mode

Description:

When a periodic snapshot occurs in 32-bit periodic mode, only the lower 16-bit are stored into the **GPTM Timer A (GPTMTAR)** register.

Workaround:

If both the **TASNAPS** and **TBSNAPS** bits are set in the **GPTM Timer A Mode (GPTMTAMR)** register, the entire 32-bit snapshot value is stored in the **GPTMTAR** register.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

7 Watchdog Timers

7.1 Writes to Watchdog Timer 1 module WDTLOAD register sometimes fail

Description:

Due to the independent clock domain of the Watchdog Timer 1 module, writes to the **Watchdog Load (WDTLOAD)** register may sometimes fail, even though the **WRC** bit in the **WDTCTL1** register is set after the write occurs.

Workaround:

After performing a write to the **WDTLOAD** register, read the contents back and verify that they are correct. If they are incorrect, perform the write operation again.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

8 ADC

8.1 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:

C3, C5

Fixed:

Not yet fixed.

8.2 Differential pair encodings are incorrect

Description:

When using differential mode, the MUX_n fields in the **ADCSSMUXn** registers should be configured to be "i" where the paired inputs are "2i" and "2i + 1". This encoding does not work for AIN8 - AIN15.

Workaround:

Use the encodings shown in the following table:

Adjacent Channels	i	MUXn Encoding
AIN0 and AIN1	0	0x0
AIN2 and AIN3	1	0x1
AIN4 and AIN5	2	0x2
AIN6 and AIN7	3	0x3
AIN8 and AIN9	4	0x8
AIN10 and AIN11	5	0x9
AIN12 and AIN13	6	0xA
AIN14 and AIN15	7	0xB

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

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 **RTIM** enable 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 on 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:

C3, C5

Fixed:

Not yet fixed.

10 Ethernet Controller

10.1 Encoding error in the Ethernet MAC LED Encoding (MACLED) register

Description:

Configuring the LED0 or LED1 field of the **Ethernet MAC LED Encoding (MACLED)** register to 0x8 should cause the corresponding LED to report a combined link + activity status. However, it instead only reports activity status (i.e. exactly the same as encoding 0x1).

Workaround:

None.

Silicon Revision Affected:

C3, C5

Fixed:

Not yet fixed.

11 USB

11.1 USB compliance test issue: USB full-speed, far-end signal compliance tests fail with 5 m cable

Description:

While USB packet loss has not been observed, the device is unable to pass the following USB compliance tests:

- USB Host Test B.3.3.2 Full-speed Downstream Signal Quality Test
- USB Device Test B.6.3.1 Signal Integrity Test – Upstream Signal test (full speed)

Compliance testing is based on the “USB Implementers Forum Full and Low Speed Electrical and Interoperability Compliance Test Procedure” Revision 1.3 available from usb.org website. The compliance testing is performed using a 5 m USB certified cable between the host or device under test and the test SQiDD which is then connected to a USB compliant hub chain to the root hub. Under compliance test conditions, the rising edges of the USB D+/D- signals begin to violate the lower right corner of the full-speed eye diagram defined by the USB specification. USB certification cannot be obtained because of this erratum.

A full report on this issue, "USB Far End Signal Integrity Test Results," is available from your local TI FAE.

Workaround:

If a cable with a length of 1 m is used instead of a 5 m cable, the Eye diagram compliance tests all pass with adequate margin across the voltage and temperature range of the part. Under nominal voltage and temperature conditions, a cable of up to 3 m can be used and passes the eye diagram compliance tests.

Silicon Revision Affected:

C3

Fixed:

Fixed in Rev C5.

11.2 USB compliance test issue: USB embedded host low-speed, far-end signal compliance tests fail

Description:

While USB packet loss has not been observed, the device is unable to pass the following USB compliance test:

■ USB Host Test B.3.3.1 Low-Speed Downstream Signal Quality Test

USB Compliance testing is based on the "USB Implementers Forum Full and Low Speed Electrical and Interoperability Compliance Test Procedure" Revision 1.3 available from usb.org website. The rising and falling edges of the USB D+/D- signals violate the lower half of the low-speed eye diagram defined by the USB specification. This erratum applies only to systems defined as a USB embedded host that support low-speed devices. USB embedded host and OTG systems that support full-speed devices only are not affected by this erratum. USB device systems are full-speed only and thus are not affected by this erratum.

A full report on this issue, "USB Far End Signal Integrity Test Results," is available from your local TI FAE.

Workaround:

None.

Silicon Revision Affected:

C3

Fixed:

Fixed in Rev C5.

11.3 USB paired JK jitter compliance test requires automatic waiver

Description:

The USB compliance test results in a Paired JK jitter failure.

Workaround:

This failure comes with an automatic waiver from usb.org so certification can still be granted.

Silicon Revision Affected:

C5

Fixed:

Not yet fixed.

11.4 USB low speed crossover voltage compliance test requires automatic waiver

Description:

The USB compliance test results in a crossover voltage range failure when talking to low speed devices.

Workaround:

This failure comes with an automatic waiver from usb.org so certification can still be granted.

Silicon Revision Affected:

C5

Fixed:

Not yet fixed.

12 PWM

12.1 PWM fault latch does not operate correctly

Description:

If the `LATCH` bit is set in the `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:

C3, C5

Fixed:

Not yet fixed.

Copyright © 2008-2011 Texas Instruments Incorporated All rights reserved. Stellaris and StellarisWare are registered trademarks of Texas Instruments Incorporated. ARM and Thumb are registered trademarks and Cortex is a trademark of ARM Limited. Other names and brands may be claimed as the property of others.

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>

