参考資料

Application Note

TXE81xx-Q1 GPIO エクスパンダとの SPI デイジーチェーン接続

Tyler Townsend

概要

このアプリケーション ノートでは、複数の TXE8124-Q1 デバイスを使用した SPI デイジーチェーン接続の使用例を紹介します。

目次

商標

1 概要

デイジーチェーン接続は、複数の SPI プロトコルデバイスにより使用される機能です。デイジーチェーン接続を使用することで、配線コストを削減し、MCU とプロセッサから必要な IO の合計数を削減できるため、PCB がより簡潔でシンプルになります。

2 SPI デイジーチェーンとは何か？

SPI デイジーチェーン接続は、複数の SPI ペリフェラル デバイスを直列接続して通信するために使用する接続方式です。デイジーチェーン接続により、必要な配線 / パターン長が短くなり、複数のチップ セレクトのために MCU の GPIO を節約できます。

複数のペリフェラルの通常の SPI 構成では、各 SPI ペリフェラル デバイスにチップ セレクト信号が必要です。これは、システム内の各 SPI ペリフェラルのために、MCU からの GPIO を予約する必要があることを意味します。4 つのペリフェラル デバイス間の通常の SPI 構成については、以下の例を参照してください。

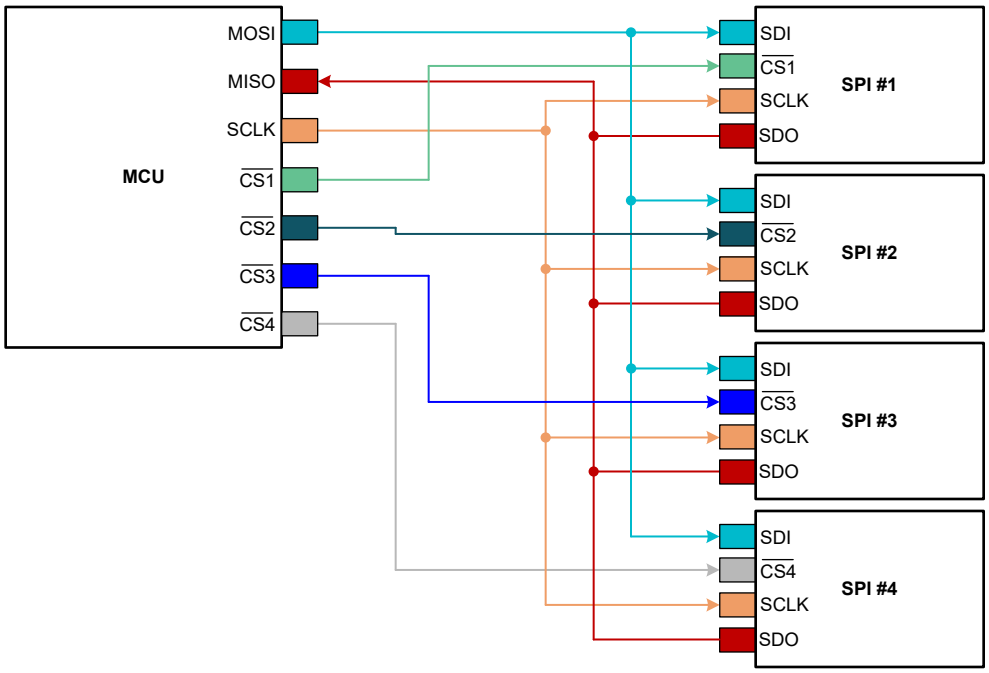

図 2-1. デイジーチェーンを使用しない SPI バスの例

4 つの SPI ペリフェラル デバイスを制御するには、MCU からのピンが 8 つ必要です。MCU からの PICO は、すべてのシリアル データ入力 (SDI) に接続されます。MISO は、シリアル データ出力 (SDO) に接続されます。クロック ピン (SCLK) は、システム内のすべてのデバイスで共有されます。個別のチップ セレクトをシステム内の各ペリフェラル専用にする必要があります。

SPI の通常の実装には、MCU からの複数の GPIO ピンが必要で、特定のシステムでは制限される可能性があります。つまり、各チップ セレクトを各ペリフェラル デバイスに配線することで、システムで使用する配線が増えることになります。言い換えると配線が物理的に増加し、システム内の重量が増加したり、PCB が複雑になる可能性があります。

必要な配線とチップ セレクト ラインの数を減らすという 2 つの問題を解決するために、SPI デイジーチェーン接続方式を実装できます。この機能を利用するには、IC デバイスのメーカーが SPI デイジーチェーンをサポートしている必要があります。

(注：SDO は SDI に接続され、1 つのチップ セレクト信号のみを使用)

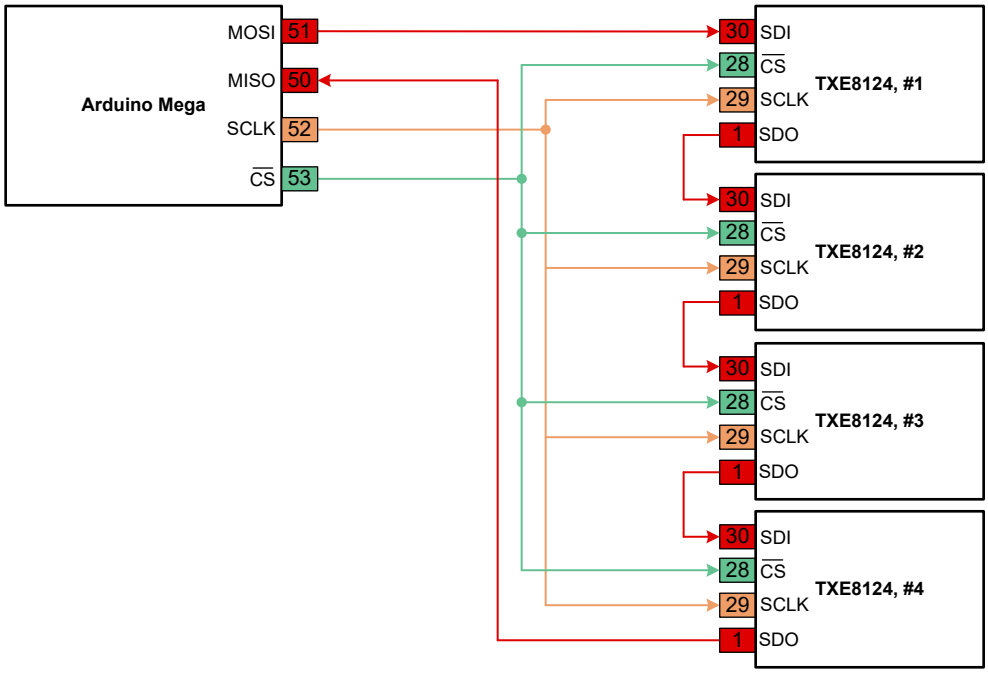

図 2-2. デイジー チェーン接続を使用した SPI バスの例

デイジーチェーンの実装では、あるペリフェラルからの出力を別のペリフェラルの入力に接続することで、シリアル データを効率化できます。この場合、MCU からの配線は必要はなく、シーケンス内の各ペリフェラルに「チェーン接続」できるため、配線を大幅に削減できます。

注

デイジー チェーン接続でデバイスを接続する場合、チェーンの最後段のデバイスからの SDO をサンプリングする際には、SCLK 周波数を低下させる必要があることがあります。これは、チェーン内の各デバイスのデータ有効時間が累積されることに加え、最も遠いデバイスまでの SCLK の伝搬遅延も加算されるためです。

3 TXE81xxEVM を使用した SPI デイジーチェーンの例

以下の使用事例では、SDO ラインと SDI ラインをチェーン内で接続することにより、4 台の TXE81XXEVM をデイジーチェーン接続しています。SCLK ピンは、MCU とチップ セレクト信号を含むすべてのデバイス間で共有されます。

(ボードには、右から左の順に 1～4 のラベルが付いています)

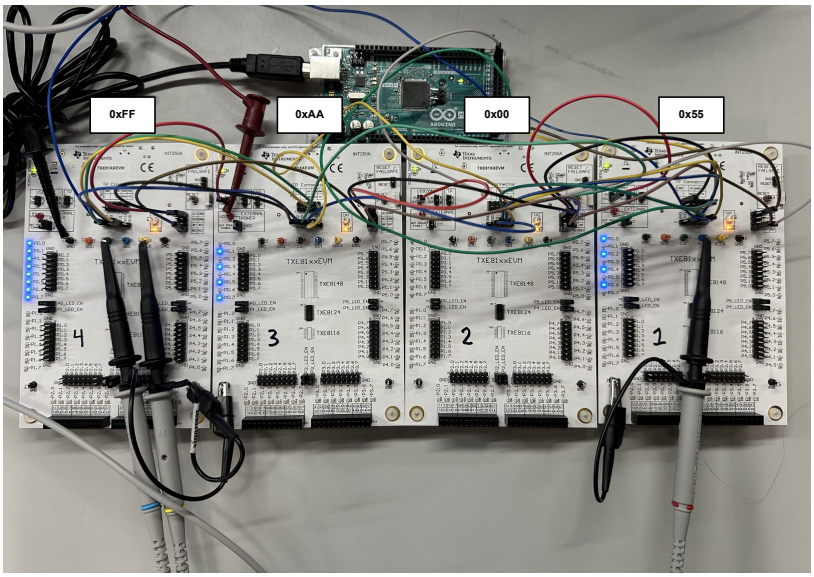

図 3-1. 4 台の TXE81XXEVM のデイジーチェーン構成での接続

各 TXE8124-Q1 デバイスの各方向構成レジスタ (0x04) にデータが書き込まれます。ポート 0 のみに書き込まれます。方向構成レジスタが 1 のとき、GPIO は出力に設定されます。方向構成レジスタが 0 のとき、GPIO は入力に設定されます。デイジーチェーンの例に記載されている正確なデータについては、表 3-1 を参照してください。

表 3-1. ボード構成

ボード番号	レジスタ・アドレス	ポート	データ	入力 / 出力
1	0x04	0	0x55	入力 = P0.1、P0.3、P0.5、P0.7 出力 = P0.0、P0.2、P0.4、P0.6
2	0x04	0	0x00	入力 = P0.0～P0.7 出力 = なし
3	0x04	0	0xAA	入力 = P0.0、P0.2、P0.4、P0.6 出力 = P0.1、P0.3、P0.5、P0.7
4	0x04	0	0xFF	入力 = なし 出力 = P0.0～P0.7

TXE81xx には、次の 4 種類の SPI セグメントがあります。ステータス、ヘッダー、アドレス、データ。表 3-2 に、デイジーチェーンで送信される各セグメントのビット単位の説明を示します。

注

ヘッダ セグメントのビット 15 と 14 はヘッダ ID であり、デバイス コントローラがヘッダ セグメントを受信していることを検出するために使用されます。ヘッダ ID ビットは値 01 を取り、これがヘッダ セグメントであることを示します。ビット 13 ～ 5 は予約済みであり、ビット 4 ～ 0 はチェーン内のデバイス数を示します。デイジーチェーン接続できるデバイスの最大数は 31 です。

表 3-2. SPI セグメントの説明

SPI セグメント タイプ	ビット割り当て
ステータス	ビット [15:14] = 1 (ステータス セグメントであることを示します) ビット [13:8] = フォルト ステータス レジスタ (0x1900) のビット 5～0 ビット [7:0] = 0 (デフォルト)
ヘッダー	ビット [15:14] = それぞれ 0 および 1 (ヘッダー セグメントであることを示します) ビット [13] = 予約済み ビット [12:0] = デイジーチェーン内のデバイス数を決定します
アドレス (レジスタ アドレス)	ビット [15] = SPI 動作モードを示します (1 = 読み取り操作、0 = 書き込み操作) ビット [14:13] = 不定 (X) ビット [12:8] = 機能アドレス ビット [7] = 不定 (X) ビット [6:4] = ポート選択 ビット [3:1] = 不定 (X) ビット [0] = マルチポート
データ	ビット [7:0] = レジスタに書き込むデータ

チェーンを介したデータ送信を開始するには、最初にヘッダー セグメントが送信され、次にチェーン内で最も遠いボードのレジスタ アドレスが送信されます。チェーンに 4 つのデバイスが存在する場合、4 番目のデバイスのレジスタ アドレスが最初に送信され、次に 3 番目のデバイスの順に送信されます。データ バイトは、レジスタ アドレス バイトの後に続きます。最初のデータ バイトは、チェーン内で最も遠いデバイスに適用されます。チェーン内に 4 つのデバイスがある場合、最初のデータ バイトは 4 番目のデバイスに適用され、次に 3 番目のデバイスの順に適用されます。SPI データがどのように送信されるかについての、より詳細なバイト単位の例は、図 3-3 を参照してください。

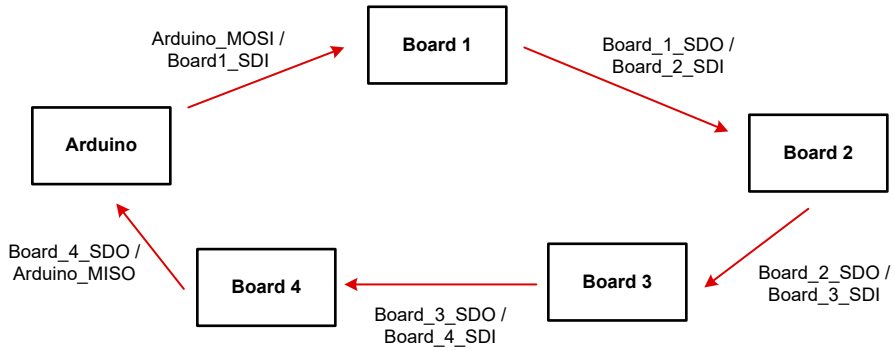

図 3-2. SPI デイジーチェーンのブロック図

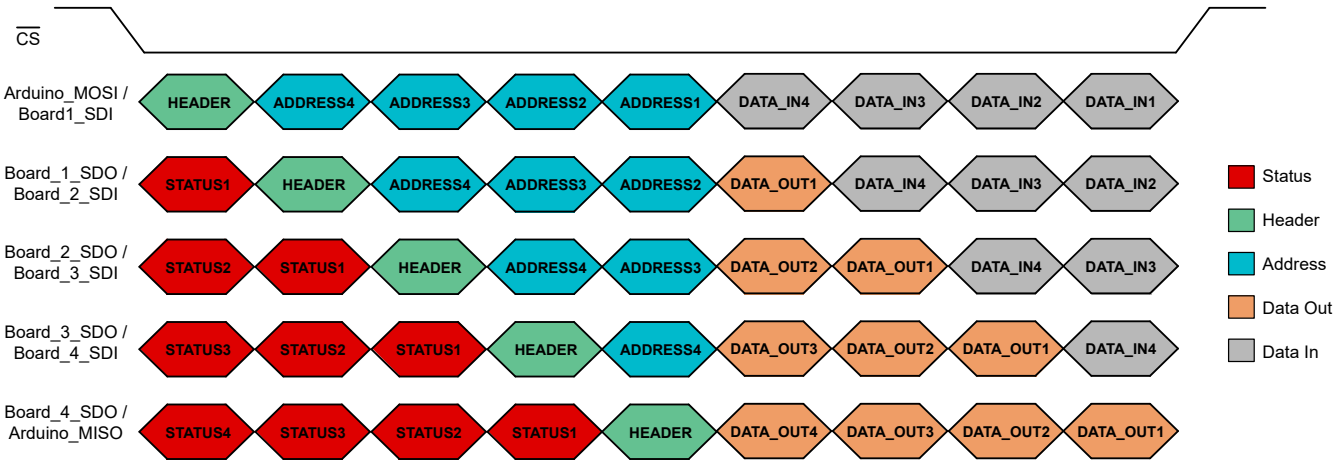

図 3-3. チェーン内の各バイトのシーケンス

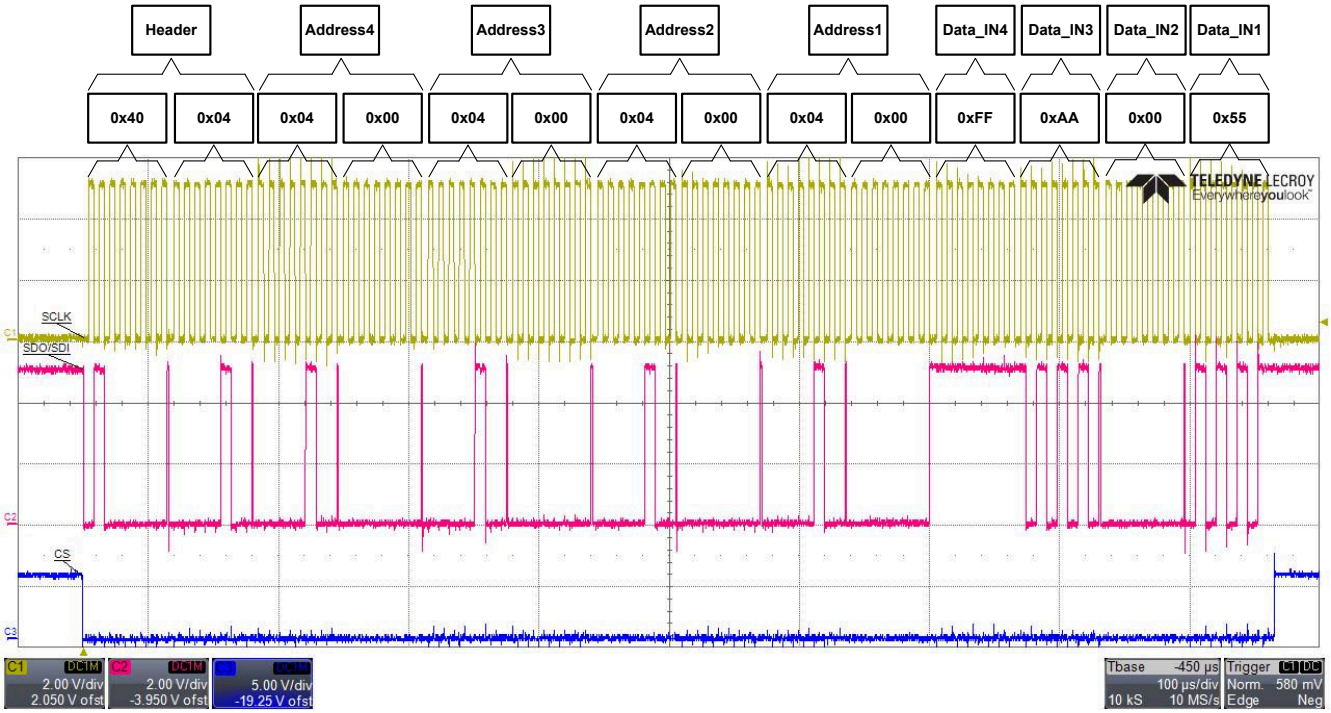

図 3-4. Arduino_MOSI と基板 1 SDI の間でプローブされたデイジーチェーン SPI 転送のアノテートされた波形

4 MSPM0 の疑似コード例

```
#include"ti_msp_dl_config.h"//MSP driver library
int main(void)
{
SYSCFG_DL_INT(); //initialize SPI driver
DL_GPIO_setPins(GPIO_LEDS_PORT, GPIO_LEDS_USER_LED_1_PIN | GPIO_LEDS_USER_TEST_PIN); //set /CS HIGH
DL_GPIO_clearPins(GPIO_LEDS_PORT, GPIO_LEDS_USER_LED_1_PIN | GPIO_LEDS_USER_TEST_PIN); //set /CS
LOW
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0b01000000); //header segment
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0b00000100);
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x04); //board 4 address
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x00);
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x04); //board 3 address
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x00);
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x04); //board 2 address
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x00);
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x04); //board 1 address
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x00);
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0xFF); //board 4 data
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0xAA); //board 3 data
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x00); //board 2 data
DL_SPI_transmitDataBlocking8(SPI_0_INST, 0x55); //board 1 data
DL_GPIO_setPins(GPIO_LEDS_PORT, GPIO_LEDS_USER_LED_1_PIN | GPIO_LEDS_USER_TEST_PIN); //set /CS HIGH
}
```

5 Arduino の疑似コード例

Arduino のコーディング例

```
#include <SPI.h>
#define CS 53

//MISO = 50
//CS = 53
//MOSI = 51
//SCLK = 52

void setup() {
  Serial.begin(115200);
  pinMode(CS, OUTPUT);

  SPI.begin();
  SPI.beginTransaction(SPISettings(125000, MSBFIRST, SPI_MODE0));
}

void loop() {
  //send SPI in 8-bit words
  uint8_t header_seg1 = 0b01000000;   //default header segment
  uint8_t header_seg2 = 0b00000100;   //4 devices in the chain
  uint8_t address_seg1 = 0b00000100;  //direction configuration register
  uint8_t address_seg2 = 0b00000000;  //port 0 selected, no multi-port

  digitalWrite(CS, LOW);
  SPI.transfer(header_seg1);
  SPI.transfer(header_seg2);
  SPI.transfer(address_seg1);         //board 4 address
  SPI.transfer(address_seg2);
  SPI.transfer(address_seg1);         //board 3 address
  SPI.transfer(address_seg2);
  SPI.transfer(address_seg1);         //board 2 address
  SPI.transfer(address_seg2);
  SPI.transfer(address_seg1);         //board 1 address
  SPI.transfer(address_seg2);
  SPI.transfer(0xFF);                 //board 4 data
  SPI.transfer(0xAA);                 //board 3 data
  SPI.transfer(0x00);                 //board 2 data
  SPI.transfer(0x55);                 //board 1 data
  digitalWrite(CS, HIGH);
}
```

6 まとめ

通常の SPI 接続と SPI デイジーチェーン接続の実装には、以下の表に示すように、それぞれいくつかの長所と短所があります。

表 6-1. 通常の SPI と SPI デイジーチェーン接続の技術的トレードオフ

	通常の SPI	SPI デイジー チェーン接続
配線	• 複数の CS ラインが必要になり配線が増える • PCB が複雑になり、潜在的な配線コストが増える • 配線の増加 = 重量の増加 = コストの増加	• 必要な CS ラインが 1 本なので配線が減る • PCB 配線と各 SPI ペリフェラルへの接続がシンプルになる
デバイス制御	• 個別のデバイス制御が可能	• 単一デバイスの制御が難しく、チェーン全体との通信が必要
データ転送	• データ転送は本質的に高速	• データがチェーン内のすべてのデバイスを通過する必要があるため、データ転送は本質的に低速
将来の設計	• 将来の設計を変更することは難しく、システム内にペリフェラルを追加するたびに CS ラインを追加する必要がある	• ペリフェラルを追加する場合、SPI デイジーチェーンの末尾に簡単に追加できる。同じ CS ラインを引き続き使用できる。
シグナル インテグリティ	• 電気的接続を複数のデバイスに展開する必要があり、パターン長 / 配線距離が長くなる可能性がある	• 直列接続によりパターン長が短くなり、SI が低減する。チェーン内の各ペリフェラル デバイスは、チェーン内の次のデバイスに SPI データをリドライブする
ソフトウェア	• ソフトウェア実装がシンプル	• チェーン内のすべてのペリフェラルで構成が必要な場合、データ更新を効率的に行える • ソフトウェア実装が複雑になる、SPI ワードが長くなる
デバッグ	• ペリフェラル障害を簡単に特定できる	• チェーン内のどのペリフェラルが壊れたかを判断するのが難しい • チェーン内の 1 つのペリフェラルが壊れると、複数のペリフェラルが影響を受ける

7 参考資料

-

8 改訂履歴

Changes from Revision * (October 2025) to Revision A (May 2026)	Page
• ドキュメント全体にわたって表、図、相互参照の採番方法を更新	2
• セクション 2 に注を追加	3
• セクション 3 に注を追加	5

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

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