

## Application Note

**Linux の動的周波数スケーリングを使用した CPU 冷却**

Keerthy J, Udit Kumar, Josiitaa RL

**概要**

このアプリケーション ノートでは、テキサス インストルメンツの TDA4VL/J721S2 プロセッサが Linux ベースの動的周波数スケーリング (DFS) とサーマル フレームワークを活用して効率的な CPU 熱管理を実装する方法を説明します。この資料では、k3\_j7xxx\_bandgap ドライバによって利用可能な電流サーマル センサのサポートについて説明し、DFS を使用してパッシブ冷却を実現する手順の概要を示します。このガイドは、組込み Linux の開発者およびシステム アーキテクトをターゲットとして、電力と熱効率が重要な車載および産業用市場のアプリケーションを対象としています。この情報は、開発者がサーマル ゾーンを監視し、TDA4VL/J721S2 ベースのシステムにソフトウェア制御のスケラブルな冷却手法を実装する方法を理解するのに役立ちます。この例では、TDA4VL への実装について説明します。他の TDA4 デバイスや Sitara デバイスにまたがる形で、Linux 上の熱管理の目的で拡張することができます。

**目次**

<b>1 はじめに</b>	<b>2</b>
1.1 TDA4VL SoC の概要	2
1.2 本書の目的	2
1.3 ターゲット オーディエンスとアプリケーション	2
1.4 問題提起	2
<b>2 Linux での動的周波数スケーリング (DFS)</b>	<b>3</b>
2.1 DFS とは	3
2.2 Linux CPUFREQ フレームワーク	3
2.3 対応 CPUFREQ ガバナー	3
2.4 TI の SoC における DFS サポート ステータス	3
<b>3 Linux サーマル フレームワーク</b>	<b>4</b>
3.1 サーマル ゾーンとトリップ ポイント	4
3.2 冷却メカニズム: パッシブとアクティブの比較	4
3.3 パッシブ冷却における DFS の役割	4
<b>4 TDA4VL デバイスでの熱特性のサポート</b>	<b>5</b>
4.1 VTM およびバンドギャップ センサの初期化	5
4.2 k3_j7xxx_bandgap ドライバによる温度監視	5
<b>5 TDA4VL での CPU 冷却の有効化</b>	<b>6</b>
5.1 CPU 冷却を有効にするパッチ	6
5.2 TDA4VL の冷却機能のテスト	7
<b>6 TDA4 と Sitara の各デバイス間でのスケラビリティ</b>	<b>9</b>
6.1 実装の適応	9
<b>7 まとめ</b>	<b>9</b>
<b>8 参考資料</b>	<b>9</b>

**商標**

すべての商標は、それぞれの所有者に帰属します。

## 1 はじめに

### 1.1 TDA4VL SoC の概要

テキサス インストルメンツの TDA4VL SoC は、K3 ファミリの一部であり、高性能車載および産業用アプリケーション向けに設計されています。デュアルコア Arm®Cortex® - A72 CPU、スケーラブルなメモリ アーキテクチャ、統合型アクセラレータを採用している SoC は、計算能力とフレキシビリティの両方を実現します。これらのアプリケーションでは、特に作業負荷や環境条件が変動する中で高性能計算が可能な Linux ベースのシステムを実行する場合に、効果的な電力と熱管理が不可欠です。

### 1.2 本書の目的

このアプリケーション ノートでは、TDA4VL デバイス上で Linux ベースの動的周波数スケーリング (DFS) と熱管理フレームワークを使用して、CPU の電力と温度を効果的に管理する方法を説明します。本書では、温度センサの現在のサポート状態を明らかにし、DFS 機能について説明し、これらの機能を組み込み Linux システムに統合するためのガイダンスを掲載しています。

また、本書では、TDA4VL プラットフォームの実装例の概要を示し、Linux の CPUFREQ とサーマル フレームワークを使用して熱的な観点で CPU 周波数をスケーリングする方法を示します。

### 1.3 ターゲット オーディエンスとアプリケーション

本書は、以下を対象としています。

- 組み込み Linux 開発者
- システム アーキテクト
- アプリケーション エンジニア
- TI の TDA4VL または他の ARM ベース SoC の開発を担当するエンジニア

これは、以下の使用事例に関連します。

- 車載 (ADAS、集中型コンピューティング ECU)
- 産業用オートメーションおよびロボティクス

### 1.4 問題提起

温度の制約が厳しい環境では、高い CPU 性能の持続によって、過熱、システム スロットル、またはシャットダウンにつながる恐れがあります。DFS は、Linux のサーマル フレームワーク統合に加えて、ソフトウェアベースの設計により、システム温度に応じて CPU 周波数を動的に制御し、大規模なハードウェア冷却設計を必要とせずに動作の信頼性と効率を検証します。

## 2 Linux での動的周波数スケーリング (DFS)

### 2.1 DFS とは

動的周波数スケーリング (DFS) は、CPU 周波数スケーリングとも呼ばれ、プロセッサが性能要求または熱条件に基づいて動作周波数を動的に調整できるパワー マネージメント技術です。周波数を低くすると、消費電力と発熱が減少し、性能が向上します。

複数の Arm®Cortex® - A72 コアを搭載したテキサス インストルメンツの TDA4VL SoC において、DFS を使用することで実行時に CPU 周波数をスケーリングし、性能と電力のバランスを確保できます。これは、さまざまなワークロードや熱的制約の下で動作する組み込みシステムで特に役立ちます。

### 2.2 Linux CPUFREQ フレームワーク

Linux カーネルは、CPUFREQ フレームワークと呼ばれる標準化されたインターフェイスを提供することで、DFS を有効にして制御します。このフレームワークを使用すると、ユーザーやシステム ソフトウェアが、実行時にハードウェアがサポートするさまざまな CPU 周波数間で切り替えることができます。

CPUFREQ フレームワークの主要部品は次のとおりです。

- CPU 周波数ドライバ:これらのインターフェイスは、CPU クロックを制御するハードウェア固有のメカニズムを備えています。
- ガバナー:CPU 周波数を変更するタイミングと方法を決定するアルゴリズム
- Sysfs インターフェイス:CPUFREQ サブシステムとのユーザー空間の相互作用を可能にする (例:/sys/devices/system/cpu/cpu \*/cpufreq/)

### 2.3 対応 CPUFREQ ガバナー

Linux カーネルは、DFS 用のガバナーを複数サポートしており、それぞれが次の異なる使用事例に対応しています。

- Performance ガバナー:
  - 常に CPU を最高対応周波数に設定します。
  - 最高レベルの性能が常に必要なアプリケーションに最適です。
- Userspace ガバナー:
  - ユーザー空間アプリケーションまたはスクリプトによって CPU の周波数を手動で設定できるようにします。
- On Demand ガバナー:
  - 現在の CPU 負荷に基づいて周波数を動的にスケーリングします。
  - 汎用システムで一般的に使用されています。

各ガバナーは、sysfs インターフェイスを介して選択および構成、またはシステム ポリシーによって自動的に選択および構成できます。

### 2.4 TI の SoC における DFS サポート ステータス

- AM62 デバイス:
  - DFS のフル サポートが利用可能です。
  - CPUFREQ フレームワークは、選択したガバナーに基づいて周波数を動的に調整できます。
- TDA4 デバイス:
  - 現在のリリース (SDK 11.0) では、DFS サポートは有効になっていません。
  - 温度監視のインフラストラクチャは k3\_j7xxx\_bandgap ドライバによって設置されていますが、ランタイム周波数スケーリングは有効化待ちです。
  - DFS をサポートすると、TDA4 ベースのシステムは熱ストレス下で消費電力と温度を低減できます。

### 3 Linux サーマル フレームワーク

Linux サーマル フレームワークは、Linux カーネル内にあるサブシステムであり、デバイスの熱状態を監視および管理できるように設計されています。このフレームワークは、サーマル ゾーンを定義する、温度センサを登録する、動的周波数スケーリング (DFS)、CPU のホット プラグ、ファン制御などの熱低減手法を適用するためのプラットフォームに依存しないインフラストラクチャを提供します。

#### 3.1 サーマル ゾーンとトリップ ポイント

サーマル ゾーンとは、CPU クラスタ、GPU、PMIC など、システム内の温度監視領域を指します。各サーマル ゾーンは、次のものに関連付けられています。

- 1 個または複数の温度センサ
- 一連のトリップ ポイント(スレッシュホールド温度)
- トリップ ポイントの到達時にトリガされる付属の冷却装置

トリップ ポイントは、アクション スレッシュホールド (パッシブ冷却やアクティブ冷却など) を定義します。トリップ ポイントは次のとおりです。

- パッシブ: ソフトウェア ベースの軽減をトリガする (CPU 周波数の低減など)
- アクティブ: ハードウェア メカニズムをトリガする (ファン速度の起動や増加など)
- クリティカル: システムのシャットダウンに使用され、ハードウェアの損傷を防止する

これらは通常、デバイス ツリーで定義され、カーネル内のサーマル ガバナー ロジックを通じて処理されます。

#### 3.2 冷却メカニズム: パッシブとアクティブの比較

サーマル フレームワークは、さまざまな種類の冷却デバイスをサポートしており、大まかに次のように分類されます。

パッシブ冷却:

- 性能や消費電力を低下させることで、システムの温度を低下させます
- 例:
  - DFS を使用した CPU 周波数の低下
  - CPU コアの無効化またはホット プラグ
- 低ノイズまたは低消費電力動作が必要な場合は、パッシブ冷却が推奨されます。

アクティブ冷却:

- 次のような外部メカニズムを使用して温度を下げます。
  - ファンの作動動
  - PWM 制御によるファン速度の向上
- 外部ハードウェアとドライバのサポートが必要です (ファンコントローラなど)

#### 3.3 パッシブ冷却における DFS の役割

TI の TDA4VL などのプラットフォームでは、DFS を Linux サーマル フレームワークの下でパッシブ冷却デバイスとして機能させることができます。サーマル ゾーンが定義されたトリップ ポイントを超えると、サーマル フレームワークから CPU 周波数の低下を要求して、温度を下げることができます。

この統合には以下を要します。

- CPU 周波数ドライバを冷却デバイスとして登録
- CPU 周波数と関連するサーマル ゾーンに関連付け
- 冷却状態にリンクするトリップ ポイントの定義

TDA4VL で DFS サポートを有効にすると、ユーザーは次のことが可能です。

- 温度上昇時に CPU 周波数を動的に下げる
- システムの安定性を維持し、過熱を防止する
- 熱的に制限された環境でも、追加のハードウェア冷却を必要とせずに動作させる

## 4 TDA4VL デバイスでの熱特性のサポート

テキサス インスツルメンツの TDA4VL プラットフォームにはオンチップの熱センシング ハードウェアが搭載されており、Linux サーマル フレームワークによる温度監視をサポートしています。A72 コアでは、完全動的周波数スケーリング (DFS) のサポートはまだ有効になっていませんが、基本的な温度監視機能はすでに利用可能です。

### 4.1 VTM およびバンドギャップ センサの初期化

TDA4VL SoC の電圧および温度モニタ (VTM) サブシステムは、熱センシングのためのハードウェア サポートを備えています。サブシステムの内容:

- 複数のオンダイ温度センサ (バンドギャップベース)
- サーマル ゾーンを監視するためのインフラ

Linux では、VTM ハードウェアは `k3_j7xxx_bandgap` ドライバを介して初期化されます。このドライバにより、以下のことが可能です。

- SoC 上の複数の熱センサから温度値の読み取り
- サーマル ゾーンを Linux のサーマル フレームワークに公開する

初期化されると、次の場所にある `sysfs` インターフェイスから温度データにアクセスできます: `/sys/class/thermal/thermal_zone*/temp`

### 4.2 k3\_j7xxx\_bandgap ドライバによる温度監視

`k3_j7xxx_bandgap` ドライバは、温度センサの低レベル初期化を実行し、ソフトウェア フックによってサーマル ゾーンを登録します。主な特長は下記の通りです。

- A72 コア、GPU、および SoC のその他の主要領域からリアルタイムで温度を読み取ります。
- 各物理センサを論理サーマル ゾーンにマッピングします。
- トリップ ポイントをトリガするため、Linux サーマル コアと統合します

この機能により、アクティブまたはパッシブな冷却メカニズムが実装される前でも、開発者は熱挙動の観測と冷却手法の計画が可能になります。

## 5 TDA4VL での CPU 冷却の有効化

このセクションでは、動的周波数スケーリング (DFS) を使用して TDA4VL デバイスのパッシブ CPU 冷却を有効にするために必要なパッチについて説明し、機能を検証するための手順を概説します。

### 5.1 CPU 冷却を有効にするパッチ

```
From e9ffcedb3b567110fda18f4a49fb8e66672910e4 Mon Sep 17 00:00:00 2001
From: Keerthy <j-keerthy@ti.com>
Date: Fri, 20 Jun 2025 14:52:39 +0530
Subject: [PATCH] arch/arm64/boot/dts/ti/k3-j721s2-thermal.dtsi: Add CPUFREQ
based cooling
Add CPUFREQ based cooling
Signed-off-by: Keerthy <j-keerthy@ti.com>
---
arch/arm64/boot/dts/ti/k3-j721s2-thermal.dtsi | 14 ++++++
arch/arm64/boot/dts/ti/k3-j721s2.dtsi | 33 ++++++
2 files changed, 47 insertions(+)
diff --git a/arch/arm64/boot/dts/ti/k3-j721s2-thermal.dtsi b/arch/arm64/boot/dts/ti/k3-j721s2-thermal.dtsi
index e3ef61c16..69727a5a5 100644
--- a/arch/arm64/boot/dts/ti/k3-j721s2-thermal.dtsi
+++ b/arch/arm64/boot/dts/ti/k3-j721s2-thermal.dtsi
@@ -39,12 +39,26 @@ main0_thermal: main0-thermal {
    thermal-sensors = <&wkup_vtm0 2>;
    trips {
+ main0_alert: main0-alert {
+    temperature = <45000>; /* millicelsius */
+    hysteresis = <2000>; /* millicelsius */
+    type = "passive";
+ };
+
    main0_crit: main0-crit {
        temperature = <125000>; /* millicelsius */
        hysteresis = <2000>; /* millicelsius */
        type = "critical";
    };
};
+
+ cpu_cooling_maps: cooling-maps {
+ map0 {
+    trip = <&main0_alert>;
+    cooling-device =
+    <&cpu0 THERMAL_NO_LIMIT THERMAL_NO_LIMIT>;
+ };
+ };
+
    main1_thermal: main1-thermal {
diff --git a/arch/arm64/boot/dts/ti/k3-j721s2.dtsi b/arch/arm64/boot/dts/ti/k3-j721s2.dtsi
index 4f416bb18..71dcb9e6f 100644
--- a/arch/arm64/boot/dts/ti/k3-j721s2.dtsi
+++ b/arch/arm64/boot/dts/ti/k3-j721s2.dtsi
@@ -51,6 +51,10 @@
    d-cache-line-size = <64>;
    d-cache-sets = <256>;
    next-level-cache = <&L2_0>;
+ clocks = <&k3_clks 202 0>;
+ clock-names = "cpu";
+ operating-points-v2 = <&cpu0_opp_table>;
+ #cooling-cells = <2>; /* min followed by max */
};
+
cpu1: cpu@1 {
@@ -65,6 +65,10 @@
    d-cache-line-size = <64>;
    d-cache-sets = <256>;
    next-level-cache = <&L2_0>;
+ clocks = <&k3_clks 203 0>;
+ clock-names = "cpu";
+ operating-points-v2 = <&cpu0_opp_table>;
+ #cooling-cells = <2>; /* min followed by max */
+ };
+ };
+
+ cpu0_opp_table: opp-table {
+ compatible = "operating-points-v2";
+ opp-shared;
```

```
+
+ opp6-2000000000 {
+ opp-hz = /bits/ 64 <2000000000>;
+ clock-latency-ns = <300000>;
+ };
+
+ opp4-1000000000 {
+ opp-hz = /bits/ 64 <1000000000>;
+ clock-latency-ns = <300000>;
+ };
+
+ opp2-500000000 {
+ opp-hz = /bits/ 64 <500000000>;
+ clock-latency-ns = <300000>;
+ };
+
+ opp1-250000000 {
+ opp-hz = /bits/ 64 <250000000>;
+ clock-latency-ns = <300000>;
+ };
+ };
+ };
--
2.17.1
```

## 5.2 TDA4VL の冷却機能のテスト

パッチが適用され、カーネルが再構築および展開された後、次の手順に従って CPU 冷却動作を検証します。

1. 冷却デバイスの表示を確認します。

```
cat /sys/class/thermal/cooling_device*/
```

CPU 周波数ドライバが冷却デバイスとして登録されていることを示す `cpu-freq` などのエントリを探します。

2. サーマル ゾーンのトリップ ポイントを確認します。

```
cat /sys/class/thermal/thermal_zone1/trip_point_0_temp
45000
```

この例では、トリップ ポイントの温度が **45°C** に設定されています。

3. 現在の状態と最大状態を確認します。

```
cat /sys/class/thermal/thermal_zone1/cooling_device0/cur_state
0
cat /sys/class/thermal/thermal_zone1/cooling_device0/max_state
3
```

4. 現在および利用可能なスケーリング周波数を確認します。

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
2000000
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
250000 500000 1000000 2000000
```

現在の CPU 周波数は **2GHz** で、利用できる周波数範囲は **250MHz ~ 2GHz** です。

5. 現在の温度を確認します。

```
cat /sys/class/thermal/thermal_zone2/temp
44753
```

現在の温度は約 **44.753°C** です。

6. CPU 負荷をトリガします。

```
cpuloadgen 100 100 &
[1] 1246
```

`cpuloadgen` コマンドを使用すると、CPU コアを 100 秒間 100% に固定して CPU 負荷を増加できます。



## 7. 現在の温度を確認します。

```
cat /sys/class/thermal/thermal_zone2/temp  
45209
```

CPU 負荷が上昇すると、SoC の温度が **45.209°C** に上昇し、周波数スケーリングを使用して動的 CPU 冷却がトリガされます。これにより、温度は **44.981°C** に低下します。

```
cat /sys/class/thermal/thermal_zone2/temp  
44981
```

## 8. 現在のスケーリング頻度と冷却状態を確認します。

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq  
250000  
cat /sys/class/thermal/thermal_zone1/cooling_device0/cur_state  
3
```

現在の周波数が **250MHz** まで低下し、デバイスは最大冷却状態に達しています。

---

**注**

室温でのすべての機能を提示するために、トリップ ポイントは **45°C** として意図的にプログラムされています。ユーザーは、システムに基づいてトリップ温度をキャリブレーションする必要があります。

---



## 6 TDA4 と Sitara の各デバイス間でのスケーラビリティ

このアプリケーション ノートに記載の手法は、TDA4VL プラットフォームで DFS ベースのパッシブ CPU 冷却を実現するためのものであり、類似の CPUFREQ とサーマル フレームワークをサポートしているその他のテキサス インストルメンツの TDA4 および Sitara ファミリー SoC に適応できます。

### 6.1 実装の適応

TDA4 または Sitara ファミリーの他のデバイスに設計を移植する場合：

1. デバイス ツリー ノードの更新し
  - ・ ターゲット SoC でサポートされている周波数に応じて CPU OPP 表を調整します。
  - ・ 利用可能なテスト条件に基づいてサーマル ゾーンを構成します
2. DFS サポートの確認
  - ・ ターゲット SoC の CPUFREQ ドライバがランタイム周波数の変更をサポートしていることを確認します。
  - ・ カーネル構成で適正なガバナーを有効にします。
3. サーマル ゾーン マッピング
  - ・ CPU 冷却デバイスを新しい SoC の適切なサーマル ゾーンに関連付けます。
4. 負荷試験
  - ・ [TDA4VL の冷却機能のテスト](#)で説明されている機能テスト手順を繰り返して、正常な冷却動作を確認します。

## 7 まとめ

このアプリケーション ノートでは、Linux サーマル フレームワークを使用して、TDA4VL プラットフォームに DFS ベースのパッシブ CPU 冷却機能を実装する方法を提示します。デバイス ツリー構成によって CPUFREQ 冷却デバイスとオンダイの熱センサを統合することで、システムは CPU 周波数を動的に調整し、アクティブな冷却部品を必要とせずに安全な動作温度を維持できます。

公開されているパッチとテスト手順は、類似の TDA4 と Sitara の各デバイスで設計を複製するための実践的なリファレンスとして活用できます。CPU OPP 表、サーマル ゾーン構成、CPUFREQ ガバナーに小規模な変更を加えるだけで、同じアプローチをさまざまな TI SoC に適応させることができます。

この方法を採用すると、システムの信頼性と耐用期間が向上するだけでなく、熱的に制限された環境でも低ノイズで電力効率の高い動作を実現できます。

## 8 参考資料

1. Samsung Electronics Co, 『[CPU 冷却 API の使い方](#)』
2. インテル, 『[Generic Thermal Sysf ドライバの使い方](#)』
3. テキサス インストルメンツ, 『[TDA4VE TDA4AL TDA4VL Jacinto™ プロセッサ、シリコン リビジョン 1.0](#)』データシート

## 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](https://www.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  
Copyright © 2025, Texas Instruments Incorporated

## 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、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 © 2025, Texas Instruments Incorporated

最終更新日：2025 年 10 月