

## Application Note

# Bluetooth® Low Energy を使用した SimpleLink™ CC2340R5 マイコンによる BLDC モーター制御



Ryan Brown and Tarek Darwiche

Connectivity LPCS

## 概要

このアプリケーション ノートは、1 つのマイコン設計でブラシレス DC (BLDC) モーター設計に Bluetooth® Low Energy (LE) 無線プロトコルを追加する機能について説明します。BLDC モーターは一般的に、電動ドリル、リモート コントロール (RC) 自動車、電動歯ブラシなどの組込み無線の利点を活用できるアプリケーションで使用されています。この資料で説明する資料は、[CC2340R5](#) が [DRV8329](#) を使用してこのタスクをどのように実行できるかを示しています。

この資料が提供する実装では、[TI.com](#) で入手可能なハードウェア評価モジュール (EVM) と、[SimpleLink 低消費電力 F3 Demos GitHub](#) で無償で提供されているファームウェアを使用します。必要なハードウェア接続とファームウェア動作の両方について詳しく説明します。これにより、開発者はデモを動作させ、BLDC モーターを取得した後に要件を満たすようにプロジェクトを変更できるようになります。また、追加のテスト データも提供されているため、ユーザーは動作条件とアプリケーション拡張のオプションを十分に理解できます。

## 目次

1 はじめに.....	3
1.1 CC2340R5.....	3
1.2 DRV8329A.....	3
1.3 BLDC モーター.....	4
2 BLDC ハードウェア.....	5
2.1 ハードウェア設定.....	5
2.2 接続図.....	6
3 例を実行する.....	8
3.1 依存関係.....	8
3.2 ファームウェアのロード.....	8
3.3 モーターの接続テスト.....	9
3.4 Bluetooth® LE を使用した、BLDC モーターのホール センサ付きトラップ動作.....	10
4 ファームウェア設計.....	13
4.1 コードフローの説明.....	13
4.2 カスタマイズ済み TI ドライバ.....	14
4.3 アプリケーション イベント.....	15
4.4 整流テーブル.....	15
4.5 モーター加速.....	15
4.6 ADC での動作.....	15
4.7 スピン検出機能.....	16
4.8 レポート統計.....	16
4.9 Bluetooth® LE スタック.....	16
5 テストと結果.....	17
6 まとめ.....	19
7 参考資料.....	19

## 図の一覧

図 1-1. LP-XDS110ET および LP-EM-CC2340R5 の接続.....	3
図 1-2. DRV8329AEVM.....	4

図 1-3. ホール エフェクト センサを使用したブラシレス BLDC.....	4
図 2-1. DRV8329A のハードウェア設定.....	5
図 2-2. 物理ハードウェアのセットアップ.....	7
図 3-1. CCS のプロパティ.....	8
図 3-2. CCS ロード オプション.....	8
図 3-3. PuTTY UART のオプション.....	9
図 3-4. Predefined Symbols.....	10
図 3-5. BLDC モーター、ホール テストなし.....	10
図 3-6. SimpleLink Connect アプリケーション.....	11
図 4-1. BLDC モーターのコード図.....	13
図 5-1. BLDC モーター オシロスコープのスクリーンショット.....	17
図 5-2. BLDC モーターの遷移.....	18

## 表の一覧

表 2-1. CC2340R5 と DRV8329AEVM の間の接続.....	6
表 2-2. BLDC モーターと DRV8329AEVM の間の接続.....	6
表 3-1. SimpleLink™ Connect アプリと UART インターフェイスのアクション.....	12
表 4-1. BLDC モーター アプリケーションの定義.....	14
表 4-2. アプリケーションのルーチンと関数.....	15
表 4-3. Bluetooth® LE スタック構成.....	16
表 5-1. BLDC モーター アプリケーションの性能.....	17

## 商標

SimpleLink™, BoosterPack™, LaunchPad™, and Code Composer Studio™ are trademarks of Texas Instruments.

Bluetooth® is a registered trademark of Bluetooth SIG, Inc.

Arm® and Cortex® are registered trademarks of Arm Limited.

Wi-Fi® is a registered trademark of Wi-Fi Alliance.

Windows® is a registered trademark of Microsoft Corporation.

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

**CC2340R5** は、512kB のフラッシュと 36kB または 64kB の SRAM を搭載した、強力な低コストのマイコン (MCU) で、Arm® Cortex®-M0+ と 2.4GHz の無線を採用しています。これらの機能により、さまざまな無線プロトコルに対応する多数の最終アプリケーションを、単一チップ設計で実現することが可能です。このアプリケーション ノートでは、1 つのインスタンスを取り上げ、このデバイスを使用してより広い可能性を証明します。

**DRV8329** と組み合わせると、Bluetooth® LE 無線通信経由の **CC2340R5** と、ホール エフェクト センサトラップ設計を使用して BLDC モーターを制御できます。このドキュメントでは、このアプリケーションを実現するために必要なハードウェアとソフトウェアの実装、および実現されているオプション機能について詳しく説明します。この資料を読むことで、ユーザーは BLDC モーター制御と **CC2340R5** 開発の両方の詳細を確認し、同様のコンセプトを設計に確実に使用できるようになります。

### 1.1 CC2340R5

**CC2340R** は SimpleLink™ MCU プラットフォームの製品です。本プラットフォームは、シングル コア SDK (ソフトウェア開発キット) と豊富なツール セットを備えた使いやすい共通の開発環境を共有する Wi-Fi®, Bluetooth LE, Thread, Zigbee, Sub-1GHz MCU, ホスト MCU で構成されています。これらのデバイスは、ビルディング オートメーション (ワイヤレス センサ、照明制御、ビーコン)、アセットトラッキング、医療、リテール EPOS (電子 POS)、ESL (電子棚札)、パーソナル エレクトロニクス (玩具、HID、スタイラス ペン) の市場における低消費電力のワイヤレス通信と OAD (Over the Air Download) サポートに最適化されています。

**LP-EM-CC2340R5** 開発キットは、Bluetooth 5 LE および 2.4GHz 専用プロトコルに対応した SimpleLink Bluetooth LE MCU の開発を加速するために使用されます。ソフトウェア サポートは、Code Composer Studio™ IDE を使用して構築できる **SIMPLELINK-LOWPOWER-F3-SDK** によって提供されています。機能には、BoosterPack™ プラグイン モジュール コネクタを介したすべての I/O 信号へのアクセスや、TI SimpleLink Connect を使用して LaunchPad™ 開発キットをスマートフォンに接続する機能が含まれます。プログラミング、デバッグ、および RF 評価を行うには、**LP-XDS110ET** または **LP-XDS110** デバッガ (別売) が必要です。

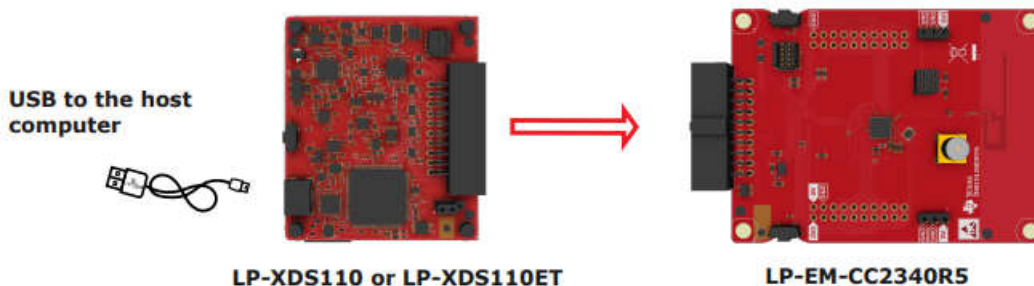


図 1-1. LP-XDS110ET および LP-EM-CC2340R5 の接続

### 1.2 DRV8329A

**DRV8329** ファミリのデバイスは、3 相アプリケーション用の統合型ゲートドライバです。このデバイスには、3 つのハーフブリッジ ゲートドライバがあり、それぞれがハイサイドとローサイドの N チャネル パワー MOSFET を駆動できます。このデバイスは、内部のチャージ ポンプを使用して適切なゲート駆動電圧を生成し、ブートストラップ回路を使用してハイサイド MOSFET をオンさせます。100% デューティ サイクルをサポートするためにトリクル チャージ ポンプを備えています。このゲートドライバのアーキテクチャは、最大でソース (供給) 1A、シンク (吸い込み) 2A のピーク ゲート駆動電流をサポートします。**DRV8329** は単一電源で動作でき、4.5V~60V (DC) の広い入力電源電圧範囲に対応します。

**DRV8329AEVM** は、BLDC モーター向けの **DRV8329A** ゲートドライバをベースとする、30A の 3 相ブラシレス DC ドライブ段です。**DRV8329** は、ブートストラップ動作向けの 3 個のダイオードを内蔵しており、外部ダイオードは必要ありません。このデバイスは、ローサイド電流測定用の電流シャント アンプ、80mA 低ドロップアウトレギュレータ (LDO)、デッドタイム制御ピン、VDS 過電流レベル ピン、ゲートドライバ シャットオフ ピンを搭載しています。この評価基板 (EVM) には、これらの設定を評価するためのスイッチ、ポテンシオメータ、抵抗が付属しており、**DRV8329** デバイスの A バリエーション (6 個の PWM) と B バリエーション (3 個の PWM) に対応する構成能力もあります。最大 60V の電圧をこの評価基板 (EVM) に供給できます。また、**DRV8329** の内蔵 LDO は、ブートストラップの GVDD 電源電圧が必要とするゲート電圧

を生成します。ユーザーへのフィードバック目的で、各電源に対応する複数のステータス LED と 1 個の障害 LED を搭載しています。

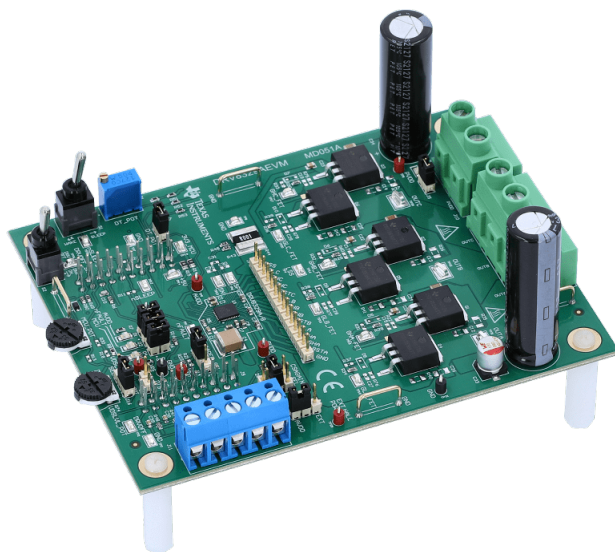


図 1-2. DRV8329AEVM

### 1.3 BLDC モーター

ブラシレス DC (BLDC) 電動モーターは、直流 (DC) 電源を使用する同期モーターです。これらには、永久磁石を回転させるための磁界を生成する 3 つのモーター ステータに DC 電流を切り替えるコントローラが必要です。このドキュメントでは、インターフェイス回路はユニポーラドライブ方式で 3 つの個別の位相のハイ サイドとロー サイドに対して 6 つの PWM を駆動します。このアプリケーション ノートで主に取り扱っているのは、ホール エフェクト センサの設計を使用して回転子の位置を測定し、それに応じて整流テーブルを沿った遷移を測定することです。このため、プロジェクトを意図したとおりに実行するには、ホール エフェクト接続を備えたモーターが必要です。ホール エフェクト センサを使用せずに基本的な BLDC モーターをサポートするには、さらに変更が必要です。

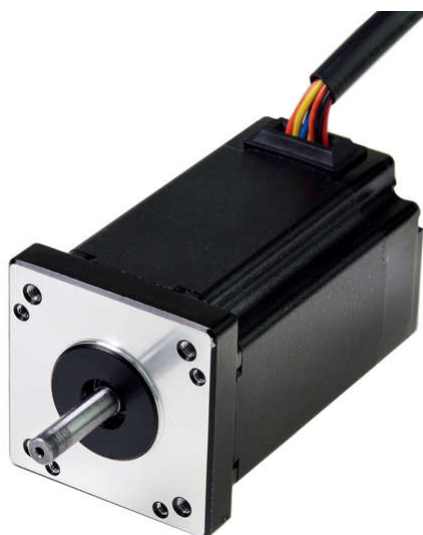


図 1-3. ホール エフェクト センサを使用したブラシレス BLDC



## 2 BLDC ハードウェア

### 2.1 ハードウェア設定

以下のセクションでは、入手が必要なハードウェア、EVM 上で実施する設定変更、デフォルトのファームウェア設計を変更せずにサンプルを実行するために必要な接続について説明します。

#### 2.1.1 DRV8329AEMV の設定

図 2-1 に、適切な場所にジャンパを実装した DRV8329AEMV 基板を示します。また、BLDC モーターを動作させようとするときに問題を回避するために、ボード スイッチとスライダをどのように配置するかについてのコメントも掲載しています。このハードウェアとのインターフェイス方法の詳細については、『DRV8329AEMV ユーザー ガイド』を参照してください。

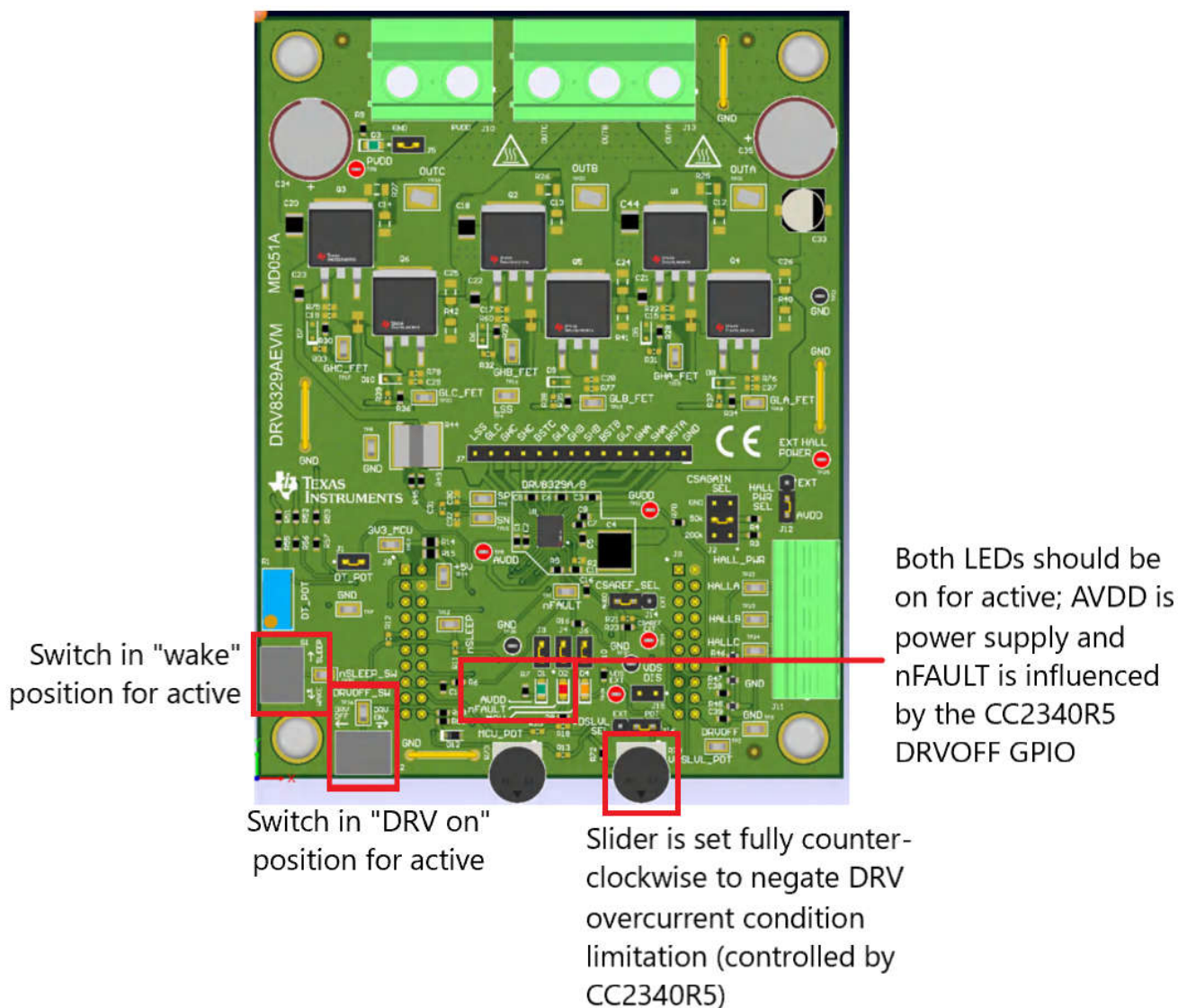


図 2-1. DRV8329A のハードウェア設定

## 2.2 接続図

表 2-1 は、ステッパ モーター デモを実現するための DRV8329AEVM と CC2340R52 間の接続です。

**表 2-1. CC2340R5 と DRV8329AEVM の間の接続**

接続	CC2340R5 機能	CC2340R5 ピン	DRV8329AEVM
位相 A HS	PWM 出力	DIO24	INHA
位相 A LS	PMW 出力	DIO8	INLA
位相 B HS	PWM 出力	DIO12	INHB
位相 B HL	PWM 出力	DIO21	INLB
位相 C HS	PWM 出力	DIO6	INHC
位相 C LS	PWM 出力	DIO11	INLC
HALL A	デジタル割り込み入力	DIO23	HALLA
HALL B	デジタル割り込み入力	DIO18	HALLB
HALL C	デジタル割り込み入力	DIO13	HALLC
バス電圧	ADC 入力	DIO7	VSENPVDD
A 相の電圧	ADC 入力	DIO1	VSENA
B 相の電圧	ADC 入力	DIO2	VSENB
C 相の電圧	ADC 入力	DIO5	VSENC
シャント電流	ADC 入力	DIO0	ISENA
DRV フォルト	デジタル割り込み入力	DIO14	nFAULT_49C
UART TX	UART 送信出力	DIO20	該当なし
UART RX	UART 受信入力	DIO22	該当なし
共通の GND	GND 接続	GND	GND

BLDC モーターの配線は、指定の DRV8329AEVM ピンに接続する必要があります。必要なモーター ワイヤと対応するピン接続を表 2-2 に示します。特定のモーターについては、どのワイヤがこれらの機能を実装しているかを確認し、それに応じて接続してください

**表 2-2. BLDC モーターと DRV8329AEVM の間の接続**

モーター ワイヤ	DRV8329AEVM コネクタ	モーター イメージの例の色
VCC	HALL_PWR	赤
HU/PH1	HALLA	オレンジ
HV/PH2	HALLB	黄
HW/PH3	HALLC	青
GND	GND	黒
U	OUTA	オレンジ
V	OUTB	黄
W	OUTC	青

最終結果は、図 2-2 のようになります。これらの GPIO は他の目的で使用されているため、LED ヘッダー ジャンパは LP-EM-CC2340R5 から取り外されていることに注意してください。DRV8329AEVM 上の J10 を介して有効な電源を供給すると、緑色の AVDD (D1) および赤色の nFAULT (D2) LED が点灯するように、トグル スイッチ S1 と S2 の両方をそれぞれ *wake* 位置と *on* 位置に向ける必要があります。

CC2340R5 が DRVOFF ピンを High に設定することで DRV モーター ドライバをイネーブルにすると、nFAULT LED はクリアされます。nSLEEP ラインと DRVOFF ラインは DRV8329A への入力であり、EVM のトグル スイッチ ハードウェアを使用して制御しますが、ファームウェア開発を重ね、CC2340R5 を使用して制御することもできます。

CC2340R5 DIO6 ピンは、BLDC モーターを制御する PWM として使用されますが、LP-EM-CC2340R5 はこのピンをオンボードの外部フラッシュ デバイスの nCS に接続することもできます。そのため、LaunchPad の R24 と R25 を実装しない限り、消費電力が増大します。

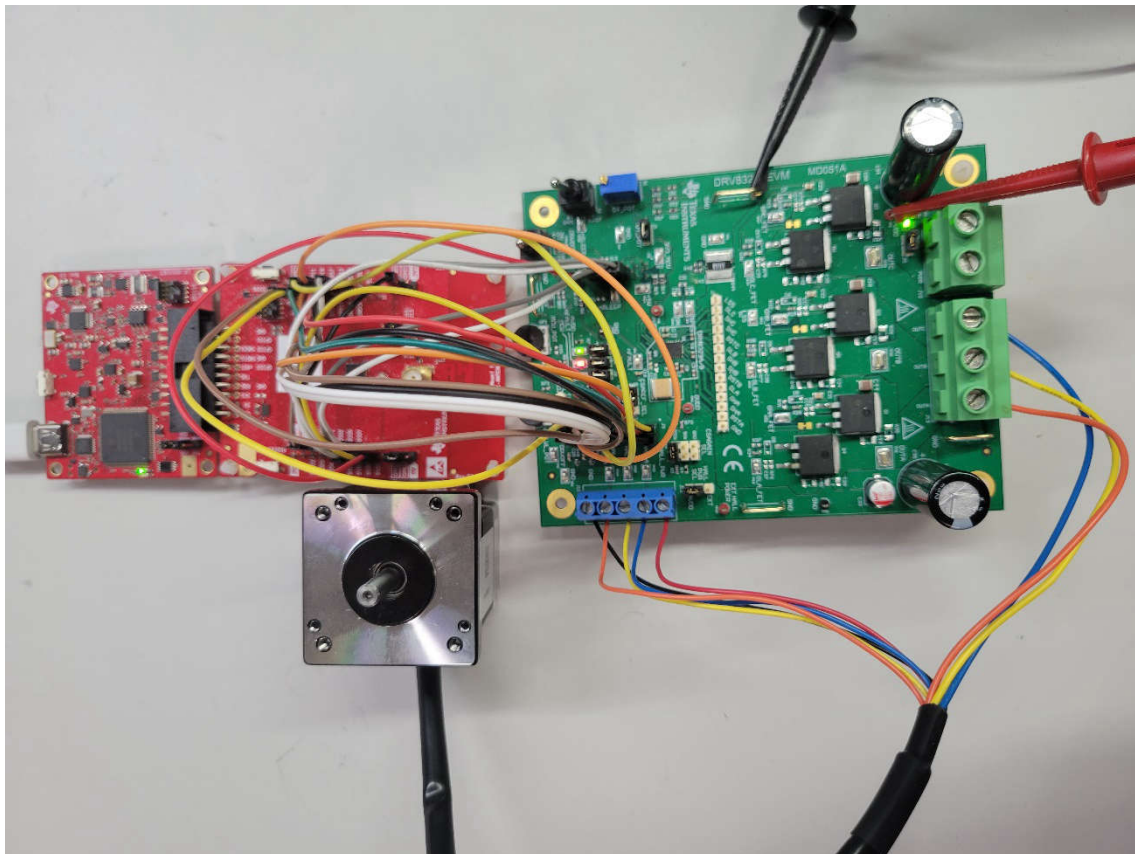


図 2-2. 物理ハードウェアのセットアップ

## 3 例を実行する

次のセクションでは、ファームウェアの詳細、および各コンポーネントが BLDC モーターの駆動とデータの収集にどのように機能するかについて説明します。

### 3.1 依存関係

SimpleLink 低消費電力 F3 デモ [GitHub](#) で提供されるコードプロジェクトは、SimpleLink F3 SDK v8.40.0.61、SYSCONFIG、TI CLANG v4.0.0 コンパイラを使用しています。プロジェクトを Code Composer Studio™ (CCS) v20 以降にインポートする前に、これらの依存関係がすべてマシンにインストールされていることを確認します。環境設定の詳細な例については、[SimpleLink Academy for CC23xx](#) を参照してください。上記以外の依存バージョンの移行およびサポートは、ユーザーが責任を負うことに注意してください。

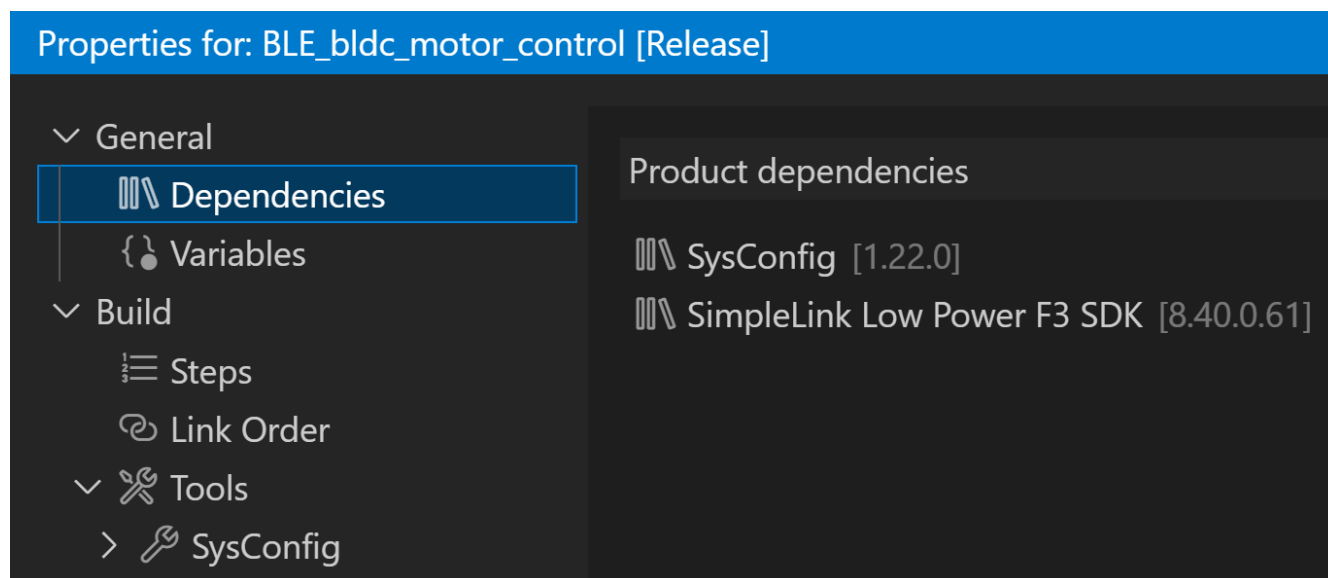


図 3-1. CCS のプロパティ

### 3.2 ファームウェアのロード

CCS 内でビルドされたプロジェクトは、IDE 上で「Run」(実行) → 「Flash Project」(プロジェクトのフラッシュ) (Ctrl + F5) または「Debug Project」(プロジェクトのデバッグ) (F5) を選択することで直接ロードできます。プロジェクトをアクティブにデバッグしていない場合は、フリーラン実行を許可するためにデバッグ モードを終了することを推奨します。バイナリイメージをロードするために、[UNIFLASH](#) ソフトウェア ツールの使用を検討してください。

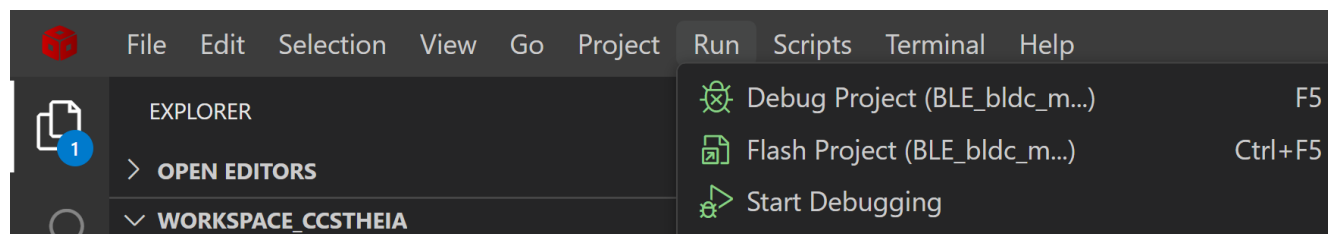


図 3-2. CCS ロード オプション



### 3.3 モーターの接続テスト

プロジェクトに付属するテスト用ファームウェア セットアップが付属しており、DRV8329AEVM (OUTA、OUTB、OUTC) のモーター位相接続の向きをテストし、ホール エフェクト センサの割り込み (HALLA、HALLB、HALLC) を監視して、接続方向が正しいかどうかを判定します。これを発生させるには、位相テーブルとホール センサの両方の位置を出力し、UART 端末で設定(921600 ボー、8 データビット、1 ストップビット、パリティなし、フロー制御なし)を使用します。COM ポートは、コンピュータ システムに表示されるアプリケーション UART と同じです。PuTTY ソフトウェアは、以下の例では Windows® オペレーティング システムで使用していますが、正しい設定を使用している限り、任意の UART 端末またはオペレーティング システムを使用できます。

モーターが回転していない間にスタンバイ消費電力を確保するため、アプリケーションで UART を完全に無効化する必要があります。それ以外の場合、RX ピンを常に監視するためにアクティブ モードが必要です。

`app_blcdc_motor_control.c` ファイルで `USE_UART` を定義解除することで、UART をアプリケーション コードから削除できます。

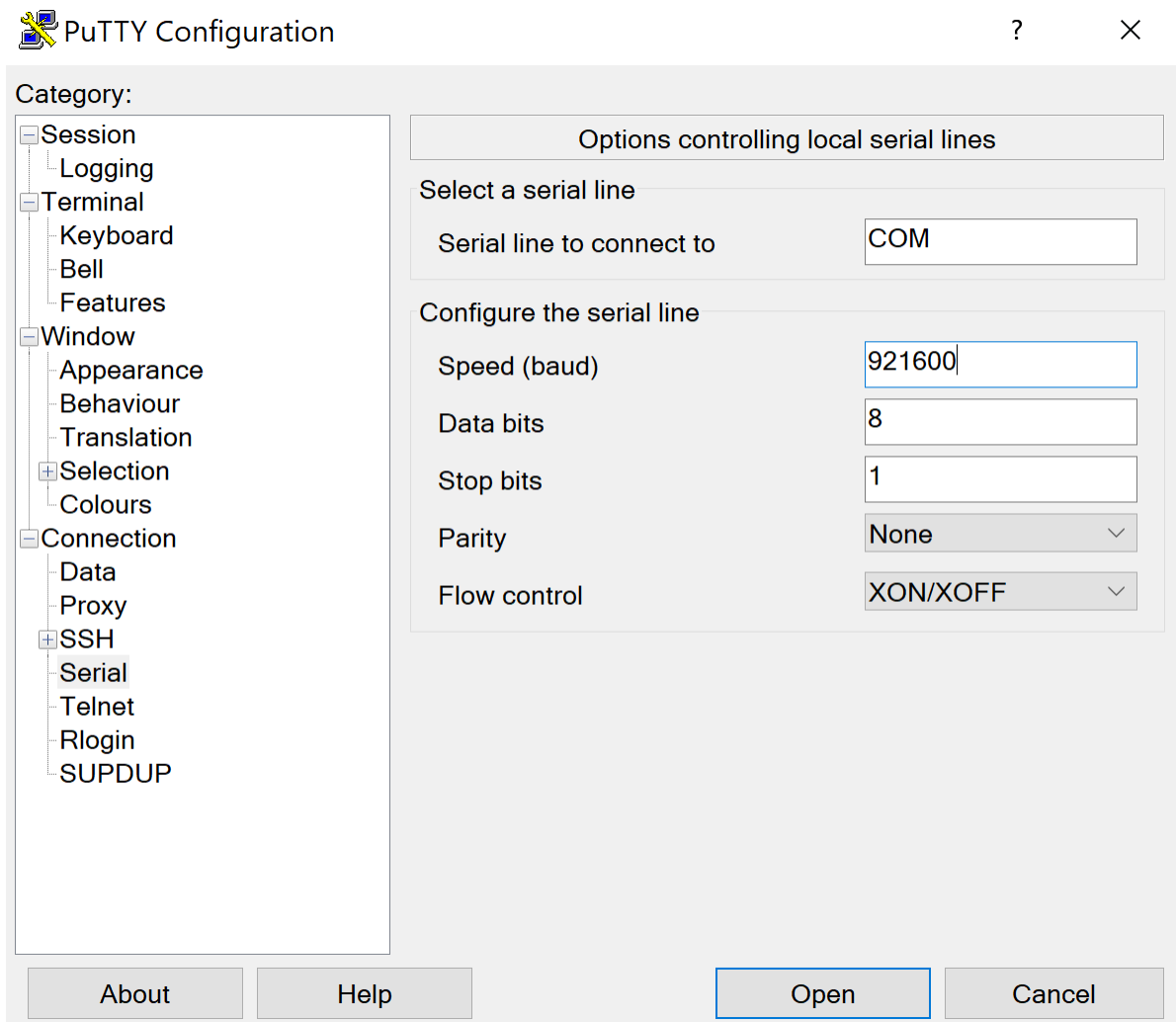


図 3-3. PuTTY UART のオプション

このオプションを有効にするには、「Project Properties」(プロジェクトのプロパティ) → 「Build」(ビルド) → 「Tools」(ツール) → 「Arm Compiler」(Arm コンパイラ) → 「Predefined Symbols」(事前定義済みシンボル) の中に `TEST_NO_HALL` を追加します。次に、プロジェクトを再ビルドしてロードします。

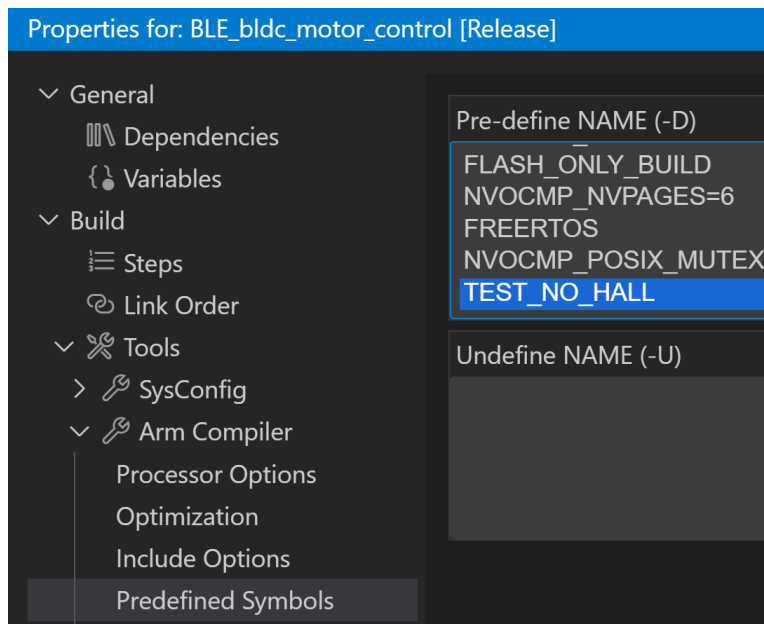


図 3-4. Predefined Symbols

BLDC モーターのメイン メニューが表示されたら、キーボードの **s** を押し (UART 端末ウィンドウがアクティブに選択されていることを確認してください)、正方向の 10% のデューティ サイクルでモーターを起動します (回転子シャフトを直接見ているとき、フォワードは反時計回りになっていると見なされます)。モーターの方向が予想値と逆になっている場合、OUT モーター位相のうち 2 つを入れ替えることを検討してください。それ以外の場合は、**s** をもう一度押してモーターを停止し (または LaunchPad をリセットし)、UART 出力が図とまったく同じシーケンスに従うまで DRV8329AEVM の HALL 入力に繰り返し変更を加えます。これにより、ホール センサがイネーブルになったときに BLDC モーターが想定どおりに動作するようになります。

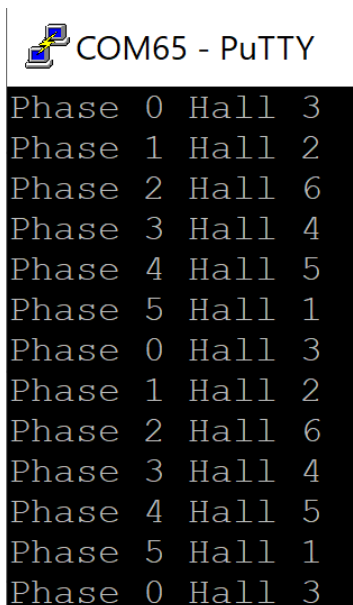


図 3-5. BLDC モーター、ホール テストなし

### 3.4 Bluetooth® LE を使用した、BLDC モーターのホール センサ付きトラップ動作

正しいモーター接続がすべて確認されたら、ユーザーはデフォルトの例を実行できます。プロジェクトを再ビルドしてロードする前に、TEST\_NO\_HALL 事前定義が削除されていることを確認してください。

### 3.4.1 SimpleLink™ Connect 電話アプリケーション

このデモは、(SimpleLink™ Apple Store または SimpleLink™ Play Store のどちらかで) すぐに使用できる SimpleLink Connect 電話アプリケーションを使用し、BLDC モーターのファームウェアと対話します。CC2340R5 デバイスを実行した後、スキャナ機能を使用して Bluetooth LE/BLDC モーター デバイスを見つけて接続することができます。次に、TI Simple Peripheral Service (TI シンプル ペリフェラル サービス) を選択し、関連する特性へのアクセスをオープンにします。

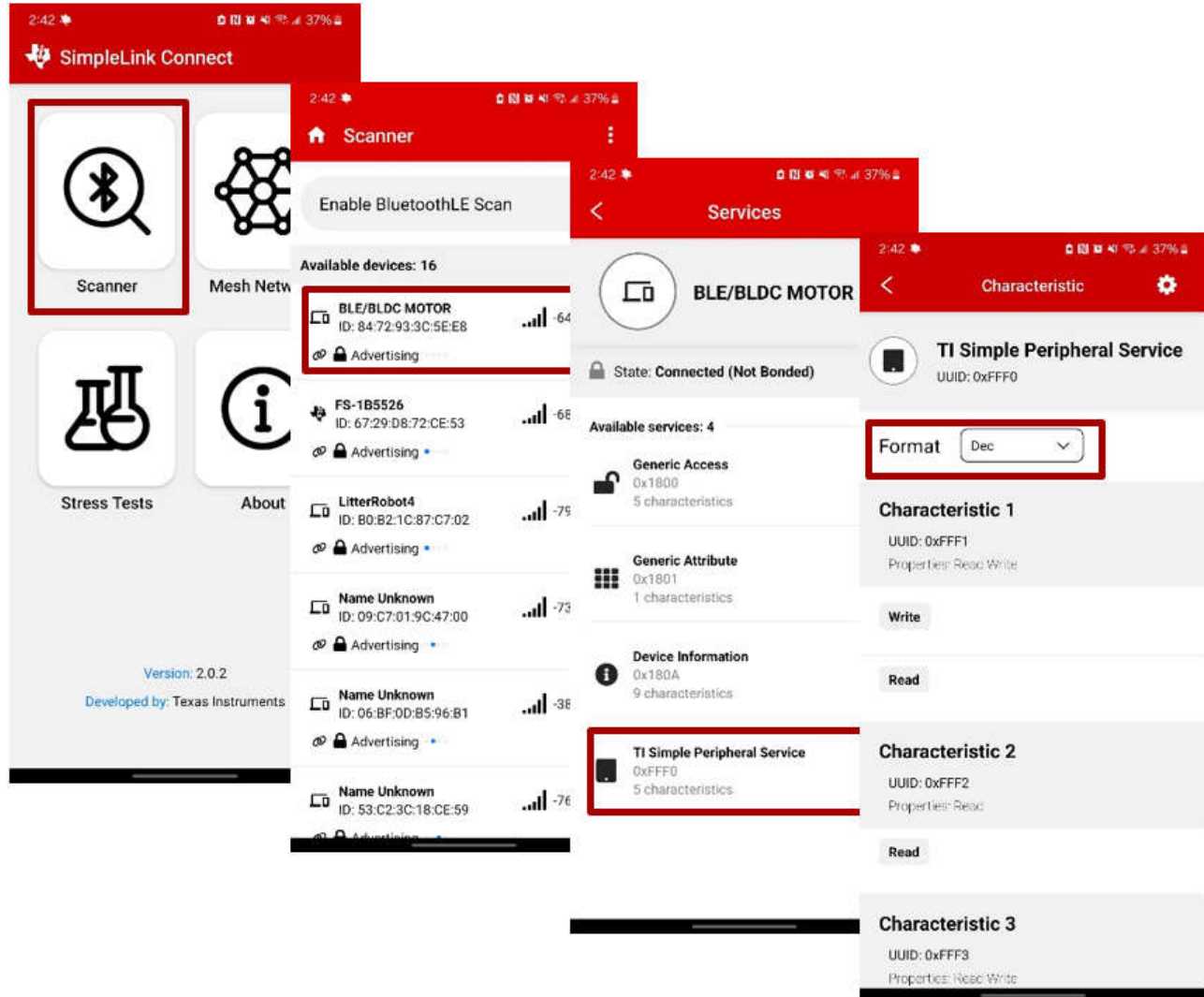


図 3-6. SimpleLink Connect アプリケーション

このプロジェクトでは、カスタム Bluetooth LE プロファイルを作成しました。このプロファイルは、SDK の基本的な Bluetooth LE の例 (*simple\_gatt.c*) にある Simple GATT プロファイルに基づいており、各特性のプロパティを大幅に変更して、このプロジェクトの目的の機能を実現しています。SimpleLink™ Connectivity 電話アプリケーションでは、このアプリケーションは TI Simple Peripheral Service と表示されます。表 3-1 は、モーターと相互作用するために使用可能なすべての Bluetooth LE オプションを反映します。該当するのは、以前のモーター接続テスト セクションで説明した Bluetooth LE 特性や UART インターフェイスなどです。

**表 3-1. SimpleLink™ Connect アプリと UART インターフェイスのアクション**

アクション	SimpleLink Connect インターフェイス	UART キーボード エントリ
モーターを始動します	特性 1 に 1 を書き込みます	s (モーターが停止している場合)
モーターを停止します	特性 1 に 0 を書き込みます	s (モーターが動作している場合)
RPM を表示	読み取り特性 2 (16 進形式)	l (文字)
順方向	特性 3 に 1 を書き込みます	f
逆方向	特性 4 に 0 を書き込みます	r
デューティ サイクル パーセンテージ	特性 5 を書き込みます (Dec 形式)	該当なし
デューティ サイクルを増やします	該当なし (特性 5 を参照)	u
デューティ サイクルを減らします	該当なし (特性 5 を参照)	d

## 4 ファームウェア設計

デフォルトの BLDC モーター プロジェクトが意図したとおりに動作しているため、ファームウェアをさらに分析します。

### 4.1 コードフローの説明

図 4-1 は、CC2340R5 コード内で使用されるプロセスのシンプルなコード ブロック図です。この機能は、`app_bldc_motor_control.c` ファイルで実現されます。

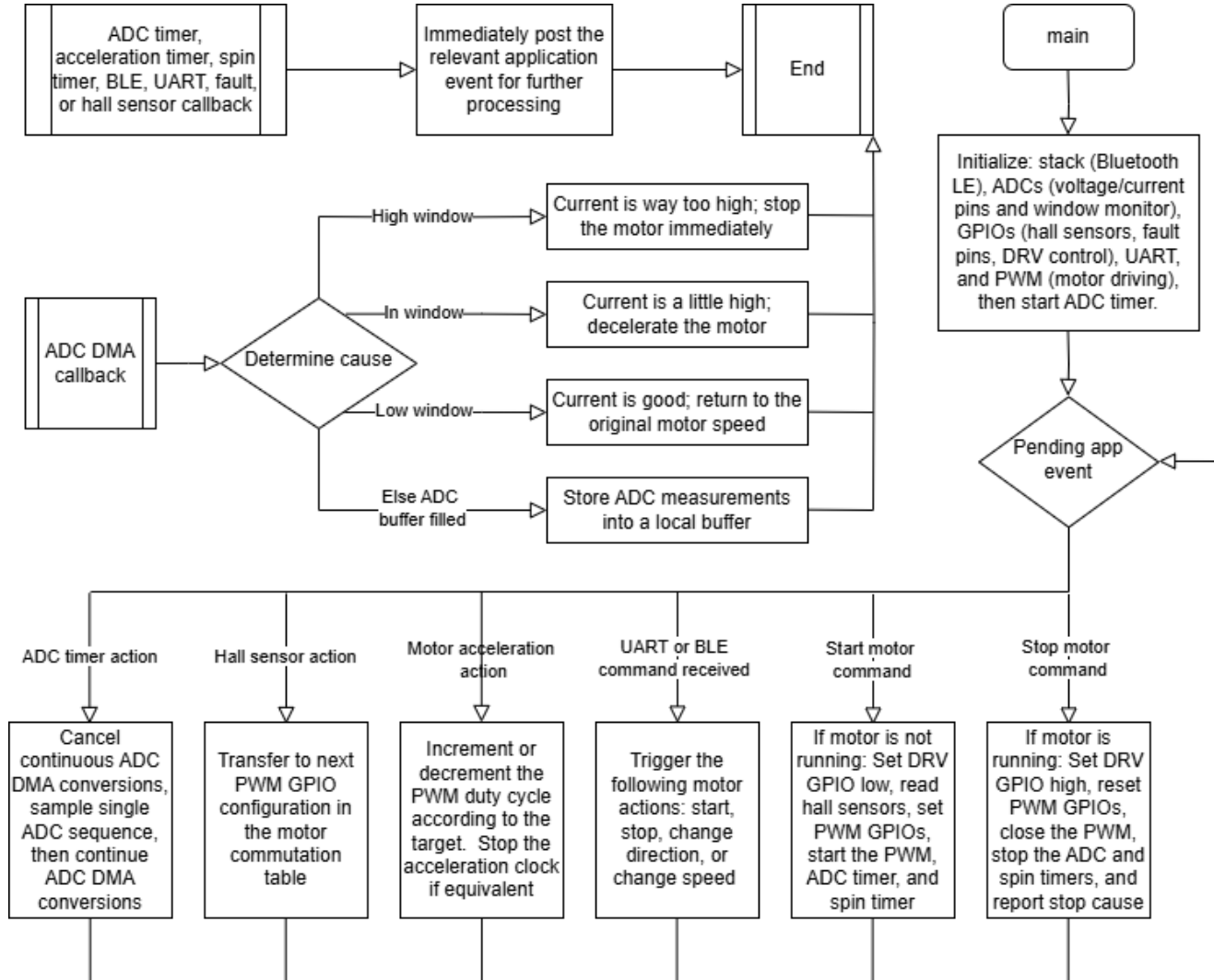


図 4-1. BLDC モーターのコード図

`main` 関数は、BLDC モーターのサンプルが動作するために必要なすべての TI ドライバとタイマを初期化します。`main` の `while` ループに入ると、ハードウェア コールバックによってイベントが設定されるまで、後続の処理を保留できます。サブルーチンを介して対応するアクションを処理した後、イベントがリセットされ、プロセスが繰り返されます。

いくつかの例外を除き、すべてのハードウェア コールバックは単にイベントをポストし、メイン アプリケーションで処理されます。例外は `ADCBuf` コールバックであり、このコールバックはステータスを即座に処理し、ウィンドウ コンパレータのスレッショルドを超えた場合を除いて、メイン アプリケーション ループでの追加処理を要求しません。

モーターの動作を定義する複数の定義が実装されています。これらの定義については、以下のセクションで参照してください。表 4-1 に、`app_bldc_motor_control.h` にある構成可能な定義を示します。



**表 4-1. BLDC モーター アプリケーションの定義**

定義	デフォルト	単位	機能
PWM_PERIOD	40000	Hz	PWM 周期を 2 で割った値 (アップ モードまたはダウン モードの場合)、したがって 20kHz
DUTY_MAX	80	%	最大許容 PWM デューティ サイクル
DUTY_MIN	10	%	最大許容 PWM デューティ サイクル
DUTY_INC	5	%	UART コマンドによる PWM デューティ サイクル インクリメント
MAX_STRING	20	整数	1 つの UART スtring 内の最大文字数
SPIN_TIMEOUT	1000000	μs	この期間中にモーターが 1 回転しなければ、モーター ドライバは停止します
RPM_INTERVAL	500000	μs	RPM、CPU 負荷、ADC 測定カウントが報告される時間間隔
ACCEL_INTERVAL	10000	μs	PWM デューティ サイクルを増加または減少させる前の時間遅延
CHANGE_DELAY	1000000	μs	方向を変更する前に、モーターを停止した後の遅延
ADC_COUNT	5	整数	有効な ADC チャンネル数
ADC_INTERVAL	100000	μs	単一の ADC シーケンスをサンプリングする時間間隔 (ADC DMA バッファを停止してから再起動する必要があります)
VSEN_THRESHOLD	2000	整数	モーター停止をトリガするために必要な未加工電源の ADC 値
WINDOW_LOW	1500	整数	ロー ウィンドウ モニタの未加工電流 ADC 値
WINDOW_HIGH	3000	整数	ハイ ウィンドウ モニタの未加工電流 ADC 値
ADCSAMPLESIZE	128	整数	DMA によって転送される ADC バッファのサイズ

## 4.2 カスタマイズ済み TI ドライバ

BLDC モーター動作に適した方法で機能するようにカスタマイズ済みの TI ドライバがいくつかあります。修正されたドライバ アプリケーション ファイルがプロジェクト ディレクトリに含まれているため、SDK リリースにパッケージされているオリジナル ファイルの代わりにファイルが使用されます。

### 4.2.1 PWM

PWM TI ドライバは元々、タイマから 1 チャンネルの出力のみをアップ モードで駆動するように設計されています。BLDC モーターの目的のために、*PWMTimerLPF3.c* が変更され、(3 つのモーター位相を表す) 3 つのチャンネル出力すべてを 1 つのタイマから駆動します。また、ドライバは、デューティ サイクルの中間にモーター電流の測定をトリガするために重要なため、アップ モードまたはダウン モードも使用します。PWM\_PERIOD (アップ モードまたはダウン モードで 2 で分周した後の 20kHz) は、BLDC モーターを駆動するときの標準的な PWM 周波数です。モーター設計に深く精通していない限り、この値を変更することはお勧めしません。

### 4.2.2 ADCBuf

デフォルトの ADCBuf は、反復シングル モードで 1 つのチャンネルで動作します。このアプリケーションでは、電流 (ISEN) と電圧 (VSEN PVDD) という 2 つの連続的なチャンネル変換が保証されているため、反復シーケンス モードが推奨されます。したがって、*ADCBufLPF3.c* は、データが単一のメモリ レジスタではなく、ADC ペリフェラルの FIFO を介して転送されるようにも構成されています。

元の TI ドライバは、その後の ADC 変換を自動的に開始することも選択します。PWM デューティ サイクルの途中で電流を測定する必要があるモーター設計の目的において、これは優れた選択肢ではありません。したがって、アップ モードまたはダウン モードで目標値に達すると、PWM TI ドライバの LGPT で最初の ADC チャンネル (ISEN) の変換がトリガされるように ADC が設定されます。次の ADC チャンネル (VSEN PVDD) はその直後に測定されます。このプロセスは、バッファが満たされるまで続行されます。

ADCBuf コールバック動作には、*app\_bldc\_motor\_control.c* ファイルで初期化されるウィンドウ モニタの上限割り込みのステータス処理も含まれます。これにより、ウィンドウモニタが変更されたこと、または DMA を介して ADC バッファが完了したことをアプリケーションに通知できます。

### 4.2.3 電源

カスタム ポリシー関数が `app_bldc_motor_control.c` に追加され、CPU がアクションを実行していないときにアイドル状態に費やされた時間数をカウントします。これは、アプリケーションの合計 CPU 使用率の測定値としてユーザーに報告されます。`bldc_motor.syscfg` SysConfig ファイル パワー モジュールは、このカスタム ポリシー関数を直接参照して、この関数が使用されます。この機能は、テストおよび評価目的でのみ使用します。

### 4.3 アプリケーション イベント

ユーザー操作またはハードウェア コールバックがイベントを設定すると、次のルーチンが処理されます。

表 4-2. アプリケーションのルーチンと関数

ルーチン	機能
ACTION_UART	すでに述べたユーザー入力を通じて受信した Bluetooth LE または UART 機能のいずれかを処理します。開発者は、このセクションを変更して、機能とユーザー インターフェイスを決定できます。
ACTION_CONTINUE	DRV ピンを Low に駆動し、最小デューティ サイクルを設定し、正しいモーター GPIO 状態を初期化すること、アプリケーション タイマを開始して、PWM TI ドライバを開くことで、モーターを起動します。PWM TI ドライバがすでに開かれている場合、このルーチンは実行されません。
ACTION_STOP	DRV ピンを High に駆動し、モーターの GPIO ピンをリセットし、関連するアプリケーション クロックを停止し、PWM TI ドライバを停止して、モーターを停止します。これにより、UART 経由で停止の原因も報告します。
ACTION_HALL	ホール センサ ピンのコールバック後、すべてのホール センサ状態を読み取り、それに応じてモーター整流テーブルに従います。
ACTION_PWM	これが目標と一致するまで、加速クロックが経過するたびにデューティ サイクル値をインクリメントまたはデクリメントし、その後、達成されると加速クロックを停止します。

### 4.4 整流テーブル

ホール センサの GPIO が High から Low、または Low から High に遷移するたびに、モーター整流テーブルを使用して、次のモーター位相状態が選択されます。したがって、`reversePhaseTable` 配列または `forwardPhaseTable` 配列は、3 つのホール センサすべてのビットマスク値を受け取り、次のケースを `forwardDirection` 関数または `reverseDirection` 関数で処理することを決定します。これらの関数は、ハードウェア レジスタの直接コマンドを使用して、各モーターの相を正しく、かつ想定の構成に設定するだけです。

### 4.5 モーター加速

たとえば 20% ~ 80% のように、モーターは瞬間的なデューティ サイクルを大幅に変化させることはできないため、加速クロックを使用して `ACCEL_INTERVAL`  $\mu$ s ごとにデューティ サイクルを 1% ずつインクリメントまたはデクリメントします。デューティ サイクル出力が意図した目標と一致すると、このクロックは停止します。アプリケーションは、デューティ サイクルが `DUTY_MIN` および `DUTY_MAX` 内に維持されるように、注意を払って行われます。`ACCEL_INTERVAL` を小さくすると、モーター遷移のデューティ サイクルは速くなりますが、そうすることで電流スパイクが増加する可能性があります。

### 4.6 ADC での動作

メモリレジスタ (4 つ) よりも ADC 入力 (`ADC_COUNT` は 5) が多く、各 PWM デューティ サイクル出力 (つまり `PWM_PERIOD/2` のレートで) で `ISEN` ピン (モーター電流) を測定する必要があることを考慮すると、1 つの `CC2340R5` ADC ペリフェラルを使用して、関連するすべての ADC チャンネルを測定することが課題となります。修正された実装では、改良型 `ADCBuf` TI ドライバを使用して、ADC と DMA を利用して、CPU を中断せずにサイズの `ADCSAMPLESIZE` バッファを迅速かつ効率的に充填し、`ISEN` と `VSEN` の両方 (システム電圧) を連続的に測定しながら、ときどきこのプロセスを中断し、`ADC_INTERVAL` のレートで残りの `VSENA`、`VSENB`、`VSENC` ADC ピンを測定します。その結果、`ADC_INTERVAL` タイマが満了するたびにトリガされる 1 つの ADC 測定の時間と周波数により、`ADCBuf` を使用するチャンネルごとの ADC 測定数は `PWM_PERIOD / 2` よりもわずかに低くなります。`ADCBuf` ドライバは PWM LGPT アクティビティから ADC をトリガするよう構成されているため、モーターが停止している間、ドライバは測定を行いません。一方、単一の ADC 測定は、モーターの動作に関係なく、`ADC_INTERVAL` が発生するたびに有効となります。

ADC ペリフェラルでもウィンドウ モニタがイネーブルになるため、`adcBufCallback` でさらに処理されるように、最新の `ISEN` 測定の値がウィンドウ制限範囲内またはそれ以上になった場合にアクションを即座に実行できます。

WINDOW\_LOW の値を超えた場合、ADC\_INT\_INIFG の ADC コールバック ステータスによって、サイクルがウィンドウ値を下回るまで PWM デューティ サイクルが低下します。この時点で、ADC\_INT\_LOWIFG ステータスは、デューティ サイクルを元の値に戻します。WINDOW\_HIGH を超えた場合、ADC\_INT\_HIGHIFG ステータスはモーターを直ちに停止します。ADC メモリ レジスタはウィンドウ モニタをトリガできますが、設定できるのは、下限ウィンドウおよび上限ウィンドウ スレッシュホールドの値は 1 つだけです。ウィンドウ モニタが有効になっているメモリ レジスタ 2 を使用するように ISEN のみが構成されているため、ISEN のみがウィンドウ モニタをトリガできます。ISEN が 2 つの ADC メモリ レジスタ (0 と 2) を切り替えると、ウィンドウ モニタは約 10kHz の速度で ISEN を評価します。

ウィンドウ モニタのステータスが原因ではなく、ADC コールバックがトリガされると、ADC バッファは充填されました。アプリケーションは、ADC 変換の結果を outputBuffer アレイに入力し、VSEN\_THRESHOLD を上回る受信 VSENPVDD ADC 変換が原因でモーターを停止する必要があるかどうかを評価します。すべての変換は 12 ビットの分解能 (最大値 4096) で、未加工形式のままになります。ただし、これらの値をさらに調整し、それらの値を  $\mu\text{V}$  形式に変換するために、追加の処理 API を TI ドライバで使用することもできます。これらのプロセスがモーター ドライブ機能を妨げることがないように注意する必要があります。

## 4.7 スピン検出機能

モーターが起動するたびにアプリケーション タイマが起動し、SPIN\_TIMEOUT  $\mu\text{s}$  ごとにモーターが 1 回転を完了しない場合、タイムアウトします。これは、モーターがアクティブに回転していない場合に、開放された PWM TI ドライバがモーター相にアクティブに出力されるのを防止するためです。整流テーブルが完全に繰り返されるたびにタイマは再起動し、他のアプリケーション方式によってモーターを停止させるよう命令されると、タイマは停止します。

## 4.8 レポート統計

各 RPM\_INTERVAL ( $\mu\text{s}$ ) 統計は、それぞれのサブプロセスの数に基づいて更新されます。現在有効になっている統計情報には、CPU 負荷をパーセンテージ (cpuLoad) で表したものと RPM を 1 分あたりの回転数 (rpmValue) で表したものが含まれます。別のオプションは、1 秒あたりで取得される ADCBuf 測定 of 合計 (adcTotal) です。これらは、UART または Bluetooth LE インターフェイスで決定されたとおりにユーザーに通知されますが、アクティブなデバッグ セッションの [Expressions] ウィンドウ内で表示することもできます。約 25k RPM を超えるモーターを使用する場合、統計情報が報告される方法に注意してください。これによりフォルトが発生する可能性があるためです (DRV8329A がモーターの動作をシャットダウンし、CC2340R5 への nFAULT ピン入力を Low に駆動することが示されます)。そのため、テキサス インストルメンツでは、アプリケーションに対しては Bluetooth LE を使用して報告されるデータを短く簡潔にすることを推奨しています。

## 4.9 Bluetooth® LE スタック

この例は、SimpleLink 低消費電力 F3 SDK v8.40 にある、すぐに使用できる [Basic Bluetooth® LE \(BLE\) プロジェクト](#) と [Data Stream Bluetooth LE \(BLE\)](#) の各サンプルをベースにしています。ただし、このアプリケーション ノートで説明しているプロジェクトの目標を達成できるように、Bluetooth LE プロファイルに大幅な変更が加えられました。Bluetooth LE スタックに関連しないすべてのプロジェクト機能は、タスクの飢餓や Bluetooth LE 接続に影響を及ぼすリスクを防止するために、Bluetooth LE タスク コンテキストの外部で処理されました。このデバイスは Bluetooth LE ペリフェラルとして機能し、以下のパラメータを使用してアドバタイズしますが、これらは高度に構成可能です。Bluetooth LE でのスキャンとアドバタイズの詳細については、SimpleLink Academy Scanning and Advertising ラボを参照してください。

表 4-3. Bluetooth® LE スタック構成

Bluetooth® LE コンポーネント	値
アドバタイズメント タイプ	Bluetooth LE を接続し、Bluetooth LE 非指向性アドバタイズメントをスキャンします
プライマリ PHY 間隔の最小値と最大値	100ms、100ms
プライマリ PHY	1M PHY

app\_connection.c ファイル内で接続確立済みイベントを受信すると、CC2340R5 は Bluetooth LE セントラルに交換 MTU 要求を送信し、Bluetooth LE プロファイルの Bluetooth LE 特性と連携しているときに送信できるデータの最大サイズを増やすことができます。これを実行するために、Syscfg ファイル内の Bluetooth LE スタックの一般構成では、PDU の最大サイズを 255 に増やしました。これは、[セクション 5](#) で説明しているスループット テストで大規模に利用されました。

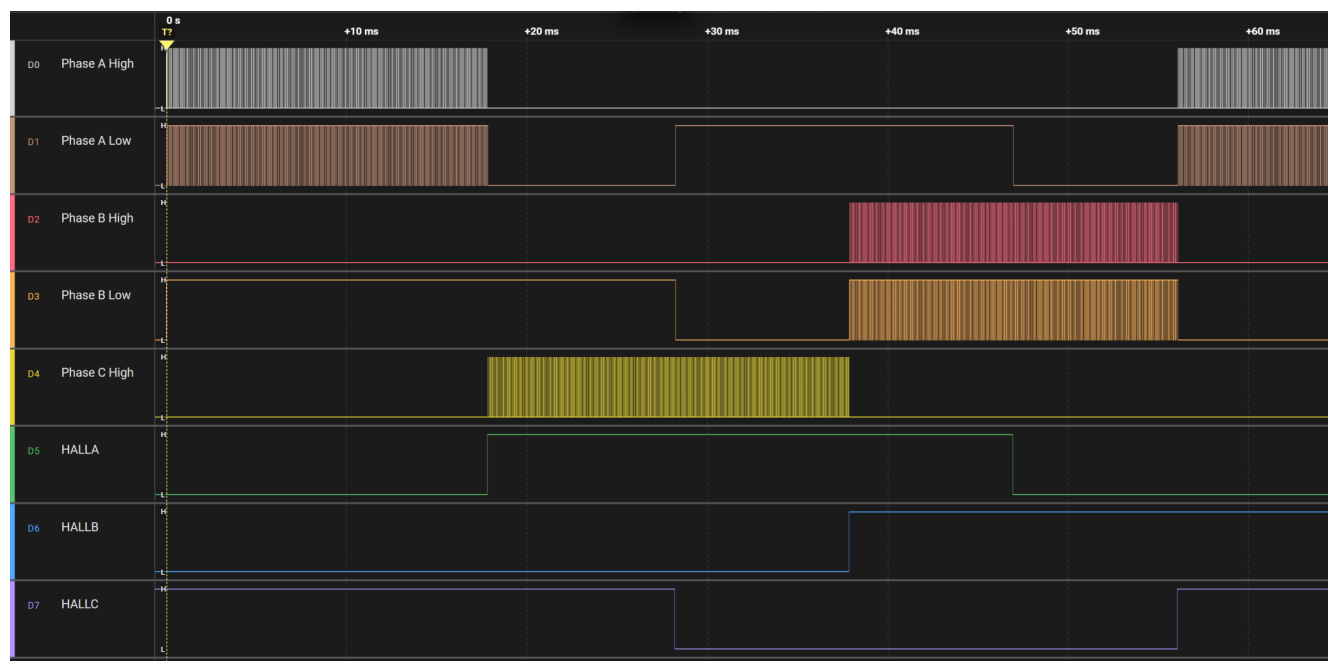
## 5 テストと結果

さまざまなデューティ サイクル出力に対応するために、アクティブ Bluetooth LE 接続を使用した CPU 負荷と RPM の測定値を記録しており、表 5-1 に示されています。デューティ サイクルに対する RPM は、使用する BLDC モーターの物理的特性と特性に大きく依存することに留意してください。

**表 5-1. BLDC モーター アプリケーションの性能**

デューティ サイクル (%)	RPM	CPU 負荷 (%)
10	1080	3
30	5040	5
50	10800	8
70	15120	11
90	23520	15

ロジック アナライザを使用すると、ホール エフェクト センサのレベル遷移間の遷移とともに、3 相のハイ サイドとロー サイドを使用して駆動されている BLDC モーターを簡単に視覚的に確認できます



**図 5-1. BLDC モーター オシロスコープのスクリーンショット**

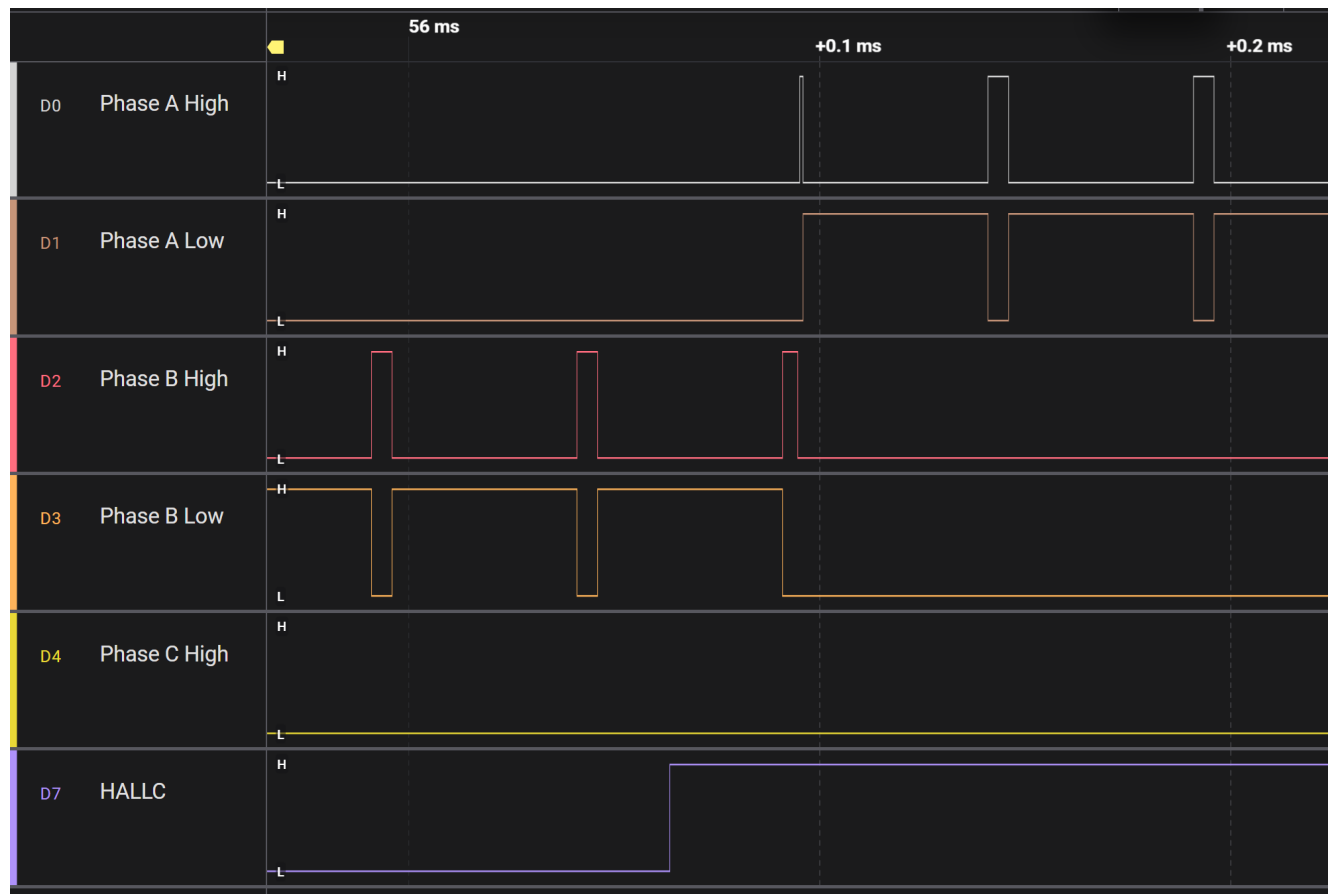


図 5-2. BLDC モーターの遷移

UART ペリフェラルを無効にし、Bluetooth LE の動作を有効にせず、BLDC モーターをアクティブに回転していない場合、CC2340R5 LaunchPad の動作で消費する電力は 1 $\mu$ A 未満です。

CCS 内のメモリ割り当てビューから決定されるように、デフォルトのプロジェクトには、184KB のフラッシュ (不揮発性メモリ用に予約されている 16KB は含まれません) と 32KB の SRAM が必要です。UART を無効にすると、3KB のフラッシュが節約されます。Bluetooth LE スタックを使用せずに構築されたプロジェクトでは、消費するのはわずか 22KB のフラッシュと 15KB の SRAM です。

Bluetooth LE プロファイルの特性 5 で送信できる最大データ量の概算値を取得するため、スループット テストを追加しました。このテストを実行するために、クロック インスタンスを初期化して、247 バイトのデータを通知で送信しました。デバイスは中央に接続されており、その接続間隔は 45 ms でした。この接続で、80% のデューティ サイクルを維持しながら、380 Kbps のスループットを達成しました。スループット テストは、接続間隔、CPU オーバーヘッド (この場合はモーターデューティ サイクル) などの多くの要因に依存するため、これらの数値は概算値であり、適切な変更を加えることで改善することができます。ストレス テストはデフォルトで無効になっていますが、`app_peripheral.c` で検出された ClockP インスタンスの開始フラグを `True` に設定することで簡単に追加できます。



## 6 まとめ

このアプリケーション ノートでは、SimpleLink CC2340R5 と DRV8329A を使用して BLDC モーターのホール エフェクト センサトラップの定義を完全に記述しました。必要なハードウェア接続と MCU のプログラミング手順について説明しており、ユーザーがすぐにデモを実行できるようになっています。この実装の安定性と堅牢性を確認するためのテスト結果も示されています。ソースコードは自由にアクセス可能であり、コードフローも詳細に説明されているため、開発者はプロジェクトの動作を理解し、自身のアプリケーション要件に合わせてプロジェクトをさらに改良することができます。この資料についてテキサス インストルメンツへ追加の質問やサポートが必要な場合は、E2E フォーラムへの投稿を推奨します。

## 7 参考資料

1. テキサス インストルメンツ『[CC2340R SimpleLink™ 2.4GHz ワイヤレス MCU ファミリ](#)』、データシート
2. テキサス インストルメンツ、『[Meet the CC2340R5 LaunchPad Development Kit](#)』、クイック スタート ガイドをご覧ください
3. テキサス インストルメンツ、『[DRV8329 4.5~60V、3 相 BLDC ゲートドライバ](#)』データ シート
4. テキサス インストルメンツ、『[DRV8329AEVM ユーザー ガイド](#)』
5. テキサス インストルメンツ、『[BLE5-Stack ユーザー ガイド](#)』
6. テキサス インストルメンツ、『[SimpleLink Academy for CC23xx](#)』
7. テキサス インストルメンツ、『[CC13xx と CC26xx の消費電流の測定](#)』、アプリケーション ノート
8. GitHub、『[SimpleLink 低消費電力 F3 デモ](#)』、サンプルコード

## 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