参考資料

Errata

MSPM0L222x、*MSPM0L122x*、*MSPM0L222x-Q1*、*MSPM0L122x-Q1* マイコン

TEXAS INSTRUMENTS

概要

この文書では、機能仕様に対する既知の例外 (アドバイザリ) について説明します。

目次

商標

エラッタのリストおよび各項目を読む前に、以下の注意事項を確認してください:

- I2C_ERR_01 は、このデバイスに影響を与えません

1 機能アドバイザリ

デバイスの動作、機能、またはパラメータに影響するアドバイザリ。

✓ チェック マークは、指定されたリビジョンに問題が存在することを示します。

エラッタ番号	リビジョン A	Rev B
ADC_ERR_05	✓	✓
ADC_ERR_06	✓	
AES_ERR_01	✓	✓
COMP_ERR_03	✓	✓
COMP_ERR_04	✓	✓
CPU_ERR_01	✓	✓
CPU_ERR_02	✓	✓
CPU_ERR_03	✓	✓
CPU_ERR_04	✓	✓
CRC/CRCP_ERR_01	✓	✓
FLASH_ERR_04	✓	✓
FLASH_ERR_05	✓	✓
FLASH_ERR_08	✓	✓
FLASH_ERR_09	✓	✓
GPIO_ERR_03	✓	✓
GPIO_ERR_04	✓	✓
GPIO_ERR_06	✓	✓
I2C_ERR_01		
I2C_ERR_03	✓	✓
I2C_ERR_04	✓	✓
I2C_ERR_05	✓	✓
I2C_ERR_06	✓	✓
I2C_ERR_07	✓	✓
I2C_ERR_08	✓	✓
I2C_ERR_09	✓	✓
I2C_ERR_10	✓	✓
I2C_ERR_13	✓	✓
I2C_ERR_15	✓	✓
KEYSTORE_ERR_01	✓	✓
LCD_ERR_01	✓	
LFSS_ERR_01	✓	
LFSS_ERR_02	✓	
LFSS_ERR_03	✓	✓
LFXT_ERR_01	✓	✓
LFXT_ERR_02	✓	✓
PMCU_ERR_08	✓	✓
PMCU_ERR_09	✓	✓

エラッタ番号	リビジョン A	Rev B
PMCU_ERR_10	✓	✓
PMCU_ERR_12	✓	✓
RST_ERR_01	✓	✓
RST_ERR_02	✓	✓
RTC_A_ERR_02	✓	✓
SPI_ERR_03	✓	✓
SPI_ERR_04	✓	✓
SPI_ERR_05	✓	✓
SPI_ERR_06	✓	✓
SPI_ERR_07	✓	✓
SPI_ERR_10	✓	✓
SRAM_ERR_01	✓	✓
SYSCTL_ERR_01	✓	✓
SYSCTL_ERR_02	✓	✓
SYSCTL_ERR_03	✓	✓
SYSCTL_ERR_04	✓	✓
SYSCTL_ERR_05	✓	✓
SYSCTL_ERR_06	✓	✓
SYSOSC_ERR_01	✓	✓
SYSOSC_ERR_02	✓	✓
SYSOSC_ERR_04	✓	✓
SYSOSC_ERR_05	✓	✓
SYSOSC_ERR_06	✓	✓
TAMPERIO_ERR_01	✓	✓
TIMER_ERR_01	✓	✓
TIMER_ERR_04	✓	✓
TIMER_ERR_06	✓	✓
TIMER_ERR_07	✓	✓
UART_ERR_01	✓	✓
UART_ERR_02	✓	✓
UART_ERR_03	✓	✓
UART_ERR_04	✓	✓
UART_ERR_05	✓	✓
UART_ERR_06	✓	✓
UART_ERR_07	✓	✓
UART_ERR_08	✓	✓
UART_ERR_09	✓	✓
UART_ERR_10	✓	✓
UART_ERR_11	✓	✓

2 プログラム済みのソフトウェア アドバイザリ

工場出荷時にプログラムされたソフトウェアに影響を及ぼすアドバイザリ。

✓ チェック マークは、指定されたリビジョンに問題が存在することを示します。

3 デバッグ専用のアドバイザリ

デバッグ動作のみに影響するアドバイザリ。

✓ チェック マークは、指定されたリビジョンに問題が存在することを示します。

エラッタ番号	リビジョン A	Rev B
GPIO_ERR_03	✓	✓

4 コンパイラ アドバイザリによって修正

コンパイラの回避方法により解決されるアドバイザリ各アドバイザリについては、回避策が適用されている IDE およびコンパイラのバージョンを参照してください。

✓ チェック マークは、指定されたリビジョンに問題が存在することを示します。

5 デバイスの命名規則

製品開発サイクルの段階を示すため、TI はすべての MSP MCU デバイスの型番に接頭辞を割り当てています。MSP MCU 商用ファミリの各番号には、MSP、X のいずれかの接頭辞があります。MSP または XMS。これらの接頭辞は、製品開発の進展段階を表します。段階には、エンジニアリング プロトタイプ(XMS)から、完全認定済みの量産デバイス(MSP)までがあります。

XMS - 実験段階のデバイスであり、必ずしも最終製品の電気的特性を表しているとは限りません

MSP — 完全に認定済みの量産版デバイス

サポートツールの名前付けプレフィックス:

X:開発サポート製品。テキサス・インスツルメンツの社内認定試験はまだ完了していません。

null:完全に認定済みの開発サポート製品です。

XMS デバイスと MSPX 開発サポート ツールは、以下の免責事項に基づいて出荷されます:

「開発中の製品は、社内での評価用です。」

MSP デバイスの特性は完全に明確化されており、デバイスの品質と信頼性が十分に示されています。TI の標準保証が適用されます。

プロトタイプ デバイス (XMS) は、標準の量産デバイスよりも故障率が高いことが予想されます。これらのデバイスは、予測される最終使用時の故障率が未定義であるため、テキサス・インスツルメンツはそれらのデバイスを量産システムで使用しないよう推奨しています。認定済みの量産デバイスのみを使用する必要があります。

TI デバイスの項目表記には、デバイス ファミリ名の接尾辞も含まれます。この接尾辞は、温度範囲、パッケージ タイプ、配布形式を示しています。

5.1 デバイスの記号表記とリビジョンの識別

次のパッケージ図はパッケージ記号化スキームを示すとともにし表 5-1、デバイスリビジョンからバージョン ID へのマッピングを定義しています。

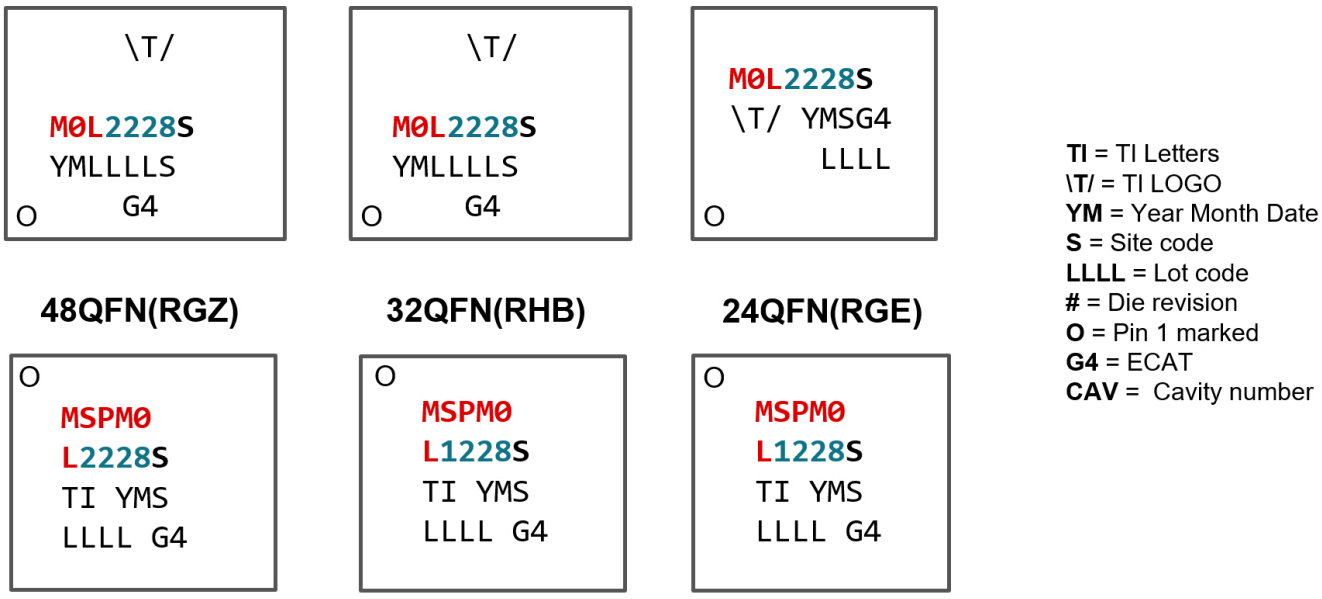

図 5-1. パッケージの記号表記

表 5-1. ダイ リビジョン

リビジョンレター	バージョン（デバイスの工場出荷時定数メモリ内）
A	0x0
B	0x1

リビジョン文字は、製品のハードウェアの改訂版を示します。このドキュメントのアドバイザリには、リビジョン文字に基づいて、特定のバイスに該当するか否かがマークされています。この文字は、デバイスのメモリに保存された整数にマップされ、アプリケーションソフトウェア または接続されたデバッグプローブによるリビジョンの検索に使用できます。

6 アドバイザリの説明

ADC_ERR_05

ADC モジュール

カテゴリ

機能

機能

IP（周辺モジュール）が有効化される前にハードウェアイベントが生成された場合、その ADC トリガはキューに保持されたままになります

説明

ADC が HW イベントトリガを受信すると、CTL0.ENC = 0x0 であっても、保留中のイベントは ADC トリガキューに入ります。ADC が CTL1.TRIGSRC = 0x1 (HW トリガ) かつ CTL0.ENC = 1 に設定されると、キューに設定された HW トリガは、ADC サンプリングおよび変換プロセスを開始します。保留中の HW 要求は、CTL1.TRIGSRC = 0x0 (SW トリガ) の場合でもキューに入ることができますが、CTL1.TRIGSRC = 0x1 かつ CTL0.ENC = 0x1 の場合のみ ADC サンプリングを開始します。

回避方法

HW トリガを使用することを予期した場合にのみ ADC F_SUB を構成し、そうでない場合、保留中の要求がキューに登録されます。SW トリガモードと HW トリガモードを切り替える場合、ADC (RSTCTL) をリセットすると保留中のキューがクリアされますが、ADC の再構成が必要となります。

ADC_ERR_06

ADC モジュール

カテゴリ

機能

機能

ADC 出力コードが DNL/INL の仕様の劣化を招きます

説明

変換エラーが発生すると、ADC の入力電圧に対応する変化がないにもかかわらず、デジタル出力コード上に ±64LSB の固定的なジャンプとしてそのエラーが現れます。

最悪条件下 (-40℃) においては、12 ビット モードで変換された 12M 回のサンプルにつき 1 回、エラーが発生します。
0℃時のエラーレートは 24M 回に 1 回、55℃時は 60M 回に 1 回です (使用された VDD 電圧と基準電圧はエラー レートに影響しない)。

回避方法

アプリケーションのニーズに応じて、最適な回避策は異なりますが、本ソフトウェアでは、以下の回避策を提案します。最適な回避方法の選択は、システム設計者の判断に委ねられます。

回避方法 1：ADC の結果がアプリケーションで設定されたスレッショルドを外れた場合 (ADC ウィンドウ コンパレータやソフトウェアによるスレッショルド判定を使用)、重要なシステム判断を行う前に、もう一度 ADC の結果を取得するか、次の変換結果を待つようにします

回避方法 2: 後処理時に、ADC 値の中央値または予測値からかけ離れた ADC 値を破棄します。期待値は、システム内で取得された実際のサンプルの平均に基づいて設定し、除外のためのスレッショルドは、測定されたシステム ノイズの大きさに基づいて決定する必要があります。

ADC_ERR_06 (続き) *ADC モジュール*

回避方法 3: 単一の誤変換結果の影響を最小限に抑えるために、ADC サンプルの平均化を使用します。

AES_ERR_01 *AES モジュール*

カテゴリ

機能

機能

AES Saved Context Ready 割り込みが予想どおりに生成されていません

説明

Saved Context Ready 割り込みが生成されていません。いずれかの AES レジスタに対してアクセス (読み取りまたは書き込み) が行われた場合に、割り込みが生成されます。

回避方法

ポーリングベースのメカニズムを使用して、割り込みをせず、CTRL レジスタの保存済みコンテキストレディのステータスビットを確認します。

COMP_ERR_03 *COMP モジュール*

カテゴリ

機能

機能

入力交換機能を使用する場合、COMP ヒステリシス機能は機能しません

説明

COMP モジュールのヒステリシス機能を使用している状態で、COMP の入力を入れ替える (COMPx.CTL1.EXCH = 1) と、COMP モジュールが不安定になる可能性があります。

回避方法

入力交換機能で COMP モジュールを利用する場合は、内部ヒステリシスの方法を適用しないでください。

COMP_ERR_04 *COMP モジュール*

カテゴリ

機能

機能

スタンバイ中に温度センサをコンパレータ入力チャネルとして接続すると、コンパレータ出力が連続的にトグルする

説明

スタンバイ モードで温度センサはオフです。スタンバイ モードで温度センサをコンパレータ入力チャネルとして接続すると、コンパレータ出力が連続的にトグルします。

回避方法

スタンバイ モードでは、温度センサをコンパレータ入力として使用しないでください。

CPU_ERR_01 *CPU モジュール*

カテゴリ

機能

機能

CPU キャッシュ コンテンツが破損する可能性があります

概要

メインフラッシュメモリへのアクセスと、NONMAIN やキャリブレーションデータ領域など他のメモリ領域の間を切り替えると、キャッシュ破損が発生するおそれがあります。

回避方法

メイン メモリ以外の領域に安全にアクセスするには、次の手順に従います:
1.CTL.ICACHE を「0」に設定して、キャッシュを無効にします。
2.メモリ領域への必要なアクセスを実行します。
3.CTL.ICACHE を「1」に設定して、キャッシュを再び有効にします。

CPU_ERR_02 *CPU モジュール*

カテゴリ

機能

機能

CPUSS のプリフェッチ機能を無効にする制限

説明

保留中のフラッシュメモリアクセスがある場合、CPU プリフェッチを無効にしても無効にはなりません。

回避方法

プリフェッチャーを無効にし、SYSCTL でシャットダウンメモリへのメモリアクセス (SHUTDNSTORE) を発行します。これは SYSCTL.SOCLOCK.SHUTDNSTORE0; で実行できます。

メモリアクセスが完了すると、プリフェッチャーは無効になります。
例:
CPUSS.CTL.PREFETCH = 0x0、プリフェッチャーを無効にします
SYSCTL.SOCLOCK.SHUTDNSTORE0、シャットダウンメモリへのメモリアクセス

CPU_ERR_03 *CPU モジュール*

カテゴリ

機能

機能

低電力モードへの遷移時に、プリフェッチャが誤った命令を読み取る可能性がある

説明

低電力モードへ遷移する際に保留中のプリフェッチがある場合、プリフェッチャが誤って正しくないデータ (すべて 0) をフェッチする可能性があります。デバイスがウェイクアップした際、もしプリフェッチャおよびキャッシュが ISR コードによって上書きされない場合、フラッシュから実行されるメイン コードが破損する可能性があります。たとえば、ISR が SRAM 内にある場合、フラッシュからプリフェッチされた誤ったデータは上書きされません。ISR から復帰する際に、プリフェッチャ内

CPU_ERR_03 (続き) ***CPU モジュール***

の破損したデータが CPU によってフェッチされ、誤った命令が実行されるおそれがあります。ハードウェア イベント ウェイクアップは、デバイスをウェイクアップするがプリフェッチャをフラッシュしないプロセスのもう 1 つの例です。

回避方法

低電力モードに入る前にプリフェッチャを無効にします。

例:
CPUSS.CTL.PREFETCH = 0x0; // プリフェッチャを無効化
SYSCTL.SOCLOCK.SHUTDNSTORE0; // シャットダウン メモリから読み出し
__WFI(); // または __WFE(); この関数は低電力モードへの遷移を呼び出す
CPUSS.CTL.PREFETCH = 0x1; // プリフェッチャを再有効化

CPU_ERR_04 ***CPU モジュール***

カテゴリ

機能

機能

キャッシュが、ハード フォルトの原因となった FLASH 内のアドレスから正しいデータを返しません

説明

デバイスが FLASH アドレスからハード フォルトを引き起こした後、デバイスがハードフォルトから復旧し、(その FLASH アドレスのデータを格納している) キャッシュから情報を取得しようとすると、戻り値がゼロになります。
さらに、同じ FLASH アドレスにアクセスすると、新しいハード フォルトはトリガされません。

回避方法

CPUSS.CTL.ICACHE ビットを変更して、キャッシュを無効化してから再度有効化します。
例:
CPUSS->CTL &= ~(CPUSS_CTL_ICACHE_ENABLE); // 命令キャッシュを無効化します
CPUSS->CTL |= CPUSS_CTL_ICACHE_ENABLE; // 命令キャッシュを有効化します

CRC/ CRCP_ERR_01 ***CRC/CRCP モジュール***

カテゴリ

機能

機能

LPM が中断している状態では、DMA をトリガして CRC/CRCP モジュールにアクセスすることはできません

説明

RUN0/1/2 および SLEEP0/1/2 モードでは、DMA は通常 CRC/CRCP モジュールにアクセスできます。STOP/STANDBY モードでは、DMA がトリガされると、デバイスは中断状態の低消費電力モードに移行します。ただし、一部のデバイスでは、このモードになると、DMA は CRC/CRCP モジュールにアクセスできなくなります。

回避方法

1.CRC/CRCP への DMA アクセスが必要な場合は、STOP/STANDBY モードにしないでください。2.STOP/STANDBY モードでは、別のウェークアップ機能 (割り込みなど) を使用して、デバ

CRC/CRCP_ERR_01 (続き)

CRC/CRCP モジュール

イスを STOP/STANDBY モードから復帰させ、CRC/CRCP への DMA アクセスをトリガするか、CRC/CRCP へのアクセスに DMA ではなく CPU を使用します。

FLASH_ERR_04

FLASH モジュール

カテゴリ

機能

機能

エラーがメイン フラッシュ領域以外で発生した場合、SYSCTL_DEDERRADDR には誤ったアドレスが報告されます。

説明

回避方法

SysCtl_DEDERRADDR の戻りアドレスで 0x00Cxxxxx が返る場合は、0x41000000 で OR 演算を実行して、NONMAIN または工場出荷時領域の復帰アドレスに適切なアドレスを取得します。たとえば、SYSCTL_DEDERRADDR = 0x00C4013C の場合、実際のアドレスは 0x41C4013C となります。

SYSCTL_DEDERRADDR の復帰アドレスが 0x00Dxxxxx を返した場合、Databank の正しい復帰アドレスを得るために、0x41000000 との OR 演算を行います。たとえば、SYSCTL_DEDERRADDR = 0x00D0012A の場合、実際のアドレスは 0x00D0012A となります。

FLASH_ERR_05

FLASH モジュール

カテゴリ

機能

機能

DEDERRADDR に誤ったリセット値が設定される可能性があります

説明

SYSCTL -> DEDERRADDR のリセット値では、正しい 0x00000000 のかわりに 0x00C4013C が返されることがあります。エラーが発生している場所はファクトリ トリム領域であり、故障を示すものではありません。そのため、この値は無視して問題ありません。デバイスに NONMAIN をプログラムすされると、リセット値が変化する傾向があります。

回避方法

0x00C4013C を別のリセット値として受け入れ、ブートからのデフォルト値を 0x00000000 または 0x00C4013C にすることができます。戻り値はデバイス上の MAIN フラッシュの範囲外であるため、実際のフラッシュ DED ステータスから返された可能性はありません。

FLASH_ERR_08

FLASH モジュール

カテゴリ

機能

機能

通常の無効なメモリ領域に対してハード フォルトは生成されません

FLASH_ERR_08
(続き)

FLASH モジュール

説明

不正なメモリ アドレス空間へのアクセス中は、以下に示すようにハード フォルトは生成されません。1.0x010053FF ～ 0x20000000 2。0x40BFFFFF ～ 0x41C00000 3。0x41C007FF～0x41C40000

回避方法

番号

FLASH_ERR_09

FLASH モジュール

カテゴリ

機能

機能

特定のフラッシュ スワップ ポリシー下では、bank1 の消去時に、BANK1 の静的書き込み保護は機能しません

説明

デバイスのフラッシュ メモリが最大容量のデバイスでない場合 (例：MSPM0L111x ファミリには、MSPM0L1117 と MSPM0L1116 として 2 つの異なるデバイスがありますが、MSPM0L1116 にのみこの問題があります)、以下の条件で bank1 が予期せず消去されるという問題があります：失敗ケース 1：BANK1 の静的書き込み保護が有効、フラッシュ スワップ ポリシーが無効、SECCFG_Reg.FLBANKSWP レジスタの USEUPPER ビットが 0 または 1 の場合、バンク消去コマンドで BANK1 が予期せず消去されます。失敗ケース 2：BANK1 の静的書き込み保護が有効、フラッシュ スワップ ポリシーが有効、SECCFG_Reg.FLBANKSWP レジスタの USEUPPER ビットが 0 の場合、バンク消去コマンドで BANK1 が予期せず消去されます。

回避方法

1.Bank1 にはバンク消去の代わりにセクタ消去を使用します。または 2.同じデバイス ファミリ内で最大容量のフラッシュ デバイスを使用します。

GPIO_ERR_03

GPIO モジュール

カテゴリ

機能

機能

デバッガで GPIO EVENT0 IIDX を読み取ると、割り込みがクリアされます。

説明

GPIO の EVENT0 の IIDX をデバッガで読み取ると、CPU による読み取りと見なされ、割り込みがクリアされます。

回避方法

デバッグ中、event0 の IIDX は、ソフトウェアで RIS を読み取ることで確認できます。

GPIO_ERR_04

GPIO モジュール

カテゴリ

機能

機能

グローバル ファストウェイクの設定により、PAD データが DIN レジスタに入力されません。

GPIO_ERR_04 (続き)

GPIO モジュール

説明

CTL レジスタのファストウエィクのみのビットを設定し、実行モードでデータを PAD に強制的に入力しても、PAD のデータは DIN レジスタに反映されません。これは、CTL レジスタの設定により、PAD から DIN レジスタへのデータの流れが阻止されるためです。

回避方法

PAD のデータが DIN レジスタに入力されることが予想される場合は、GPIO ファストウエィクのみの機能を使用しないでください。

GPIO_ERR_06

GPIO モジュール

カテゴリ

機能

機能

システム内で DMA 同時アクセスが発生すると、GPIO DOUT、DOUTSET、および DOUTCLR への CPU 書き込みアクセスが無視される場合があります。

説明

GPIO DMAMASK は、DOUT MMR 内の DMA が変更できるレジスタ ビットを制御します。実装の誤りにより、システム内で DMA 同時アクセスが発生すると、GPIO DOUT、DOUTSET、DOUTCLR への CPU アクセスにも DMAMASK が適用されます。その結果、DMAMASK = 1 を設定して DMA に割り当てられたレジスタ ビットは、CPU によって設定されない可能性があります。

回避方法

1.CPU と DMA からアクセスする必要のある GPIO を分離してください。DMAMASK を使用して、DMA に割り当てる GPIO と、CPU 用に定義する GPIO を定義します。CPU から DMA に割り当てられたビットにアクセスしないでください。
2.CPU と DMA で同じ GPIO へ同時にアクセスする必要がある場合、CPU が GPIO にアクセスする前に DMAMASK をクリアする必要があります。疑似コード例:
I. DL_GPIO_disableDMAAccess(GPIO_REG, SELECTED_PINS); // 特定の GPIO レジスタからのピン グループに対する DMA アクセスを無効にする
II./* GPIO の DOUT、DOUTSET、および / または DOUTCLR レジスタの変更を実装する */
III.DL_GPIO_enableDMAAccess(GPIO_REG, SELECTED_PINS); // 特定の GPIO レジスタからのピン グループに対する DMA アクセスを有効にする

I2C_ERR_01

I2C モジュール

カテゴリ

機能

機能

SMBUS クイック コマンドが発行されると、I2C モジュールが SBMUS モードで SDA ラインをホールドし続ける場合があります

説明

I2C モジュールがターゲット モードで SBMUS に設定されている場合、バス コントローラがデバイス宛に SMBUS クイック コマンド (I2C START コンディション → 7 ビット アドレス → 1 ビットの R/W ビット → 1 ビットの ACK → I2C STOP 条件) を発行すると、I2C モジュールが SDA ライン

I2C_ERR_01 (続き) ***I2C モジュール***

を Low に引き下げようとすることがあります。これがバス コントローラによる I2C STOP 条件の生成と同時に起こると、STOP 条件が正しく完了せず、通信が妨げられる可能性があります。

回避方法

I2C モジュールが SDA ラインを Low に駆動するのを防ぐために、アドレス ACK が完了する前に、MSB を 1 に設定したデータを I2C モジュールの送信 FIFO にロードします。これにより、バス コントローラは STOP 条件を正常に発行し、SMBUS クイック コマンドを完了できます。

I2C_ERR_03 ***I2C モジュール***

カテゴリ

機能

機能

ソースに MFCLK を使用している場合、I2C ペリフェラル モードを起動できません

説明

I2C モジュールがペリフェラル モードに設定されており、かつ I2C が MFCLK (中周波クロック) をソースとして使用していて、さらにデバイスが STOP2 または STANDBY0/STANDBY1 の電力モードにある場合、データを受信しても I2C はデバイスをウェイクアップできません。

回避方法

I2C をペリフェラル モードで使用し、データ受信時に低電力モードからのウェイクアップを必要とする場合は、I2C のクロック ソースを MFCLK ではなく BUSCLK に設定します。

I2C_ERR_04 *I2C モジュール*

カテゴリ

機能

機能

SCL が Low になり、ターゲットのウェイクアップが有効になっている場合、デバイスが無期限にクロック ストレッチを行う可能性があります。

説明

デバイスがターゲットモードにあり、クロック ストレッチや外部によるグランド接続により SCL が Low に保たれている場合、コントローラがバスを解放しても、I2C ターゲットはクロック ストレッチを解除できない状態になることがあります。

回避方法

ターゲット ウェークアップ イネーブル ビット (SWUEN) をディセーブルにします。

I2C_ERR_05 *I2C モジュール*

カテゴリ

機能

機能

進行中のトランザクション中に ACTIVE ビットをトグルすると、I2C SDA が 0 に固定化されるおそれがあります

概要

進行中の転送中にアクティブビットがトグルされると、ステートマシンはリセットされます。ただし、コントローラによって駆動される SDA と SCL 出力はリセットされません。SDA が 0 の状態でコントローラが IDLE 状態に遷移すると、コントローラは IDLE 状態から先へ進めず、SDA の値も更新できなくなります。ターゲットの BUSUSY がセットされ(アクティブビットのトグルによってライン上で開始が検出されます)、BUSY はクリアされません。これは、コントローラが停止を駆動してクリアできないためです。

回避方法

進行中のトランザクション中は、ACTIVE ビットをトグルしないでください。

I2C_ERR_06 *I2C モジュール*

カテゴリ

機能

機能

SMBus の High タイムアウト機能は、I2C クロックが 24 kHz 未満になると動作しません

説明

SMBus の High タイムアウト機能は、I2C クロックレートが 24 kHz 未満(20 kHz、10 kHz など)では正常に動作しません。SMBus 仕様から、アクティブトランザクション中の SCL High 時間の上限は 50µs です。開始 MMR ビットの書き込みから SCL Low までに要する合計時間は 60µs で、50µs 以上です。タイムアウトイベントのトリガがかかりされ、転送開始時にトランザクションを完了することなく I2C コントローラが IDLE に移行できます。以下は詳細な説明です。SCL が 20 kHz に構成されている場合、SCL の Low 期間と High 期間はそれぞれ 30 µs および 20 µs です。まず、High タイムアウトカウンタでデクリメントが開始し、同時に MMR ビットの書き込みが開始します。その後、START MMR ビットの書き込みから SDA が Low(スタート条件)になるまでに、1 SCL Low 期間(30 µs)かかります。次に、SDA が Low(スタート条件)になってから SCL が Low になる(データ転送が開始)までにさらに別の SCL Low 期間(30µs)がかかり、この時点

I2C_ERR_06 (続き) ***I2C モジュール***

で High タイムアウトカウンタが停止します。合計で、カウンターの開始から終了まで 60µs かかります。ただし、高タイムアウトカウンタには上限（50µs）により、I2C トランザクションは問題なく正常に動作しますが、タイムアウトイベントがトリガされます。

回避方法

I2C クロックが 24KHz 未満の場合は、SMBus 高タイムアウト機能を使用しないでください。

I2C_ERR_07 ***I2C モジュール***

カテゴリ

機能

機能

コントローラの制御レジスタへの連続書き込みを行うと、I2C 通信が開始されない可能性があります。

説明

連続 CTR レジスタへの書き込みでは、次の CTR .START によって正しく開始条件が発生しません。

回避方法

CTR.START を含むすべての CTR ビットを 1 回の書き込みで書き込むか、CTR 書き込みと CTR.START 書き込みの間に 1 クロック サイクル待機します。

I2C_ERR_08 ***I2C モジュール***

カテゴリ

機能

機能

RXDONE 割り込みの直後に FIFO を読み出すと、誤ったデータが取得されます

概要

RXDONE 割り込みが発生したとき、FIFO は最新のデータに対して常に更新されるとは限りません。

回避方法

最新のデータが FIFO に確実に反映されるように、2 つの I2C クロックサイクル分待機してください。I2C CLK は、I2C レジスタの CLKSEL レジスタに基づいています。

I2C_ERR_09 ***I2C モジュール***

カテゴリ

機能

機能

I2C を低速で動作させている場合、割り込みサービスルーチン（ISR）内での読み取り時に、開始アドレス一致ステータスがタイミング的に更新されていない可能性があります。

説明

I2C 速度が 100kHz 未満で動作している場合、ADDRMATCH ビット (TSR レジスタのアドレス一致) が割り込みによる読み取りに間に合うように設定されない可能性があります。

I2C_ERR_09 (続き) ***I2C モジュール***

回避方法

I2C で 100kHz 未満で実行している場合は、ADDRMATCH ビットを読み取る前に少なくとも 1 つの I2C CLK サイクルを待機します。

I2C_ERR_10 ***I2C モジュール***

カテゴリ

機能

機能

低消費電力に移行しないよう、I2C ビジーステータスは有効になっています

概要

I2C ターゲットモードでは、STOP ビットがない場合、トランザクションの後、I2C ビジーステータスは High のままです。

回避方法

STOP ビットを送信するように I2C コントローラをプログラムします。最後のバイトに対して NACK を送信しないでください。すべての I2C 転送は STOP 条件で終了し、適切な BUSY ステータスと非同期クロック要求の動作を維持してください（低消費電力モードへの再移行に備えるため）。

I2C_ERR_13 ***I2C モジュール***

カテゴリ

機能

機能

I2C BUSY ビットのポーリングでは、コントローラ転送の完了を確実に保証できない場合がある

説明

CCTR.BURSTRUN ビットを設定して I2C コントローラ転送を開始した後、BUSY ステータスがアサートされるまでに、約 3 クロック分の I2C 機能クロックサイクルを要します。転送完了を待つように CCTR.BURSTRUN を設定した直後に BUSY ビットをポーリングすると、BUSY ステータスが設定される前にチェックされることがあります。この問題は、CLKDIV の値が大きい (その結果、I2C 機能クロックが遅くなる) 場合やコンパイラ最適化レベルが高い場合に、発生する可能性がより高くなります。

回避方法

BUSY ステータスをポーリングする前に、ソフトウェア遅延を追加します。ソフトウェア遅延 = 3 x CPU CLK / I2C の機能クロック = 3 x CPU CLK / (CLKSEL / CLKDIV)。たとえば、クロック分周器 (CLKDIV) が 8MHz、クロックソース (MFCLK) が 4MHz、CPU CLK が 32MHz の場合、ソフトウェア遅延 = 3 x 32 MHz / (4 MHz/ 8) = 192 CPU サイクル。

I2C_ERR_15 ***I2C モジュール***

カテゴリ

機能

機能

I2C SDA のグリッチにより、低消費電力モードで I2C ペリフェラルがアクティブ状態になる

説明

I2C SDA 上の 4µs 以内のグリッチにより、低消費電力モード (STOP/STANDBY) で I2C ペリフェラルがアクティブ ステータスに切り替わり、低消費電力モードに戻らなくなります。

I2C_ERR_15 (続き) ***I2C モジュール***

回避方法

1.I2C バスの容量を増やします。ただし、合計が I2C 規格で許容される値を超えないようにしてください。2.I2C バスの電圧を上昇させます。3.I2C ラインをノイズ発生源から離します。4.I2C ペリフェラルのステータスを確認するために、デバイスを定期的にウェークアップし、I2C ペリフェラルのステータスが IDLE ステータスでない場合は、I2C をリセットして再度初期化します。

KEYSTORE_ERR_01 ***キーストアモジュール***

カテゴリ

機能

機能

STATUS.STAT の値は、キーアクセスがない場合、0 または 1 になります。

説明

STATUS.STAT のリセット値は 1 で、以下の条件で 0 になります。1.リセット後、レジスタウィンドウを介したデバッガアクセスは 0x00 を返します。2.リセット後、最初の CPU 読み取りは 0x01 を返し、その後の CPU 読み取りは 0x00 を返します。3) リセット後、最初に他の キーストアレジスタを読み取り、次に STATUS.STAT を読み取ると、0x00 が返されます。

回避方法

STATUS.STAT=0x0 は「エラーなし」を意味します。スロットが有効かどうか (キーが存在するかどうか) を確認するには、STATUS.VALID を確認してください。

LCD_ERR_01 ***LCD モジュール***

カテゴリ

機能

機能

LCD モジュールがオフのときに低消費電力モード仕様を達成するために時間を延長

説明

LCD モジュールがオフのときに低消費電力モードの IDD 電流仕様を達成するために時間を延長。

回避方法

LCD モジュールをスタンバイ モードに電源を入れると、正確な IDD 電流引き込みを迅速に実現できます。シャットダウン モードでの回避方法はありません。

LFSS_ERR_01 ***LFSS モジュール***

カテゴリ

機能

機能

VDD が VBAT を下回ると VBAT 電流が増加する

説明

VBAT が VDD を上回ると、VBAT をからより多くの電流が流れます。

LFSS_ERR_01 (続き)

LFSS モジュール

回避方法

回避方法はありません。低い VBAT 電流を維持するには、VDD を VBAT より大きくする必要があります。

LFSS_ERR_02

LFSS モジュール

カテゴリ

機能

機能

LFXT と IWDT を同時に使用すると、VBAT 電流が増加する

説明

VBAT のクロック ソースとして LFXT を有効化して IWDT をイネーブルにすると、VBAT 電流は大幅に増加します。

回避方法

IWDT を実行しながら、LFXT を使用しないでください。

LFSS_ERR_03

LFSS モジュール

カテゴリ

機能

機能

LFSS 割り込みフラグは、EVENT0/1 の割り込み設定に対応して IMASK がイネーブルの場合にのみ、RIS で生成されます。

説明

LFSS 割り込みフラグは、EVENT0/1 の割り込み設定に対応して IMASK がイネーブルの場合にのみ、RIS で生成されます。これは、すべての LFSS 割り込みで見られます。

回避方法

一般的に、割り込みのステータスを把握するために RIS ステータスを直接ポーリングすることはお勧めしません。ただし、CPU NVIC レベルで LFSS 割り込みをイネーブルせずに、ポーリングによって割り込みステータスを確認したい場合、そのイベントに対応する IMASK ビットを EVENT0 と EVENT1 MMR のいずれかでイネーブルにできます。これにより、RIS と MIS が予想どおりに更新されます。

LFXT_ERR_01

LFXT モジュール

カテゴリ

機能

機能

LFCLKCFG mmr に何かを書き込むまで、LFXT クロックはアクティブになりません

説明

STARTLFXT を設定した後、CLKSTATUS は期待どおりに更新されますが、LFCLKCFG への書き込みが行われるまで、LFCLK_OUT に LFXT クロックは検出されません。LFXT が LFCLK ツリーのソースとして構成されている場合、VBAT 電源がダウンしてから復帰すると、LFCLKCFG への書き込みが行われるまで、LFCLK はブランクです。

LFXT_ERR_01 (続き)

LFXT モジュール

回避方法

LFXT がブランキング状態から起動するたびに、LFCLKCFG レジスタ (XT1DRIVE フィールドなど) を手動で構成して LFXT を有効にします。

LFXT_ERR_02

LFXT モジュール

カテゴリ

機能

機能

VDD ドメインの電源サイクル後の LFXT の LFCLKMUX ステータスが正しくない

説明

LFXT (LFSS 内) はユーザーによって設定されます。VDD の パワー サイクルが発生すると、VDD 起動後に、LFCLKMUX ステータスが誤ってデフォルト状態 (LFOSC を示す LFCLKMUX ステータス) にリセットされます。デバイスは引き続き、LFXT を LFCLK ソースとして使用します。

回避方法

1. VDD のパワーサイクルが発生した場合は、LFCLKMUX ビット ステータスは無視します。2. VDD のパワーサイクルが発生した場合は、LFXT を再構成します。

PMCU_ERR_08

PMCU モジュール

カテゴリ

機能

機能

デバイスが LPM へ移行中にトリガが与えられると、想定よりもウェイクアップ時間が長くなることがあります

説明

デバイスが低電力モードへ移行中にウェイクアップ信号が与えられると、約 3us の追加ウェイクアップ時間が発生します。

回避方法

回避方法はありません。

PMCU_ERR_09

PMCU モジュール

カテゴリ

機能

機能

POR (NRST > 1s) リセット後、RSTCAUSE が誤って 0xC に更新されます。

説明

NRST を使用して POR (NRST > 1s) リセットをトリガーすると、RSTCAUSE が誤って 0xC に更新されます。これは 0x2 に更新されるべきです。

回避方法

リセット原因を確認する必要がある場合は、RSTCAUSE ステータスを取得した後、利用可能な SHUTDOWN メモリバイト (SHUTDNSTOREx) の 1 つを使用して、ゼロでないのデータを保存

PMCU_ERR_09 (続き)

PMCU モジュール

してください。RSTCAUSE が 0xC を返す場合、SHUTDNSTOREx データがクリアされると POR が発生し、SHUTDNSTOREx データが維持されると BOR が発生します。

PMCU_ERR_10

PMCU モジュール

カテゴリ

機能

機能

特定の動作条件下では、VBOOST により大きな遅延が発生する可能性があります

概要

アナログマルチプレクサの VBOOST は、VDD < 1.8V で大きな遅延が発生しました。このため、HFXT、COMP、SYSOSC（FCL-EXTERNAL R）、OPA、GPAMP などのその他のモジュールのセトリングタイムが遅延します。

回避方法

VDD を 1.8V 以上に維持し、GENCLKCFG[23:22] = 0x2 を設定して、VBOOST を ONALWAYS モードで使用します。

PMCU_ERR_12

PMCU モジュール

カテゴリ

機能

機能

BOR コールドブート仕様違反

概要

BOR コールドブート仕様に違反するおそれがあり、最大仕様は 1.65V です。

回避方法

回避策はありません。デバイスをコールドブートする際は、1.65V 以上の電圧で動作させてください。

RST_ERR_01

RST モジュール

カテゴリ

機能

機能

LFCLK_IN が LFCLK のソースとして選択されており、かつ LFCLK_IN が無効になっている場合、NRST リリースは検出されません

説明

LFCLK = LFCLK_IN で、LFCLK_IN を無効にすると、NRST パルスエッジ検出を見逃されし、デバイスがリセットから復帰しないコーナーシナリオが発生します。この問題は、NRST パルス幅が 608µs 未満のときに見られます。NRST パルスが 608µs を超える場合は、リセットは通常どおり表示されます。

回避方法

この問題を回避するため、608µs よりも高い NRST パルス幅を維持します。

RST_ERR_02 *RST モジュール*

カテゴリ

機能

機能

NRST デアサートが最初の LFOSCGOOD アサートと一致した場合、デバイスは起動に失敗する可能性があります

説明

起動イベント中、内部コア電圧 (VCORE) が安定したレベルに達すると、内部低周波数発振器 (LFOSC) が有効になります。LFOSCGOOD 信号は、VCORE が安定状態に達してから 250µs (最小) から 1.5ms (最大) の範囲内で High にアサートされます。VDD の VBOR+ スレッショルドは、VCORE が安定したときのリファレンス ポイントとして使用できます。NRST のデアサート (Low から High への遷移) が 200ns ウィンドウ内に LFOSCGOOD の立ち上がりエッジと一致した場合、NRST デアサートは検出されず、デバイスはリセット状態のままとなり、スタートアップ シーケンスのブート フェーズを終了できません。NRST デアサート タイミングは、VDD の立ち上がり速度と、NRST 回路上の外部抵抗およびコンデンサ (RC) ネットワークの両方によって制御されるため、実際にはこの状態をトリガできます。この現象が発生しやすいことが知られている構成例として、NRST 回路上に 47k の抵抗と 10nF のコンデンサが含まれている場合が挙げられます。

回避方法

回避方法 1：RC フィルタには十分小さなプルアップ抵抗を使用して、LFOSCGOOD の最小アサート時間 (250µs) よりも少なくとも 50µs 前に、NRST 信号が VDD の 90% まで確実に立ち上がるようにします。これにより、NRST デアサート エッジが LFOSCGOOD のアサート ウィンドウから十分に離れるようにします。NRST 回路上に 1kΩ の抵抗と 10nF のコンデンサを配置した構成が、準拠していることが確認されています。Workaround2：外部リセット コントローラを採用し、VDD 立ち上がりの開始から少なくとも 2.0ms 間、NRST をロジック Low レベルに保持します。これにより、LFOSCGOOD の最大アサート時間 (1.5ms) 後に NRST デアサートが発生することが保証されます。

RTC_A_ERR_02 *RTC モジュール*

カテゴリ

機能

機能

LFSS SW POR リセット時に、LFSS-RTC TS データ レジスタの内容がリセットされない (LFSSRST)

説明

LFSS SW POR リセット (LFSSRST) では、RTC TS (タイムスタンプ) データ レジスタは 0x0 にリセットされません。

回避方法

TSCLR レジスタを使用して、RTC TS データ レジスタの値を手動でクリアします。

SPI_ERR_03 *SPI モジュール*

カテゴリ

機能

機能

ペリフェラルとして構成した場合、CSCLR を有効にすると、受信データは SPH = 0 モードで 1 ビット右シフトされる

SPI_ERR_03 (続き) ***SPI モジュール***

説明

ペリフェラル モードで CSCLR を有効にした場合、CS 信号がアクティブまたは非アクティブのときに SCK ラインにグリッチが発生すると、次の最初のフレームで受信データが 1 ビット右方向にシフトします。この問題は、SPH = 0 の Motorola SPI フレーム フォーマットで発生し、CS が非アクティブの時に SCK がトグルするマルチ ペリフェラル モードに影響します。

回避方法

1. CSCLR = 0h に設定します。
2. SPH = 0 モードで CSCLR = 1h に設定すると、常に最初のフレームをドロップします。

SPI_ERR_04 ***SPI モジュール***

カテゴリ

機能

機能

SPI ペリフェラルが受信モードのみの場合、各フレーム受信後の IDLE/BUSY ステータストグル。

概要

SPI ペリフェラルが受信モードのみの場合、SPI がデータを連続的に受信している間に、各フレーム受信の後で、IDLE 割り込みおよび BUSY ステータスがトグルされます（SPI_PHASE = 1）。ここでは、ペリフェラルの TXFIFO にロードされるデータはなく、TXFIFO は空です。

回避方法

SPI ペリフェラルのみの受信モードを使用しないでください。SPI ペリフェラルを送受信モードに設定します。TX FIFO のデータを SPI 用に設定する必要はありません。

SPI_ERR_05 ***SPI モジュール***

カテゴリ

機能

機能

SPI ペリフェラルの受信タイムアウト割り込みは、RXFIFO のデータの有無にかかわらず発生します

概要

SPI タイムアウト割り込みを使用すると、最終的な SPI CLK を受信した後でも RXTIMEOUT でデクリメントが継続するため、誤った RXTIMEOUT が発生するおそれがあります。

回避方法

最後のパケットを受信した後は、RXTIMEOUT を無効にします（これは ISR 内で実行可能です）。その後、SPI 通信が再開されるときに、RXTIMEOUT を再度有効にしてください。

SPI_ERR_06 ***SPI モジュール***

カテゴリ

機能

機能

デバッグ HALT がアサートされている場合、IDLE/BUSY ステータスは SPI IP の正しい状態を反映しません

SPI_ERR_06 (続き) *SPI モジュール*

概要

IDLE/BUSY は HALT とは無関係で、RXFIFO/TXFIFO の書き込み/読み取りストローブのみをゲーティングします。つまり、コントローラがデータ送信中であっても、そのデータが FIFO にラッチされていない状態で BUSY ステータスが設定されてしまいます。POCI 回線は、停止中に以前に送信されたデータを回線上で送信します

回避方法

SPI IP が停止しているときは、IDLE/BUSY ステータスを使用しないでください。

SPI_ERR_07 *SPI モジュール*

カテゴリ

機能

機能

SPI ペリフェラルで TXFIFO への読み取り / 書き込みが同時に発生した場合、SPI アンダーフロー イベントは生成しない場合があります。

説明

SPI.CTL0.SPH = 0 であり、本デバイスが SPI ペリフェラルとして構成されている場合。

SPI コントローラからの読み取り要求がある間に TXFIFO への書き込みが発生した場合、読み取り / 書き込み要求が同時に発生するため、アンダー フロー イベントが生成されない可能性があります。

回避方法

SPI コントローラによるデバイスのアドレス指定中、TXFIFO が確実に空でないようにします。これは、同じ TXFIFO アドレスへの書き込みと読み取りを避けるために、データを事前ロードすることで実現できます。あるいは、CRC のようなデータチェック戦略を使用してパケットが確実に正しく送信されるようにし、CRC が一致しない場合にデータを再送信することもできます。

SPI_ERR_10 *SPI モジュール*

カテゴリ

機能

機能

DMA は、最新の SPI トリガ カウントをサンプリングしていません

説明

受信タイムアウト (RTOUT) イベントで DMA 転送をトリガするように SPI が構成されている場合、トリガ要求の時点で DMA チャネルが別の転送の処理をビジー状態にしている場合、DMA が要求をアクノリッジする前に SPI が受信した追加のバイトは転送に含まれません。DMA は、トリガ要求が発行されたときに受信されたバイト数のみを転送し、その後受信したバイトは RX FIFO で未読のままになります。

回避方法

RX トリガ条件を RTOUT と組み合わせて使用するように DMA_TRIG_RX を構成します。RX トリガは、RX FIFO が設定されたスレッショルド レベルに達すると起動し、DMA 処理が開始される前に、予測されるバイト数がすべて FIFO に存在することを保証します。これにより、DMA チャネルによる要求のアクノリッジが遅延するかどうかに関係なく、DMA 転送カウントが正確であることが保証されます。

SRAM_ERR_01 *SRAM モジュール*

カテゴリ

機能

機能

CPU/DMA からの非 ECC / パリティ アパーチャ アクセスと ECC アパーチャ アクセスを混在させると、機能エラーが発生する可能性があります

説明

CPU アクセスが非 ECC / パリティ アパーチャを使用し、DMA が ECC アパーチャを使用するように設定されている場合は、断続的な故障の原因となることがあります。

回避方法

CPU と DMA の間で非 ECC / パリティ / ECC アパーチャ アクセスを混在させないようにします。CPU アクセスと DMA アクセスの両方に同じタイプのアパーチャを使用します。安全メカニズムでフォルトの注入が必要な場合は、CPU ソフトウェアによってこの安全診断メカニズムが実行されているは、DMA を同時に使用しないでください。

SYSCTL_ERR_01 *SYSCTL モジュール*

カテゴリ

機能

機能

SW-POR 機能は、HW_POR と組み合わせて使用できます

説明

ソフトウェア トリガ POR を生成するために正しいキーを使って LFSSRST レジスタに書き込むと、RSTCAUSE レジスタには、予測される 0x3 (ソフトウェア トリガ POR) ではなく 0x2 (NRST トリガ POR) が表示されます。これは、SW-POR 機能が HW-POR パスと組み合わされているためです。

回避方法

番号

SYSCTL_ERR_02 *SYSCTL モジュール*

カテゴリ

機能

機能

BOOTRST の後には、SYSSTATUS.FLASHSEC はゼロ以外になります

説明

BOOTRST/ブートコード完了後、SYSSTATUS.FLASHSEC はゼロ以外になります。これは、お客様がブートコードが完了した後に表示されます。

回避方法

番号

SYSCTL_ERR_03 *SYSCTL モジュール*

カテゴリ

機能

機能

DEDERRADDR は、SYSRESET または SYSSTATUSCLR への書き込みの後にも持続します

詳細

SYSRESET または SYSSTATUSCLR レジスタへの書き込みの後も、DEDERRADDR は持続します。この値は、新しい FLASHDED エラーが発生した場合にのみ上書きされます。この挙動は、初期リセット値をゼロに規定されているテクニカル リファレンス マニュアル (TRM) に矛盾します。

回避方法

回避方法はありません。

SYSCTL_ERR_04 *SYSCTL モジュール*

カテゴリ

機能

機能

SYSRESET 後に SYSSTATUS.FLASHSEC がクリアされません。

説明

SYSSTATUS.FLASHSEC は、SYSRESET 後にクリアされず、SYSSTATUSCLR レジスタに書き込まれることでのみクリアされます。

回避方法

番号

SYSCTL_ERR_05 *LFCLK モジュール*

カテゴリ

機能

機能

シャットダウンからの終了時に LFCLK が動作しない

説明

LFCLK_IN ピンがプルアップ付きの汎用入力 (または) LFCLK_IN 機能として構成されている場合、この構成では、シャットダウン モードを終了すると、LFCLK がスタックします。

回避方法

次のいずれかを選択：1. この LFCLK_IN I/O では、プルアップの代わりにプルダウンをイネーブルにする、2. 入力として構成するのを避ける

SYSCTL_ERR_06 *CLK_OUT モジュール*

カテゴリ

機能

機能

非同期クロックが CLK_OUT ソースとして選択されているときに、ユーザーが CLK_OUT をディスエーブルにすると、外部クロック出力 (CLK_OUT) でグリッチが発生する可能性があります

SYSCTL_ERR_06
(続き) ***CLK_OUT モジュール***

説明

CLK_OUT のクロック ソースが現在のバス クロックと非同期である場合 (たとえば、バス クロックが SYSOSC で、CLK_OUT のクロック ソースが LFCLK に選択されている場合)、この場合、CLK_OUT ピンが一定時間イネーブルになってからディスエーブルになるときにグリッチが発生することがあります

回避方法

外部ピンへのグリッチ出力を回避するには、以下のシーケンスに従ってください:CLK_OUT を有効化するケース:IOMUX の有効化設定は、CLK_OUT の有効化設定の後に行う必要があります。CLK_OUT を無効化するケース:IOMUX の無効化設定は、CLK_OUT の無効化設定より前に行う必要があります。

SYSOSC_ERR_01 ***SYSOSC モジュール***

カテゴリ

機能

機能

STOP1 モードと SYSOSC の FCL を併用すると、MFCLK にドリフトが発生する可能性があります

説明

MFCLK が有効になっており、SYSOSC が周波数補正ループ (FCL) モードを使用しており、STOP1 の低電力動作モードが使用されている場合、SYSOSC が 4MHz から 32MHz に切り替わる際 (STOP1 モードから RUN モードへの移行時、または SYSOSC を 32MHz に強制する非同期の高速クロック要求時)、MFCLK が 2 サイクル分ドリフトする可能性があります。

回避方法

STOP1 モードではなく STOP0 モードを使用します。STOP0 モードを使用する場合、MFCLK ドリフトは発生しません。

または

STOP1 を使用する場合は、FCL モードで SYSOSC を使用しないでください(FCL を無効のままにしておきます)。

SYSOSC_ERR_02 ***SYSOSC モジュール***

カテゴリ

機能

機能

SYSOSC が FCL モードで無効化されている LPM 中に非同期クロック要求を受信しても、MFCLK は動作しません

説明

以下のシナリオでは、MFCLK はトグルを開始しません:
1.FCL モードを有効にした後、MFCLK を有効にします 2.SYSOSC が無効になる低消費電力モードに移行します(SLEEP2/STOP2/STANDBY0/STANDBY1)。
3.MFCLK を機能クロックとして使用する一部のペリフェラルから非同期要求が受信されます。

SYSOSC_ERR_02
(続き)

SYSOSC モジュール

ASYNC 要求を受信すると、SYSOSC は有効になり、ulpclk は 32MHz になります。ただし、デバイスが依然として LPM に設定されているため、MFCLK はゲートオフの状態となり、一切トグルしません。

回避方法

SYSOSC が FCL モードを使用している場合は、通常 SYSOSC がオフになる LPM モードへ移行する際に、ペリフェラル用の MFCLK を有効にしないでください。

SYSOSC_ERR_04 ***SYSOSC モジュール***

カテゴリ

機能

機能

FCL ON モードでは、SYSOSC の精度が低下します

説明

内部発振器 SYSOSC の FCL ON モードを使用する場合、精度が最大 ±3% 低下する可能性があります。精度の低下は、4MHz FCL サンプリング クロックと、システム内のノイズとの間の同期に起因します。

回避方法

SYSPLL FCL ON モードで使用する場合は、次のように SYSPLL 周波数に 4MHz の倍数以外の値を使用します。例:78MHz または 79MHz
SYSPLL を 16、32、48、40、64、80MHz などにしないでください。
78MHz の場合:
SYSPLLCFG1.PDIV = 0x3 と SYSPLLCFG1.QDIV を 38 に設定します。
4MHz の倍数でない SYSPLL 動作周波数であっても、FCL サンプリング クロックに対して少なくとも 1 マイクロ秒に 1 回は同期します。2 つのクロックが共通係数を共有する場合は、さらに高い頻度で同期します。

SYSOSC_ERR_05 ***SYSOSC モジュール***

カテゴリ

機能

機能

FCL が有効な場合、LPM からの非同期高速ウェークアップ中、SYSOSC はベース周波数より低く動作する場合があります

説明

FCL = 有効な場合、低消費電力モード時およびアクティブな非同期高速クロック要求がある場合、SYSOSC はベース周波数より最大 10% 低い値で動作する可能性があります。これは、デバイスが低消費電力モードにある間に発生します。なぜなら、FCL = 有効であっても、FCL = 無効トリムが常に適用されてしまい、FCL = 有効トリムが使用されないためです。デバイスがウェークアップし、LPM を終了して要求を処理すると、SYSOSC は正しいトリムを適用して、目的のターゲット周波数で通常の FCL = 有効動作を再開します。

ほとんどの使用事例では大きな影響はなく、必要に応じて FCL および非同期の高速クロック要

SYSOSC_ERR_05
(続き)

SYSOSC モジュール

求の両方を引き続き使用できます。このエラッタが大きな影響を与える主なアプリケーションは、UART が非同期高速クロック要求のソースであるアプリケーションです。この一時的なシフトにより、デバイスが完全にウェークアップするまで実効ボーレートに影響を及ぼす可能性があるためです。

回避方法

1. FCL = 無効のままにします
2. FCL = 有効にする必要がある場合、非同期高速クロック要求を無効化します。

SYSOSC_ERR_06 ***SYSOSC モジュール***

カテゴリ

機能

機能

SYSOSC が FCL モードで無効化されている LPM 中に非同期クロック要求を受信した場合、MFCLK は 4MHz ではなく 32MHz で動作します

説明

以下のシナリオでは、MFCLK が誤って 32MHz に構成されています:
1.FCL モードを有効にした後、MFCLK を有効にします 2.SYSOSC が無効になる低消費電力モードに移行します (SLEEP2/STOP2/STANDBY0/STANDBY1)。
3.MFCLK を機能クロックとして使用する一部のペリフェラルから非同期リクエストを受信します。非同期リクエストを受信すると、SYSOSC が有効になり、ULPCLK は 32MHz になります。その場合、MFCLK が 32MHz に誤って構成されています。

回避方法

SYSOSC が FCL モードを使用している場合は、通常 SYSOSC がオフになる LPM モードへ移行する際に、ペリフェラル用の MFCLK を有効にしないでください。

TAMPERIO_ERR_01

TAMPERIO モジュール

カテゴリ

機能

機能

VDD 電源が失われると、改ざん IOMUX 状態ではラッチステータスが失われる

説明

LFSS I/O 状態 (IOMUX モードに設定されている場合) は、SoC のシャットダウン モード遷移中もラッチされ、保持されます。ただし、VDD のパワーサイクルを実行すると、状態はが失われます。

回避方法

VDD パワー サイクル中も LFSS I/O 状態を維持するには、改ざんモードに設定します。

TIMER_ERR_01 *TIMx モジュール*

カテゴリ

機能

機能

ハードウェア イベントでタイマを開始する場合、キャプチャ モードが誤った値を取得することがあります

説明

いずれかのタイマ インスタンスをキャプチャ モードで使用している場合、ゼロ条件 (ZCOND) やロード条件 (LCOND) によってタイマを開始すると、本来キャプチャすべき値ではなく、ゼロ値やロード値が該当する TIMx.CC レジスタにキャプチャされてしまうことがあります。この問題は、周期やパルス幅のキャプチャといった周期的な使用ケースに影響を及ぼします。

回避方法

以下のソフトウェア フローを使用して、周期またはパルス幅を計算します。回避方法の例については、MSPM0-SDK の timx_timer_mode_capture_duty_and_period を参照してください。

1. 0h に設定して、ZCOND または LCOND を無効化します。
2. キャプチャが発生した場合、キャプチャ値は正しく TIMx.CC に格納されます
3. TIMx.CTR をリロード値 (load または 0) に設定してタイマを再起動します。

TIMER_ERR_04 *TIMER モジュール*

カテゴリ

機能

機能

TIMER をゼロ イベントの直前に再有効化すると、再有効化が失われる可能性があります

説明

タイマーをワンショット モードで使用している場合、ゼロ イベント付近で再有効化を行うと再有効化が失われる可能性があります。タイマー有効ビットのハードウェア更新には、1 機能クロック サイクルが必要です。たとえば、タイマーのクロック ソースが 32.768kHz で、クロック分周比が 3 の場合、有効ビットが正しく 0 に設定されるまでに約 100µs かかります。

回避方法

タイマーを再有効化する前に 1 機能クロック サイクル分待機するか、一度タイマーを無効化してから再度有効化してください。

CTRCTL.EN = 0 でカウンタを無効化してから、CTRCTL.EN = 1 で再度有効化します

TIMER_ERR_06 *TIMG モジュール*

カテゴリ

機能

機能

CLKEN ビットに 0 を書き込んでも、カウンタは無効化されません

説明

カウンタ クロック制御レジスタ (CCLKCTL) のクロック イネーブル ビット (CLKEN) に 0 を書き込んでも、タイマは停止しません。

TIMER_ERR_06 (続き)

TIMG モジュール

回避方法

カウンタ制御 (CTRCTL) イネーブル (EN) ビットに 0 を書き込むことで、タイマを停止します。

TIMER_ERR_07

初期リピート カウンタの周期は、次のリピート モジュールより 1 回だけ少なくなる

カテゴリ

機能

機能

TIMER

説明

タイマ リピート カウンタ モードを使用する場合、以下のリピート カウンタには 0 とロード値の間の遷移が含まれるため、最初のリピートのカウントは後続のリピートより 1 回少なくなります。たとえば、TIMx.RCLD = 0x3 の場合、観測可能な 3 つのゼロ イベントが最初のリピート カウンタに現れ、観測可能な 4 つのゼロ イベントが後続するリピート カウンタ シーケンスに現れます。

回避方法

初期 RCLD 値を想定される RCLD より 1 だけ大きく設定し、リピート カウンタ ゼロ イベント (REPC) の ISR 内で、RCLD を目的の値に設定します。たとえば、4 回の繰り返しを行う場合は、初期 RCLD 値を RCLD = 0x5 に設定し、REPC 割り込み用のタイマ ISR 内で、RCLD = 0x4 に設定します。これで、すべてのタイマーの繰り返しで、ゼロ / ロード イベントの数が同一になります。

UART_ERR_01

UART モジュール

カテゴリ

機能

機能

STANDBY1 モードへの遷移時に、UART のスタート条件が検出されないことがあります

概要

デバイスが STANDBY1 モードのときに、UART 送信によって開始された非同期高速クロック要求を処理した後、デバイスは STANDBY1 モードに戻ります。STANDBY1 モードへの復帰中に別の UART 送信が開始されると、デバイスはそのデータを正しく検出および受信できません。

回避方法

UART のスタート条件が繰り返し発生することが想定される場合は、STANDBY0 モードまたはそれ以上の低消費電力モードを使用してください。

UART_ERR_02

UART モジュール

カテゴリ

機能

機能

TXE のみが有効な場合、UART 送信終了の割り込みは設定されません

概要

デバイスを送信のみに設定すると (CTL0.TXE = 1、CTL0.RXE = 0)、UART 送信終了 (EOT) 割り込みのトリガはかかりません。デバイスが送受信に設定されている場合 (CTL0.TXE = 1、CTL0.RXE = 1)、EOT は正常にトリガされます

UART_ERR_02 (続き)

UART モジュール

回避方法

UART 送信終了割り込みを使用するときは、CTL0.TXE ビットおよび CTL0.RXE ビットの両方を設定します。ピンを UART 受信として割り当てる必要はないので注意してください。

UART_ERR_03

UART モジュール

カテゴリ

機能

機能

3x オーバーサンプリングと MFCLK または BUSCLK をクロック ソースとして使用して、UART RX 割り込みが誤って設定されます

説明

UART および 3x オーバーサンプリングに BUSCLK または MFCLK を使用する場合、RXINT が誤って設定されます。BUSCLK または MFCLK をソースとして UART モジュールを使用し、3x オーバーサンプリング モードの場合、RX 割り込みが誤って設定される可能性があります。これらの条件下では、TX データも破損する可能性があります。

回避方法

BUSCLK または MFCLK を使用する場合は、より高いオーバーサンプリング レートを使用してください。3x オーバーサンプリングが必要な場合は、LFCLK を利用します。

UART_ERR_04

UART モジュール

カテゴリ

機能

機能

クロックが SYSOSC から LFOSC に遷移する際、高速クロック要求が無効になっていると、UART データが誤って受信される可能性があります

概要

シナリオ:
1.UART の機能クロックとして LFCLK が選択されます 2.3 倍オーバーサンプリングで構成された 9600 のボーレート 3.UART 高速クロック要求が無効になっている状態で、UART 受信転送中に ULPCLK が SYSOSC から LFOSC に切り替わると、1 ビットが誤って読み取られることがあります

回避方法

LPM モードで UART を使用する場合は、UART 高速クロック要求を有効にしてください。

UART_ERR_05

UART モジュール

カテゴリ

機能

機能

UART モジュールのデバッグ停止機能の制限

UART_ERR_05 (続き)

UART モジュール

概要

本来は既存のフレームを完了して停止することが期待されますが、すべての Tx FIFO 要素が送信されてから通信が停止します。

回避方法

デバッグ停止がアサートされた後は、データが TX FIFO に書き込まれないようにしてください。

UART_ERR_06

UART モジュール

カテゴリ

機能

機能

UART 9 ビットモードでの予期しない RTOUT/Busy/Async の動作

説明

UART 受信タイムアウト(RTOUT)は、マルチノード構成では正しく動作しません。この構成では、1 つの UART がコントローラとして動作し、他の UART ノードはペリフェラルとして機能し、各ペリフェラルは 9 ビット UART モードで異なるアドレスに設定されます。
最初の UART コントローラが UART ペリフェラル 1 と通信し、ペリフェラル 1 のアドレスを最初のバイトとして送信してからデータを送信することで、ペリフェラル 1 がアドレスの一致を確認してデータを受信しました。コントローラがペリフェラル 1 との通信を終了した後、バス上で異なるアドレスに構成された別の UART ペリフェラル(ペリフェラル 2)との通信を直ちに開始すると、ペリフェラル 1 は設定されたタイムアウト期間が経過しても RTOUT を設定しません。ペリフェラル 2 との通信中もペリフェラル 1 の RTOUT カウンタはリセットされ続け、RTOUT が設定されるのは、コントローラがペリフェラル 2 との通信を完了した後になります。
BUSY 要求と Async 要求で同様の動作が確認観察されました。コントローラがバス上の別のペリフェラルと通信中で、アドレスが一致しない場合でも、Busy および Async 要求が設定されます。

回避方法

1 つのコントローラが複数のペリフェラルに接続されたマルチノード UART 通信では、RTOUT／BUSY／非同期クロック要求の動作は使用しないでください。

UART_ERR_07

UART モジュール

カテゴリ

機能

機能

IDLE LINE モードにおいて、RTOUT カウンタが期待どおりにカウントされません

概要

UART のアイドルラインモードでは、ラインがアイドル状態で、FIFO に何らかの要素がある場合でも、RTOUT カウンタはスタックします。つまり、IDLE LINE モードでは RTOUT 割り込みは動作しません。
アドレスが一致しない場合、Rx ラインでトグルの発生を検出すると RTOUT カウンタがリロードされます。
マルチレスポンダ構成の場合、コマンダと他のレスポンダ間で通信が行われていると、RTOUT イベントの取得に不定の遅延が発生するおそれがあります。

UART_ERR_07 (続き)

UART モジュール

回避方法

UART モジュールを IDLLINE モード/マルチノード UART アプリケーションのいずれかで使用する場合、RTOUT 機能を有効にしないでください。

UART_ERR_08

UART モジュール

カテゴリ

機能

機能

STAT BUSY は、UART モジュールの正しいステータスを表していません

概要

UART モジュールが無効で TXFIFO でデータが利用可能である場合でも、STAT BUSY は High のままです。

回避方法

TXFIFO ステータスと CTL0.ENABLE レジスタビットをポーリングして、ビジーステータスを識別します。

UART_ERR_09

UART モジュール

カテゴリ

機能

機能

UART を低速で動作させている場合、UART ADDR_MATCH ビットが読み出し時点までに設定されないことがあります。

説明

アドレス一致割り込み中に、コードが ISR にジャンプして FIFO を読み取ります。アドレス一致割り込みがストップ ビットの前に発生するため、UART は RX ラインで送信されたアドレスとしてのデータを正しく受信できないことがあります。

回避方法

ADDR_MATCH ビットがセットされるのを確実にするために、データを読み出す前に 1 UART CLK サイクル分待機します。

UART_ERR_10

UART モジュール

カテゴリ

機能

機能

UART IrDA モードの BUSY ビットの設定が遅延する

説明

IrDA モードでは、UART.STAT.BUSY ビットは IrDA スタート パルスの 2 番目のエッジで設定されます。そのため、BUSY ステータスが正しくセットされる前に、1 ビット分の送信が完了してしまう可能性があります。この間にソフトウェアが BUSY ビットをポーリングすると、IrDA スタート パルス送信中にもかかわらず UART がビジーでないと誤って認識されることがあります。この BUSY ステータスの動作は UART のボーレートに依存します。UART 送信が遅い (ボーレートが低い) ほど、BUSY が正しく設定されるまでの遅延時間が長くなります。

UART_ERR_10 (続き)

UART モジュール

回避方法

BUSY ステータスをチェックする前に、1 ビット送信の時間分の遅延を挿入します。別の方法としては、UART.STAT.BUSY == 0x0 の後に UART.STAT.BUSY == 0x1 をチェックすることで、ボーレートや他の ISR に依存しない動的遅延を実現できます。

UART_ERR_11

UART モジュール

カテゴリ

機能

機能

UART 受信タイムアウトが、STOP ビット転送中に、予期したタイミングよりも早くカウントを開始する

説明

STOP ビット転送時に、受信タイムアウトが STOP ビット転送の途中でカウントを開始する場合があります。その結果、RXTOSEL の設定値が小さすぎる場合、意図しない RTOUT 割り込みが発生する可能性があります。たとえば、ボーレートが 1Mbps で、RXTOSEL が 1 に設定されている場合、想定される RTOUT は STOP ビット転送の 1µs 後に発生するはずですが、実際には RTOUT 割り込みが 0.5µs で設定されます。

回避方法

UART.IFLS.RXTOSEL レジスタは、受信タイムアウト (RTOUT) 割り込みが発生するまでのビット時間を選択します。早期割り込みを防止するには、RXTOSEL の値を 1 より大きくする必要があります。受信タイムアウト時間は次のように計算できます。受信タイムアウト = (RXTOSEL - 0.5) / ボーレート

7 改訂履歴

資料番号末尾の英字は改訂を表しています。その改訂履歴は英語版に準じています。

Changes from NOVEMBER 30, 2025 to JULY 30, 2026 (from Revision C (November 2025) to Revision D (July 2026)) **Page**

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、 TI の販売条件 、 TI の総合的な品質ガイドライン 、 ti.com または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。 TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

最終更新日：2025 年 10 月