参考資料

Application Note

MSPM0Gx51x から MSPM33C32xx へのソフトウェア移行ガイド

Eric Rentschler

概要

MSPM0Gx51x と MSPM33C32xx では、多くのパッケージ、機能、ピン配置が共通です。そのため、同じレイアウトを使用して MSPM0Gx51x と MSPM33C32xx の両方に対応できます。このアプリケーション ノートでは、デバイス間で移行するときのソフトウェアに関する検討事項について説明します。ハードウェア移行ガイド (SLAAES2) も使用できます。

目次

商標

1 はじめに

本ドキュメントでは、MSPM0Gx51x と MSPM33C32xx 間で移行するときのソフトウェアに関する検討事項について説明します。

最新のマイコンには、ますます多くの機能やペリフェラルが搭載されています。このため、デバイス間の移行は困難な場合があります。プロセスを簡素化するため、このドキュメントでは MSPM0Gx51x と MSPM33C32xx デバイス ファミリの違いを説明しますが、すべての M0 および M33 デバイスに同じ原理を適用できます

このガイドは、移行または設計プロセスの前に使用すると最も効果的です。MSPM0Gx51x と MSPM33C32xx 間の互換性に関する検討事項の各項目に対処するには、以下の手順に従ってください。両方のデータシートを用意しておくと、移行プロセスが円滑に進みます。

1. ターゲットの MSPM0Gx51x デバイスとターゲットの MSPM33C32xx デバイスを選択します
2. 機能レベルの違いを確認します
3. ピン配置の違いを確認します

2 MSPM0Gx51x と MSPM33C32xx のソフトウェアの相違点に関するレビュー

MSPM0Gx51x ファミリと MSPM33C32x ファミリは、いくつかの機能と仕様が共通ですが、以下のセクションでは、ソフトウェアの移行に影響がある相違点について説明します。

2.1 開発環境

ソフトウェア開発ツール

MSP 製品用のソフトウェアを開発するには、以下のツールを推奨します。

MSPM33 ソフトウェア開発キット:MSPM33SDK
Code Composer Studio
サンプル コード
ドライバ ライブラリ
SysConfig
CCS 統合 SysConfig
デバッグ インターフェイス

2.2 ソフトウェア ポーティング フロー

以下の図に、MSPM0Gx51x から MSPM33C クラスのデバイスへのソフトウェア移行の手順を示します。次のセクションでは、SLAAES2 と併せて、導入環境間の違いの詳細、およびペリフェラルの相違点の概要について説明します。

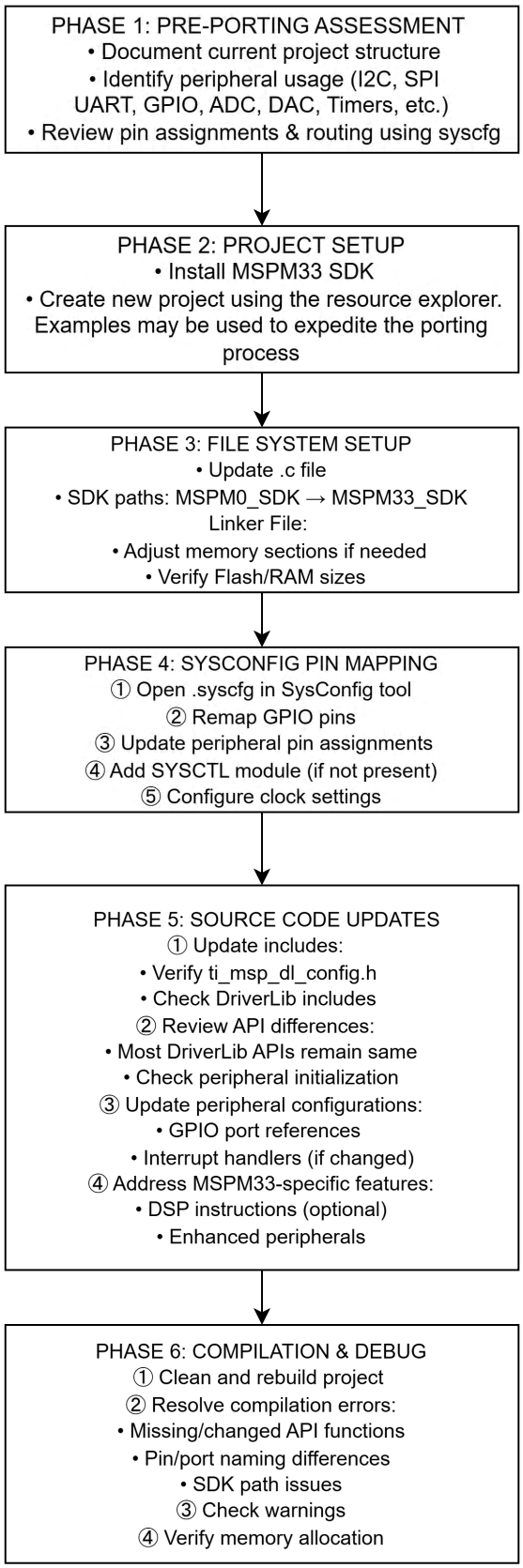

図 2-1. ソフトウェア ポーティング フロー図

このフロー チャートは、移行プロセスについて説明しています。SDK は類似していますが、エンジニアは、各ステップで性能を検証できるように、一度に 1 つのサブシステムを実装することをお勧めします。

2.3 ペリフェラルの相違点

次のセクションでは、各ペリフェラルの移行に関する検討事項について説明します。

2.3.1 割り込みの相違点

MSPM0SDK と MSPM33SDK の最も大きな違いの 1 つは、割り込みに関するフォーマットです。MSPM0G ファミリでは、デバイス割り込みは 32 個に制限されていますが、M33 コアでは最大 496 個の割り込みに対応しており、各ペリフェラルには専用の割り込みベクタが割り当てられています。これは、以前 GROUP_1_IRQHandler によって処理されていた GPIOA 割り込みが、代わりに GPIOA_IRQHandler によって処理されるということです。また、グループに関連付けられているステータス API はすべて適用されなくなり、ペリフェラル固有の API に移行する必要があります。

ほとんどのアプリケーションでは、割り込みが、もっとも大きな変更を行う必要のあるものの 1 つです。ただし、この変更は単純なものです。割り込み要求 (IRQ) が、内部に含まれるすべてのペリフェラルに対して 1 つの大きなグループを使用するのではなく、ペリフェラル自体に移管されているためです。

2.3.2 ADC の相違点

MSPM33C ファミリの ADC ペリフェラルには、MSPM0x51x の ADC12 と比較して、いくつかの顕著な変更があります。どちらも 12 ビット逐次比較型 ADC ペリフェラルですが、MSPM33C の高速 A/D コンバータ (HSADC) には、追加のシーケンシング、オーバーサンプリング、および後処理機能が搭載されており、このデバイスに移行するときに考慮する必要があります。

次については、命名規則を除いて、最小限の変更にとどめることができます。

- 基本チャネルの選択
- パワー マネージメント (有効化 / 無効化 / リセット)
- 基本割り込み
- クロック ソースの選択

次については、より大幅な再設計が必要になります。

- 変換シーケンシング
- ハードウェア平均化
- 結果の処理
- 割り込み構成
- トリガ

以下の図は、2 つのチップで各機能がどのように異なるかを示しています。移植するコードがこれらの機能を利用している場合は、SysConfig またはアプリケーション コード内の定義で、多くの定数を変更する必要があります。

ペリフェラルの仕様	ADC12 (MSPM0Gx51x)	HSADC (MSPM33Cx)
変換モード	シングル サンプル、リピート シングル サンプル、シーケンス、リピート シーケンス	優先度とプリエンプションに対応する 4 つの独立したシーケンサ (SEQ1-4)
トリガ	DL_ADC12_TRIG_SRC_SOFTWARE、DL_ADC12_TRIG_SRC_EVENT でトリガ	DL_HSADC_Trigger_TieLow_SW_Trig、DL_HSADC_Trigger_GEN_SUB_X を使用してシーケンサごとにトリガ
後処理	基本的なハードウェア平均化 (2 ～ 128 サンプル)、SysConfig を介した簡易な除算オプション	オフセット減算、2 の補数、設定可能なオーバーサンプリング、右シフト、トリップ コンパレータ、誤差計算機能を備えた 4 つの後処理ブロック (PPB)
結果の格納	12 個の MEMRES レジスタ (MEMRES0 ～ 11)、シンプル FIFO モード	16 個の結果レジスタ (ADCRESULT0 ～ 15)、4 個の PPB 結果レジスタ、シーケンサごとのステータス付き FIFO、オーバーサンプリング合計 / カウント レジスタ

ペリフェラルの仕様	ADC12 (MSPM0Gx51x)	HSADC (MSPM33Cx)
割り込み	DL_ADC12_INTERRUPT_MEM0_RESULT_LOADED、DL_ADC12_INTERRUPT_OVERFLOW、DL_ADC12_INTERRUPT_WINDOW_COMP_HI/LO	4 つの設定可能な割り込み、4 つの DMA 割り込み、ブロックごとの PPB トリップ割り込み、プリエンプション付きの早期割り込み生成
サンプル タイマ	2 個のサンプル タイマ (SCOMP0/1)、サンプリング時間 = SCOMP + 1 * CLK_DIV	シーケンサごとのサンプル ウィンドウ (1 ～ 1472 サイクル)、VrefLo と VrefHi を使用した簡易なキャパシタ リセット オプション
クロック構成	DL_ADC12_CLOCK_DIVIDE_1 ～ DL_ADC12_CLOCK_DIVIDE 48、合計 48 の分周オプション	DL_HSADC_CLOCK_DIVIDE_1_0、DL_HSADC_CLOCK_DIVIDE_2_0、DL_HSADC_CLOCK_DIVIDE_2_5...DL_HSADC_CLOCK_DIVIDE_8_5、1 ～ 8.5
チャネル	DL_ADC12_INPUT_CHAN_0～DL_ADC12_INPUT_CHAN_31	DL_HSADC_IN_0～DL_HSADC_IN_31

2.3.3 セキュリティの相違点

MSPM0Gx51x と MSPM33Cx の間のセキュリティ機能の多くは、互換性のあるラッパーを備えており、実質的に同一です。以下に、各セキュリティ ペリフェラルで必要なラッパーの変更を示します。

AESADV 機能	MSPM0SDK	MSPM33SDK
命名規則	DL_AESADV_* AESADV_REGS	「AESADV」の末尾に HP が追加されていることを除いて、すべての関数と構造は同じです DL_AESADVHP_* AESADVHP_Regs
割り込みレジスタの構造	• aesadv->CPU_INT.IMASK • aesadv->CPU_INT.RIS • aesadv->CPU_INT.MIS • aesadv->CPU_INT.ICLR • aesadv->CPU_INT.IIDX • aesadv->DMA_TRIG_DATAIN.IMASK • aesadv->DMA_TRIG_DATAOUT.IMASK	• aesadv->INT_EVENT0.IMASK • aesadv->INT_EVENT0.RIS • aesadv->INT_EVENT0.MIS • aesadv->INT_EVENT0.ICLR • aesadv->INT_EVENT0.IIDX • aesadv->INT_EVENT1.IMASK • aesadv->INT_EVENT2.IMASK

2.4 通信ペリフェラル

MSPM0Gx51x では、I2C、UART、SPI は個々のペリフェラルとして実装されますが、MSPM33Cx では、UNICOMM を通じて実装されています。各ペリフェラルの操作は類似していますが、ラッパーの実装は異なり、MSPM33Cx に移行するときに、UNICOMM 対応ペリフェラルの実装をすべて変更する必要があります。以下に、MSPM0SDK と MSPM33SDK のラッパーの相違点を列挙します。

2.4.1 SPI の相違点

SPI ペリフェラル機能	SPI (MSPM0SDK)	UNICOMM SPI (MSPM33SDK)
ラッパー	SPI_Regs を介してレジスタに直接アクセスします	SPI は UNICOMM にラップされており、UNICOMM_Inst_Regs->spi を使用してアクセスします
パワー マネージメント	DL_SPI_enablePower(SPI0)	DL_SPI_enablePower(UNICOMM0)
クロック ソース	• DL_SPI_CLOCK_BUSCLK • DL_SPI_CLOCK_MFCLK • DL_SPI_CLOCK_LFCLK	• DL_SPI_CLOCK_BUSCLK • DL_SPI_CLOCK_MFCLK • DL_SPI_CLOCK_LFCLK • DL_SPI_CLOCK_ASYNC_LFCLK • DL_SPI_CLOCK_ASYNC_SYSCLK • DL_SPI_CLOCK_ASYNC_HFCLK • DL_SPI_CLOCK_ASYNC_PLL
FIFO スレッショルド レベル	• DL_SPI_yX_FIFO_LEVEL_3_4_EMPTY • DL_SPI_yX_FIFO_LEVEL_1_2_EMPTY • DL_SPI_yX_FIFO_LEVEL_1_4_EMPTY • DL_SPI_yX_FIFO_LEVEL_ONE_ENTRY • DL_SPI_yX_FIFO_LEVEL_EMPTY	• DL_SPI_yX_FIFO_LEVEL_3_4_EMPTY • DL_SPI_yX_FIFO_LEVEL_1_2_EMPTY • DL_SPI_yX_FIFO_LEVEL_1_4_EMPTY • DL_SPI_yX_FIFO_LEVEL_ONE_ENTRY • DL_SPI_yX_FIFO_LEVEL_EMPTY • DL_SPI_yX_FIFO_LEVEL_ALMOST_EMPTY • DL_SPI_yX_FIFO_LEVEL_ALMOST_FULL
プリエンプションの追加	MSPM0 にプリエンプションはありません	• DL_SPI_INTERRUPT_DMA_PREIRQ_RX • DL_SPI_INTERRUPT_DMA_PREIRQ_TX • DL_SPI_INTERRUPT_LTOUT • DL_SPI_IIDX_PREIRQ_RX • DL_SPI_IIDX_PREIRQ_TX • DL_SPI_IIDX_LTOUT
ステータス レジスタの追加		• DL_SPI_getCurrentCDMODEValue(unicomm) • DL_SPI_isTXFIFOCleared(unicomm) • DL_SPI_isRXFIFOCleared(unicomm)
中断 / 再開		• DL_SPI_enabledSuspend(unicomm) • DL_SPI_disableSuspend(unicomm)
パッキング機能	• DL_SPI_enablePacking(spi) • DL_SPI_disablePacking(spi) • DL_SPI_isPackingEnabled(spi)	パッキング機能は削除されており、すべての 32 ビット操作はパッキングの有効化なしで動作します

2.4.2 UART の相違点

UART ペリフェラル機能	UART (MSPM0SDK)	UNICOMM UART (MSPM33SDK)
ラッパー	UART_REGS 経由でアクセスされるスタンドアロン UART ペリフェラル	UNICOMM_Inst_regs->uart
クロック ソースの拡張	• DL_UART_CLOCK_BUSCLK • DL_UART_CLOCK_MFCLK • DL_UART_CLOCK_LFCLK	• DL_UART_CLOCK_BUSCLK • DL_UART_CLOCK_MFCLK • DL_UART_CLOCK_LFCLK • DL_UART_CLOCK_ASYNC_LFCLK • DL_UART_CLOCK_ASYNC_SYSCLK • DL_UART_CLOCK_ASYNC_HFCLK • DL_UART_CLOCK_ASYNC_PLL
API 互換性マクロ		高い後方互換性、すべての DL_UART_MAIN_* および DL_UART_EXTEND_* API が Unicomm にマップされます
FIFO 管理	• DL_UART_enableFIFOs() • DL_UART_disableFIFOs()	FIFO は常に有効になっていますが、後方互換性を確保するための API が引き続き提供されます。たとえば、DL_UART_isFIFOsENABLED() を呼び出すと、常に true が返されます
FIFO スレッショルド レベル	• DL_UART_yX_FIFO_LEVEL_3_4_EMPTY • DL_UART_yX_FIFO_LEVEL_1_2_EMPTY • DL_UART_yX_FIFO_LEVEL_1_4_EMPTY • DL_UART_yX_FIFO_LEVEL_ONE_ENTRY • DL_UART_yX_FIFO_LEVEL_EMPTY	• DL_UART_yX_FIFO_LEVEL_3_4_EMPTY • DL_UART_yX_FIFO_LEVEL_1_2_EMPTY • DL_UART_yX_FIFO_LEVEL_1_4_EMPTY • DL_UART_yX_FIFO_LEVEL_ONE_ENTRY • DL_UART_yX_FIFO_LEVEL_EMPTY • DL_UART_yX_FIFO_LEVEL_ALMOST_EMPTY • DL_UART_yX_FIFO_LEVEL_ALMOST_FULL
プリエンプションの追加	MSPM0Gx51x には互換性のある機能はありません	• DL_UART_IIDX_PREEMPT_INT_TX • DL_UART_IIDX_PREEMPT_INT_RX • DL_UART_INTERRUPT_LINE_TIMEOUT • DL_UART_IIDX_LINE_TIMEOUT
レジスタの命名	• 制御: CTLx • ライン制御:LCRH • ステータス:STAT • TX データ:TXDATA • RX データ:RXDATA • ボーレート: IBRD/FBRD	• 制御:CTLx • ライン制御:LCRH • ステータス:STAT • TX データ:TXDATA • RX データ:RXDATA • ボーレート:IBRD/FBRD

UART ペリフェラル機能	UART (MSPM0SDK)	UNICOMM UART (MSPM33SDK)
パワー マネージメント	• DL_UART_enablePower(UARTx)	• DL_UART_enablePower(UNICOMM 0) • DL_UNICOMM_enablePower(unicomm)
バックアップ / 復元の構成	• DL_UART_Main_backupConfig • DL_UART_Extend_backupConfig • DL_UART_Main_saveConfiguration() • DL_UART_Extend_saveConfiguration()	• DL_UART_backupConfig • DL_UART_saveConfiguration() • DL_UART_restoreConfiguration()
アナログ グリッチ フィルタ		MSPM33C にアナログ フィルタはありません。
言及されていないコア機能	UART_Regs_* を使用	UNICOMM_Inst_regs_* を使用
API 互換性マクロ		高い後方互換性、すべての DL_UART_MAIN_* および DL_UART_EXTEND_* API が Unicomm にマップされます

2.4.3 I2C の相違点

I2C ペリフェラルの機能	I2C (MSPM0SDK)	UNICOMM I2C (MSPM33SDK)
ラッパー	I2C_Regs を介したレジスタへのダイレクト アクセス、同一のペリフェラルでのコントローラ モードとターゲット モードのサポート	UNICOMM モジュールでラップされた I2C コントローラ、UNICOMM_Inst_Regs->i2cc/i2ct 経由でアクセスするレジスタ
モード選択	同一ペリフェラル内に I2C コントローラおよびターゲット: DL_I2C_enableController(I2Cx)、DL_I2C_enableTarget(I2Cx)	• コントローラの有効化: DL_I2CC_ENABLE(UNICOMMx) • ターゲット有効化: DL_I2CT_ENABLE(UNICOMMx)
レジスタ名の変更	• 制御レジスタ: MCTR • ターゲット アドレス:MSA • ステータス レジスタ:MSR • タイマ周期:MTPR • Configuration (構成):MCR	• 制御レジスタ:CTR • ターゲット アドレス:SA • ステータス レジスタ:SR • タイマ周期:TPR • Configuration (構成):CR
クロック・ソース	• DL_I2C_CLOCK_BUSCLK • DL_I2C_CLOCK_MFCLK	• DL_I2Cx_CLOCK_BUSCLK • DL_I2Cx_CLOCK_MFCLK • DL_I2Cx_CLOCK_ASYNC_SYSCLK • DL_I2Cx_CLOCK_ASYNC_HFCLK • DL_I2Cx_CLOCK_ASYNC_PLL 「I2Cx」は、コントローラでは I2CC、ターゲットでは I2CT を指します

I2C ペリフェラルの機能	I2C (MSPM0SDK)	UNICOMM I2C (MSPM33SDK)
FIFO スレッショルド レベル	• DL_I2C_TX_FIFO_LEVEL_EMPTY • DL_I2C_TX_FIFO_LEVEL_BYTES_x • DL_I2C_RX_FIFO_LEVEL_BYTES_x LEVEL_BYTES_x:1 ～ 7	• DL_I2Cx_yX_FIFO_LEVEL_3_4_EMPTY • DL_I2Cx_yX_FIFO_LEVEL_1_2_EMPTY • DL_I2Cx_yX_FIFO_LEVEL_1_4_EMPTY • DL_I2Cx_yX_FIFO_LEVEL_ONE_ENTRY • DL_I2Cx_yX_FIFO_LEVEL_EMPTY • DL_I2Cx_yX_FIFO_LEVEL_ALMOST_EMPTY • DL_I2Cx_yX_FIFO_LEVEL_ALMOST_FULL ターゲットの構成では、I2CC は TX を使用し、I2CT は RX を使用します
グリッチ フィルタリング	• DL_I2C_ANALOG_GLITCH_FILTER_WIDTH_X • DL_I2C_DIGITAL_GLITCH_FILTER_WIDTH_X	MSPM33C にはアナログ グリッチ フィルタは実装されていません。 • DL_I2CC_DIGITAL_GLITCH_FILTER_WIDTH_X
API 関数名の変更	• DL_I2C_enableController() • DL_I2C_startControllerTransfer() • DL_I2C_transmitControllerData() • DL_I2C_receiveControllerData() • DL_I2C_getControllerStatus() • DL_I2C_fillControllerTXFIFO()	• DL_I2Cx_enable() • DL_I2Cx_startTransfer() • DL_I2Cx_transmitData() • DL_I2Cx_receiveData() • DL_I2Cx_getStatus() • DL_I2Cx_fillTXFIFO()
割り込み命名	• DL_I2C_INTERRUPT_CONTROLLER_RX_DONE • DL_I2C_INTERRUPT_CONTROLLER_TX_DONE • DL_I2C_INTERRUPT_CONTROLLER_RXFIFO_TRIGGER • DL_I2C_INTERRUPT_CONTROLLER_TXFIFO_TRIGGER • DL_I2C_INTERRUPT_CONTROLLER_NACK	• DL_I2Cx_INTERRUPT_RX_DONE • DL_I2Cx_INTERRUPT_TX_DONE • DL_I2Cx_INTERRUPT_RXFIFO_TRIGGER • DL_I2Cx_INTERRUPT_TXFIFO_TRIGGER • DL_I2Cx_INTERRUPT_NACK
プリエンプティブ DMA 割り込み	MSPM0 は、プリエンプティブ DMA 割り込みに対応していません。	• DL_I2Cx_IIDX_PREEMPT_INT_TX • DL_I2Cx_IIDX_PREEMPT_INT_RX

3 まとめ

要約すると、このアプリケーション ノートでは、MSPM0Gx51x から MSPM33C32xx にソフトウェアを移行するためのガイドラインが提供されます。これを、ハードウェア移行ガイド (SLAAES2) と併せて使用して、2 つのデバイス間の完全な移行を実行します。

4 参考資料

1. テキサス インスツルメンツ、MSPM33C321x ミックスド シグナル マイコン、データシート
2. テキサス・インスツルメンツ、『MSPM33 C シリーズ 160MHz マイコン』、技術参照マニュアル
3. テキサス インスツルメンツ、MSPM0Gx51x ミックスド シグナル マイコン、データシート
4. テキサス インスツルメンツ、『MSPM0 G シリーズ 80MHz マイコン』、テクニカル リファレンス マニュアル
5. テキサス インスツルメンツ、『MSPM0Gx51x および MSPM33C32xx ハードウェア移行ガイド』、ハードウェア移行ガイド

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