参考資料

Design Guide: TIDA-010086

*デジタル制御によるコスト最適化された **10A** バッテリの構成とテスト リファレンス デザイン*

説明

このリファレンス デザインは、マルチチャネルのバッテリ構成および試験用途向けに、コスト効率の高い設計を提供します。このデザインは、C2000™ リアルタイム マイコン (MCU) と 16 チャネルの同時サンプリング高精度 ADC を搭載しており、8 チャネルのセル試験に対応するよう拡張可能です。この設計では、高精度なデューティ サイクル制御のために HRPWM モジュールを使用し、高精度 D/A コンバータに匹敵する性能を実現しています。これにより、部品表を 30% 以上削減できます。この MCU は、定電流 (CC) と定電圧 (CV) 両方の制御動作向けに 2p2z 制御機能を内蔵しています。マルチチャネル ADC により、ソフトウェアで電流ループと電圧ループの柔軟性を最適化できます。この設計では、1 つの方式で複数の電流および電圧レベルの出力が可能です。

特長

- ±0.02% 未満の精度を持つ最大 10A の双方向電流制御
- ±5V の真のバイポーラ入力に対応し、バッテリ セルに直接接続可能な 16 チャネル SAR ADC
- レギュレーション誤差は ±1mV 未満で、充電と放電の両方で定電圧モードに対応
- ソフトウェア周波数応答アナライザ (SFRA) により、単純な閉ループ過渡応答調整が可能

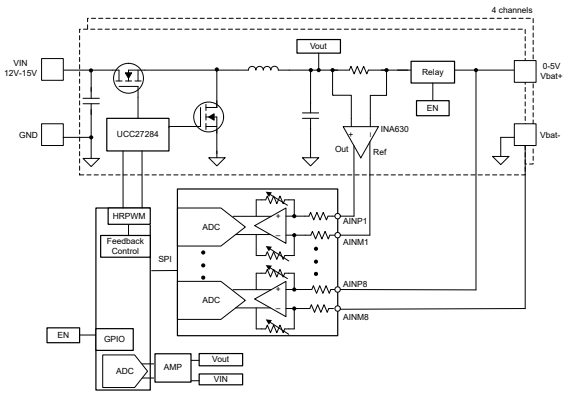
アプリケーション

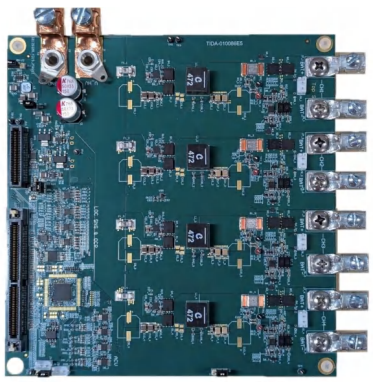
- バッテリ試験装置
- プログラマブル DC 電源

リソース

TIDA-010086	デザイン フォルダ
TMS320F28P65	プロダクト フォルダ
ADS9324、INA630	プロダクト フォルダ
UCC27284、REF50E	プロダクト フォルダ

テキサス・インスツルメンツの™ E2E サポート エキスパートにお問い合わせください。





Copyright © 2026 Texas Instruments Incorporated

1 システムの説明

1.1 リチウムイオン電池の形成機器

バッテリ テスタ装置には、シングル セル、バッテリ モジュール、および高電圧バッテリ パックのテストに使用されるさまざまな装置が含まれています。リチウムイオンまたは他の種類のバッテリ セルの性能、容量、安全性を検証するには、さまざまなテストを実施する必要があります。バッテリの形成は、セルの製造における重要なステップの 1 つを表しています。このプロセスでは、固体電解質インターフェイス (SEI) 層を形成するために、シングル セルへの高精度の充電と放電が必要です。セル グレーディングと電気試験では、各セルの容量と内部抵抗が評価されます。これらのテストでは、電流と電圧の充電放電プロファイルと、詳細な性能データをキャプチャするためのリアルタイム データ ロギングが必須です。

一般的な試験装置は、バッテリ セルの高精度な電流および電圧の充電 / 放電を実行するために高精度な双方向電源とデータ収集システムを必要とします。多くの場合、フルスケールでの ±0.05% 未満です。バッテリ容量の増加と統合性の向上に伴い、マルチチャネル電源では、バッテリ セルのテスト チャネル数が増えています。

充電 / 放電サイクルを管理し、高精度のテスト条件を提供するためには 2 つの方法を使用できます。表 1-1 に 2 つのアプローチの違いを示します。

表 1-1. アナログ制御とデジタル制御

要素	アナログ制御	デジタル制御
コントロール ロジック	アナログ コンポーネントを使用したハードウェアベースの帰還ループ	DSP 上で動作するソフトウェアベースのアルゴリズム
フレキシビリティ	固定構成可能、変更にはハードウェアの変更が必要	ソフトウェアによる高度な構成が可能、シンプルな負荷補償、複数のテスト プロファイルに対応
複雑	基本機能の設計はシンプルだが、高度な機能の設計は複雑	ソフトウェア開発は複雑だが、スケーラビリティのためのハードウェア開発はシンプル
応答時間	アナログ フィードバックを採用した、高速で連続的な応答	高速性は、マイクロ秒のレイテンシ、ADC、MCU の速度に依存
精度	高精度、コンポーネントの許容誤差とドリフトの影響を受けやすい	適切な ADC 分解能とキャリブレーションによる高精度、ドリフトの影響を受けにくい
データ ロギング	限定的、データ収集のための追加回路が必要	MCU を使用したデータ ロギング機能を内蔵、詳細な分析とトレーサビリティを実現
コスト	シンプルなシステムでは低コスト、複雑な高精度設計では高コスト	豊富な機能を搭載したスケーラブルなシステムで、コスト効果が高い

TIDA-010086 リファレンス デザインは、デジタル制御アプローチを採用して TMS320F28P650DK MCU と 16 チャネルの統合 PGA、SAR ADC を基盤としたマルチチャネル同期整流降圧コンバータを作成し、高精度、高速応答、高いシグナルチェーン密度の設計を実現します。

1.2 主なシステム仕様

表 1-2. 主なシステム仕様

パラメータ	仕様
低電圧ポート、バッテリポート	50mV ～ 5V
高電圧ポート、バス電圧	12V ～ 15V
スイッチング周波数	250kHz
チャネル双方向あたりの最大 DC 電流	10A
電流制御の精度	±0.02% FSR
電圧制御の精度	±0.02% FSR
電流過渡時間	100µs 未満
電力段と動作モード	同期整流降圧コンバータ、CCM モード

Copyright © 2026 Texas Instruments Incorporated

2 システム概要

2.1 ブロック図

図 2-1 に、リファレンス デザインのブロック図を示します。TMS320F28P650DK MCU は、同期整流降圧電力段用の高分解能 PWM を生成し、電流および電圧制御機能を実行します。INA630 電流センス アンプがバッテリ電流を検出し、電圧センスのために ADC 端子をバッテリに直接接続します。電流および電圧信号は、外部の ADS9324 ADC によってデジタル データに変換されます。C2000™ オンチップ ウィンドウ コンパレータを使用して、過電流保護機能を実装しています。

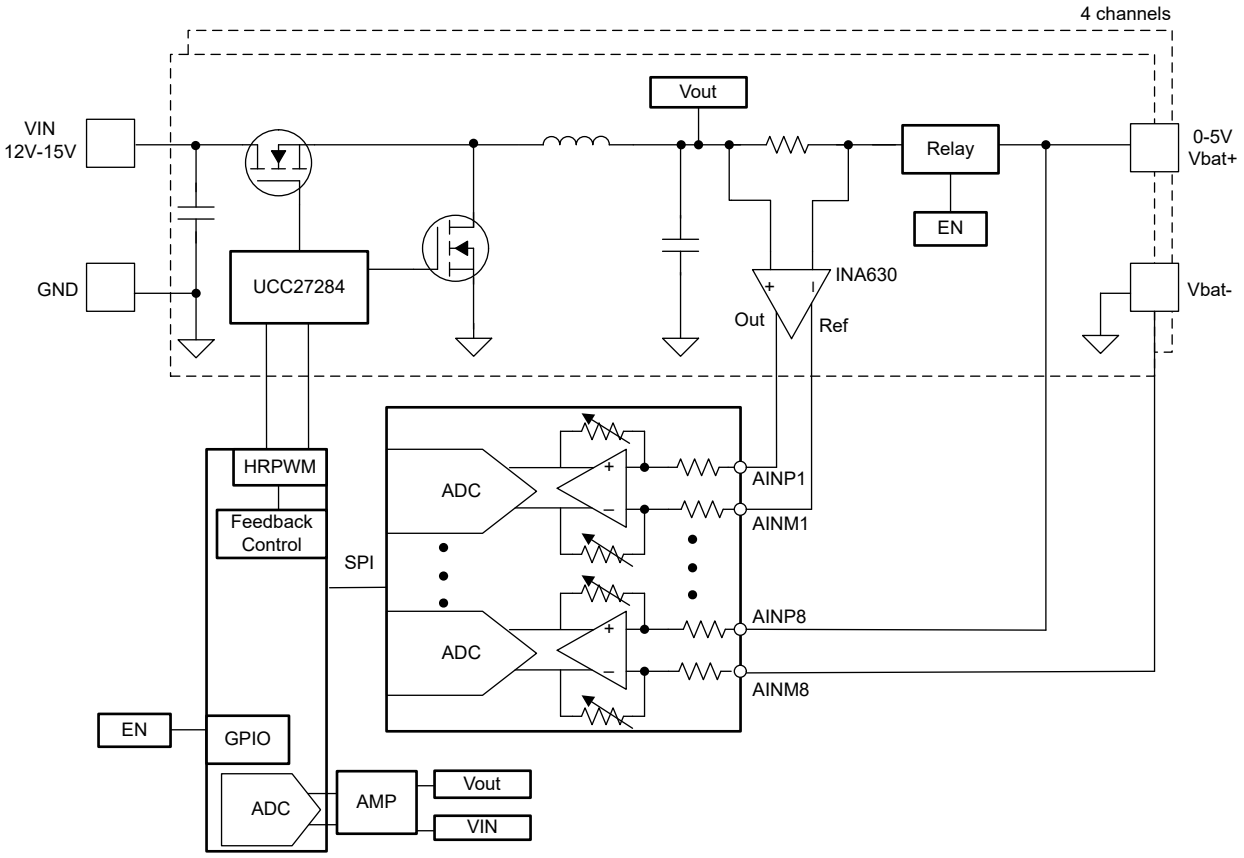


図 2-1. TIDA-010086 システム ブロック図

English Document: TIDUEY0
Copyright © 2026 Texas Instruments Incorporated

2.2 システム設計理論

2.2.1 フィードバック コントローラ

電流および電圧制御ループのソフトウェア実装を、図 2-2 に示します。電圧ループを電流にカスケード接続することで、充電モードと放電モードで定電流と定電圧の両方を実現します。バッテリ電圧が定電圧設定 (VSET) から離れている場合、電圧ループが定電流設定 (ISET) に飽和します。バッテリ電圧が VSET に近い値に達すると、電圧ループが閉じられ、ISET が低下して、バッテリ電圧が VSET 制限を超えないようにします。コントローラは充電モードと放電モードの両方で動作します。充電モードでは、VSET によって最大バッテリ電圧が制限されるため、充電が停止します。放電モードでは、VSET によって放電が停止する最小バッテリ電圧が制限されます。

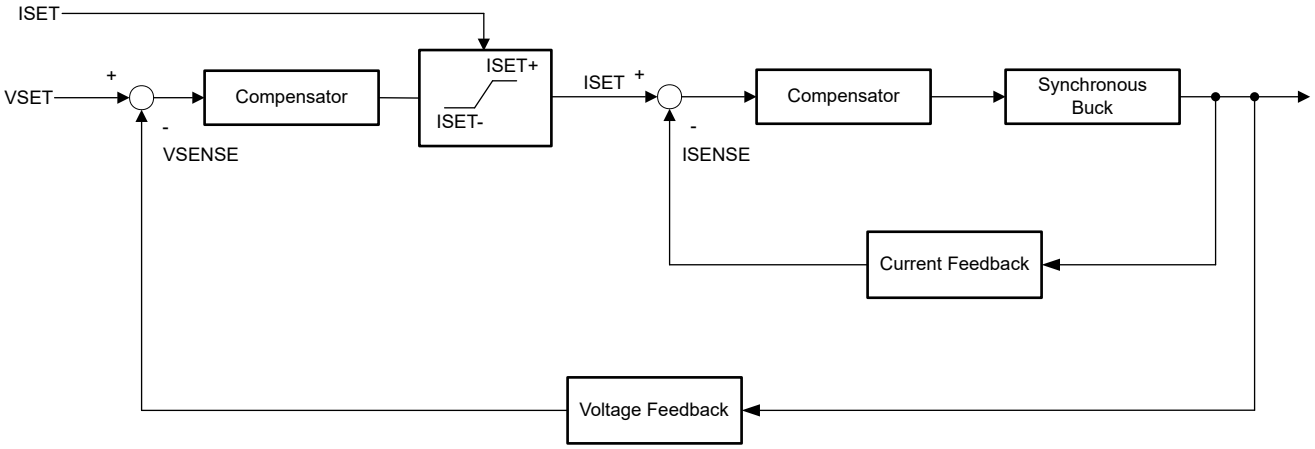


図 2-2. CCCV フィードバック コントローラ

2.2.2 DC/DC のスタートアップ

バッテリの充電と放電には、双方向電力段が使用されます。通常のスタートアップ条件では、降圧コンバータの出力が 0V から目標電圧に増加します。降圧コンバータが 0V から上昇している間にバッテリ負荷が接続されると、大電流オーバーシュートが発生する可能性があります。この問題は、2 つの方法で回避できます。1 つ目の方法では、図 2-3 に示すように、出力リレーをオープンにして降圧コンバータを起動し、降圧コンバータがバッテリ電圧に近い値に達したときに、リレーを閉位置に設定します。2 つ目の方法では、図 2-4 に示すように、DCM モードで降圧コンバータを起動し、充電中はローサイド スイッチをオフにし、放電中はハイサイドをオフにします。非同期モードから同期モードに切り替えるには、タイマが必要です。

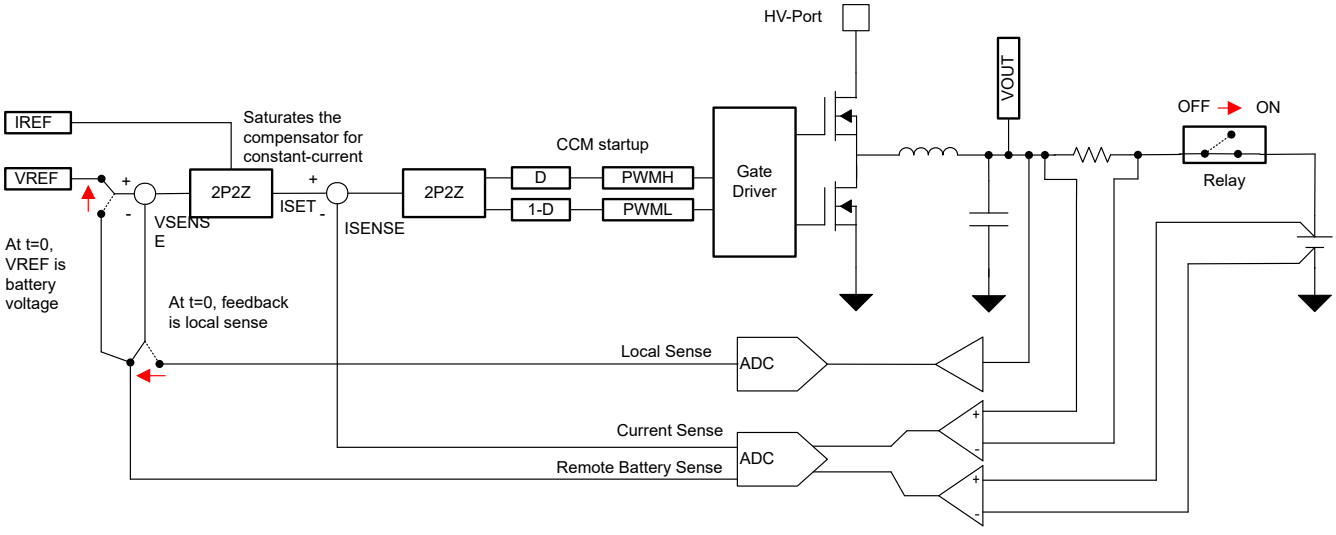


図 2-3. 同期スタートアップ

Copyright © 2026 Texas Instruments Incorporated

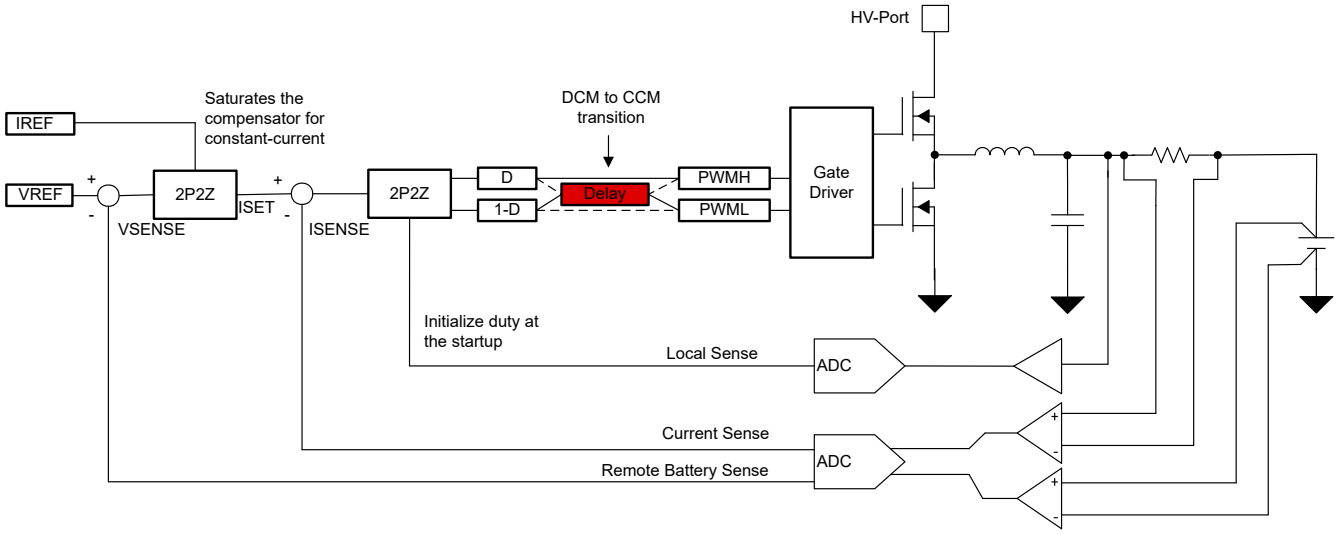


図 2-4. 非同期スタートアップ

2.2.3 高分解能 PWM 生成

高分解能 PWM は、高精度のフィードバック制御を実現します。高分解能カウンタにより、150ps のタイムステップ機能を利用することができます。250kHz のスイッチング周波数と 200MHz EPWMCLK の場合、このアプローチでは約 14.7 ビットの分解能が実現されます。さまざまなスイッチング周波数での PWM 分解能を、表 2-1 に示します。

表 2-1. PWM と HRPWM の C2000™ マイコンの分解能

PWM 周波数	通常の分解能 (PWM)		高分解能 PWM	
	100MHz EPWMCLK			
(kHz)	ビット	%	ビット	%
20	12.3	0.02	18.1	0
50	11	0.05	16.8	0.001
100	10	0.1	15.8	0.002
150	9.5	0.15	15.2	0.003
200	9	0.2	14.8	0.004
250	8.6	0.25	14.4	0.005

2.2.4 出力インダクタとコンデンサの選択

出力コンデンサの値およびコンデンサの ESR により、出力電圧リップルと負荷過渡性能が決まります。このコンデンサは、可能な限り最高の出力電圧リップルと負荷過渡性能を実現するように設計されています。

式 1 により、降圧電流のデューティ サイクルを計算します。

$$D = \frac{V_{OUT(max)}}{V_{IN(min)}} \times \text{Efficiency} = \frac{6\ V}{12\ V} \times 90\% = 55.5\% \quad (1)$$

インダクタのリップル電流は通常、出力電流の 0.2～ 0.3 倍の間で定義されます。制御ループは高速である必要があるため、インダクタはより小さい値で選択できます。そのため、リップル電流係数として 0.2 を選択します。計算では 12A のフルスケールを使用してある程度のマージンを確保するため、インダクタのリップル電流は 2.4A と推定されます。

降圧コンバータの連続導通モード (CCM) のワーストケース シナリオは、T_{off} が式 2 に示す最大値のときに発生します。

$$\Delta I_L = \frac{\left(V_{OUT(max)}\right) \times \left(1 - D\right)}{f_S \times L} \quad (2)$$

Copyright © 2026 Texas Instruments Incorporated

この式を解くと、インダクタの値は 4.45µH より大きくする必要があります。インダクタンスが小さいインダクタは飽和電流が大きくなります。この設計では、4.7µH のインダクタを選択しています。式 3 を使って、最大インダクタ電流を計算できます。

$$\Delta I_L = \frac{6\,V \times (1 - 0.555)}{250\,kHz \times 4.7\,\mu H} \cong 2.27\,A \quad (3)$$

式 4 は ESR を考慮せずに、出力キャパシタンスを計算します。

$$C_{out} = \frac{\Delta I_L}{8 \times f_{sw} \times \Delta V_{Out\_Ripple}} \quad (4)$$

出力電圧リップルは、最大出力電圧の 0.1%、すなわち 6mV を目標としています。式 4 に代入すると、出力コンデンサは 192µF で計算されます。この設計では、容量に合わせて 4 つの 47µF および 2 つの 1µF セラミック コンデンサを並列に配置しています。

総出力電圧リップルを決定するには、出力コンデンサの ESR 要件も重要です。ESR が大きいと、全体的に出力電圧リップルが大きくなる可能性があります。ESR 要件を計算するには、次の式を使用して、コンデンサのみによる電圧リップルと ESR のみによる電圧リップルを計算します。

$$V_{o\_pp} = \sqrt{\left(\frac{\Delta I_L}{8 \times C_{out} \times f_{sw}}\right)^2 + (\Delta I_L \times R_{ESR})^2} \quad (5)$$

$$6\,mV = \sqrt{\left(\frac{2.4\,A}{8 \times 190\,\mu F \times 250\,kHz}\right)^2 + (2.4\,A \times R_{esr})^2} \quad (6)$$

$R_{esr} \cong 0.5\ m\Omega$ について解きます。セラミック コンデンサを並列接続すると、この値を実現できます。各コンデンサは、250kHz のスイッチング周波数において ESR が約 1.5mΩ になります。コンデンサの接続を並列接続すると、全体的な ESR を 1mΩ 未満に低減できます。

2.2.5 電流帰還と電圧帰還

この設計は、±0.02% の FS 電流と電圧の制御と、±5°C の温度変化に対して測定精度を達成することを目標としています。シャントの両端での電圧を測定して、ADC 入力電圧にスケーリングする計測アンプを使用して電流センスを実行します。

オフセット誤差とゲイン誤差は、機器でキャリブレーションできるため、大きな問題ではありません。ただし、±0.02% または ±200ppm の精度を実現するためには、ゲインとオフセットのドリフトは重要なパラメータです。シャント抵抗を小さくすると、出力段経由で大きな電流が流れる場合に放熱量と合計温度ドリフトを低減できます。このリファレンス デザインでは、10A のフルスケール電流範囲に対して、25ppm/°C の電流シャントである 2mΩ が使用されます。INA630 は電流センス用で、この計測アンプの間接電流帰還 (ICFB) トポロジで使用され、内部高精度トリム抵抗をレーザー トリミングするプロセスを排除し、ディスクリートの外付け抵抗によってゲインを設定するプロセスを排除しているため、コスト効率の優れたアプローチが実現されます。

INA630 の入力オフセット ドリフトは 0.5 µ V/°C、ゲイン ドリフトは 3ppm/°C (標準値) で、2-mΩ 検出抵抗と 10A のフルスケール電流定格の場合、信号チェーンのドリフトは 25.18ppm/°C です。ゲインは外付けの分割抵抗のマッチングによって決定されるため、温度係数をさらに改善することでゲイン ドリフトを低減できます。マッチング抵抗ペアを使用し、ゲイン ドリフト誤差を最小化することによって、最大限の性能を利用できます。

ADC の合計ドリフトは電圧リファレンスから得られます。REF50E の場合、温度ドリフトは 2.5ppm/°C です。電流パスの総合未調整誤差は $\pm 5\,℃ \times \sqrt{(25^2 \times (2.5)^2}\ \ ppm/\,℃ = \pm 126.5 ppm$ です。これは、設計要件と一致しています。

電圧センス パスでは、温度変化に対する誤差がさらに改善されます。ADS9324 は 1MΩ の入力インピーダンスを備えており、内蔵のプログラマブル ゲイン アンプにより ±5V の入力範囲に対応できます。正と負の両方のアナログ入力で、5V バッテリに直接接続できるため、外部差動アンプが不要になり、BOM (部品表) コストを削減できます。また、高精度の電圧検出をサポートするための最小 CMRR は 100dB です。5V セルのスイングが ±1V の場合、CMRR による誤差は、フ

Copyright © 2026 Texas Instruments Incorporated

ルスケール電圧範囲の ±5µV または ±10ppm です。オフセットドリフトは 0.5ppm/°C であるため、合計誤差ではより簡単に 50ppm 未満のマージンを達成できます。

Copyright © 2026 Texas Instruments Incorporated

2.3 主な使用製品

2.3.1 TMS320F28P650DK

TMS320F28P650DK C2000 デバイスを使用して、同期整流降圧電力段を制御します。このデバイスには、18 の降圧コンバータを制御するのに十分な 36 の HRPWM チャネルが搭載されています。詳細については、*TMS320F28P65x リアルタイム マイクロコントローラ*のデータシートを参照してください。

2.3.2 ADS9324

ADS9324 には、同時サンプリング可能な 16 ビット逐次比較型 (SAR) A/D コンバータ (ADC) が 16 チャネル組み込まれています。このデバイスは、最大 750ksps のサンプリング レートを実現し、15kHz および 30kHz の内蔵 LPF によりノイズを低減しています。真のバイポーラ入力範囲は ADC のフルスケールとして ±5V、±6.25V を受け入れることができるため、高精度の電圧センス アンプは不要です。バッテリ セルに接続すると、ADS9324 は ±0.01% の精度と 1.5kHz のループ帯域幅を確保できます。詳細については、「ADS9324」を参照してください。

2.3.3 INA630

INA630 は、高精度でコスト最適化された間接電流帰還計測アンプであり、ゲイン 100 の入力オフセット (0.7µ V/C) と標準ゲイン ドリフト (2.5ppm/°C) が低く、±5°C の温度変化に対して ±0.02% の電流制御精度を達成しています。*INA630 高精度、126dB CMRR、間接電流帰還計測アンプ* データシートも参照してください。

2.3.4 UCC27284

UCC27284 は堅牢な N チャネル MOSFET ドライバで、最大スイッチ ノード (HS) 電圧定格は 100V です。ハーフブリッジまたは同期整流降圧構成ベースのトポロジで、2 つの N チャネル MOSFET を制御できます。3A のソースおよびシンク能力を備え、標準伝搬遅延は 16ns であるため、デッドタイム要件が最小化され、効率がさらに向上されます。

2.3.5 REF50E

REF50xxE は、低ノイズ、低ドリフト、非常に高精度の電圧リファレンスのファミリです。独自の設計手法により、優れた温度ドリフト (2.5ppm/℃) と高精度 (0.025%) を実現しています。超低フリッカー ノイズ (0.5µVPP/V) の組み合わせにより、REF50xxE は高精度のデータ収集システムでの使用に最適です。

Copyright © 2026 Texas Instruments Incorporated

3 ハードウェア、ソフトウェア、テスト要件、テスト結果

3.1 ハードウェア要件

図 3-1 に、TIDA-010086 ハードウェアのさまざまな部分を示します。このボードを使用するには、ハードウェアとソフトウェアの性能をテストするために F28P65 controlCARD 評価基板が必要です。

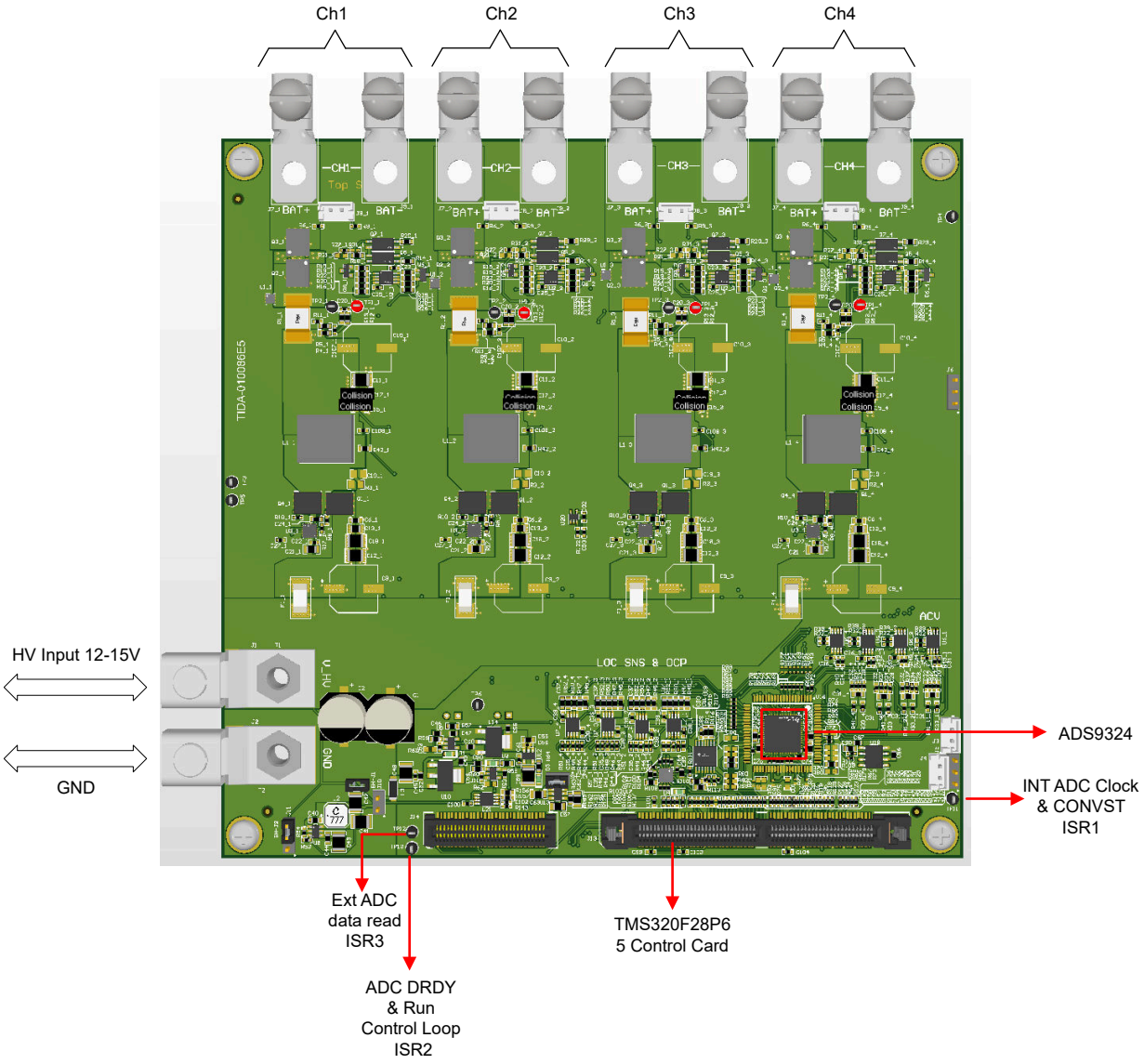


図 3-1. TIDA-010086 ハードウェア

このリファレンス デザインを評価するために、以下のハードウェア機器を使用しました。

- DC システムの電源:HP 6675A
- ラボ電源: Agilent E3634A
- DC 負荷:クロマ、63102A
- 6.5 桁分解能のマルチメータ:Agilent 34401A
- オシロスコープ: Tektronix MDO34

Copyright © 2026 Texas Instruments Incorporated

3.2 ソフトウェア

このソフトウェアは Code Composer Studio (CCS) 統合開発環境 (IDE) を使用し、C2000WARE-DIGITALPOWER-SDK ライブラリの一部として利用できます。

3.2.1 Code Composer Studio™ 内でプロジェクトを開く

Code Composer Studio (CCS) でプロジェクトを開始するには、次の手順に従います。

1. Code Composer Studio (CCS) 統合開発環境 (IDE) ツール フォルダから Code Composer Studio をインストールします。バージョン 12.4 またはそれ以降をお勧めします。
2. C2000WARE-DIGITALPOWER-SDK を次の 2 つのいずれかの方法でインストールします。
   a. CCS にアクセスし、[View] → [Resource Explorer] をクリックします。テキサス・インスツルメンツの Resource Explorer 下で C2000WARE-DIGITAL-POWER-SDK にアクセスし、[Install] ボタンをクリックします。
   b. C2000Ware Digital Power SDK ツール フォルダからダウンロードします。
3. インストールが完了したら、CCS を閉じて、新しいワークスペースを開きます。CCS により自動的に powerSUITE が検出されます。変更を有効にするために CCS を再起動しなければならない場合があります。

注

デフォルトでは、powerSUITE は SDK のインストールと同時にインストールされます。

ファームウェア プロジェクトは、次のいずれかの方法でインポートできるようになりました。

- Resource Explorer を使用する
  1. Resource Explorer の C2000WARE-DIGITAL-POWER-SDK で、[powerSUITE] → [Solution Adapter Tool] をクリックします。
  2. [DC-DC] セクションに表示される設計のリストから TIDA-010086 を選択します。
  3. 開発キット ページが表示されます。プロジェクトを実行するためのアイコンがトップ バーに表示されます。[Run Project] をクリックします。
  4. この操作によりプロジェクトがワークスペース環境にインポートされ、GUI が図 3-2 のような設定ページが表示されます。
  5. この GUI ページが表示されない場合は、C2000WAREDIGITAL-POWER-SDK Resource Explorer の powerSUITE 下の FAQ セクションを参照してください。
- solution フォルダから直接インポートする
  1. CCS 内で [*Project*] → [*Import CCS Projects*] をクリックし、/solutions/tida_010086/f28p65x/ccs にある solution フォルダを参照して、プロジェクトを直接インポートすることもできます。
  2. 2 つのプロジェクト仕様が表示されます。1 つは powerSUITE 付きで、もう 1 つは powerSUITE なしです。いずれかをクリックすると、プロジェクトの自己完結型フォルダが作成され、その中にすべての依存関係が含まれます。
  3. 以下のすべての手順では、この設計ガイドに記載されている settings.h および user_settings.h ファイルで、関連する #defines を変更する方法について説明します。

Copyright © 2026 Texas Instruments Incorporated

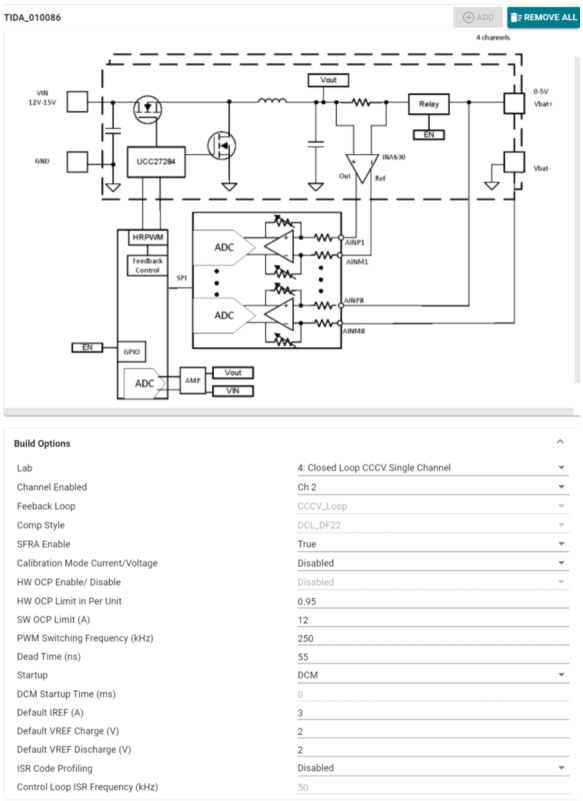


図 3-2. 設計の powerSUITE GUI

English Document: TIDUEY0
Copyright © 2026 Texas Instruments Incorporated

3.2.2 プロジェクト構造

プロジェクトの一般構造を、図 3-3 に示します。プロジェクトがインポートされると、図 3-4 に示すように CCS 内に Project Explorer が表示されます。

注

図 3-4 は F28p65x のプロジェクトを示していますが、ページから別のデバイスを選択しても、その構造は同様です。

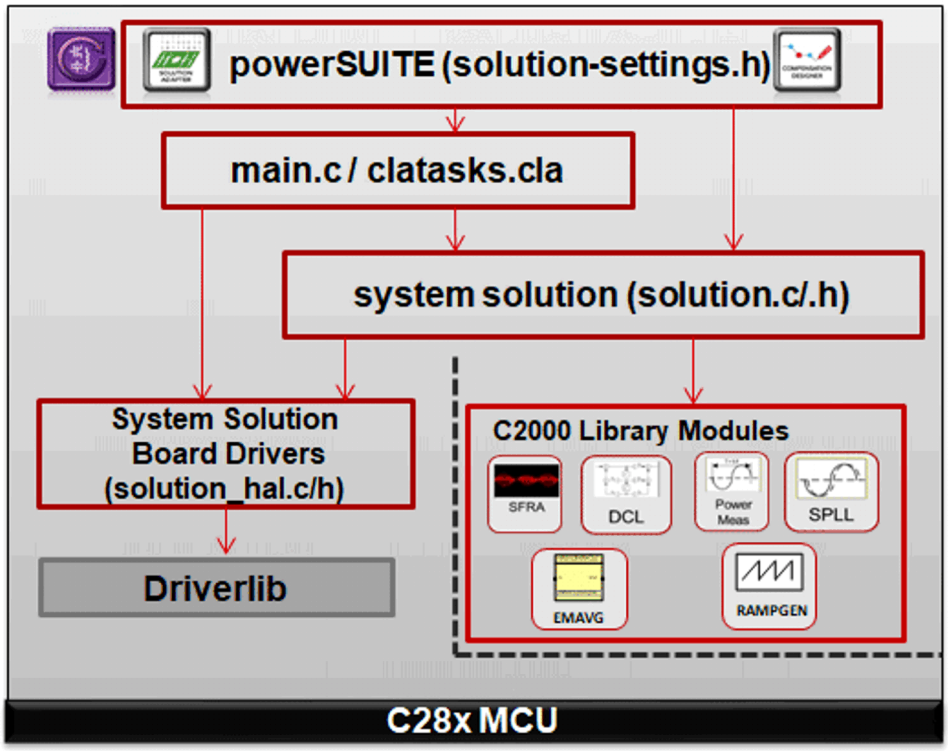


図 3-3. プロジェクト構造の概要

コア アルゴリズム コードで構成されるソリューション専用でデバイスに依存しないファイルは .c/h **.c/h** . にあります。

基板専用かつデバイス専用のファイルは **_hal.c/h** にあります。このファイルは、ソリューションを実行するデバイス特定のドライバで構成されています。別の変調方式やデバイスを使用する場合、プロジェクト内のデバイス サポート ファイルを変更する以外に変更を加える必要があるのは、これらのファイルのみです。

-main.c ファイルは、プロジェクトのメイン フレームワークで構成されています。このファイルは、システム フレームワークの作成に役立つボード ファイルとソリューション ファイルの呼び出し、割り込みサービス ルーチン (ISR)、低速なバックグラウンド タスクで構成されています。

この設計では、ソリューションは **bt4ch** です。

powerSUITE ページは、Project Explorer に表示される main.syscfg ファイルをクリックすると開きます。このページでは **_settings.h** ファイルが生成されます。このファイルは、powerSUITE ページで生成されたプロジェクトのコンパイル時に使用する唯一の C 言語を使用したファイルです。プロジェクトが保存されるたびに powerSUITE によって変更内容が上書きされるため、このファイルを手動で変更しないでください。_settings.h ファイルには、動作モード選択と、電流および電圧制御ループの補償設定機能が含まれています。**_user_settings.h** は **_settings.h** に含まれており、ADC マッピングの #defines や GPIO など、powerSUITE ツールの範囲外の設定を保持するために使用できます。

_cal.h ファイルは、電流と電圧を測定するためのゲイン値とオフセット値で構成されています。

Copyright © 2026 Texas Instruments Incorporated

_.ccxml は、リファレンス デザインでどのデバッガを使用するかを設定します。この設計では、XDS110 USB デバッグ プローブの接続を使用し、ターゲット デバイスは TMS320F28P65DK9 です。

Kit.json ファイルと solution.js ファイルは、powerSUITE により内部で使用されるため、ユーザーが変更することはできません。これらのファイルを変更すると、プロジェクトが正常に機能しなくなります。

ソリューション名は、ソリューションで使用されるすべての変数のモジュール名および定義としても使用されます。したがって、すべての変数および関数呼び出しの先頭には ***BT4CH*** 名が付きます (たとえば、*BT4CH_userParam_chX*)。この命名規則により、名前の競合を回避しながら、異なるソリューションを組み合わせることができます。

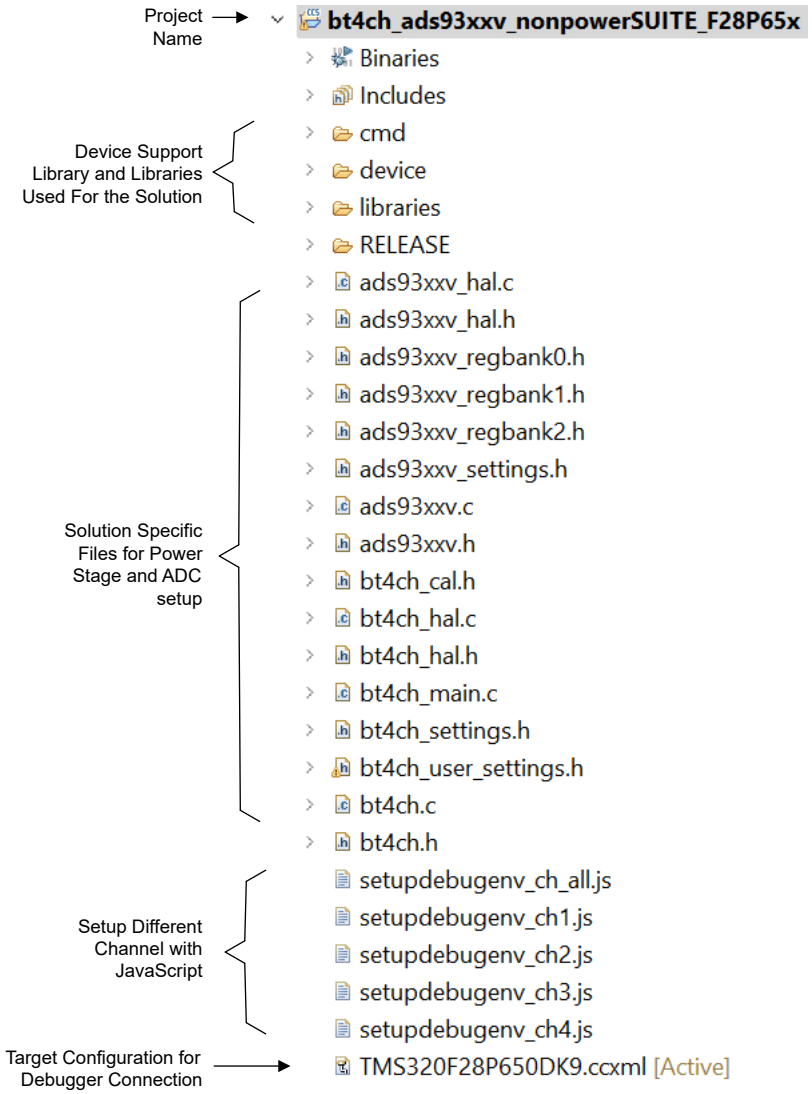


図 3-4. BT4PH プロジェクトの [Project Explorer] ビュー

bt4ch プロジェクトは 3 つの ISR (ISR1、ISR2、および ISR3) で構成されています。

- 降圧コンバータの入力電源と出力コンデンサ電圧を検出するには、*ISR1* を使用します。ISR1 は ADCC 変換完了によってトリガされ、外部 ADC CONVST と同じ周波数で実行されます。ADCC はコンバータの入力電圧と出力電圧を検出し、その出力は DC/DC のソフトスタートの実装に使用されます。
- *ISR2* は、ADS9324 の BUSY 信号によってトリガされます。外部 ADC は 400kSPS のサンプルレート (CONVST) にプログラムされ、オーバーサンプリング レート (OSR) は 8 に設定され、ISR 周波数が 50kHz に設定されます。
- *ISR3* は、SPI 受信 FIFO 割り込みによってトリガされます。ISR を使用して、FIFO レジスタから外部 ADC データを読み出し、制御ループ関数を実行します。

Copyright © 2026 Texas Instruments Incorporated

図 3-6 に、4 つのチャネルすべてがオンの場合の ISR1、ISR2、ISR3 の所要時間を示します。3 つの ISR に要する合計時間は 6µs 未満となり、これは 50kSPS 制御ループのサンプル レートでは、CPU 使用率の 30% 未満です。図 3-5 および図 3-7 に、1 つのチャネルのみがオンですべてのチャネルがオフのときの ISR 時間を示します。

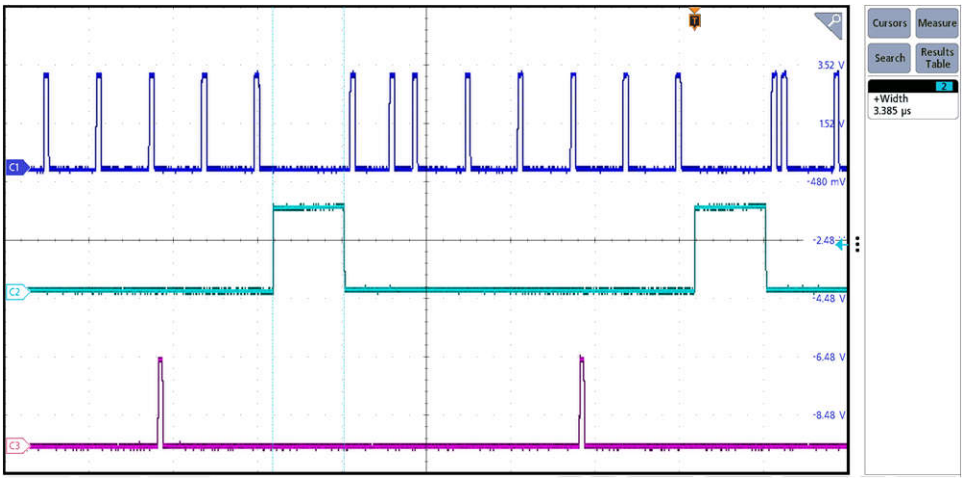


図 3-5. 1 チャネルの ISR 実行時間

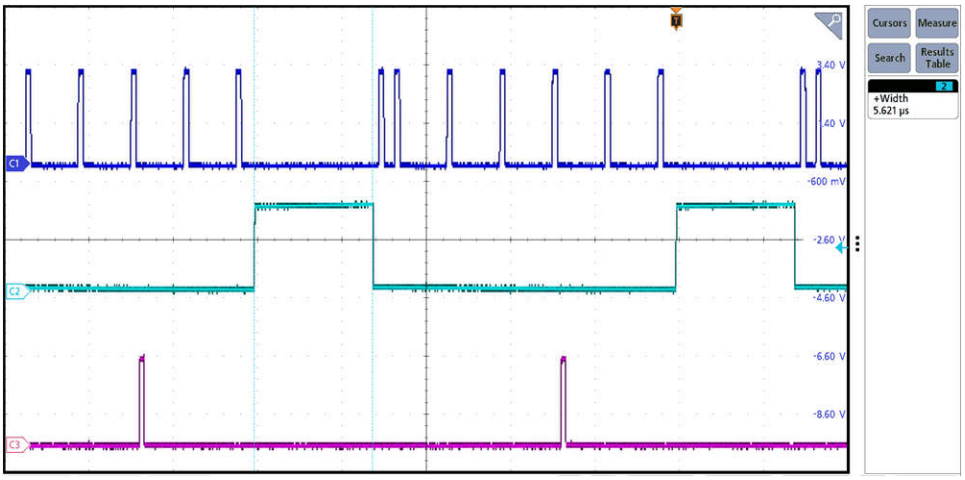


図 3-6. 4 チャネルの ISR 実行時間

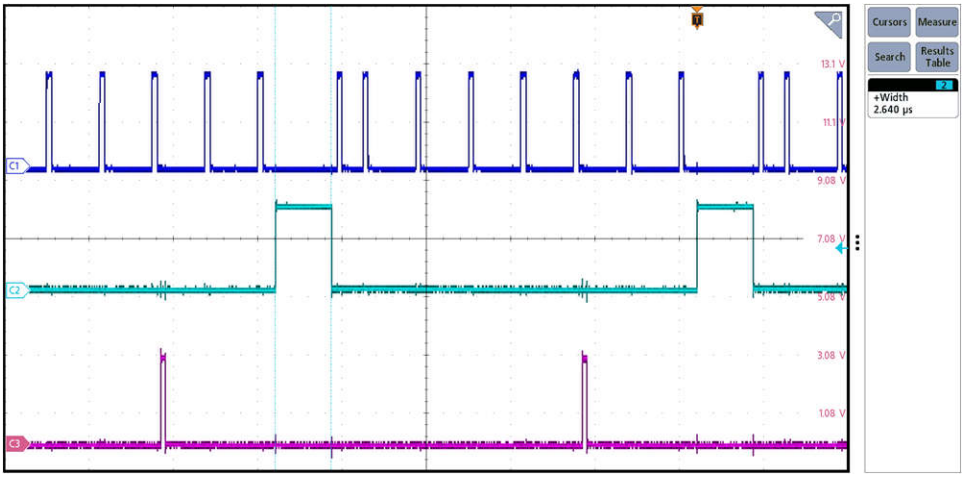


図 3-7. すべてのチャネルがオフのときの ISR 実行時間

Copyright © 2026 Texas Instruments Incorporated

3.2.3 ソフトウェア フロー図

図 3-8 に、ソフトウェア フロー図を示します。

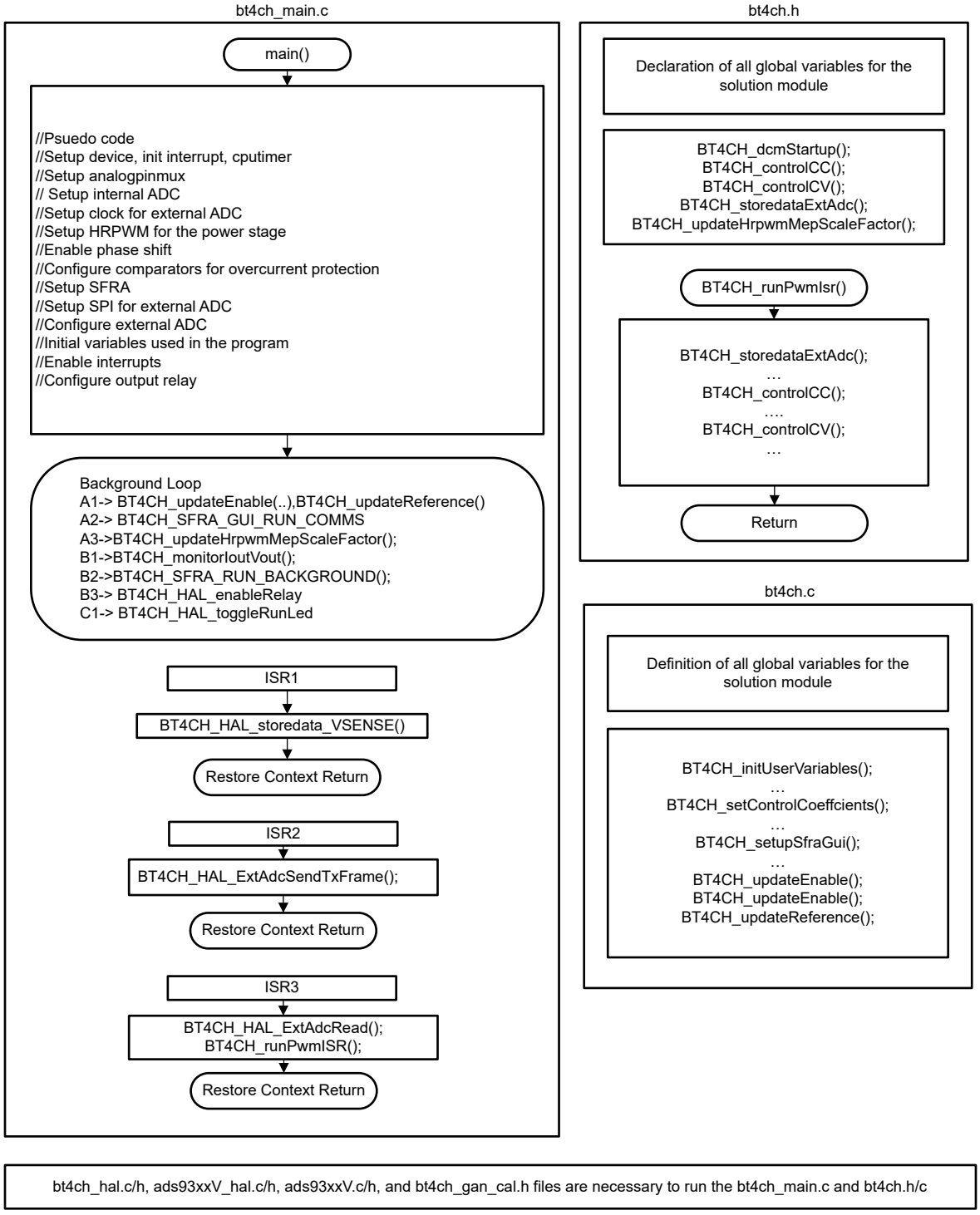


図 3-8. ソフトウェア フロー図

Copyright © 2026 Texas Instruments Incorporated

3.3 テスト設定

3.3.1 双方向の電力フローをテストするためのハードウェア設定

双方向電源をテストするための設定を 図 3-9 に示します。入力 DC 電源は、入力バス電圧として 12V ～ 15V で動作します。入力電源が双方向電流フローをサポートできない場合、電子負荷 (e-load) は放電モードを提供します。バッテリ側では、バッテリ セルの充電または放電をシミュレートするために、システムをラボ用電源と電子負荷に接続します。

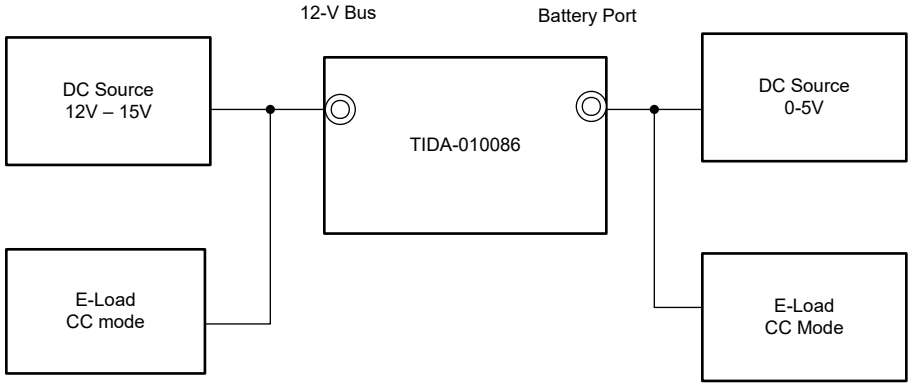


図 3-9. 双方向の電力フローをテストするためのハードウェア設定

3.3.2 電流および電圧ループをチューニングするためのハードウェア設定

制御ループは、シングル チャネル動作中のみ調整が必要です。エンジニアはこの方式を使用して電流ループと電圧ループを調整します。バッテリ負荷は、m Ω ～ 数百 m Ω の範囲に対応する小型インピーダンス ケーブルとして現れます。外部機器を使用しないインピーダンス整合ケーブルは、実際のバッテリ セルよりも制御ループを簡単に調整するためのものです。

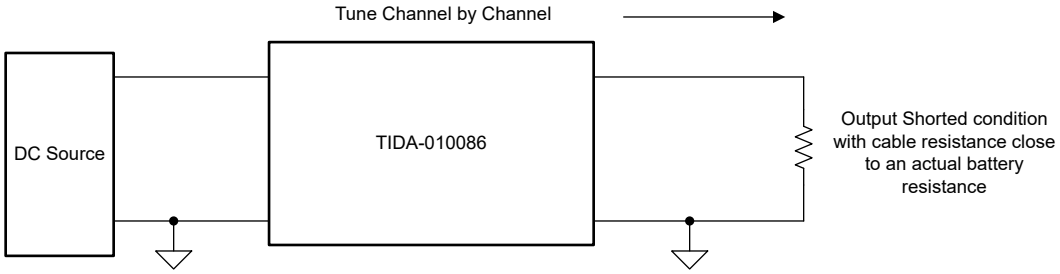


図 3-10. 電流および電圧ループをチューニングするためのハードウェア設定

3.3.3 電流および電圧キャリブレーションのハードウェア設定

電流と電圧のキャリブレーションのために、6.5 桁のマルチメータが、出力端子の電圧と、電流シャントの両端間の電圧降下を測定します。電流キャリブレーションでは、電子負荷定電流モードまたは短絡状態が使用されます。電圧キャリブレーションでは、開路条件が使用されます。

Copyright © 2026 Texas Instruments Incorporated

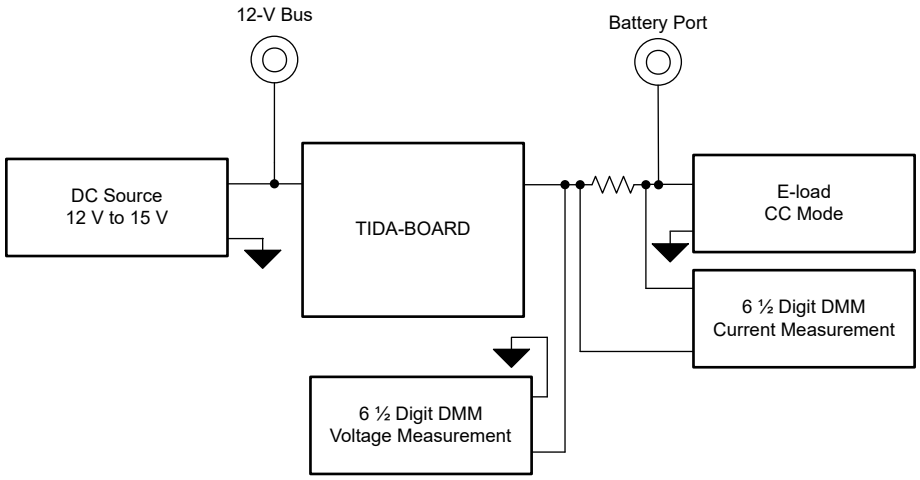


図 3-11. 電流および電圧キャリブレーションのハードウェア設定

3.3.4 ラボ変数の定義

BT4CH_userParam_ch1,2,3,4 変数は、さまざまなラボで電力段を制御するために使用されます。パラメータの定義については、表 3-1 を参照してください。

表 3-1. BT4CH_userParam_chX 定義

BT4CH_userParam_chX	データの種類	備考
Iref_A	フローティング	充電モードと放電モードの両方の電流を設定 [0、10]
VrefCharge_V	フローティング	充電モードで電圧を設定 [0、5]
VrefDischarge_V	フローティング	放電モードで電圧を設定 [0、5]
Dir_bool	符号なし整数	充電モードの場合、このパラメータを 1 に設定します
		放電モードの場合、このパラメータを 0 に設定します
Relay_ON	符号なし整数	リレーをイネーブルにする場合、このパラメータを 1 に設定します
		リレーをディセーブルにする場合、このパラメータを 0 に設定します
En_bool	符号なし整数	チャネルをイネーブルにする場合、このパラメータを 1 に設定します
		チャネルをディセーブルにする場合、このパラメータを 0 に設定します
remote_sense	符号なし整数	閉ループ制御にリモート センスを使用する場合、このパラメータを 1 に設定します
		閉ループ制御にローカル センスを使用する場合、このパラメータを 0 に設定します
DutyRef_pu	フローティング	開ループ モードの基準デューティ サイクル範囲 = 0～1.0
IbatCal_pu	フローティング	キャリブレーション モードで出力電流を設定するには、このパラメータを使用します。範囲 = 0～1.0
VbatCal_pu	フローティング	キャリブレーション モードで出力電圧を設定するには、このパラメータを使用します。範囲 = 0～1.0
IoutGain_pu	フローティング	変数には、電流ゲインのキャリブレーション データが格納されます
IoutOffset_pu	フローティング	変数には、バッテリ電流オフセットのキャリブレーション データが格納されます
IoutGain_A	フローティング	変数には、バッテリ電流ゲインのキャリブレーション データが格納されます
IoutOffset_A	フローティング	変数には、バッテリ電流オフセットのキャリブレーション データが格納されます
VoutGain_pu	フローティング	この変数には、バッテリ降圧コンバータの出力電圧ゲインのキャリブレーション データが格納されます
VoutOffset_pu	フローティング	この変数には、降圧コンバータの出力電圧オフセットのキャリブレーション データが格納されます
VoutGain_V	フローティング	この変数には、降圧コンバータの出力電圧オフセットのキャリブレーション データが格納されます
VoutOffset_V	フローティング	この変数には、降圧コンバータの出力電圧オフセットのキャリブレーション データが格納されます

Copyright © 2026 Texas Instruments Incorporated

表 3-1. BT4CH_userParam_chX 定義 (続き)

BT4CH_userParam_chX	データの種類	備考
VbatGain_pu	フローティング	変数には、バッテリ電圧のキャリブレーション データが格納されます
VbatOffset_pu	フローティング	変数には、バッテリ電圧オフセットのキャリブレーション データが格納されます
VbatGain_V	フローティング	変数には、バッテリ電圧ゲインのキャリブレーション データが格納されます
VbatOffset_V	フローティング	変数には、バッテリ電圧オフセットのキャリブレーション データが格納されます

Copyright © 2026 Texas Instruments Incorporated

3.3.5 テスト方法

3.3.5.1 ラボ 1.開ループ電流制御単相

3.3.5.1.1 ラボ 1 のソフトウェア オプションの設定

1. セクション 3.2.1 の概要に従って CCS プロジェクトを開きます。powerSUITE を使用している場合は、手順 2 に進みます。それ以外の場合は手順 3 に進みます。
2. SYSCONFIG ページを開き、[Build Options] セクションで以下を選択します。
   - ラボは [Lab 1: *Open Loop CC Single Channel]* を選択します。
   - 4 つのチャネルのいずれかを選択します。
   - SFRA をイネーブルにします。
   - ページを保存します。
3. powerSuite のないバージョンのプロジェクトを使用する場合、上記の設定は solution_settings.h ファイルで直接変更されます。

```
#define LAB_NUMBER (1)
#define CHANNEL_NUMBER (1)
#define SFRA_ENABLED (true)
```

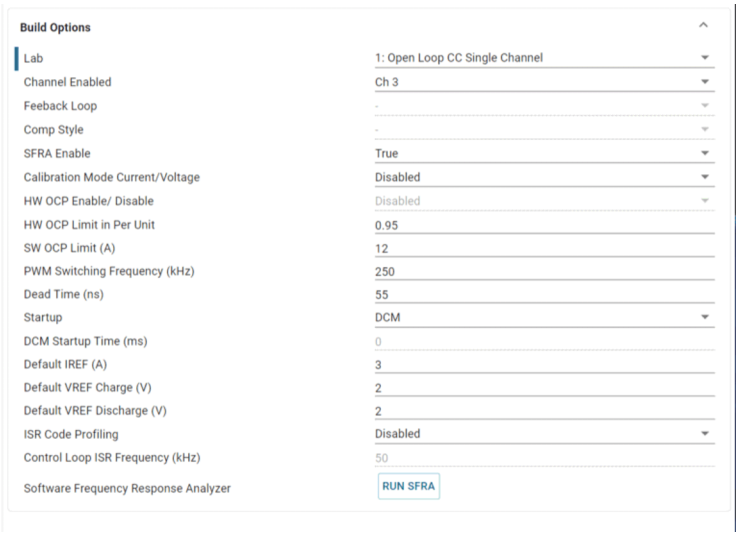


図 3-12. ラボ 1 のビルド オプション

3.3.5.1.2 プロジェクトのビルドおよびロードとデバッグ環境の設定

プロジェクトをビルドおよびロードし、デバッグ環境を設定するには、次の手順を実行します。

1. プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
2. プロジェクトが正常にビルドされます。
3. Project Explorer で、targetConfigs 下で適切な目標構成ファイルが有効になっていることを確認します。
4. **[Run]** → **[Debug]** をクリックしてデバッグ セッションを開始します。
5. するとプロジェクトがデバイスにロードされ、CCS デバッグ ビューが有効になります。メイン ルーチンの開始時にコードは停止します。
6. [Watch] および [Expressions] ウィンドウに変数を追加するには、**[View]** → **[Scripting Console]** をクリックして、[Scripting Console] ダイアログ ボックスを開きます。このコンソールの右上隅で、[Open] をクリックして、プロジェクト フォルダ内にある **setupdebugenv_chX.js** スクリプト ファイルを参照します。これにより、[Watch] ウィンドウに、システムをデバッグするのに必要な適切な変数が入力されます。
7. [Watch] ウィンドウで **[Continuous Refresh]** ボタン ( ) をクリックして、コントローラからの値の連続更新を有効にします。

Copyright © 2026 Texas Instruments Incorporated

3.3.5.1.3 コードの実行

ラボ 1 のコードを実行するには、次の手順に従います。

1. セクション 3.3.2 に示すようにテスト設定を使用します。
2. メニュー バーの  をクリックしてプロジェクトを実行します。
3. [Watch] ビューの [Expression] ウィンドウで BT4H_InputVoltageSense_V が 12V～15V の範囲内にあるかどうかを確認します。
4. オシロスコープを使用して、周波数が 50kHz の場合に外部 ADC の DRDY 信号をチェックします。MCU が動作しているときの ADS9324 の DRDY および CONVST 信号を、図 3-13 に示します。
5. [Expression] ウィンドウで次のパラメータを設定します。
   - BT4CH_userParam_chX->dutyRef_pu = 0.02
   - BT4CH_userParam_chX->en_bool = 1 を設定します
   - BT4CH_userParam_chX->Relay_ON を 1 に設定して、出力リレーを有効にします
   - [Expression] ウィンドウの設定については、図 3-14 を参照してください。
6. BT4CH_measure_V_I_chX 変数は、DC/DC コンバータの出力電流と電圧を示します。BT4CH_userParam_chX->DutyRef_pu を調整し、電流が約 4.5A であることを確認します。
7. 図 3-15 に、開ループ電流制御用のプラント モデルを抽出するための SFRA 設定を示します。SYSCONFIG ページで Run SFRA アイコンをクリックします。 SFRA GUI がポップアップ表示されます。
8. SFRA GUI でデバイスのオプションを選択します。たとえば、F28P65x の場合は浮動小数点を選択します。[Setup Connection] ボタンをクリックします。ポップアップ ウィンドウで [Boot on Connect] オプションのチェックを外し、適切な COM ポートを選択します。[OK] ボタンをクリックします。SFRA GUI に戻り、[Connect] ボタンをクリックします。
9. SFRA GUI がデバイスに接続します。これで [Start Sweep] をクリックして、SFRA 掃引を開始できるようになりました。SFRA 掃引が完了するまでには数分かかります。完了すると、図 3-16 に示すように測定値が表示されたグラフが表示されます。
10. また、周波数応答データは SFRA データ フォルダ下のプロジェクト フォルダに保存され、SFRA 実行時のタイムスタンプが記録されます。
11. ラボ演習が終了したら、*[Expression] ウィンドウ*から次のパラメータを設定してコードを停止します。
    - BT4CH_userParam_chX->dutyRef_pu = 0
    - BT4CH_userParam_chX->en_bool = 0 を設定します
    - BT4CH_userParam_chX->Relay_ON を 0 に設定して、出力リレーを無効化します
    - プログラムを終了します

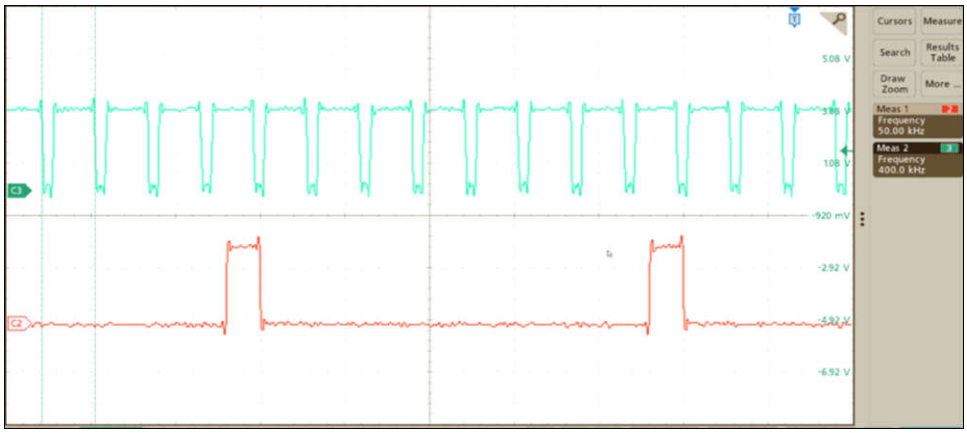


図 3-13. ADS9324 の CONVST および DRDY 信号

Copyright © 2026 Texas Instruments Incorporated

Expression	Type	Value
BT4CH_lab	enum <unname...	Lab1_singleChannelOpenLoopCC
BT4CH_sfraStatus	enum <unname...	SFRA_Disabled
BT4CH_calibrationStatus	enum <unname...	Calibration_Disabled
BT4CH_calibrationMode	enum <unname...	Calibration_CC
BT4CH_startupMode	enum <unname...	DCM_Startup
BT4CH_HAL_InputVoltageSense_V	float	13.3130159
BT4CH_ISR2_Loading	float	0.126220882
BT4CH_ISR2_LoadingMax	float	0.12963891
BT4CH_userParam_ch3	struct <unname...	{Iref_A=5.0,VrefCharge_V=3.0,VrefDi...
Iref_A	float	5.0
VrefCharge_V	float	3.0
VrefDischarge_V	float	3.0
dir_bool	unsigned int	1
Relay_ON	unsigned int	1
Start_bool	unsigned int	1
remote_sense	unsigned int	0
DutyRef_pu	float	0.0199999996
IbatCal_pu	float	0.200000003
VbatCal_pu	float	0.400000006
IoutGain_pu	float	0.0805866644
IoutOffset_pu	float	0.000886499882
IoutGain_A	float	12.4090014
IoutOffset_A	float	-0.0110005783
VoutGain_pu	float	0.24248305
VoutOffset_pu	float	-0.000581979752
VoutGain_V	float	4.1239996
VoutOffset_V	float	0.00240008417
VbatGain_pu	float	0.200000018
VbatOffset_pu	float	-0.000600039959
VbatGain_V	float	4.99999952
VbatOffset_V	float	0.00300019956
BT4CH_measureVI_ch3	struct <unname...	{Ibat_A=4.24398136,Vout_V=0.0790...
Ibat_A	float	4.24188375

図 3-14. ラボ 1 [Expression] ウィンドウ、開ループ

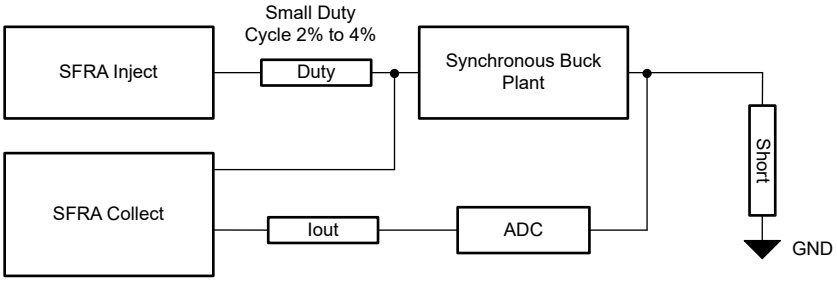


図 3-15. 開ループ電流制御用の SFRA 設定

Copyright © 2026 Texas Instruments Incorporated

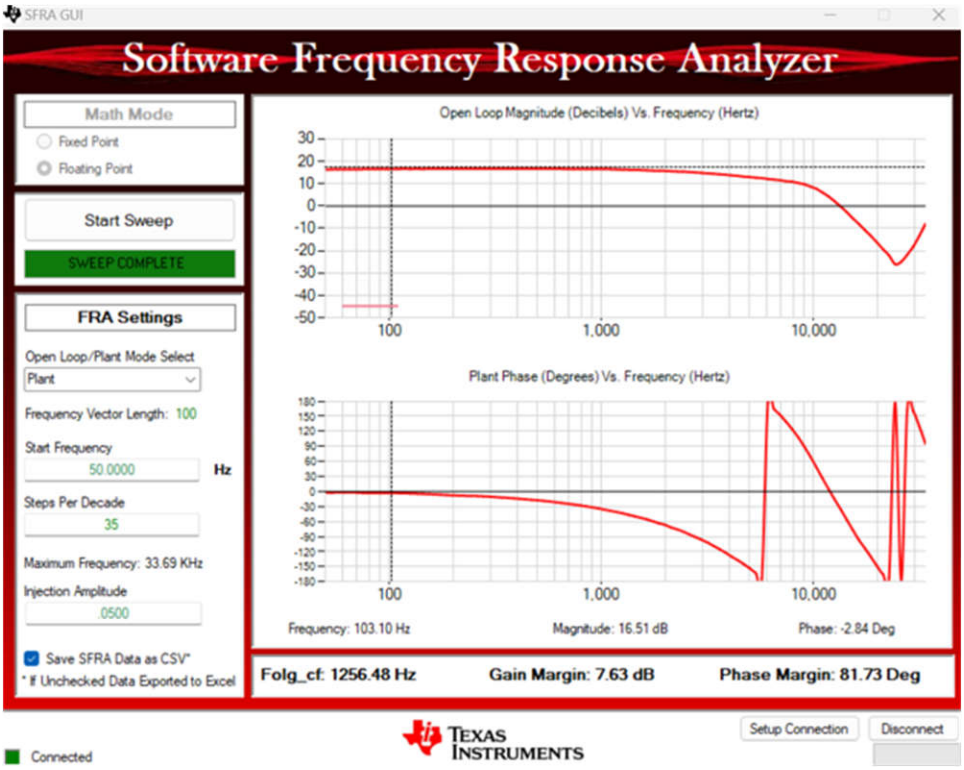


図 3-16. 電流制御の開ループ周波数応答

3.3.5.2 ラボ 2.閉ループ電流制御シングル チャネル

3.3.5.2.1 ラボ 2 のソフトウェア オプションの設定

1. このラボを実行するには、前のセクション、セクション 3.3.2 で説明したようにハードウェアが設定されていることを確認します。
2. セクション 3.2.1 の概要に従って CCS プロジェクトを開きます。powerSUITE を使用している場合は、手順 3 に進みます。それ以外の場合は手順 4 に進みます。
3. SYSCONFIG ページを開き、[Build Options] セクションで以下を選択します。
   - ラボは [Lab 2: *Closed Loop CC Single Channel]* を選択します。
   - チャンネルを選択します。
   - SFRA をイネーブルにします。
   - [Run Compensation Design] ボタンをクリックして、Compensation Designer を開きます。
   - 補償デザイナが起動し、有効な SFRA データ ファイルを選択するように求められます。ラボ 1 の実行から補償デザイナに SFRA データをインポートし、2 極、2 ゼロの補償器を設計します。この設計の繰り返し中により多くのマージンを確保して、ループが閉じたときにシステムが安定するようにします。
4. このデザインは 2p2z 制御を使用しており、OTA を使用した Type II 補償器としてモデル化することもできます。制御ループの調整のため、この設計では以下の手順を使用します。
   - まず、目的のクロスオーバー周波数 f_c と位相マージン p_0 を選択します。クロスオーバー周波数応答アナライザの開ループ プラント ゲイン G_{f_c} とプラント位相 p_1 を書き留めます。
   - 位相ブーストは、目的のクロスオーバー周波数 f_c である p_0 を実現するための補償された調整であり、式 7 で計算できます。
   - 補償極周波数 f_p を計算するには、式 8 を使用します
   - 位相が極とゼロの間の幾何平均でピークになるので、式 9 でゼロ f_z の補償を計算します。
   - OTA を備えた Type II 補償器の場合、クロスオーバー周波数の伝達関数は、リンク テキストを自動生成できないものとして表現できます。
   - 式 8 と f_z から極とゼロを適用すると、KDC は式 11 のように計算できます。
   - 2 番目のゼロ配置は、最初のゼロよりもはるかに高い位置に配置できます。

Copyright © 2026 Texas Instruments Incorporated

- 式 8、f_z、KDC から極とゼロを Compensation Designer に適用すると、PowerSUITE GUI で 2p2z 補償を計算できます。
- *電流ループ*の補償パラメータを図 3-18 に示します。
- [Save Comp] ボタンをクリックして、補償を保存します。Compensation Designer ツールを閉じます。
- SYSCONFIG ページを保存します。

5. powerSuite のないバージョンのプロジェクトを使用する場合、[*Build Settings*] は solution_settings.h ファイルで直接変更されます。*Compensation Designer* は、*C2000Ware_DigitalPower_Install_Location\powerSUITE\source\utils* に配置されています。

```
#define LAB_NUMBER (2)
#define CHANNEL_NUMBER (3)
#define SFRA_ENABLED (true)
```

以下の式は上記のリストで参照されています。

$$\text{boost} = -(p_1 - p_0 + 90) \tag{7}$$

$$f_p = \left(\tan\left(\text{boost} \times \frac{\pi}{180}\right) + \sqrt{\tan\left(\text{boost} \times \frac{\pi}{180}\right)^2 + 1}\ \right) \times f_c \tag{8}$$

$$f_z = \frac{{f_c}^2}{f_p} \tag{9}$$

$$|G(f_c)| = \text{Kdc}\frac{\sqrt{1 + \left(\frac{f_z}{f_c}\right)^2}}{\sqrt{1 + \left(\frac{f_c}{f_p}\right)^2}} \tag{10}$$

$$\text{Kdc} = 2\pi f_c\frac{\sqrt{1 + \left(\frac{f_c}{f_p}\right)^2}}{\sqrt{1 + \left(\frac{f_z}{f_c}\right)^2}} \tag{11}$$

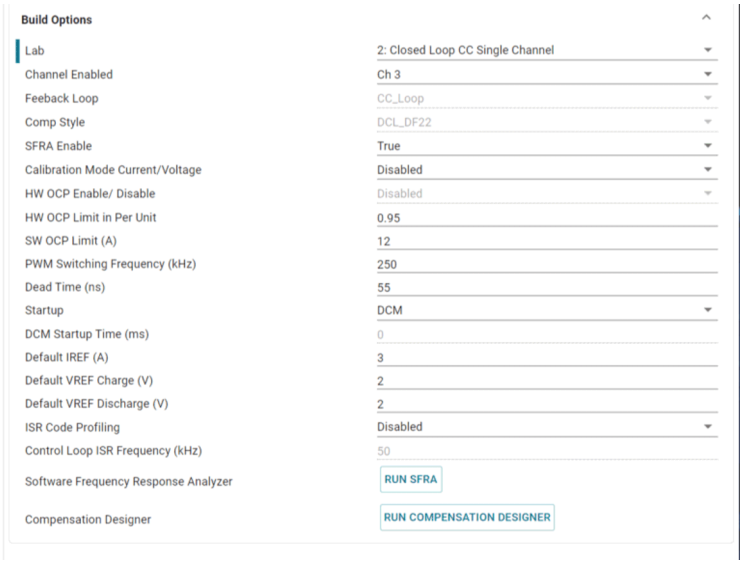


図 3-17. ラボ 2 のビルド オプション

Copyright © 2026 Texas Instruments Incorporated

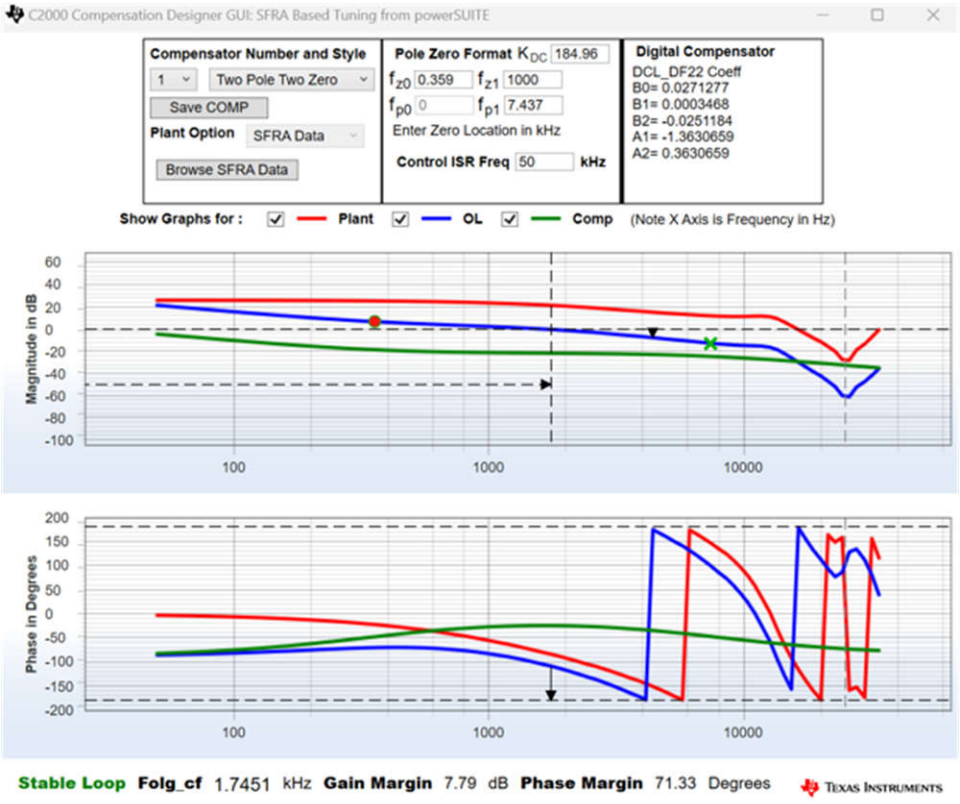


図 3-18. Compensation Designer を使用した電流ループのチューニング

3.3.5.2.2 プロジェクトのビルドおよびロードとデバッグ環境の設定

1. プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
2. プロジェクトが正常にビルドされます。
3. Project Explorer で、targetConfigs 下で適切な目標構成ファイルが有効になっていることを確認します。
4. **[Run] → [Debug]** をクリックしてデバッグ セッションを開始します。
5. するとプロジェクトがデバイスにロードされ、CCS デバッグ ビューが有効になります。メイン ルーチンの開始時にコードは停止します。
6. [Watch] および [Expressions] ウィンドウに変数を追加するには、**[View] → [Scripting Console]** をクリックして、[Scripting Console] ダイアログ ボックスを開きます。このコンソールの右上隅で、[Open] をクリックして、プロジェクト フォルダ内にある **setupdebugenv_chX.js** スクリプト ファイルを参照します。これにより、[Watch] ウィンドウに、システムをデバッグするのに必要な適切な変数が入力されます。
7. [Watch] ウィンドウで **[Continuous Refresh]** ボタン ( ) をクリックして、コントローラからの値の連続更新を有効にします。

3.3.5.2.3 コードの実行

ラボ 2 のコードを実行するには、次の手順に従います。

1. このラボを実行するには、セクション 3.3.1 で説明したようにハードウェアが設定されていることを確認します。
2. メニュー バーの をクリックしてプロジェクトを実行します。
3. [Watch] ビューの *[Expression] ウィンドウ* で BT4CH_InputVoltageSense_V が 12V ～ 15V の範囲内にあるかどうかを確認します。
4. [Expression] ウィンドウで次のパラメータを設定します。
   - BT4CH_userParam_chX->Relay_ON を 1 に設定して、出力リレーを有効にします
   - BT4CH_userParam_chX->iref_A = 7.0
   - BT4CH_userParam_chX->en_bool = 1 を設定します

Copyright © 2026 Texas Instruments Incorporated

- [Expression] ウィンドウの設定については、図 3-19 を参照してください。

5. BT4CH_measureVI_chX 変数は、DC/DC コンバータの出力電流と電圧を示します。Isense1_A の表示値は iref_A 設定に近く、誤差は ±1mA です。
6. ループ安定性をテストするための SFRA を 図 3-20 に示します。SYSCONFIG ページで Run SFRA アイコンをクリックします。SFRA GUI がポップアップ表示されます。
7. SFRA GUI でデバイスのオプションを選択します。たとえば、F28P65x の場合は浮動小数点を選択します。[Setup Connection] ボタンをクリックします。ポップアップ ウィンドウで [Boot on Connect] オプションのチェックを外し、適切な COM ポートを選択します。[OK] ボタンをクリックします。SFRA GUI に戻り、[Connect] ボタンをクリックします。
8. SFRA GUI がデバイスに接続します。これで [Start Sweep] をクリックして、SFRA 掃引を開始できるようになりました。SFRA 掃引が完了するまでには数分かかります。完了すると、図 3-21 に示すように測定値が表示されたグラフが表示されます。
9. また、周波数応答データは SFRA データ フォルダ下のプロジェクト フォルダに保存され、SFRA 実行時のタイムスタンプが記録されます。
10. ラボ演習が終了したら、*[Expression] ウィンドウ* から次のパラメータを設定してコードを停止します。
    - BT4CH_userParam_chX->en_bool = 0 を設定します
    - BT4CH_userParam_chX->Relay_ON を 0 に設定して、出力リレーを無効化します
    - プログラムを終了します

Expression	Type	Value
BT4CH_lab	enum <unname...	Lab2_singleChannelClosedLoopCC
BT4CH_sfraStatus	enum <unname...	SFRA_Enabled
BT4CH_calibrationStatus	enum <unname...	Calibration_Disabled
BT4CH_calibrationMode	enum <unname...	Disabled
BT4CH_startupMode	enum <unname...	DCM_Startup
BT4CH_HAL_InputVoltageSense_V	float	12.4000731
BT4CH_ISR2_Loading	float	0.149474204
BT4CH_ISR2_LoadingMax	float	0.17483741
BT4CH_userParam_ch3	struct <unname...	{Iref_A=7.0,VrefCharge_V=3.0,VrefDi...
Iref_A	float	7.0
VrefCharge_V	float	3.0
VrefDischarge_V	float	3.0
dir_bool	unsigned int	1
Relay_ON	unsigned int	1
Start_bool	unsigned int	1
remote_sense	unsigned int	0
DutyRef_pu	float	0.0
IbatCal_pu	float	0.200000003
VbatCal_pu	float	0.400000006
IoutGain_pu	float	0.0730994046
IoutOffset_pu	float	0.00584799051
IoutGain_A	float	13.6800022
IoutOffset_A	float	-0.0800005198
VoutGain_pu	float	0.24248305
VoutOffset_pu	float	-0.000581979752
VoutGain_V	float	4.1239996
VoutOffset_V	float	0.00240008417
VbatGain_pu	float	0.200000018
VbatOffset_pu	float	-0.000600039959
VbatGain_V	float	4.99999952
VbatOffset_V	float	0.00300019956
BT4CH_measureVI_ch3	struct <unname...	{Ibat_A=6.99993467,Vout_V=0.2138...
Ibat_A	float	6.99981737
Vout_V	float	0.2136783
Vbat_V	float	0.196476862

図 3-19. ラボ 2 [Expression] ウィンドウ、閉ループ

Copyright © 2026 Texas Instruments Incorporated

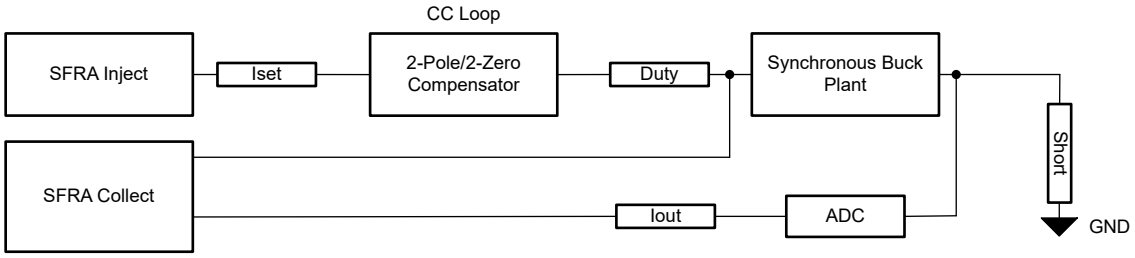


図 3-20. 閉ループ電流制御の SFRA 設定

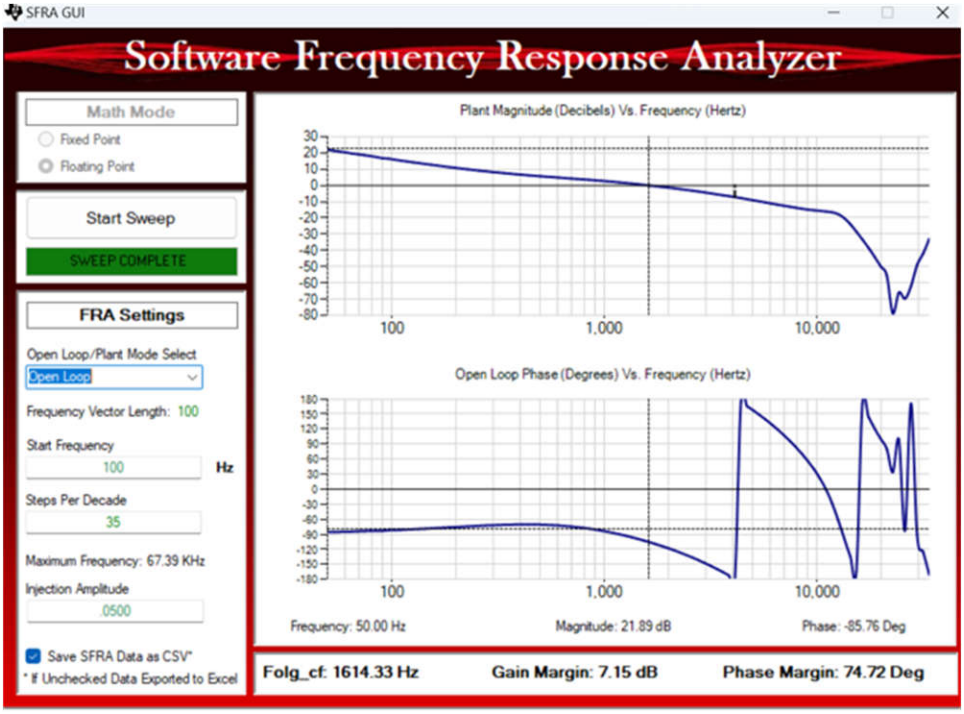


図 3-21. 電流制御の閉ループ周波数応答

3.3.5.3 ラボ 3.開ループ電圧制御 1 チャネル

3.3.5.3.1 ラボ 3 のソフトウェア オプションの設定

1. セクション 3.2.1 の概要に従って CCS プロジェクトを開きます。powerSUITE を使用している場合は、手順 2 に進みます。それ以外の場合は 3 に進みます。
2. SYSCONFIG ページを開き、[*Build Options*] セクションで以下を選択します。
   - ラボは [Lab 3: *Single Channel Open-Loop CV Control*] を選択します
   - 4 つのチャネルのいずれかを選択します。
   - SFRA をイネーブルにします。
   - ページを保存します。
3. powerSuite のないバージョンのプロジェクトを使用する場合、上記の設定は solution_settings.h ファイルで直接変更されます。

```
#define LAB_NUMBER (3)
#define CHANNEL_NUMBER (3)
#define SFRA_ENABLED (true)
```


Copyright © 2026 Texas Instruments Incorporated

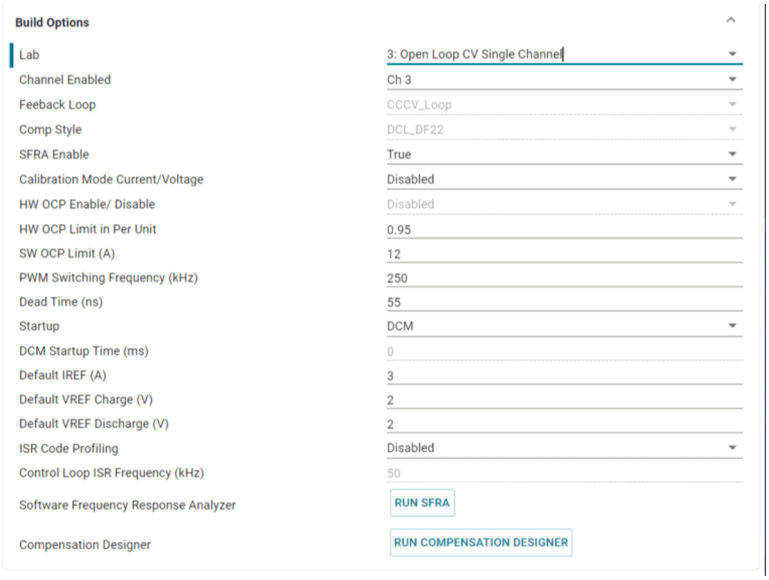


図 3-22. ラボ 3 のビルド オプション

3.3.5.3.2 プロジェクトのビルドおよびロードとデバッグ環境の設定

プロジェクトをビルドおよびロードし、デバッグ環境を設定するには、次の手順を実行します。

1. プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
2. プロジェクトが正常にビルドされます。
3. Project Explorer で、targetConfigs 下で適切な目標構成ファイルが有効になっていることを確認します。
4. **[Run] → [Debug]** をクリックしてデバッグ セッションを開始します。
5. するとプロジェクトがデバイスにロードされ、CCS デバッグ ビューが有効になります。メイン ルーチンの開始時にコードは停止します。
6. [Watch] および [Expressions] ウィンドウに変数を追加するには、**[View] → [Scripting Console]** をクリックして、[Scripting Console] ダイアログ ボックスを開きます。このコンソールの右上隅で、[Open] をクリックして、プロジェクト フォルダ内にある **setupdebugenv_chX.js** スクリプト ファイルを参照します。これにより、[Watch] ウィンドウに、システムをデバッグするのに必要な適切な変数が入力されます。
7. [Watch] ウィンドウで **[Continuous Refresh]** ボタン () をクリックして、コントローラからの値の連続更新を有効にします。

3.3.5.3.3 コードの実行

ラボ 3 のコードを実行するには、次の手順に従います。

1. セクション 3.3.1 に示すようにテスト設定を使用します。
2. メニュー バーの  をクリックしてプロジェクトを実行します。
3. [Watch] ビューの *[Expression] ウィンドウ* で BT4CH_InputVoltageSense_V が 12V ～ 15V の範囲内にあるかどうかを確認します。
4. [Expression] ウィンドウで次のパラメータを設定します。
   - BT4CH_userParam_chX->Relay_ON を 1 に設定して、出力リレーを無効化します
   - BT4CH_userParam_V_I_chm->iref_A = 15.0
   - BT4CH_userParam_chX->en_bool = 1 を設定します
   - [Expression] ウィンドウの設定については、図 3-23 を参照してください。
5. BT4CH_measureVI_chX 変数は、DC/DC コンバータの出力電流と電圧を示します。Ibatsense_A の表示値は iref_A 設定に近く、誤差は ±1mA です。
6. 図 3-24 に、開ループ電圧制御周波数応答を測定するための SFRA 設定を示します。
7. SYSCONFIG ページで Run SFRA アイコンをクリックします。SFRA GUI がポップアップ表示されます。

Copyright © 2026 Texas Instruments Incorporated

8. SFRA GUI でデバイスのオプションを選択します。たとえば、F28P65x の場合は浮動小数点を選択します。[Setup Connection] ボタンをクリックします。ポップアップ ウィンドウで [Boot on Connect] オプションのチェックを外し、適切な COM ポートを選択します。[OK] ボタンをクリックします。SFRA GUI に戻り、[Connect] ボタンをクリックします。
9. SFRA GUI がデバイスに接続します。これで [Start Sweep] ボタンをクリックして、SFRA 掃引を開始できるようになりました。SFRA 掃引が完了するまでには数分かかります。完了すると、図 3-25 に示すように測定値が表示されたグラフが表示されます。
10. また、周波数応答データは SFRA データ フォルダ下のプロジェクト フォルダに保存され、SFRA 実行時のタイムスタンプが記録されます。
11. ラボ演習が終了したら、*[Expression] ウィンドウ*から次のパラメータを設定してコードを停止します。
    - BT4CH_userParam_V_I_chm->iref_A = 0
    - BT4CH_userParam_chX->en_bool = 0 を設定します
    - BT4CH_userParam_chX->Relay_ON を 0 に設定して、出力リレーを無効化します
    - プログラムを終了します

BT4CH_calibrationMode	enum <unname...	Disabled
BT4CH_startupMode	enum <unname...	DCM_Startup
BT4CH_HAL_InputVoltageSense_V	float	12.4089355
BT4CH_ISR2_Loading	float	0.149675161
BT4CH_ISR2_LoadingMax	float	0.174261391
BT4CH_userParam_ch3	struct <unname...	{Iref_A=7.0,VrefCharge_V=1.0,VrefDi...
Iref_A	float	7.0
VrefCharge_V	float	1.0
VrefDischarge_V	float	3.0
dir_bool	unsigned int	1
Relay_ON	unsigned int	1
Start_bool	unsigned int	1
remote_sense	unsigned int	0
DutyRef_pu	float	0.0
IbatCal_pu	float	0.200000003
VbatCal_pu	float	0.400000006
IoutGain_pu	float	0.0730994046
IoutOffset_pu	float	0.00584799051
IoutGain_A	float	13.6800022
IoutOffset_A	float	-0.0800005198
VoutGain_pu	float	0.24248305
VoutOffset_pu	float	-0.000581979752
VoutGain_V	float	4.1239996
VoutOffset_V	float	0.00240008417
VbatGain_pu	float	0.200000018
VbatOffset_pu	float	-0.000600039959
VbatGain_V	float	4.99999952
VbatOffset_V	float	0.00300019956
BT4CH_measureVI_ch3	struct <unname...	{Ibat_A=7.00024748,Vout_V=0.1169..
Ibat_A	float	7.00016308
Vout_V	float	0.116786078
Vbat_V	float	0.0993618295

図 3-23. ラボ 3 [Expression] ウィンドウ、閉ループ

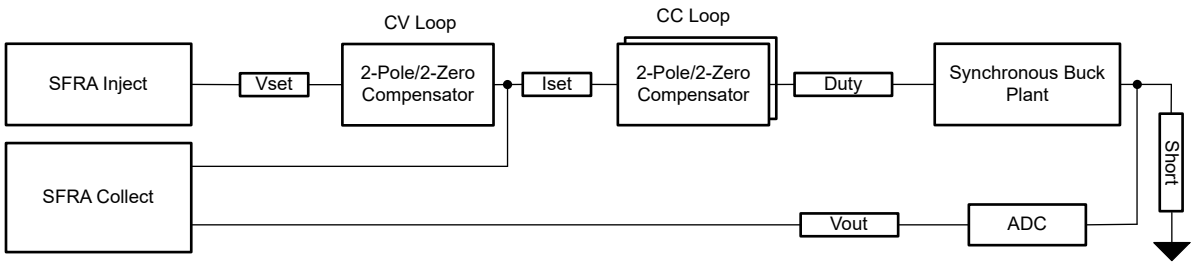


図 3-24. 開ループ電流制御の SFRA 設定

Copyright © 2026 Texas Instruments Incorporated

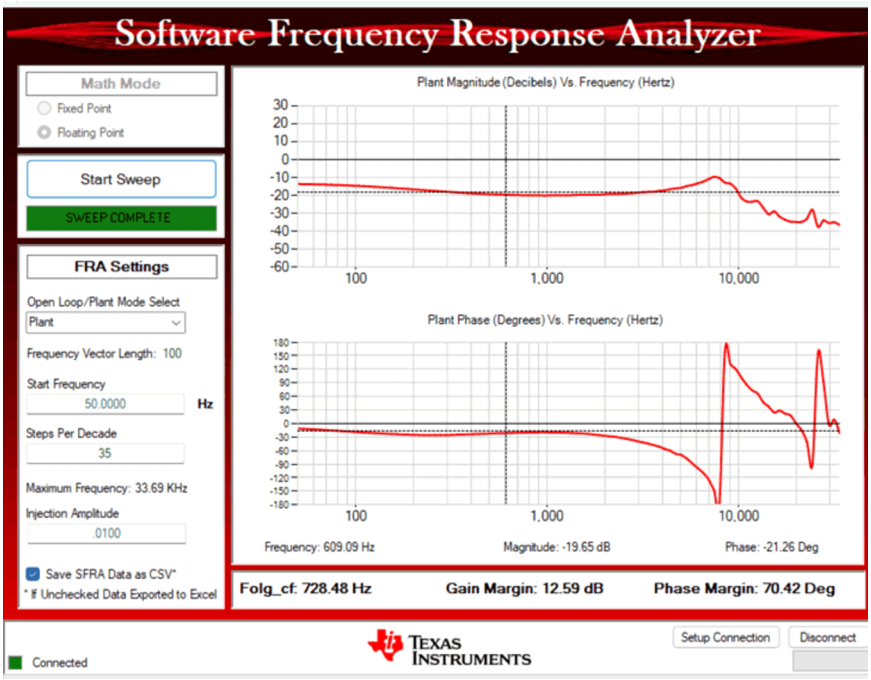


図 3-25. 電圧制御の開ループ周波数応答

3.3.5.4 ラボ 4.閉ループ電流および電圧制御 1 チャネル

3.3.5.4.1 ラボ 4 のソフトウェア オプションの設定

1. このラボを実行するには、前のセクション、セクション 3.3.2 で説明したようにハードウェアが設定されていることを確認します。
2. セクション 3.2.1 の概要に従って CCS プロジェクトを開きます。powerSUITE を使用している場合は、手順 3 に進みます。それ以外の場合は手順 4 に進みます。
3. SYSCONFIG ページを開き、[Build Options] セクションで以下を選択します。
   - ラボは [Lab 4: *閉ループ CCCV シングル チャネル*
   - チャンネルを選択します。
   - SFRA をイネーブルにします。
   - Compensation Designer を開きます
4. powerSUITE 以外のバージョンのプロジェクトを使用する場合、[*Build Settings*] は solution_settings.h ファイルで直接変更されます。Compensation Designer は C2000Ware_DigitalPower_Install_Location\powerSUITE\source\utils にあります。

```
#define LAB_NUMBER (4)
#define CHANNEL_NUMBER (3)
#define SFRA_ENABLED (true)
```


Copyright © 2026 Texas Instruments Incorporated

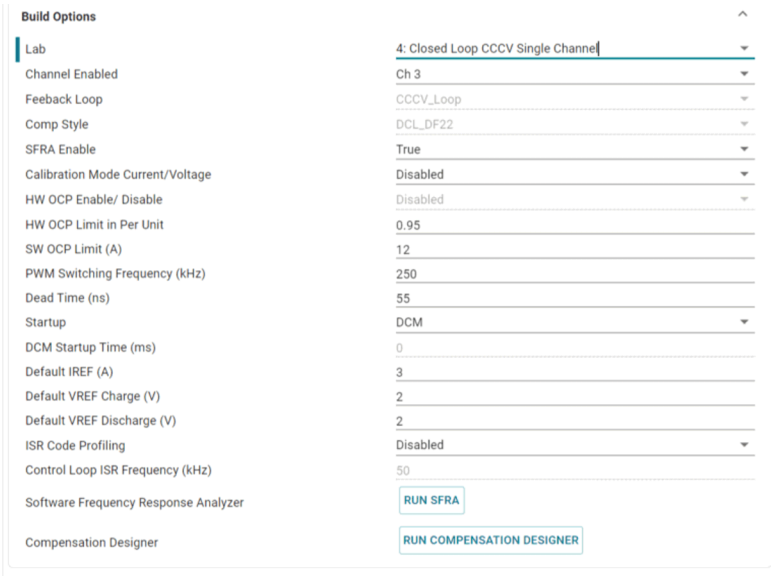


図 3-26. ラボ 4 のビルド オプション

図 3-27. 電圧ループの調整

3.3.5.4.2 プロジェクトのビルドおよびロードとデバッグ環境の設定

1. プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
2. プロジェクトが正常にビルドされます。
3. Project Explorer で、targetConfigs 下で適切な目標構成ファイルが有効になっていることを確認します。
4. 次に、**[Run]** → **[Debug]** をクリックしてデバッグ セッションを起動します。
5. するとプロジェクトがデバイスにロードされ、CCS デバッグ ビューが有効になります。メイン ルーチンの開始時にコードは停止します。

Copyright © 2026 Texas Instruments Incorporated

6. [Watch] および [Expressions] ウィンドウに変数を追加するには、**[View] → [Scripting Console]** をクリックして、[Scripting Console] ダイアログ ボックスを開きます。このコンソールの右上隅で、[Open] をクリックして、プロジェクト フォルダ内にある **setupdebugenv_chX.js** スクリプト ファイルを参照します。これにより、[Watch] ウィンドウに、システムをデバッグするのに必要な適切な変数が入力されます。
7. [Watch] ウィンドウで **[Continuous Refresh]** ボタン () をクリックして、コントローラからの値の連続更新を有効にします。

3.3.5.4.3 コードの実行

ラボ 4 のコードを実行するには、次の手順に従います。

1. セクション 3.3.1 に示すようにテスト設定を使用します。
2. メニュー バーの  をクリックしてプロジェクトを実行します。
3. [Watch] ビューの *[Expression] ウィンドウ*で BT4CH_InputVoltageSense_V が 12V～15V の範囲内にあるかどうかを確認します。
4. [Expression] ウィンドウで次のパラメータを設定します。
   - BT4CH_userParam_chX->Relay_ON を 1 に設定して、出力リレーをイネーブルにします
   - BT4CH_userParam_chX->iref_A = 8.5
   - BT4CH_userParam_chX->vrefCharge_V = 0.075
   - BT4CH_userParam_chX->en_bool = 1 を設定します
   - [Expression] ウィンドウの設定については、図 3-28 を参照してください。
5. BT4CH_measureVI_chX 変数は、DC/DC コンバータの出力電流と電圧を示します。Vbatsense_V の表示値が vrefCharge_V に近く、誤差は ±0.5mV です。
6. 図 3-29 に、閉ループ電圧制御周波数応答を測定するための SFRA 設定を示します。
7. SYSCONFIG ページで Run SFRA アイコンをクリックします。 SFRA GUI がポップアップ表示されます。
8. SFRA GUI でデバイスのオプションを選択します。たとえば、F28P65x の場合は浮動小数点を選択します。[Setup Connection] ボタンをクリックします。ポップアップ ウィンドウで [Boot on Connect] オプションのチェックを外し、適切な COM ポートを選択します。[OK] ボタンをクリックします。SFRA GUI に戻り、[Connect] ボタンをクリックします。
9. SFRA GUI がデバイスに接続します。これで [Start Sweep] ボタンをクリックして、SFRA 掃引を開始できるようになりました。SFRA 掃引が完了するまでには数分かかります。完了すると、図 3-30 に示すように測定値が表示されたグラフが表示されます。
10. また、周波数応答データは SFRA データ フォルダ下のプロジェクト フォルダに保存され、SFRA 実行時のタイムスタンプが記録されます。
11. ラボ演習が終了したら、*[Expression] ウィンドウ*から次のパラメータを設定してコードを停止します。
    - BT4CH_userParam_chX->iref_A = 0
    - BT4CH_userParam_chX->vrefCharge_V = 0
    - BT4CH_userParam_chX->en_bool = 0 を設定します
    - BT4CH_userParam_chX->Relay_ON を 0 に設定して、出力リレーを無効化します
    - プログラムを終了します

Copyright © 2026 Texas Instruments Incorporated

Expression	Type	Value
BT4CH_calibrationMode	enum <unname...	Disabled
BT4CH_startupMode	enum <unname...	DCM_Startup
BT4CH_HAL_InputVoltageSense_V	float	12.4133673
BT4CH_ISR2_Loading	float	0.165582791
BT4CH_ISR2_LoadingMax	float	0.179018527
BT4CH_userParam_ch3	struct <unname...	{Iref_A=8.5,VrefCharge_V=0.075000...
Iref_A	float	8.5
VrefCharge_V	float	0.075000003
VrefDischarge_V	float	3.0
dir_bool	unsigned int	1
Relay_ON	unsigned int	1
Start_bool	unsigned int	1
remote_sense	unsigned int	1
DutyRef_pu	float	0.0
IbatCal_pu	float	0.200000003
VbatCal_pu	float	0.400000006
IoutGain_pu	float	0.0730994046
IoutOffset_pu	float	0.00584799051
IoutGain_A	float	13.6800022
IoutOffset_A	float	-0.0800005198
VoutGain_pu	float	0.24248305
VoutOffset_pu	float	-0.000581979752
VoutGain_V	float	4.1239996
VoutOffset_V	float	0.00240008417
VbatGain_pu	float	0.200000018
VbatOffset_pu	float	-0.000600039959
VbatGain_V	float	4.99999952
VbatOffset_V	float	0.00300019956
BT4CH_measureVI_ch3	struct <unname...	{Ibat_A=5.27749491,Vout_V=0.0879...
Ibat_A	float	5.27365923
Vout_V	float	0.0880440623
Vbat_V	float	0.0750049874

図 3-28. ラボ 4 [Expression] ウィンドウ、閉ループ

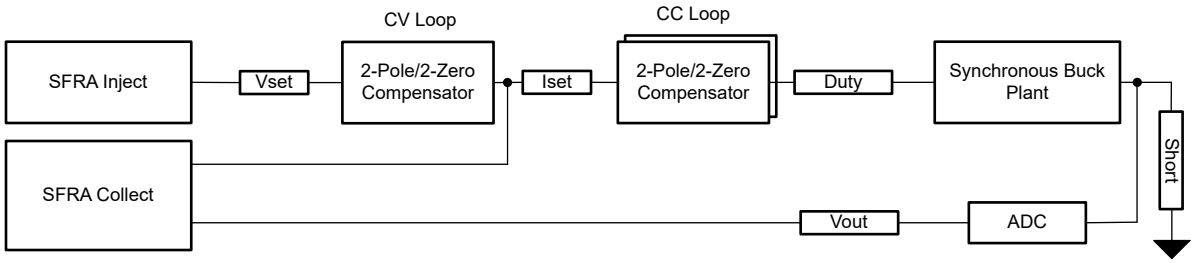


図 3-29. 閉ループ電流制御の SFRA 設定

Copyright © 2026 Texas Instruments Incorporated

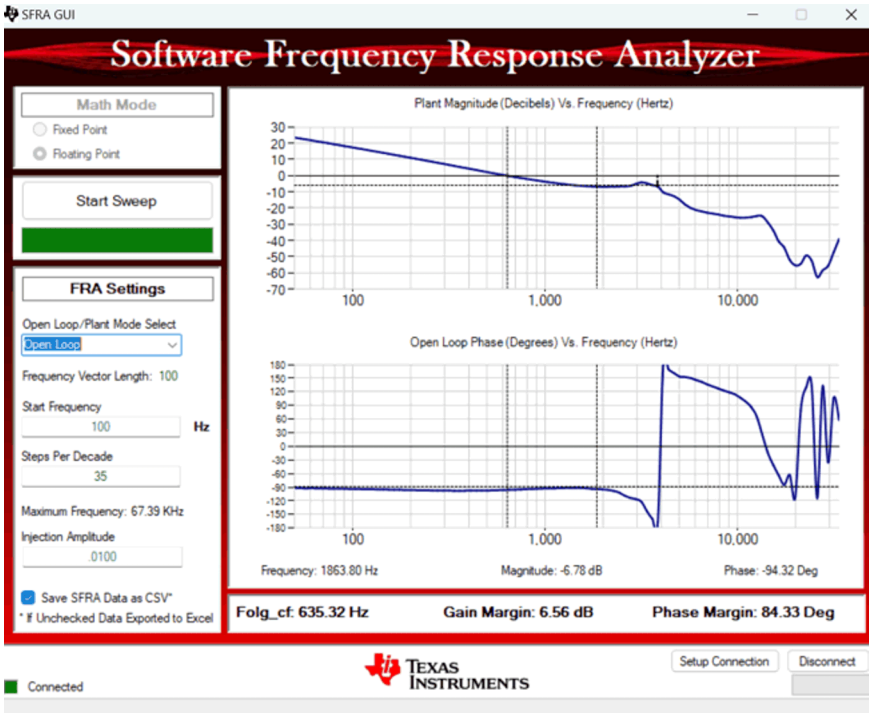


図 3-30. 電圧制御の閉ループ周波数応答

3.3.5.5 ラボ 5.閉ループ電流および電圧制御 4 チャネル

3.3.5.5.1 ラボ 5 のソフトウェア オプションの設定

1. このラボを実行するには、前のセクション、セクション 3.3.2 で説明したようにハードウェアが設定されていることを確認します。
2. セクション 3.2.1 の概要に従って CCS プロジェクトを開きます。powerSUITE を使用している場合は、手順 3 に進みます。それ以外の場合は手順 4 に進みます。
3. SYSCONFIG ページを開き、[Build Options] セクションで以下を選択します。
   - ラボは [Lab 5: *Closed-Loop CCCV Dual Phase]* を選択します
   - すべてのチャネルを選択します
   - SFRA オプションを無効化します
   - Compensation Designer を開きます
4. powerSUITE 以外のバージョンのプロジェクトを使用する場合、[*Build Settings*] は solution_settings.h ファイルで直接変更されます。

```
#define LAB_NUMBER (5)
#define CHANNEL_NUMBER (5)
#define SFRA_ENABLED (false)
```


Copyright © 2026 Texas Instruments Incorporated

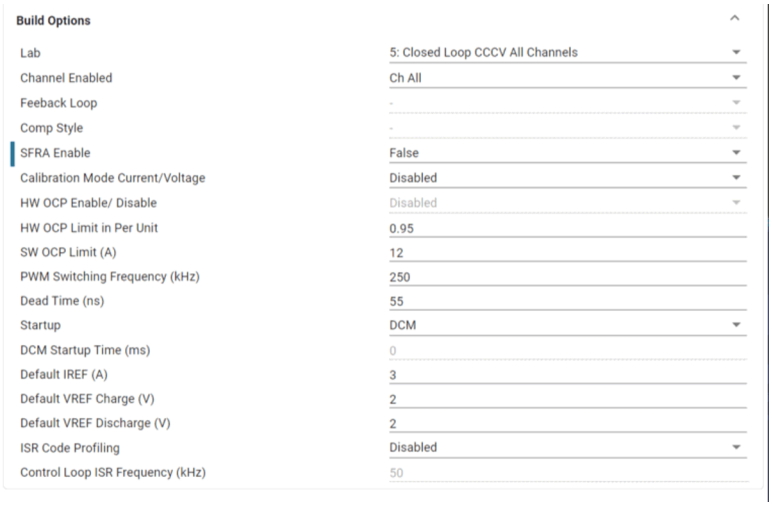


図 3-31. ラボ 5 のビルド オプション

3.3.5.5.2 プロジェクトのビルドおよびロードとデバッグ環境の設定

プロジェクトをビルドおよびロードし、デバッグ環境を設定するには、次の手順を実行します。

1. プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
2. プロジェクトが正常にビルドされます。
3. Project Explorer で、targetConfigs 下で適切な目標構成ファイルが有効になっていることを確認します。
4. **[Run]** → **[Debug]** をクリックしてデバッグ セッションを開始します。
5. するとプロジェクトがデバイスにロードされ、CCS デバッグ ビューが有効になります。メイン ルーチンの開始時にコードは停止します。
6. [Watch] および [Expressions] ウィンドウに変数を追加するには、**[View]** → **[Scripting Console]** をクリックして、[Scripting Console] ダイアログ ボックスを開きます。このコンソールの右上隅で、[Open] をクリックして、プロジェクト フォルダ内にある **setupdebugenv_chX.js** スクリプト ファイルを参照します。これにより、[Watch] ウィンドウに、システムをデバッグするのに必要な適切な変数が入力されます。
7. [Watch] ウィンドウで **[Continuous Refresh]** ボタン ( ) をクリックして、コントローラからの値の連続更新を有効にします。

3.3.5.5.3 コードの実行

ラボ 5 のコードを実行するには、次の手順に従います。

1. セクション 3.3.1 に示すようにテスト設定を使用します。
2. メニュー バーの をクリックしてプロジェクトを実行します。
3. [Watch] ビューの *[Expression] ウィンドウ* で BT4CH_InputVoltageSense_V が 12V ～ 15V の範囲内にあるかどうかを確認します。
4. [Expression] ウィンドウで次のパラメータを設定します。
   - BT4CH_userParam_chX->Relay_ON を 1 に設定して、出力リレーを有効にします
   - BT4CH_userParam_chX->iref_A = 5.0
   - BT4CH_userParam_chX->vrefCharge_V = 3
   - BT4CH_userParam_chX->en_bool = 1 を設定します
   - [Expression] ウィンドウの設定については、図 3-32 を参照してください。
5. BT4CH_measureVI_chX 変数は、DC/DC コンバータの出力電流と電圧を示します。
6. Iref と Vref を変更して、定電流モードと定電圧モードの間の遷移を確認します。
7. 電流の方向を変更するには、BT4CH_userParam_chX->Dir_bool を切り替えます。
8. ラボ演習が終了したら、*[Expression] ウィンドウ* から次のパラメータを設定してコードを停止します。
   - BT4CH_userParam_chX->iref_A = 0
   - BT4CH_userParam_chX->vrefCharge_V = 0

Copyright © 2026 Texas Instruments Incorporated

- BT4CH_userParam_chX->en_bool = 0 を設定します
- BT4CH_userParam_chX->Relay_ON を 0 に設定して、出力リレーを無効化します
- プログラムを終了します
- すべてのチャネルがシャットダウンされるまで、すべてのチャネルで繰り返します

Expression	Type	Value
BT4CH_lab	enum <unname...	Lab5_AllChannelClosedLoopCV
BT4CH_sfraStatus	enum <unname...	SFRA_Disabled
BT4CH_calibrationStatus	enum <unname...	Calibration_Disabled
BT4CH_calibrationMode	enum <unname...	Disabled
BT4CH_startupMode	enum <unname...	DCM_Startup
BT4CH_HAL_InputVoltageSense_V	float	13.2465382
BT4CH_ISR2_Loading	float	0.280758649
BT4CH_ISR2_LoadingMax	float	0.292939395
BT4CH_userParam_ch1	struct <unname...	{Iref_A=5.0,VrefCharge_V=2.0,VrefDi...
Iref_A	float	5.0
VrefCharge_V	float	2.0
VrefDischarge_V	float	3.0
dir_bool	unsigned int	1
Relay_ON	unsigned int	1
Start_bool	unsigned int	1
remote_sense	unsigned int	0
DutyRef_pu	float	0.0
IbatCal_pu	float	0.200000003
VbatCal_pu	float	0.400000006
IoutGain_pu	float	0.0789889395
IoutOffset_pu	float	0.00829383731
IoutGain_A	float	12.6600008
IoutOffset_A	float	-0.104999982
VoutGain_pu	float	0.24248305
VoutOffset_pu	float	-0.000581979752
VoutGain_V	float	4.1239996
VoutOffset_V	float	0.00240008417
VbatGain_pu	float	0.200000018
VbatOffset_pu	float	-0.000600039959
VbatGain_V	float	4.99999952
VbatOffset_V	float	0.00300019956
BT4CH_measureVI_ch1	struct <unname...	{Ibat_A=-0.0642971396,Vout_V=1.6...
Ibat_A	float	-0.0635485798

図 3-32. ラボ 5 [Expression] ウィンドウ

3.3.5.6 較正

1. このラボを実行するには、セクション 3.3.3 で説明したようにハードウェアが設定されていることを確認します。ゲイン誤差とオフセット誤差の較正には、2 点較正法を使用します。
2. 電流を測定する方法は 3 種類あります。
   a. 電圧モードでは、外付けの高精度抵抗と 6.5 桁の DMM を使用して、抵抗の両端での電圧降下を測定することにより、電流を計算できます。
   b. TIDA-010086 ボード上のセンス抵抗の両端での電圧を測定します
   c. 基板上の TP1 と TP2 をプローブして、シャントの両端での電圧降下を測定します
   d. 電子負荷読み取り値を使用しますが、この方法では高精度の電子負荷またはソース測定ユニット (SMU) 機器が必要です。
3. 電圧を測定するには、降圧コンバータの出力電圧とリモート センス端子 J8 の間に DMM を使用します
4. SYSCONFIG ページを開き、ラボ 5 を選択し、[Calibration Mode] を [Current Calibration] に設定します。*図 3-33* に、電流キャリブレーションの SYSCONFIG ページの設定を示します。
   - SYSCONFIG ページを保存し、コードを実行します。
   - [Expression] ウィンドウを開きます。
   - 出力電流は、BT4PH_userParam_V_I_chX->ibatCal_pu パラメータを使用して更新します。
   - *BT4CH_userParam_chX->Relay_ON* を 1 に設定して、出力リレーをイネーブルにします。

Copyright © 2026 Texas Instruments Incorporated

- BT4CH_userParam_chX->en_bool = 1 を設定します。
- BT4CH_userParam_chX->ibatCal_pu を「0.3」および「0.5」に設定し、出力電流読み取り値を書き留めます。
- bt4ch_gan_cal.h ファイルの実際の出力電流読み取り値を更新します。

```
#define BT4CH_IBAT_ACTUAL_CH1_P1_A ((float32_t)3.59)
#define BT4CH_IBAT_ACTUAL_CH1_P2_A ((float32_t)6.02)
```

- チャネル 2、3、4 について、この手順を繰り返します。

5. SYSCONFIG ページを開いてラボ 5 を選択し、[Calibration Mode] を [Voltage Calibration] に設定します。図 3-34 に、電圧キャリブレーションの SYSCONFIG ページの設定を示します。
   - SYSCONFIG ページを保存し、コードを実行します。
   - [Expression] ウィンドウを開きます。
   - 出力電流は、BT4PH_userParam_V_I_chX->vbatCal_pu パラメータを使用して更新します。
   - *BT4CH_userParam_chX->Relay_ON* を 1 に設定して、出力リレーをイネーブルにします。
   - BT4CH_userParam_chX->en_bool = 1 を設定します。
   - BT4CH_userParam_chX->vbatCal_pu を「0.2」および「0.6」に設定し、出力電流の測定値を書き留めます。bt4ch_cal.h ファイルの実際の出力電流読み取り値を更新します。

```
#define BT4CH_VBAT_ACTUAL_CH1_P1_V ((float32_t)0.9976)
#define BT4CH_VBAT_ACTUAL_CH1_P2_V ((float32_t)2.998)
```

   - チャネル 2、3、4 について、この手順を繰り返します。
6. 較正が完了したら、プロジェクトを停止して SYSCONFIG ページを開き、較正モードを無効化します。
7. powerSUITE 以外のバージョンのプロジェクトを使用する場合、[*Build Settings*] は solution_settings.h ファイルで直接変更されます。電流キャリブレーションの場合は CALIBRATION_MODE を 1 に設定し、電圧キャリブレーションの場合は 2 に設定します。

```
#define LAB_NUMBER (5)
#define CHANNEL_NUMBER (5)
#define CALIBRATION_ENABLED (true)
#define CALIBRATION_MODE (1)
```

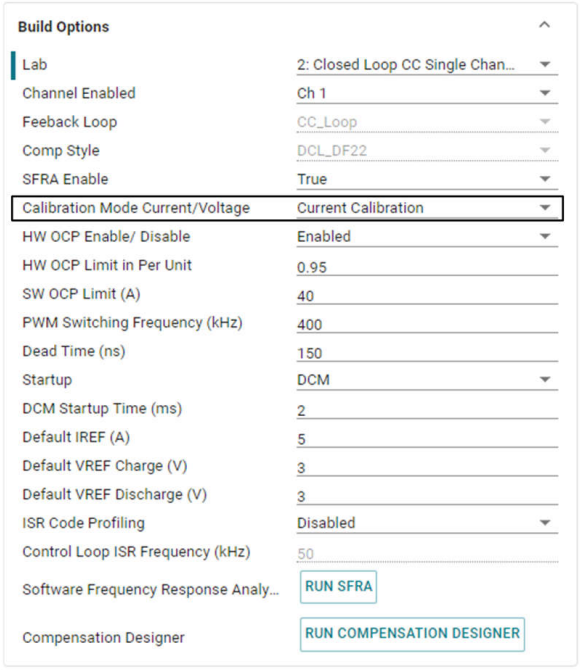


図 3-33. 電流キャリブレーションのビルド オプション

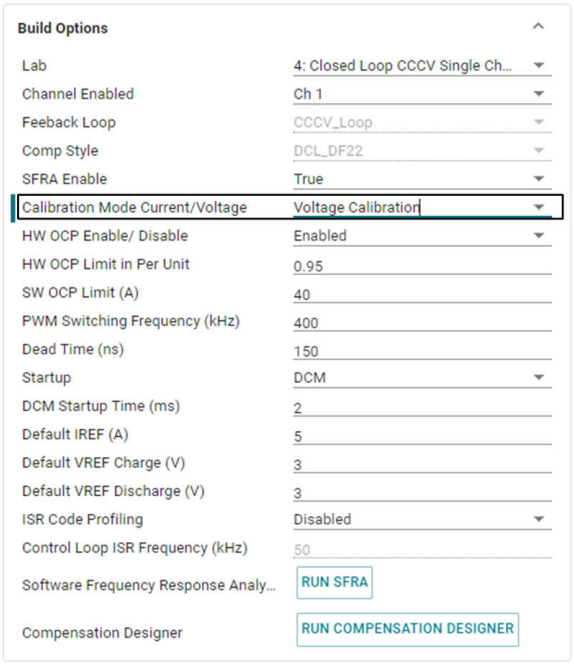


図 3-34. 電圧キャリブレーションのビルド オプション

Copyright © 2026 Texas Instruments Incorporated

3.4 テスト結果

3.4.1 電流負荷レギュレーション

セクション 3.3.3 のセットアップを使用して、電流負荷レギュレーションをテストします。

表 3-2. 電流負荷レギュレーション

FSR (A)	10							
出力モード	充電				放電			
ISET (A)	0.1	1	5	10	0.1	1	5	10
E ロード CV モード	端子電圧読み取り値							
VSET(V)	I_{actual} (A)	I_{actual} (A)	I_{actual} (A)	I_{actual} (A)	I_{actual} (A)	I_{actual} (A)	I_{actual} (A)	I_{actual} (A)
1	0.0996	0.9998	4.99993	10.0005	-0.1005	-1.0001	-5.001	-10.001
2	0.0995	0.9996	4.99995	10.0008	-0.1002	-1.00025	-5.0011	-10.001
3	0.09985	0.9995	4.99995	10.0008	-0.1005	-1.0002	-5.001	-10.0008
4	0.09925	0.9995	4.99995	10.0008	-0.1005	-1.0002	-5.001	-10.0011
誤差 (mA)	0.7500	0.5	0.05	-0.8	-0.5	-0.25	-1.1	-1.1
誤差 (%FSR)	0.0075	0.0050	0.0005	0.0080	0.0050	0.0025	0.0110	0.0110

3.4.2 電圧負荷レギュレーション

表 3-3. 電圧負荷レギュレーション

FSR (V)	5					
VSET(V)	0.2	1	2	3	4	5
E–LOAD CC モード	電流読み取り値					
ISET(A)	V_{actual} (V)	V_{actual} (V)	V_{actual} (V)	V_{actual} (V)	V_{actual} (V)	V actual(V)
無負荷	0.20003	1.00001	2.00001	2.99997	3.99994	4.99991
1	0.20002	0.99998	1.99998	2.99998	3.99994	5.0001
4	0.20002	1.0002	2.00004	3.00006	3.99997	5.00006
8	0.20002	1.00005	2.00002	3.00002	3.99993	5.00002
10	0.20002	1.00004	2.00002	3.00004	3.99994	5.00002
誤差 (mV)	0.0300	0.0500	0.0400	0.0600	-0.0700	-0.0900
誤差 (%FSR)	0.0003	0.0005	0.0004	0.0006	0.0007	0.0009

Copyright © 2026 Texas Instruments Incorporated

3.4.3 電流直線性テスト

電流制御の精度は、電流検出抵抗、電流センスアンプのゲイン、オフセット、ドリフトによって異なります。このテストでは、充電モードで 2 点較正を適用します。全体の電流誤差は、較正後も ±0.02% 未満です。図 3-35 に結果を示します。

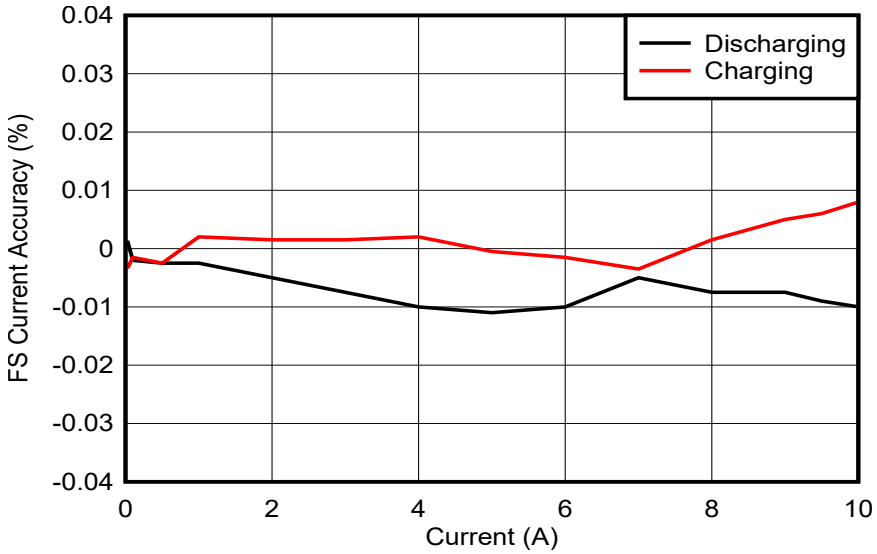


図 3-35. 電流制御の精度テスト

3.4.4 電圧ループ直線性テスト

電圧ループの較正は、無負荷状況で行われます。図 3-36 の黒い曲線は、無負荷条件下での電圧制御の直線性を示しています。赤い曲線は 10A 負荷条件下での電圧精度を示しています。このテストを行うことで、さまざまな出力負荷条件で、電圧ループ レギュレーション誤差が ±0.01% 未満に維持されることが証明されています。

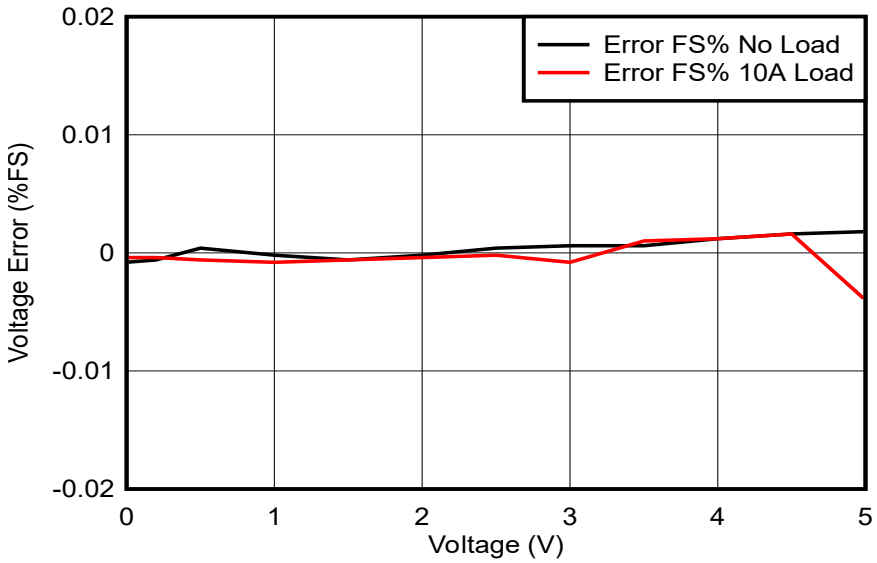


図 3-36. 電圧制御直線性テスト

Copyright © 2026 Texas Instruments Incorporated

3.4.5 双方向電流スイッチング時間

充電遷移と放電遷移の両方について (図 3-37 および図 3-38 を参照)、電流応答時間は高速かつ滑らかに維持されます。適切な調整を行うと、スイッチング電流動作の過渡応答時間は 400µs 内に達する可能性があります。

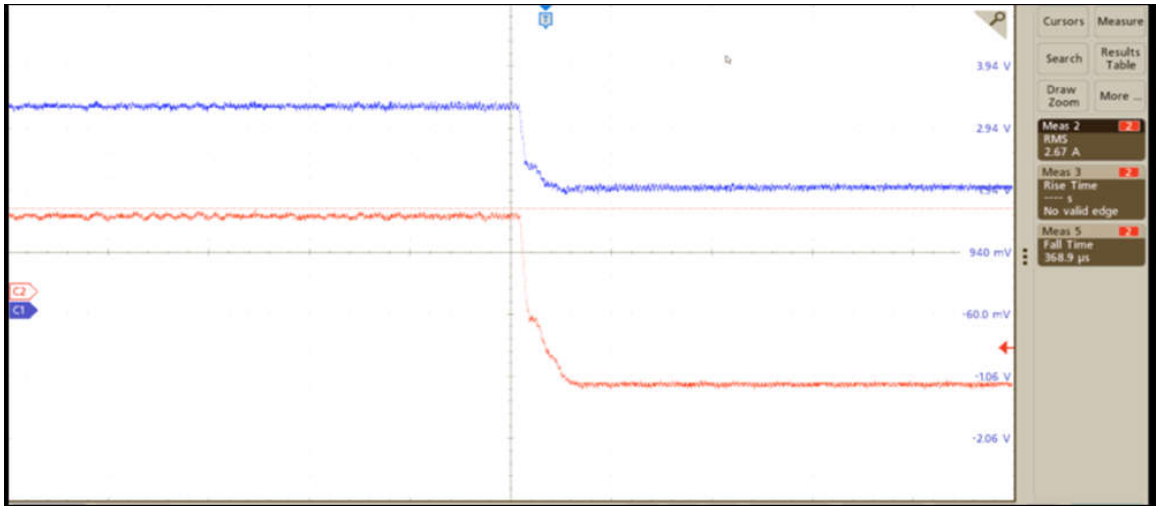


図 3-37. 電流遷移、充電から放電

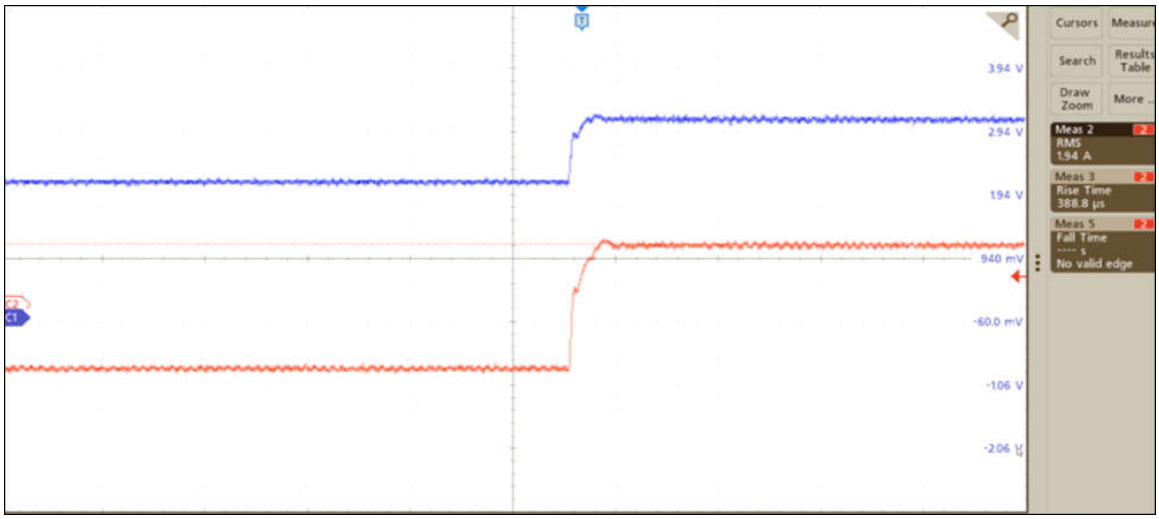


図 3-38. 電流遷移、充電から放電

Copyright © 2026 Texas Instruments Incorporated

3.4.6 電流ステップ応答

制御ループの最適化は、15mΩ の出力負荷時に行われ、100µs の立ち上がり時間と立ち下がり時間内に達成されます。バッテリ負荷が異なると、結果も異なります。目的の結果を達成するには、バッテリ負荷を使用してループを最適化することが重要です。電力遷移波形については、図 3-39 と図 3-40 を参照してください。

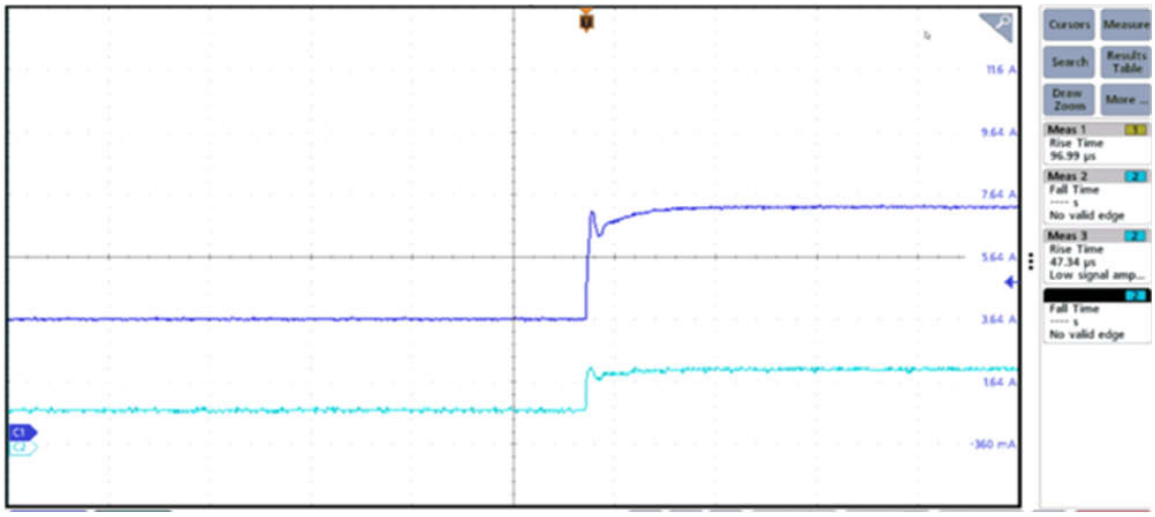


図 3-39. 電流ステップ立ち上がり時間

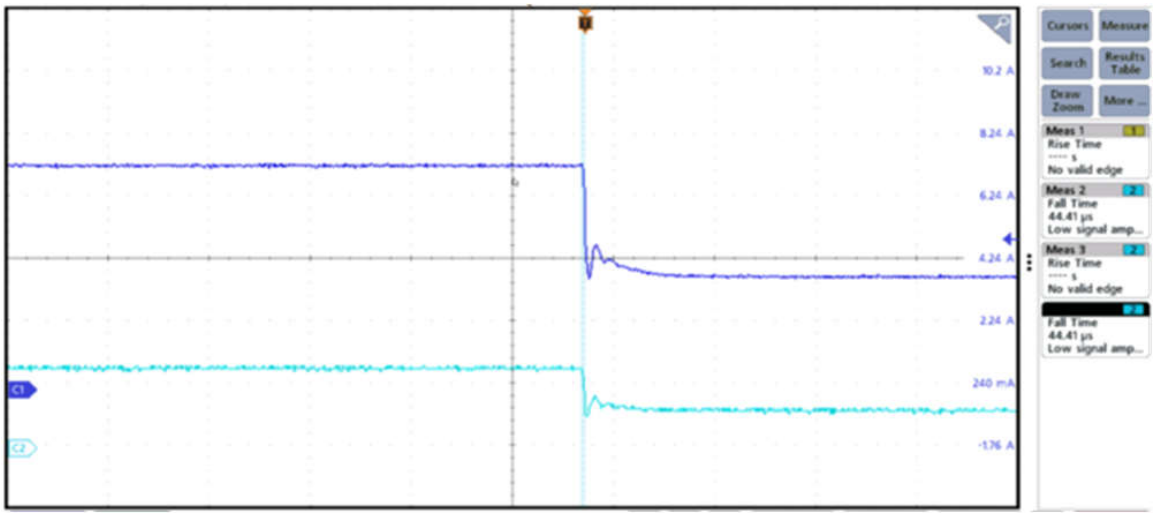


図 3-40. 電流負荷ステップの立ち下がり時間

Copyright © 2026 Texas Instruments Incorporated

4 設計とドキュメントのサポート

4.1 デザイン ファイル

4.1.1 回路図

回路図をダウンロードするには、TIDA-010086 のデザイン ファイルを参照してください。

4.1.2 BOM

部品表 (BOM) をダウンロードするには、TIDA-010086 の設計ファイルを参照してください。

4.2 ツールとソフトウェア

ツール

TMDSCNCD28P65X	F28P65 controlCARD 評価基板

ソフトウェア

CCSTUDIO	Code Composer Studio (CCS) 統合開発環境 (IDE)
C2000WARE-DIGITALPOWER-SDK	C2000™ マイコン向け DigitalPower ソフトウェア開発キット (SDK)

4.3 ドキュメントのサポート

1. テキサス・インスツルメンツ、『*TMS320F28P65x リアルタイム マイクロコントローラ データ シート*』SPRSP69
2. テキサス・インスツルメンツ、『*ADS9324 16 チャネル、16 ビット、1MSPS、アナログ フロントエンド内蔵の同時サンプリング SAR ADC*』データシート
3. テキサス・インスツルメンツ、『*INA630 高精度、126dB CMRR、間接電流帰還計測アンプ*』データシート

4.4 サポート・リソース

テキサス・インスツルメンツ E2E™ サポート・フォーラムは、エンジニアが検証済みの回答と設計に関するヒントをエキスパートから迅速かつ直接得ることができる場所です。既存の回答を検索したり、独自の質問をしたりすることで、設計で必要な支援を迅速に得ることができます。

リンクされているコンテンツは、各寄稿者により「現状のまま」提供されるものです。これらはテキサス・インスツルメンツの仕様を構成するものではなく、必ずしもテキサス・インスツルメンツの見解を反映したものではありません。テキサス・インスツルメンツの使用条件を参照してください。

4.5 商標

テキサス・インスツルメンツの™, C2000™, and Code Composer Studio™, and テキサス・インスツルメンツ E2E™ are trademarks of Texas Instruments.
すべての商標は、それぞれの所有者に帰属します。

5 著者について

ETHAN YU は、テキサス・インスツルメンツのシステム エンジニアであり、テストおよび測定アプリケーションのリファレンス デザイン開発を担当しています。Ethan はテキサス A&M 大学で電気工学の学士号を取得しています。

著者は、このリファレンス デザインに対するサポートについて、SHAURY ANAND、MEGHANA MANAVAZHI、TIM PRICE に対して感謝しています。

Copyright © 2026 Texas Instruments Incorporated

6 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 * (November 2020) to Revision A (December 2025) **Page**

- リチウムイオン バッテリの形成セクションを「*リチウムイオンセルの形成機器*」に置き換え。 2
- 「*主なシステム仕様*」の表を更新。 2
- 図 2-1 を更新。 3
- セクション 2.2 を更新。 4
- 「*出力インダクタおよびコンデンサの選択*」セクションを追加。 5
- 「*主な使用製品*」を新しいデバイスで更新。 8
- *TIDA-010086 ハードウェア* の画像を更新。 9
- 「*ソフトウェア*」セクションを更新。 10
- 「*テスト方法*」セクションを更新。 19
- 「*テストと結果*」セクションを更新 37

Copyright © 2026 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 © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月