参考資料

Errata

MSPM0L110x、MSPM0L13xx 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

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	Rev C	Rev D
ADC_ERR_01	✓	✓
ADC_ERR_02	✓	✓
ADC_ERR_03	✓	✓
ADC_ERR_04	✓	✓
ADC_ERR_05	✓	✓
ADC_ERR_06	✓	✓
COMP_ERR_01	✓	✓
COMP_EER_02	✓	✓
COMP_ERR_03	✓	✓
COMP_ERR_05	✓	✓
CPU_ERR_01	✓	✓
CPU_ERR_02	✓	✓
CPU_ERR_03	✓	✓
FLASH_ERR_02	✓	✓
FLASH_ERR_04	✓	✓
FLASH_ERR_05	✓	✓
FLASH_ERR_06	✓	✓
GPIO_ERR_01	✓	✓
GPIO_ERR_02	✓	✓
GPIO_ERR_03	✓	✓
I2C_ERR_01	✓	✓
I2C_ERR_02	✓	✓
I2C_ERR_03	✓	✓

エラッタ番号	Rev C	Rev D
I2C_ERR_04	✓	✓
I2C_ERR_05	✓	✓
I2C_ERR_06	✓	✓
I2C_ERR_07	✓	✓
I2C_ERR_08	✓	✓
I2C_ERR_09	✓	✓
I2C_ERR_10	✓	✓
PMCU_ERR_01	✓	✓
PMCU_ERR_02	✓	✓
PMCU_ERR_03	✓	✓
PMCU_ERR_13	✓	✓
PWREN_ERR_01	✓	✓
RST_ERR_01	✓	✓
SPI_ERR_01	✓	✓
SPI_ERR_03	✓	✓
SPI_ERR_04	✓	✓
SPI_ERR_05	✓	✓
SPI_ERR_06	✓	✓
SPI_ERR_07	✓	✓
SYSOSC_ERR_01	✓	✓
SYSOSC_ERR_02	✓	✓
TIMER_ERR_01	✓	✓
TIMER_ERR_04	✓	✓
TIMER_ERR_06	✓	✓
UART_ERR_01	✓	✓
UART_ERR_02	✓	✓
UART_ERR_04	✓	✓
UART_ERR_05	✓	✓
UART_ERR_06	✓	✓
UART_ERR_07	✓	✓
UART_ERR_08	✓	✓
UART_ERR_09	✓	✓
VREF_ERR_01	✓	✓

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。

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。テキサス・インスツルメンツ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。

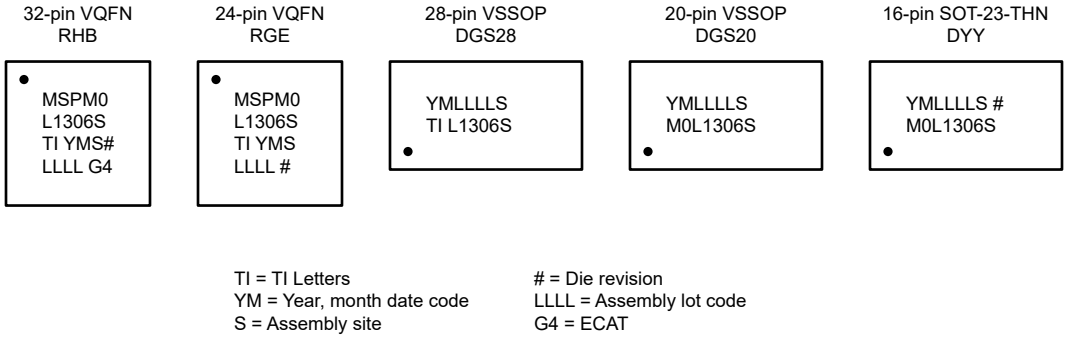

図 5-1. パッケージの記号表記

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)
C	1
D	1

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_01

ADC モジュール

カテゴリ

機能

機能

ADC は STANDBY1 モードでは高速クロックをトリガできません

説明

デバイスが STANDBY1 モードで動作している場合、ADC モジュールがイベント システム経由でトリガされた際 (たとえば、タイマなどのイベント パブリッシャが ADC のイベント サブスクライバ ポートを通じてイベントを生成した場合)、非同期の高速クロック要求が正しくアサートされないことがあります。

回避方法

イベント ファブリック経由で ADC 変換をトリガする場合は、STANDBY0 以上の電力モードを使用してください。

ADC_ERR_02

ADC モジュール

カテゴリ

機能

機能

ADC が反復モードのときの低消費電力モードでの電流の増加

説明

ADC が低電力モード (LPM) で EVENT トリガを使用し、リピート シーケンス モードまたはリピート シングル モードに設定されている場合、ADC モジュールは無効化されるまで追加の電流を消費し続けます。

回避方法

ADC をリピートなしのシングル モードまたはシーケンス モードに設定し、変換後に割り込みを使用して次のトリガに備えて ADC を再有効化します。

ADC_ERR_03

ADC モジュール

カテゴリ

機能

機能

ADC グランド基準に VSSA を使用する場合、ADC の SNR と DNL 性能が低下します。

説明

ADC がグランド基準として VSS を使用している場合、VSS グランドのノイズ源は ADC 結果に影響を及ぼし、SNR と DNL の性能が低下します。

回避方法

回避方法 1:ADC リファレンスには VREF+ ピンと VREF- ピンを使用します。VREF ピンを使用すると、ADC はアナログ グランド基準電圧を備えているため、外部ノイズ源の影響は減少します。回避方法 2:VSS のノイズ源を低減します。一般的な原因は CPU またはデジタル スイッチング ノイズです

ADC_ERR_04 *ADC モジュール*

カテゴリ

機能

機能

ADC が反復モードのときの低消費電力モードでの電流の増加

説明

ADC が低電力モード (LPM) で EVENT トリガを使用し、リピート シーケンス モードまたはリピート シングル モードに設定されている場合、ADC モジュールは無効化されるまで追加の電流を消費し続けます。

回避方法

ADC をリピートなしのシングル モードまたはシーケンス モードに設定し、変換後に割り込みを使用して次のトリガに備えて ADC を再有効化します。

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

概要

ADC を HW イベントトリガモードに構成されていて、ADC が有効になる前にトリガが生成されると、ADC トリガはキュー内にとどまります。ADC が有効になると、サンプリングおよび変換がトリガされます。

回避方法

ADC をハードウェア トリガ モードで設定した後、外部トリガを与える前に、まず ADC を有効にします。

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

ADC は、12 ビットモードにおいて最大で約 200 万回に 1 回の割合で誤差が発生する可能性があります。

変換エラーが発生すると、ADC の入力電圧に変化がないにもかかわらず、デジタル出力に +/-64LSB のランダムなジャンプが生じます。

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

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

ADC_ERR_06 (続き) *ADC モジュール*

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。

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。

COMP_ERR_01 *COMP モジュール*

カテゴリ

機能

機能

コンパレータ出力は、STANDBY0 モードでトグルを維持します

説明

コンパレータの反転入力マルチプレクサ (IMSEL) が 0 (COMP0_IN0−) に設定されており、かつデバイスの動作モードが STANDBY0 に設定されている場合、コンパレータの出力が入力電圧に関係なく予期せずトグルする可能性があります。

回避方法

STANDBY0 モードでコンパレータを使用し、負入力端子に外部電圧を印加する場合は、コンパレータのチャネル 1 (IMSEL = 1) を使用してください。
STANDBY0 モードでコンパレータを使用し、内部リファレンス電圧(DAC8、VREF など) を負の入力に適用する場合は、IMSEL を 0 以外の値に設定してください (IMSEL 0)。

COMP_ERR_02 *COMP モジュール*

カテゴリ

機能

機能

DACCODEx をヒステリシスに使用すると、COMP 出力がトグルします

説明

COMP モジュール内の 8 ビット DAC を COMP の入力として使用し、DAC の出力が DACCODEx に設定された値間で切り替わる場合、COMP の出力は、新しい基準値を直ちに上回ったかのようにトグル動作します。

これは、COMPx.CTL2.DACCTL ビットの設定に関係なく発生します。

この現象は、COMP モジュールにカスタムまたは非対称のヒステリシスを実装する目的で、2 つの DACCODEx コードを使用しているアプリケーションで最も一般的に見られます。

回避方法

COMPx.CTL1.HYST レジスタ ビットに設定されているヒステリシス値を利用します。

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_05 *COMP モジュール*

カテゴリ

機能

機能

コンパレータが有効になると、出力により立ち上がりおよび立ち下がり割り込みが設定されます

概要

コンパレータが有効なとき、コンパレータによって立ち上がりと立ち下がりが設定されします。

回避方法

1.ICLR ビットを使用して CPU 割り込みをクリアします。
ICLR は一般的イベントのクリアには機能しません。COMP の汎用イベントをクリアするには、以下の手順に従ってください（以下は DriverLib 関数です。当社の MSPM0 SDK の機能の内容を確認して、ビット操作を確認できます）
a. COMP を有効にする前に、COMP パブリッシャを何らかのダミー ID で構成します。
DL_COMP_setPublisherChanID(COMP_0_INST, 0); // 実際のパブリッシャを削除します
b. DL_COMP_enableEvent(COMP_0_INST, (DL_COMP_EVENT_OUTPUT_EDGE)); // IMASK で COMP イベントを有効にします
c. DL_COMP_enable(COMP_0_INST); // COMP モジュールを有効にします。この手順で RIS のイベントがクリアされます。
d. DL_COMP_disableEvent（COMP_0_INST、（DL_COMP_EVENT_OUTPUT_EDGE））;// IMASK におけるクリアで COMP イベントが無効になります。
DL_COMP_setPublisherChanID（COMP_0_INST、COMP_0_INST_PUB_CH）//実際のパブリッシャ f を構成。DL_COMP_enableEvent(COMP_0_INST, (DL_COMP_EVENT_OUTPUT_EDGE)); // IMASK で COMP イベントを再度有効化

または
コンパレータを有効後、割り込みを読み取り、コンパレータが有効になったために最初の割り込みが発生したことを検出します。

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

CPU_ERR_01 (続き) *CPU モジュール*

説明

メイン フラッシュ メモリと、NONMAIN や較正データ領域などの他の不揮発性メモリ領域との間でアクセスを切り替える際に、キャッシュの破損が発生する可能性があります。

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

概要

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

プリフェッチを無効にした後、SYSCTL のシャットダウンメモリ（SHUTDNSTORE）へのメモリアクセスを実行してください。これは SYSCTL->SOCLOCK.SHUTDNSTORE0 にアクセスすることで行えます。メモリアクセスが完了すると、プリフェッチャは無効化されます。

CPU_ERR_03 *CPU モジュール*

カテゴリ

機能

機能

プリフェッチャは、SLEEP モードへの遷移時にデータ整合性の問題を引き起こす可能性があります

概要

SLEEP0 に移行するとき、プリフェッチャで不正なデータ（すべて 0）が誤ってフェッチされることがあります。スリープモードから復帰したときに、プリフェッチャとキャッシュが ISR コードによって上書きされない場合、フラッシュからのメインコード実行が破損するおそれがあります。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。

回避方法

SLEEP に入る前にプリフェッチャーを無効にします。

FLASH_ERR_02 *FLASH モジュール*

カテゴリ

機能

機能

NONMAIN でのデバッグ無効は、デフォルトのパスワードで再度有効にできます

FLASH_ERR_02 (続き)

FLASH モジュール

説明

NONMAIN 構成（DEBUGACCESS = 0x5566）でデバッグが無効になっている場合でも、デバイスにはデフォルトのパスワードでアクセスできます。

回避方法

1. DEBUGACCESS を Debug Enabled with Password オプション（DEBUGACCESS = 0xCCDD）に設定し、PWDDEBUGLOCK フィールドに一意のパスワードを入力します。より高度のセキュリティを確保するために、暗号化されたランダムなデバイス固有のパスワードを使用することをお勧めします。これにより、適切な 128 ビットのパスワードでデバッグアクセスが可能になりますが、一部のデバッグコマンドで、CFG-AP と SEC-AP にアクセスすることもできます。
2. SWDP_MODE を無効にして、物理的な SW デバッグポートを完全に無効にします。これにより、デバイスへのデバッグアクセスや要求は完全に防止されますが、Failure Analysis やリターンフローに影響が出るおそれがあります。

FLASH_ERR_04

FLASH モジュール

カテゴリ

機能

機能

誤ったアドレスが SYSCTL -> DEDERRADDR で報告される

説明

FLASHDED エラーが表示されると、データの最上位バイト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 フラッシュまたは工場出荷時領域では、MSB は 0x41xx.xxxx と表記されます

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、実数アドレスは 0x41C4013C です。

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

概要

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_06 *フラッシュ モジュール*

カテゴリ

機能

機能

CPU と DMA は、同時にフラッシュにアクセスすることはできません

詳細

CPU と DMA はフラッシュメに同時にアクセスすることができません。これらが同時にアクセスすると、フラッシュから誤ったデータが読み出される可能性があります。

回避方法

CPU と DMA 経由で同時にフラッシュにアクセスすることはできません。通常のフラッシュ操作 (プログラム/ 消去/ 読み取り検証/ ブランク検証動作など) や DMA によるフラッシュからの読み出しを行う場合、ソフトウェアは、フラッシュがビジー状態の間に CPU がフラッシュにアクセスしないようにする必要があります。これを回避する方法としては、フラッシュ操作の実行中にコードを SRAM 上に配置するか、DMA が読み出す必要のあるデータをフラッシュ メモリから SRAM に移しておく方法があります。

GPIO_ERR_01 *GPIO モジュール*

カテゴリ

機能

機能

STANDBY モードでは、GPIO のウェイクアップ エッジが失われる可能性があります

説明

一度 GPIO のエッジでウェイクアップした後、STANDBY/STOP/SLEEP モードでは、その後の GPIO ウェイクアップ エッジが見逃される可能性があります。

ケース 1:
STANDBY0 ウェイクアップ: MCU が STANDBY/STOP/SLEEP モードに入り、ある IO が「ウェイク」状態にあるときに、IO を「非ウェイク」状態に 3 LFCLK サイクル未満の間だけ戻し、再度アサートした場合、次のウェイクアップ エッジは検出されない可能性があります。

ケース 2:
STANDBY1 ウェイクアップ: GPIO エッジを使ってウェイクアップした場合に、デバイスが STANDBY1 に戻る時点で GPIO パルスがまだアクティブであると、その後のウェイクアップ エッジは検出されません。

回避方法

ケース 1:
デバイスがアクティブ モードの間に GPIO をディアサートしておく
または GPIO のウェイクアップ パルスを 3 LFCLK サイクルより長くなるようにします
ケース 2:
GPIO ウェークアップ エッジを立ち下がりエッジと立ち上がりエッジの両方に設定する、または、STANDBY1 に入力する前に GPIO ウェークアップ パルスがアクティブでないことを確認します

GPIO_ERR_02 *GPIO モジュール*

カテゴリ

機能

GPIO_ERR_02 (続き)

GPIO モジュール

機能

PA2 ピンに負の電流が注入された後の大電流

説明

PA2 ピンに負の電流が注入されると、デバイスに予期しない持続的な大電流消費が発生する可能性があります。

この正誤表は、PA2 が ROSC モードで使用されている場合 (ROSC と VSS の間に 100kΩ の抵抗が実装されている場合) は有効ではありません。

回避方法

PA2 ピンの近くに抵抗を配置し、PA2 ピンと接続されている回路との間に、標準値 100Ω の直列抵抗を配置します。この抵抗は、PA2 ピンで観測される高速過渡を制限することを目的としており、この状況が発生するのを防止するのに十分です。

OR

PA2 ピンの使用は避け、代わりに代替ピンを使用してください。

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。

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。

I2C_ERR_01 (続き) ***I2C モジュール***

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_02 ***I2C モジュール***

カテゴリ

機能

機能

I2C クイック コマンド読み取りモードは、特定の条件でのみ機能します

説明

読み取りモードでの I2C のクイック コマンドは、TXFIFO に MSB が 1 に設定されたデータがあり、かつクロック ストレッチが無効の場合のみ動作します。

回避方法

MSB が 1 に設定されたダミー データを TXFIFO に入れ、クロック ストレッチ (CLKSTRETCH = 0) を無効にします。

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 で SDA が High の状態では、ターゲット I2C はストレッチを解除できません。

概要

1:SCL ラインを接地して解放し、デバイスは無制限に SCL を Low にプルします。

2:ポストクロックストレッチ、タイムアウト、解放。ライン上に別のクロック Low がある場合、本デバイスは無期限に SCL を Low にプルします。

I2C_ERR_04 (続き) ***I2C モジュール***

回避方法

I2C ターゲットアプリケーションで、非同期高速クロック要求を使用した低電力モードでのデータ受信が不要な場合は、SWUEN をデフォルトで無効にすることを推奨します (リセット時や電源サイクル時を含む)。この場合、バグの説明 1 と 2 は発生しません。

I2C ターゲットアプリケーションで、非同期高速クロック要求を使用した低電力モードでのデータ受信が必要な場合は、低電力モードへ移行する直前に SWUEN を有効にし、復帰後に SWUEN をクリアします。このシナリオでも、I2C ターゲットが低消費電力のときにバグ説明 1 および 2 が発生するおそれがあります。バス上の他のデバイスによって連続的なクロックストレッチングまたはタイムアウトが発生すると、SCL ラインが無期限にストレッチされます。この状況から回復するには、I2C ターゲットデバイスで Low タイムアウト割り込みを有効にし、低タイムアウト ISR 内で I2C モジュールをリセットして再初期化します。

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

説明

進行中の転送中に ACTIVE 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 USBUSY がセットされ (ACTIVE 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) 、BUSBUSY はクリアされません。これは、コントローラが STOP 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

I2C_ERR_06 (続き) *I2C モジュール*

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。

概要

CTR レジスタへの連続書き込みを行うと、次の CTR.START によって正しく開始条件が発生しない可能性があります。

回避方法

CTR.START を含むすべての CTR ビットは、1 回の書き込みでまとめて設定するか、CTR ビットの書き込み後に十分な待機時間を挟んでから CTR.START を書き込む必要があります。

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更新されない場合があります。

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

標準的な I2C 速度 (100kHz 未満) で動作している場合、割り込みを通過する読み出しの時間内に ADDRMATCH ビット (TSR レジスタのアドレス一致) が設定されないおそれがあります。

I2C_ERR_09 (続き) ***I2C モジュール***

回避方法

非標準的な I2C 速度で動作する場合、ADDRMATCH ビットを読み取る前に、少なくとも 1 つの I2C クロックサイクル分の遅延を入れてください。

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。任意の I2C 転送を必ず 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にしてください (低消費電力モードへの再移行に備えるため)。

PMCU_ERR_01 ***GPIO モジュール***

カテゴリ

機能

機能

PA2/ROSC ピンの電流リーク

説明

ROSC ではない機能構成でピン PA2/ROSC を使用する場合、ピンが high に駆動されていると、大きなリーク電流が観測されます。これは、約 17kΩ のインピーダンスを経由するグランドに意図しない接続が原因です。

このエラッタは、ROSC 機能には影響しません。

回避方法

ROSC の代替機能として PA2/ROSC を使用する場合、ピンが外部または内部のいずれかで high に駆動されているとき、エネルギー バジェットにおける追加のリーク電流を考慮する必要があります。

抵抗を使用してピンを外部でプルアップすると、このエラッタにより分圧回路が形成されるため、外部電圧は想定よりも低くなります。

PMCU_ERR_02 ***GPIO モジュール***

カテゴリ

機能

機能

デバイスのスタートアップ時の過渡電圧出力

PMCU_ERR_02 (続き)

GPIO モジュール

説明

ROSC ではない機能構成でピン PA2/ROSC を使用する場合、起動時に PA2/ROSC ピンに意図しない過渡電圧パルスが観測されることがあります。

このエラッタは、ROSC 機能には影響しません。

回避方法

回避策はありません。

PMCU_ERR_03

BOR モジュール

カテゴリ

機能

機能

BOR1、BOR2、BOR3 は、スタンバイ モードでは動作しません

説明

代替 BOR スレッショルド (BOR1、BOR2、BOR3) は、デバイスがスタンバイ モードで動作している場合、ユーザーが選択可能では機能しないため、これらのスレッショルドを超えるとデバイスは正しくリセットされません。

回避方法

スタンバイ モードで動作する場合は、BOR1、BOR2、BOR3 を使用しないでください。スタンバイ モードに移行する前に、デフォルトの BOR0 レベルを使用するようデバイスを構成します。スタンバイ モードを終了すると、代替 BOR レベルを再度イネーブルにすることができます。

PMCU_ERR_13

PMCU モジュール

カテゴリ

機能

機能

特定のシナリオにおいて、STOP2 または STANDBY0 からのウェークアップ時に MCU がスタックする可能性があります

概要

デバイスが STOP2 および STANDBY0 に遷移する前に、保留中のプリフェッチアクセスがある場合。タイマなどの保留中のプリフェッチアクセスが完了し、DMA が GPIO からのイベントを受信した直後のシナリオでは、DMA 転送もタイマの ISR 実行も行われず、CPU がスタック状態になります。この問題は、WFI 命令がハーフワードアライメント、デバイスのウェイト状態が 2 であり、デバイスが LPM に遷移する前に保留中のプリフェッチアクセスが存在するときに発生します。

回避方法

LPM に移行する前に、プリフェッチを無効にして、シャットダウンレジスタ読み取りまたはペリフェラル読み取りなどのいくつかのダミー命令を実行することができます。これにより、プリフェッチアクセスが無効になり、LPM からのウェークアップ時にデバイスがハングすることを防止できます。

PWREN_ERR_01 *GPIO モジュール*

カテゴリ

機能

機能

PWREN レジスタを無効にした後でも、ペリフェラル レジスタには引き続きアクセス可能な状態となります

説明

PWREN レジスタを 0 に設定してペリフェラルの電源を無効にした場合でも、ペリフェラルのレジスタは読み出すとデータ値を保持しているように見えることがあります。PWREN が 0 の場合にレジスタを読み書きしても、ペリフェラルは動作していないため、影響はありません。

影響を受けるペリフェラル: コンパレータ (COMP)、オペアンプ(OPA)、タイマ A、タイマ G、汎用入出力 (GPIO)、ウィンドウ付きウォッチドッグ タイマ (WWDT)。

回避方法

ペリフェラルの PWREN レジスタが 0 に設定されている場合、ペリフェラルに関連するレジスタの値は無視するか、無効なものとして扱う必要があります。

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。

SPI_ERR_01 *SPI モジュール*

カテゴリ

機能

機能

SPI パリティ ビットは機能しません

説明

SPI ハードウェア パリティ モードは機能しません。

回避方法

SPI モジュールの CTL1 レジスタ内にある PTEN ビットまたは PREN ビットによってハードウェア パリティを有効にしないでください。パリティの計算とチェックは、アプリケーション ソフトウェアを使用して実装できます。

SPI_ERR_03 *SPI モジュール*

カテゴリ

機能

機能

マルチペリフェラル アプリケーションで周辺機器として構成された場合、受信データに右シフトが生じることがあります

説明

マルチペリフェラルのシナリオでは、SPI コントローラは最初にペリフェラル 0 と通信し、その後ペリフェラル 1 と通信します。ペリフェラル 1 との通信が終了した後、コントローラは再びペリフェラル 0 との通信を行います。2 回目のペリフェラル 0 との通信時、最初のフレームで受信データに右シフトが発生する可能性があります。コントローラがデータ 0x76 を送信すると、ペリフェラル 0 は最初のデータを 0x3B として取得します。

回避方法

マルチペリフェラルのシナリオをサポートするには、アクティブな通信が行われていない間 (ペリフェラルの CS が無効になっている間) に、ペリフェラル側で CSCLR を有効にして、RX および TX のビット カウンタをリセットする必要があります。

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

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できません

概要

SPh = 0 で、デバイスが SPI ペリフェラルとして構成されている場合：読み取り要求の発行中に TXFIFO への書き込みが発生すると、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を生成できません。

回避方法

コントローラでペリフェラルがアドレス指定されているとき、ペリフェラルの TXFIFO が空にならないことを確認する必要があります。さらに、CRC などのデータチェック戦略で、パケットが正しく送信されたことを確認できます。

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

概要

MFCLK が有効で、SYSOSC が周波数補正ループ（FCL）モードを使用しており、STOP1 の低消費電力動作モードを使用している場合、SYSOSC が 4MHz から 32MHz に戻るとき（STOP1 から RUN モードへの終了時、または SYSOSC を 32MHz に強制的に強制的に印加する非同期高速クロック要求時のいずれか）、MFCLK がは 2 サイクルドリフトすることがあります。

回避方法

STOP1 モードの代わりに STOP0 モードを使用してください。STOP0 モードでは MFCLK のドリフトは発生しません。

SYSOSC_ERR_01 *SYSOSC モジュール*
(続き)

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

概要

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。
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。

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 *TIMG 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

概要

GP TIMER をワンショット モードで使用し、CLKDIV.RATIO が 0 でない場合、ゼロ イベント直前に TIMER を再有効化すると、再有効化が失われることがあります。

回避方法

タイマは、最初に再度イネーブルにする前に無効にできます。

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。

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

UART_ERR_02 (続き)

UART モジュール

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

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意してください。

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

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

UART_ERR_06 (続き)

UART モジュール

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。

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。

UART_ERR_08 (続き)

UART モジュール

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。

VREF_ERR_01

VREF モジュール

カテゴリ

機能

機能

VREF を無効化した後、VREF READY ビットはクリアされません

説明

SYSRST 後に VREF モジュールを初めて有効にする際は、VREF READY ビットを本来の機能として使用することができます。アプリケーション内で VREF モジュールを無効にした場合でも、VREF READY ビットはクリアされません。このエラッタの結果として、VREF モジュールを再度有効にした際には、VREF モジュールの安定性を示すために VREF READY ビットを使用することができません。

回避方法

アプリケーション内で VREF モジュールを再度有効にする場合は、VREF モジュールを使用する前に、TIMER モジュールを用いて最大の VREF 立ち上がり時間待機してください。VREF の立ち上がり時間については、デバイスのデータシートを参照してください。

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 Revision C (July 2025) to Revision D (August 2025)	Page
• リビジョン C と同じ正誤表の項目を持つ最新のシリコンリビジョン D を追加	1

重要なお知らせと免責事項

テキサス・インスツルメンツは、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、テキサス・インスツルメンツ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 テキサス・インスツルメンツ製品の選定、(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 テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、テキサス・インスツルメンツ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。テキサス・インスツルメンツ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、テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、テキサス・インスツルメンツの販売条件、または ti.com やかかる テキサス・インスツルメンツ製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。テキサス・インスツルメンツがこれらのリソースを提供することは、適用されるテキサス・インスツルメンツ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。

お客様がいかなる追加条項または代替条項を提案した場合でも、テキサス・インスツルメンツはそれらに異議を唱え、拒否します。

郵送先住所:Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

重要なお知らせと免責事項

テキサス・インスツルメンツは、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、 テキサス・インスツルメンツ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 テキサス・インスツルメンツ製品の選定、(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 テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、 テキサス・インスツルメンツ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。 テキサス・インスツルメンツ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、 テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、 テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、 テキサス・インスツルメンツの販売条件、または ti.com やかかる テキサス・インスツルメンツ製品の関連資料などのいずれかを通じて提供する適用可能な条項の下で提供されています。 テキサス・インスツルメンツがこれらのリソースを提供することは、適用される テキサス・インスツルメンツ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。

お客様がいかなる追加条項または代替条項を提案した場合でも、 テキサス・インスツルメンツはそれらに異議を唱え、拒否します。

郵送先住所：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265