

Errata

MSPM0L112x および MSPM0L211x マイコン



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

1 機能アドバイザリ.....2

2 プログラム済みのソフトウェア アドバイザリ.....3

3 デバッグ専用のアドバイザリ.....3

4 コンパイラ アドバイザリによって修正.....3

5 デバイスの命名規則.....4

 5.1 デバイスの記号表記とリビジョンの識別.....4

6 アドバイザリの説明.....6

7 改訂履歴.....22

商標

すべての商標は、それぞれの所有者に帰属します。

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
AES_ERR_01	✓
COMP_ERR_06	✓
CPU_ERR_02	✓
CPU_ERR_03	✓
FCC_ERR_01	✓
FLASH_ERR_03	✓
FLASH_ERR_04	✓
FLASH_ERR_05	✓
FLASH_ERR_08	✓
KEYSTORE_ERR_01	✓
PMCU_ERR_14	✓
RST_ERR_01	✓
RST_ERR_02	✓
SYSCTL_ERR_01	✓
SYSCTL_ERR_02	✓
SYSCTL_ERR_03	✓
SYSCTL_ERR_04	✓
SYSCTL_ERR_05	✓
SYSCTL_ERR_06	✓
SYSCTL_ERR_11	✓
SYSCTL_ERR_12	✓
SYSOSC_ERR_01	✓
SYSOSC_ERR_02	✓
SYSOSC_ERR_04	✓
SYSOSC_ERR_05	✓
TIMER_ERR_04	✓
TIMER_ERR_06	✓
TIMER_ERR_07	✓
UNICOMMI2CC_ERR_01	✓
UNICOMMI2CC_ERR_02	✓
UNICOMMI2CC_ERR_03	✓
UNICOMMSPI_ERR_02	✓
UNICOMMSPI_ERR_03	✓
UNICOMMSPI_ERR_04	✓
UNICOMMUART_ERR_01	✓
UNICOMMUART_ERR_02	✓
UNICOMMUART_ERR_03	✓
UNICOMMUART_ERR_04	✓
UNICOMMUART_ERR_05	✓
UNICOMMUART_ERR_06	✓
UNICOMMUART_ERR_07	✓
UNICOMMUART_ERR_09	✓

エラッタ番号	リビジョン A
UNICOMMUART_ERR_10	✓

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。

このカテゴリに該当するエラッタは、このデバイスには存在しません。

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。

このカテゴリに該当するエラッタは、このデバイスには存在しません。

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气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。

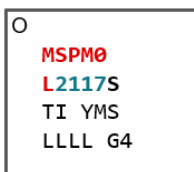
プロトタイプ デバイス (**XMS**) は、標準の量産デバイスよりも故障率が高いことが予想されます。これらのデバイスは、予測される最終使用時の故障率が未定義であるため、テキサス・インスツルメンツはそれらのデバイスを量産システムで使用しないよう推奨しています。認定済みの量産デバイスのみを使用する必要があります。

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同時に図 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。

48QFN (RGZ):



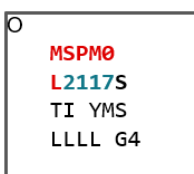
64QFP (PM):



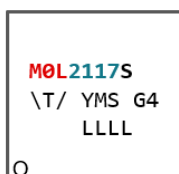
32VSSOP (DGS32):



32QFN (RHB):



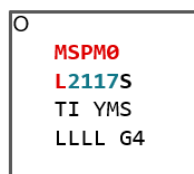
48QFP (PT):



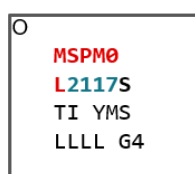
28VSSOP (DGS28):



24QFN (RGE):



28QFN (RUY):



TI = TI Letters
\T/ = TI LOGO
YM = Year Month Date
S = Site code
LLLL = Lot code
T = "Q" character

LLLL = Lot code
= Die revision
O = Pin 1 marked
G4 = ECAT
CAV = Cavity number

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	製品開発ステータス	バージョン(デバイスの工場出荷時定数メモリ内)
A	RTM	0

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ーには、リビジョン文字に基づいて、特定のバースに該当するか否かがマークされています。この文字は、デバイスのファクトリー定数メモリ (DEVICEID.VERSION フィールド) に格納されている整数値に対応しており、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を使用して、この値からリビジョンを確認できます。

注

DEVICEID.VERSION フィールドは、RTM (Release to Market) 後に製品ハードウェアのリビジョンが変更された際にインクリメントされます。

6 アドバイザリの説明

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_06 COMP モジュール

カテゴリ

機能

機能

出力が準備完了する前に SYSOSC を無効にすると、COMP 出力レディ信号および割り込みの生成が遅延する

説明

COMP をイネーブルにするか、その入力構成 (IPSEL/IMSEL) を変更すると、COMP 出力の準備ができた時点で OUTRDYIFG ビットがセットされるようになります。OUTRDYIFG は、出力の準備ができるまで、すべての COMP 割り込みの起動を阻止します。

COMP OUTRDYIFG ビットでは、SYSOSC が 4MHz 以上で設定される必要があります。SYSOSC を無効にすると、OUTRDYIFG は 2 LFCLK サイクル (約 60us) 遅延します。

回避方法

2 LFCLK サイクルの遅延を回避するには、SYSOSC を無効にする LPM に移行する前、またはクロックを LFCLK に切り替える前に、OUTRDYIFG ビットがセットされるまで待機してください。

- 1.コンパレータをイネーブルにする
- 2.必要に応じて IPSEL/IMSEL を設定する
- 3.OUTRDYIFG 割り込みを待つ
- 4.必要に応じて、低消費電力モード (LPM) に移行するか、クロックを LFCLK に変更します

CPU_ERR_02 CPU モジュール

カテゴリ

機能

機能

CPUSS のプリフェッチ無効化に関する制限

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。

CPU_ERR_02 (続き) CPU モジュール

回避方法

プリフェッチを無効にした後、SYSCTL 内のシャットダウン メモリ (SHUTDNSTORE) に対してメモリ アクセスを実行してください。これは、SYSCTL.SOCLOCK.SHUTDNSTORE0 で行えます。メモリ アクセスが完了すると、プリフェッチャが無効化されます。

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 メモリへのアクセス
```

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
```

FCC_ERR_01 FCC モジュール

カテゴリ

機能

機能

BUSCLK が LFCLK から供給されている場合、FCC が誤って動作する

説明

BUSCLK が LFCLK から供給される場合、FCC は期待どおりに動作しません。FCC 完了信号がアサートされないか、または誤った FCC 値が報告されます。

回避方法

回避方法はありません。

FLASH_ERR_03 *FLASH モジュール*

カテゴリ

機能

機能

2 待機状態のフラッシュ アクセスの直後に無効なブートコード領域へのアクセスが行われると、次のフラッシュ アクセスでも違反が発生する可能性があります

説明

2 待機状態が設定されている状態で、フラッシュ アクセスの直後に **BOOTCODE** 領域へのアクセスを行うと、その次のフラッシュ アクセスでも違反が発生する可能性があります。

回避方法

ブート フェーズ終了後は、ブートコード領域へのアクセスを行わないでください。そうしない場合、ブートコード違反の後に正しいフラッシュ アクセスを行うまでに、少なくとも 4 クロック サイクルの間隔を空ける必要があります。

FLASH_ERR_04 *FLASH モジュール*

カテゴリ

機能

機能

NONMAIN または **Factory** 領域でエラーが発生した場合、**SYSCTL_DEDERRADDR** に誤ったアドレスが報告されます

説明

FLASHDED エラーが発生すると、データの最上位バイト (**MSB**) が切り捨てられます。デバイスのメモリ制限では、最上位バイトは **MAIN** フラッシュの復帰アドレスに影響を与えません。**NONMAIN** フラッシュまたは **Factory** 領域の場合、**MSB** は **0x41xx.xxxx** である必要があります。

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。メインフラッシュ **DED** の場合、**SYSCTL_DEDERRADDR** をそのまま使用できます。

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** をプログラムされると、リセット値が変化する傾向があります。

FLASH_ERR_05

(続き)

FLASH モジュール

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

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. 0x40BFFFFFF ~ 0x41C00000 3. 0x41C007FF ~ 0x41C40000

回避方法

いいえ

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。

PMCU_ERR_14

PMCU モジュール

カテゴリ

機能

機能

GPIO によってトリガされた DMA 転送が、FCL をイネーブルにした低消費電力モードでウェークアップするまで保留される

説明

FCL がイネーブルで低消費電力モード (STOP2/STANDBY0/STANDBY1) の場合、GPIO からのイベント チャネルによって DMA がトリガされます。イベントまたは割り込みによってデバイスが

PMCU_ERR_14 (続き)

PMCU モジュール

ウェークアップされるまで、この DMA 送信は開始されません。ウェークアップする前に追加の消費電力が確認されます。

回避方法

GPIO イベントによってトリガされる DMA を使用する代わりに、ソフトウェアで DMA 転送を開始するには GPIO 割り込みを使用します。割り込み終了後、再度低消費電力モードに移行します。

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

電源オンイベント時、NRST の最初のデアサート (Low から High への遷移) 後、内部低周波数発振器 (LFOSC) がイネーブルになります。この NRST の非アサート後、250us (最小) から 1.5ms (最大) のウィンドウ内で、LFOSCGOOD 信号がハイレベルに遷移します。次の NRST Low パルスが発生し、そのデアサートが LFOSCGOOD の立ち上がりエッジと 200ns ウィンドウ内に一致した場合、デバイスは NRST デアサート イベントを認識できない可能性があります。その結果、デバイスはリセット状態のままで、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。このシナリオは、外部回路が NRST ロジックを制御している場合に発生する可能性があります。

回避方法

外部リセットコントローラが、最初の NRST デアサートから 1.5ms の間、NRST ラインを High (デアサート) に保持するようにします。

SYSCTL_ERR_01 SYSCTL モジュール

カテゴリ

機能

SYSCTL_ERR_01

(続き)

SYSCTL モジュール

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

いいえ

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

いいえ

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

説明

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

いいえ

SYSCTL_ERR_04

SYSCTL モジュール

カテゴリ

機能

機能

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

SYSCCTL_ERR_04

(続き)

SYSCCTL モジュール**説明**

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

回避方法

いいえ

SYSCCTL_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

SYSCCTL_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

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。

SYSCCTL_ERR_11 SYSCCTL モジュール**カテゴリ**

機能

機能

FCL がイネーブルでデバイスが RUN2/SLEEP2 モードの場合、予期しない BOR リセット サイクルが発生する

SYSCTL_ERR_11

(続き)

SYSCTL モジュール

説明

FCL がイネーブルでデバイスが RUN2/SLEEP2 モードで動作する場合、予期しない BOR が発生し、BOR1/2/3 に NMI が発生します

回避方法

BOR1/2/3 モニタを使用する場合は、FCL を有効にしてデバイスを RUN2 または SLEEP2 モードで動作させることは避けてください。

SYSCTL_ERR_12 SYSCTL モジュール

カテゴリ

機能

機能

FCL が有効な状態でデバイスが RUN2/SLEEP2 モードの場合、BOR1/2/3 発生後も BOR イベントがマスクされたままとなる

説明

RUN2 または SLEEP2 電源モードで FCL が有効になっている場合、BOR マスクロジックが想定どおりに動作せず、最初の BOR 発生後に続いて発生するすべての BOR イベントを検出できなくなる。ソフトウェアによって SYSOSC が再度有効化されるまで、BOR MASK はクリアされる。

回避方法

回避方法 1: BOR1/2/3 検出が必要な場合は、FCL を有効にせず、デバイスを RUN2/SLEEP2 モードにします。回避策 2: 初期マスキング期間が経過した後、BOR/NMI ハンドラで BOR マスクフラグを手動でクリアする。回避策 3: Sysosc の有効/無効を定期的に切り替えて BOR マスクフラグをクリアすることで、ソフトウェアベースの更新メカニズムを実装しま

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を受信されます。ASYNC 要求を受信すると、SYSOSC は有効になり、ulpclock 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 SYSOSC の精度が低下する

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 $\pm 3\%$ 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、4MHz の高調波で動作する他のペリフェラルによって発生する、システム内のノイズとの同期に起因します。

回避方法

システムからノイズを完全に除去する回避方法はありません。ただし、FCL ON モードを使用する場合、ペリフェラルが 4MHz の倍数ではない周波数で動作しているときに、劣化を最小限に抑えることができます。

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

SYSOSC_ERR_05

(続き)

SYSOSC モジュール

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。

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。

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 *TIMG* モジュール

カテゴリ

機能

機能

初期リピート カウンタの周期は、次のリピートより 1 回だけ少なくなる

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タ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。

UNICOMMI2CC_E RR_01

UNICOMMI2CC 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

BUSRTRUN/FRAME_START 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 2 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 BUSRTRUN/FRAME_START 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 I2C 機能クロック = 3 x clock_divider x (CPU_CLK / 選択したクロック ソース周波数)。たとえば、clock_divider が 8、クロック ソースが 4MHz(MFCLK)、CPU_CLK が 32MHz の場合: ソフトウェア遅延 = 3 x 8 x (32MHz / 4MHz) = 192 CPU サイクル

UNICOMMI2CC_E RR_02

UNICOMMI2CC モジュール

カテゴリ

機能

機能

I2C コントローラの ACKOEN ビットが有効になっている場合、RXDONE 割り込みが 2 回発生します

UNICOMMI2CC_E RR_02 (続き)

UNICOMMI2CC モジュール

説明

I2C コントローラの CTR.ACKONE ビットが有効になっている場合、ソフトウェアは CTR.ACK ビットに 0 または 1 を書き込むことで、ACK/NACK の送信を制御します。これは、ストップ状態になる前の RXDONE 割り込み内で実行できます。CTR.ACK ビットを書き込んだ後、I2C バス上に有効な ACK/NACK 信号が出現し、その後の I2C ストップ状態で追加の RXDONE 割り込みが発生します。

回避方法

CTR.ACK ビットを書き込む前に RXDONE 割り込みを無効にし、次の送信の前に手動で有効にしてください。

UNICOMMI2CC_E RR_03

UNICOMMI2CC モジュール

カテゴリ

機能

機能

I2C コントローラーのクロックストレッチ機能を有効にすると、SCL 周波数が低下する

説明

— 回避策として ETW 機能の使用を推奨しないでください。— 本 ERRATA の提案によって、本デバイスの当該リビジョンまたは以前のリビジョンにおける既存の他の ERRATA に影響が生じないことを確認してください。クロックストレッチ機能を有効 (CR.CLKSTRETCH = 1) にすると、SCL 周波数は設定値よりも低くなります。たとえば、I2C コントローラーのバス速度を 400kHz に設定した場合、クロックストレッチ機能を有効にすると、I2C コントローラーのバスクロックは 400kHz よりも低くなり、360kHz や 390kHz などになります。同様の現象は、100kHz や 1MHz など、他のバス速度でも発生します。

回避方法

I2C 通信機能への影響は大きくないため、クロックストレッチ機能を無効にすることは推奨しません。クロックストレッチ機能を無効にすると、I2C バスクロックは期待される速度になります。

UNICOMMSPI_ER R_02

UNICOMM SPI モジュール

カテゴリ

機能

機能

コントローラー側のビジー状態が正しく読み取られない

説明

SPI のビジー状態は TXFIFO のデータ要素に依存しないため、基本的に SPI TXFIFO にデータをロードしても、CLKSEL が低速クロック (LFCLK など) または CLKDIV > 1 の場合、ビジービットはすぐにハイになりません。SPI ソースクロックが低速であるため、TXFIFO へのデータロード後から実際に通信が開始されるまでの間、ステータスが誤って読み取られることになります。

回避方法

アプリケーションコードを作成する際に、TXFIFO の STATUS ビットが空かどうかをチェックする機能を追加できます。

**UNICOMMSPI_ER
R_03**
UNICOMM SPI モジュール

カテゴリ

機能

機能

CPU を使用して RXFIFO をクリアすると、2 回目に RTOUT がセットされない

説明

rxtimeout を有効にして SPI 通信を完了し、その後 RTOUT がセットされた状態で、CPU を使用して RXFIFO のデータを読み出し、IIDX を読み出して割り込みをクリアした場合、IP をリセットまたは無効化せずに、同じ送信を 2 回目に行うと、rxtimeout カウンタが「0」であり、RXFIFO が空でないにもかかわらず、RTOUT 割り込みがセットされない。

回避方法

CPU で直接 RXFIFO のデータを読み出す代わりに、DMA を使用して RXFIFO のデータを読み出してください。または、CPU で直接 RXFIFO のデータを読み出した後、UNICOMM SPI をリセットしてください。

**UNICOMMSPI_ER
R_04**
UNICOMM SPI モジュール

カテゴリ

機能

機能

DSS の不一致が発生した状態で前回転送後に再転送を行うと、RX に保存されたデータが正しくありません

説明

コントローラ側とペリフェラル側でデータパケットサイズ設定 (DSS) が異なる状態でデータを送信した後、RXCLR を実行して FIFO をクリアし、その後、正しいデータパケットサイズで再度データを送信しても、最初の SCLK パルスが発生した際に、以前のデータが RXFIFO に格納されてしまう

回避方法

コントローラ側とペリフェラル側のデータパケットサイズ設定 (DSS) が同じであることを確認してください。

**UNICOMMUART_E
RR_01**
UNICOMMUART モジュール

カテゴリ

機能

機能

IrDA モードでグリッチが発生した場合のデータ完全性の問題

説明

UNICOMM-UART は IrDA をサポートしていますが、ノイズの多い環境では制限があります。グリッチ フィルタがないため、データラインにグリッチが発生すると、IrDA モードではデータ整合性の問題が発生する可能性があります。IrDA はエッジ検出ロジックに依存しているため、これらのグリッチが有効なパルスとして誤認識され、その結果 FIFO に予期しないデータが入り、LTOUT の計算に問題が生じます。

UNICOMMUART_E

RR_01 (続き) *UNICOMMUART* モジュール

回避方法

いいえ

UNICOMMUART_E

RR_02 *UNICOMMUART* モジュール

カテゴリ

機能

機能

IDLELINE モードにおけるグリッチによる IDLE 期間検出の問題

説明

アイドル期間中にグリッチが発生すると、受信側のアイドル検出が乱され、その結果、後続のアドレス バイトを Rx 側が破棄してしまう可能性があります。重要なのは、この故障モードがタイミング依存である点です。アイドル期間に対して特定のタイミングでグリッチが挿入された場合にのみ、この問題が発生します。それで、アイドル期間内のすべてのグリッチが破損を引き起こすわけではありません。

回避方法

いいえ

UNICOMMUART_E

RR_03 *UNICOMMUART* モジュール

カテゴリ

機能

機能

マンチェスター モードでのグリッチによるデータ完全性の問題

説明

マンチェスター符号化されたデータ ストリーム中のグリッチにより、信号の正確なエッジ タイミングが乱され、誤った受信データやデータ整合性の問題が発生します。

回避方法

いいえ

UNICOMMUART_E

RR_04 *UNICOMMUART* モジュール

カテゴリ

機能

機能

LIN モードでは、ブレーク フィールドおよび / または同期フィールド中にグリッチが発生すると、データが欠落します

説明

ブレーク フィールドのシナリオ (ハイパルス グリッチ) : ブレーク フィールド期間中に負のエッジが発生すると、カウンタがゼロにリセットされ、ブレーク フィールドの検出に失敗し、その結果データが失われます。同期フィールドのシナリオ : 同期フィールド中にグリッチが発生すると、誤った LINC0/LINC1 割り込みがトリガされ、Rx ラインの負のエッジでカウンタがリセットされ、同期フィールドの検証に失敗し、その結果データが失われます。

UNICOMMUART_E

RR_04 (続き) *UNICOMMUART モジュール*

回避方法

いいえ

UNICOMMUART_E

RR_05 *UNICOMMUART モジュール*

カテゴリ

機能

機能

ラインがアイドル状態のときにグリッチが発生すると、LTOUT/RTOUT の期間計算に問題が生じます

説明

通常の UART モードでは、アイドル ライン状態中にグリッチが発生すると、LTOUT イベントが 1 ～2 ボー クロック分ずれて発生し、LTOUT のタイミングが乱されます。

回避方法

いいえ

UNICOMMUART_E

RR_06 *UNICOMMUART モジュール*

カテゴリ

機能

機能

ストップ ビットの長さが一致していないため、RTOUT/LTOUT 周期の計算に問題が発生します

説明

レシーバ側では、機能ステート マシンがストップ ビットの途中でストップ ビット状態からアイドル状態へ遷移するように設計されています。このため、RTOUT カウンタが本来開始されるべきタイミングよりも早くカウントを開始してしまいます。この結果、RTOUT の期間が不正確となり、RTOUT 割り込みが想定よりも早く発生します。

回避方法

RTOUT カウンタに、ストップ ビット半周期分の補正を追加します

UNICOMMUART_E

RR_07 *UNICOMMUART モジュール*

カテゴリ

機能

機能

RS-232 モードでは、UART が無効化されている場合、RTS ラインは HIGH になりません

説明

UART が無効化されている場合、RTS ラインはアイドル状態 (HIGH) に戻らず、LOW のまま固定されます。

回避方法

ソフトウェアを使用して内部プルアップ抵抗を有効にし、RTS ラインの IO を Hi-Z モードに設定します。

UNICOMMUART_E

RR_08

UNICOMMUART モジュール

カテゴリ

機能

機能

LTOUT 割り込みは、タイムアウトが期限切れになるたびに継続して発生します

説明

1 回目の LTOUT 割り込みが発生した後、カウンタは再スタートしてカウントを継続し、その間に Rx フレームが存在しなくても、再び LTOUT 割り込みを発生させます。その結果、タイムアウト満了間隔ごとに LTOUT 割り込みが連続して発生することになります。この動作は RTOUT の挙動とは異なります。RTOUT 割り込みは、新たな Rx トランザクションが行われるまで一度しか発生しません。この制約は、RTOUT と LTOUT の両イベントで同一のカウンタを共有して使用していることに起因しています。

回避方法

UNICOMMUART_E

RR_09

UNICOMMUART モジュール

カテゴリ

機能

機能

ISO-7816 スマートカード モードは、57MHz UARTCLK 未満の 9600 ボーレートをサポートできません

説明

ISO-7816 スマートカード モードで 9600 のボーレートを実現するには、以下の制限を考慮して、57MHz を超える UARTCLK 周波数が必要です。1.ISO-7816 規格では、1 ビットに 372 クロック サイクルが必要であると規定されています。2.MSPM0 ISO-7816 モードでは、オーバーサンプリング レート (OVS) は UART ペリフェラルで 16x に固定されています。最小 UARTCLK の計算式を以下に示します。必要な $\text{UARTCLK} = 9600 * 372 * 16 = 57.139\text{MHz}$

回避方法

いいえ

UNICOMMUART_E

RR_10

UNICOMMUART モジュール

カテゴリ

機能

機能

CLKDIV が /1 に設定されている場合にのみ、LIN レジスタにアクセスできます

説明

UNICOMMUART (UART + LIN) バリエーションでは、CLKDIV が 1 に設定されている場合にのみ、LIN レジスタの設定が有効に維持されます。CLKDIV 値が 1 以外の場合、これらの設定は破棄されます。

回避方法

いいえ

UNICOMMUART_E**RR_11****UNICOMMUART** モジュール**カテゴリ**

機能

機能

UNICOMMUART が無効化されており、かつ txfito にデータが存在する場合、STAT.BUSY ビットは high のままになります

説明

UNICOMMUART の電源が無効化されており、かつ txfito にデータが存在する場合、STAT.BUSY ビットは high のままになります。

回避方法

すべての UNICOMMUART レジスタを初期化するには、UNICOMMUART をリセットします。

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 1, 2025 to August 18, 2026 (from Revision A (Nov 2025) to Revision B (Aug 2026))
Page

• COMP_ERR_06 カテゴリを更新しました.....	6
• COMP_ERR_06 モジュールを更新しました.....	6
• COMP_ERR_06 機能を更新しました.....	6
• COMP_ERR_06 の説明を更新しました.....	6
• COMP_ERR_06 回避策を更新しました.....	6
• FCC_ERR_01 カテゴリを更新しました.....	7
• FCC_ERR_01 回避策を更新しました.....	7
• FCC_ERR_01 機能を更新しました.....	7
• FCC_ERR_01 の説明を更新しました.....	7
• FCC_ERR_01 モジュールを更新しました.....	7
• PMCU_ERR_14 カテゴリを更新しました.....	9
• PMCU_ERR_14 モジュールを更新しました.....	9
• PMCU_ERR_14 機能を更新しました.....	9
• PMCU_ERR_14 の説明を更新しました.....	9
• PMCU_ERR_14 回避策を更新しました.....	9
• RST_ERR_02 カテゴリを更新しました.....	10
• RST_ERR_02 モジュールを更新しました.....	10
• RST_ERR_02 機能を更新しました.....	10
• RST_ERR_02 の説明を更新しました.....	10
• RST_ERR_02 回避策を更新しました.....	10
• SYSCTL_ERR_05 モジュールを更新しました.....	12
• SYSCTL_ERR_05 機能を更新しました.....	12
• SYSCTL_ERR_05 の説明を更新しました.....	12
• SYSCTL_ERR_05 回避策を更新しました.....	12
• SYSCTL_ERR_06 カテゴリを更新しました.....	12
• SYSCTL_ERR_06 モジュールを更新しました.....	12
• SYSCTL_ERR_06 機能を更新しました.....	12
• SYSCTL_ERR_06 の説明を更新しました.....	12
• SYSCTL_ERR_06 回避策を更新しました.....	12
• SYSCTL_ERR_11 カテゴリを更新しました.....	12

• SYSCTL_ERR_11 モジュールを更新しました.....	12
• SYSCTL_ERR_11 機能を更新しました.....	12
• SYSCTL_ERR_11 回避策を更新しました.....	12
• SYSCTL_ERR_11 の説明を更新しました.....	12
• SYSCTL_ERR_12 カテゴリを更新しました.....	13
• SYSCTL_ERR_12 モジュールを更新しました.....	13
• SYSCTL_ERR_12 機能を更新しました.....	13
• SYSCTL_ERR_12 の説明を更新しました.....	13
• SYSCTL_ERR_12 回避策を更新しました.....	13
• SYSOSC_ERR_04 モジュールを更新しました.....	14
• SYSOSC_ERR_04 回避策を更新しました.....	14
• SYSOSC_ERR_04 機能を更新しました.....	14
• SYSOSC_ERR_04 の説明を更新しました.....	14
• SYSOSC_ERR_05 カテゴリを更新しました.....	14
• SYSOSC_ERR_05 モジュールを更新しました.....	14
• SYSOSC_ERR_05 機能を更新しました.....	14
• SYSOSC_ERR_05 回避策を更新しました.....	14
• SYSOSC_ERR_05 の説明を更新しました.....	14
• UNICOMMI2CC_ERR_02 カテゴリを更新しました.....	16
• UNICOMMI2CC_ERR_02 モジュールを更新しました.....	16
• UNICOMMI2CC_ERR_02 機能を更新しました.....	16
• UNICOMMI2CC_ERR_02 の説明を更新しました.....	16
• UNICOMMI2CC_ERR_02 回避策を更新しました.....	16
• UNICOMMI2CC_ERR_03 カテゴリを更新しました.....	17
• UNICOMMI2CC_ERR_03 モジュールを更新しました.....	17
• UNICOMMI2CC_ERR_03 機能を更新しました.....	17
• UNICOMMI2CC_ERR_03 の説明を更新しました.....	17
• UNICOMMI2CC_ERR_03 回避策を更新しました.....	17
• UNICOMMSPI_ERR_02 カテゴリを更新しました.....	17
• UNICOMMSPI_ERR_02 モジュールを更新しました.....	17
• UNICOMMSPI_ERR_02 機能を更新しました.....	17
• UNICOMMSPI_ERR_02 の説明を更新しました.....	17
• UNICOMMSPI_ERR_02 回避策を更新しました.....	17
• UNICOMMSPI_ERR_03 カテゴリを更新しました.....	18
• UNICOMMSPI_ERR_03 モジュールを更新しました.....	18
• UNICOMMSPI_ERR_03 機能を更新しました.....	18
• UNICOMMSPI_ERR_03 の説明を更新しました.....	18
• UNICOMMSPI_ERR_03 回避策を更新しました.....	18
• UNICOMMSPI_ERR_04 カテゴリを更新しました.....	18
• UNICOMMSPI_ERR_04 モジュールを更新しました.....	18
• UNICOMMSPI_ERR_04 機能を更新しました.....	18
• UNICOMMSPI_ERR_04 の説明を更新しました.....	18
• UNICOMMSPI_ERR_04 回避策を更新しました.....	18

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みいかなる保証も、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、TI は一切の責任を拒否します。

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。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月