

Design Guide: TIDA-010090

4 チャンネル、50A、デジタル制御バッテリーセル テスタのリファレンス デザイン



概要

このリファレンス デザインでは、C2000™ マイクロコントローラ (MCU) と高精度 ADC ADS8588S を使用して、双方向降圧コンバータ電力段の電流と電圧を正確に制御する方法を紹介します。このデザインは、C2000 MCU の高分解能パルス幅変調 (PWM) 生成ペリフェラルを活用して、 $\pm 10\text{mA}$ 未満の電流レギュレーション誤差と $\pm 1\text{mV}$ の電圧レギュレーション誤差を実現しています。

リソース

TIDA-010090	デザイン フォルダ
LMG3100R017, INA241	プロダクト フォルダ
TMS320F28P650DK, REF54	プロダクト フォルダ
ADS8588S, TPSI3050, TVS0500	プロダクト フォルダ
TPS7A20, TLV1117, LM2664, TPS736	プロダクト フォルダ
TMDSCNCD28P65X	ツール フォルダ
C2000WARE-DIGITALPOWER-SDK	ツール フォルダ

特長

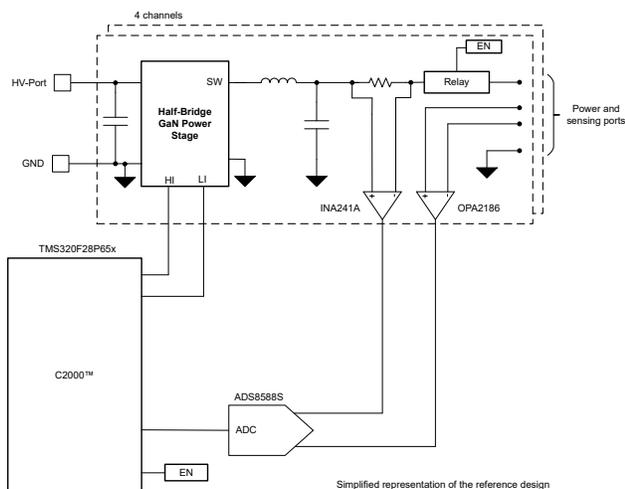
- 4 チャンネル、双方向降圧電力段
- 400kHz のスイッチング周波数
- 閉ループ制御向け外部 16 ビット SAR ADC
- レギュレーション誤差 $\pm 10\text{mA}$ 未満の定電流充放電
- レギュレーション誤差は $\pm 1\text{mV}$ 未満で、充電と放電の両方で定電圧モードに対応
- ソフトウェア周波数応答アナライザ (SFRA) および補償デザイナーにより、制御ループを簡単に調整可能
- ユーザー要件に応じた設計の適用を容易にする powerSUITE をサポート

アプリケーション

- バッテリーセル形成とテスト機器
- プログラマブル DC 電源



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。



1 システムの説明

バッテリーテスト装置には、シングルセル、バッテリーモジュール、および高電圧バッテリー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。この試験装置は、高精度の電源とデータ収集システムを搭載しており、バッテリーの充電と放電に使用され、セルのさまざまなパラメータを測定します。

図 1-1 に、リチウムイオンバッテリーを簡略化した製造プロセスを示します。最終段である、エンドオブラインコンディショニングには、セル形成とテストが含まれます。形成はリチウムイオン電池の製造において重要なステップです。形成中、セルは最初の充電と放電のプロセスを経て、固体電解質界面 (SEI) 層が形成されます。SEI 層の品質は、バッテリーセルの容量と信頼性に影響を及ぼします。形成プロセスを制御するために、セルの充電と放電に高精度のプログラマブル電源が使用されます。これらの電源は、バッテリー形成システムまたはバッテリーテストと呼ばれます。バッテリーテストに必要な電圧および電流の精度は、通常、フルスケールの $\pm 0.02\%$ ~ $\pm 0.05\%$ の範囲内です。

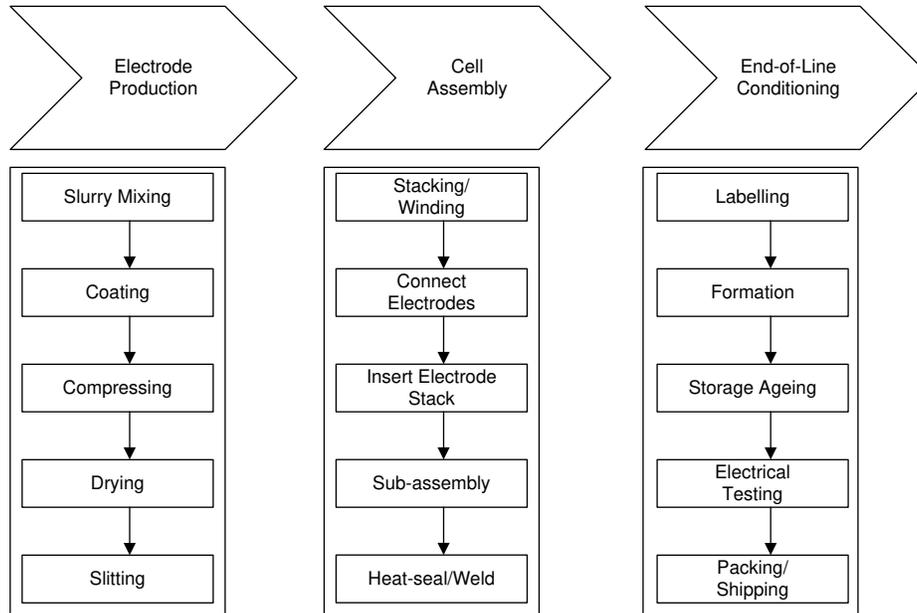


図 1-1. リチウムイオンバッテリーの製造プロセスの簡素化

1.1 主なシステム仕様

パラメータ	仕様
LV ポート - バッテリー ポート	50mV ~ 6V
HV ポート - バス電圧	12V ~ 15V
最大出力電流 (ヒートシンクなし)	$\pm 20A$
最大出力電流 (ヒートシンクあり)	$\pm 50A$
チャンネル数	4
スイッチング周波数	400 kHz
制御ループのサンプルレート	50kSPS
電流レギュレーション誤差	$\pm 10mA$ (0.02% FS) 未満
電圧レギュレーション誤差	$\pm 1mV$ (0.02% FS) 未満

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。INA241 電流センス アンプがバッテリー電流を検出し、OPA2186 オペアンプがバッテリー電圧を検出します。電流および電圧信号は、外部の ADS8588S 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。

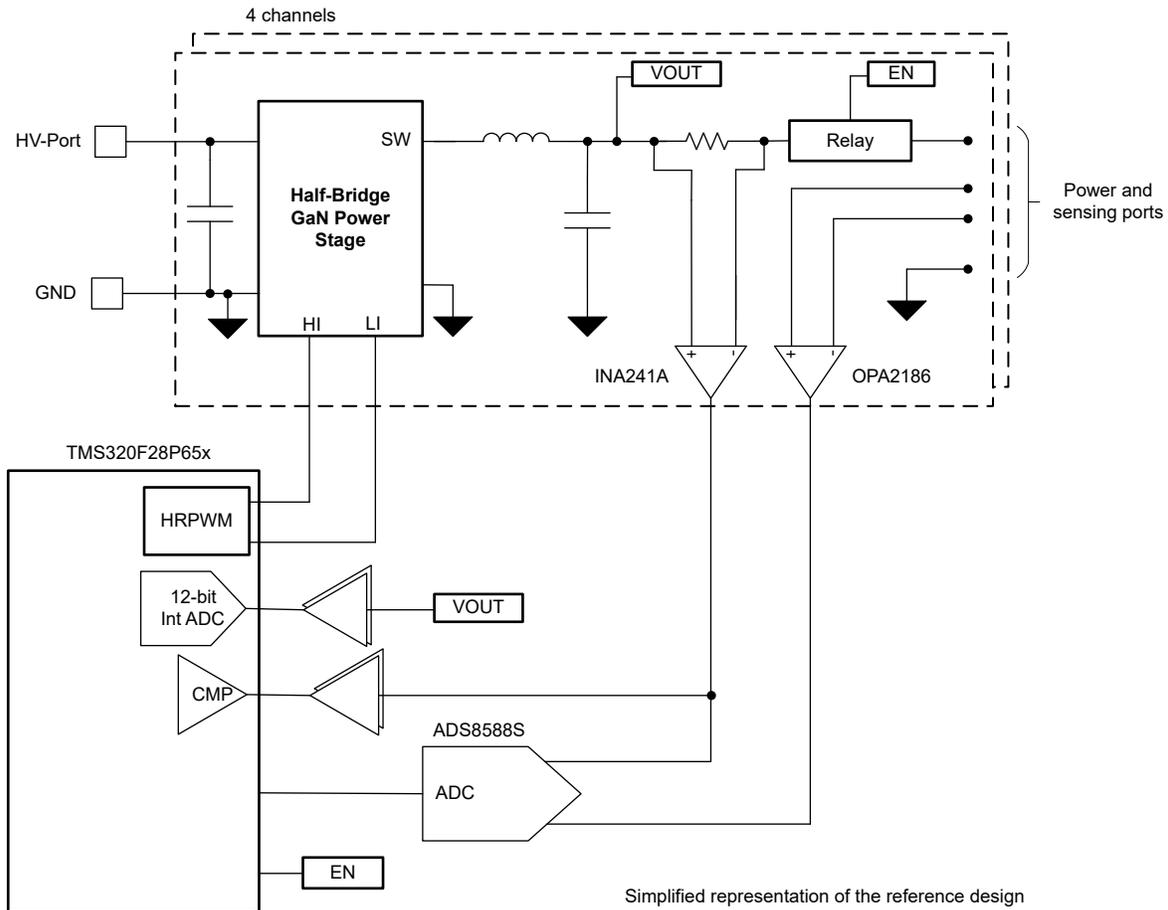


図 2-1. TIDA-010090 のブロック図

2.2 設計上の考慮事項

2.2.1 電流および電圧コントローラ

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。バッテリー電圧が定電圧設定 (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。バッテリー電圧が 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、バッテリー電圧が 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ー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ー電圧が制限されます。

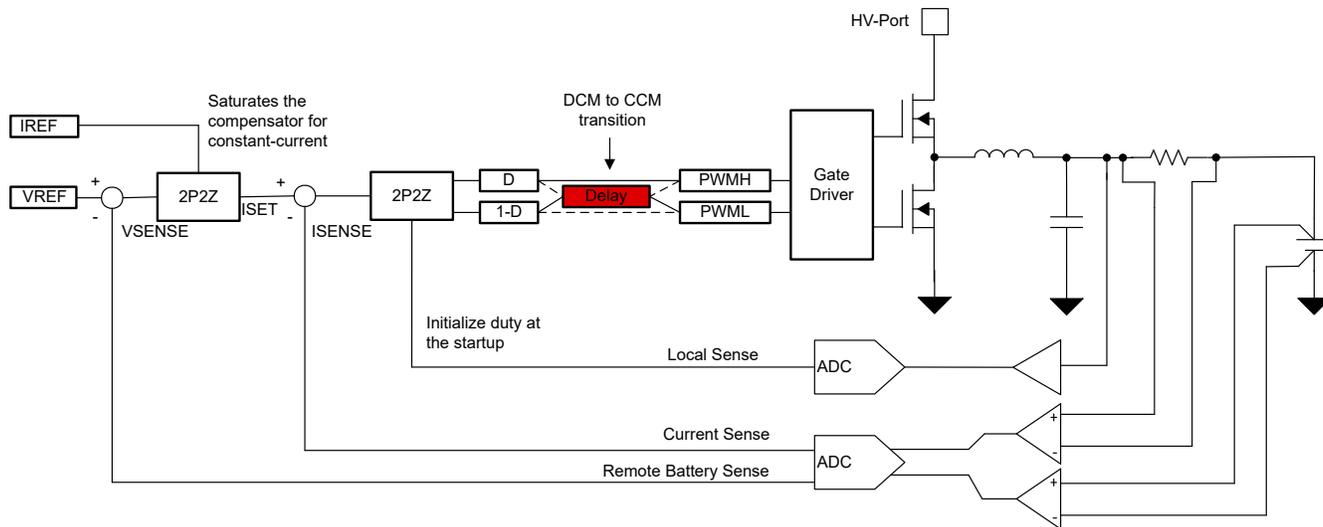


図 2-4. 非同期スタートアップ

2.2.3 高分解能 PWM 生成

高分解能を生成するには、高分解能 PWM 出力機能を持つ C2000 を使用しています。高分解能カウンタは 150ps のタイム ステップを実行でき、これは 100MHz の CPU クロックの 192kHz PWM 周波数における 15 ビットの分解能に相当します。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

2.3 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 のバッテリーテストチャンネルまたは降圧コンバータを制御するのに十分な 36 の HRPWM チャンネルが搭載されています。[『TMS320F28P65x リアルタイム マイクロコントローラ』](#)データシートも参照してください。

2.3.2 ADS8588S

ADS8588S は 8 チャンネルの同時サンプリング、16 ビット、SAR、A/D コンバータ (ADC) で、 $\pm 0.01\%$ の精度と 1kHz のループ帯域幅に十分な最大 200kSPS の最大サンプル レートを使用できます。[『ADS8588S 16 ビット、高速、8 チャンネル同時サンプリング ADC、単一電源によるバイポーラ入力』](#)データシートも参照してください。

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-010090 ハードウェアを示します。TIDA-010090 基板を動作させるには [TSM320F28P65X controlCARD](#) 評価基板 が必要です。

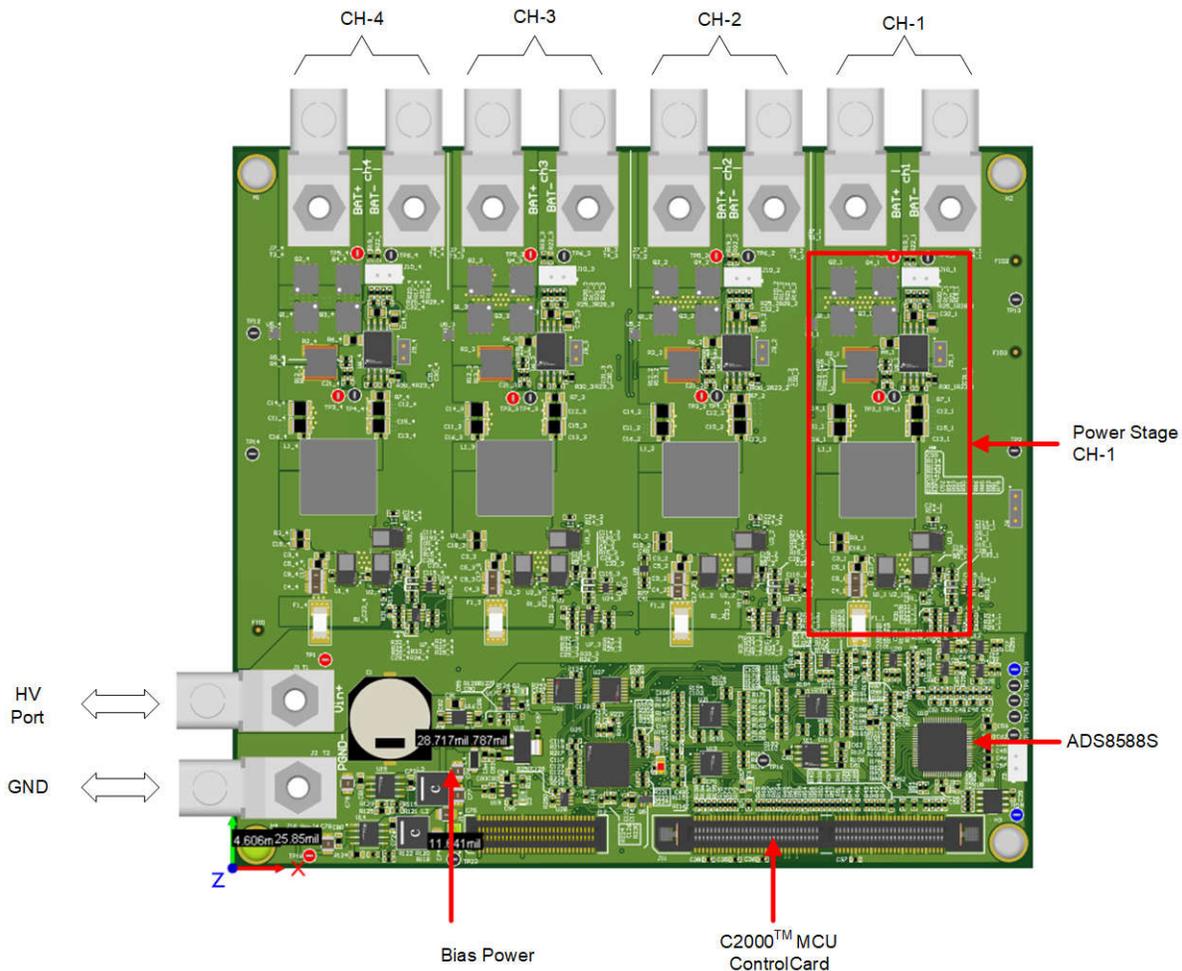


図 3-1. 基板の概要

3.2 ソフトウェア要件

この設計ソフトウェアは、C2000 マイコン向けの DigitalPower ソフトウェア開発キット (SDK) ([C2000WARE-DIGITALPOWER-SDK](#)) で供給され、powerSUITE フレームワーク内でサポートされています。

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-010090 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  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。
- ソリューション 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_010090/f28p65x/ccs にあるソリューション 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. powerSUITE のないプロジェクトは、powerSUITE GUI の制限を見つけたお客様、または量産コードから powerSUITE を削除したいお客様向けに提供されています。
 4. このドキュメントでは powerSUITE プロジェクトについて説明しますが、powerSUITE のないプロジェクトでも、この設計ガイドに記載されている powerSUITE settings.h ファイルの関連する #defines に変更を加えることで、すべての手順を繰り返すことができます。

The screenshot displays the powerSUITE (1) interface for project TIDA_010090. The top section shows a 'Simplified representation of the reference design' with a '4 channels' power stage diagram. This diagram includes a Half-Bridge GaN Power Stage, an INA241A current sensor, an OPA2188 operational amplifier, and an ADS8588S ADC. It also shows connections for HV-Port, GND, and a Relay. A 'Power and sensing ports' label points to the right. Below the diagram is a 'Power Stage Diagram' label with an arrow pointing to the right.

The bottom section shows the 'Build Options' panel. It lists various configuration parameters such as Lab, Channel Enabled, Feedback Loop, Comp Style, SFRA Enable, Calibration Mode, HW OCP, SW OCP, PWM Switching Frequency, Dead Time, Startup, DCM Startup Time, Default IREF, Default VREF Charge/Discharge, ISR Code Profiling, Control Loop ISR Frequency, Software Frequency Response Analy..., and Compensation Designer. The 'SFRA Enable' option is set to 'True' and is highlighted with an arrow pointing to the label 'SFRA Enable'. The 'Control Loop ISR Frequency' is set to 50 kHz, and there are buttons for 'RUN SFRA' and 'RUN COMPENSATION DESIGNER'. An arrow points from these buttons to the label 'Launch SFRA and Compensation Designer'.

図 3-2. 設計の powerSUITE ページ

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。

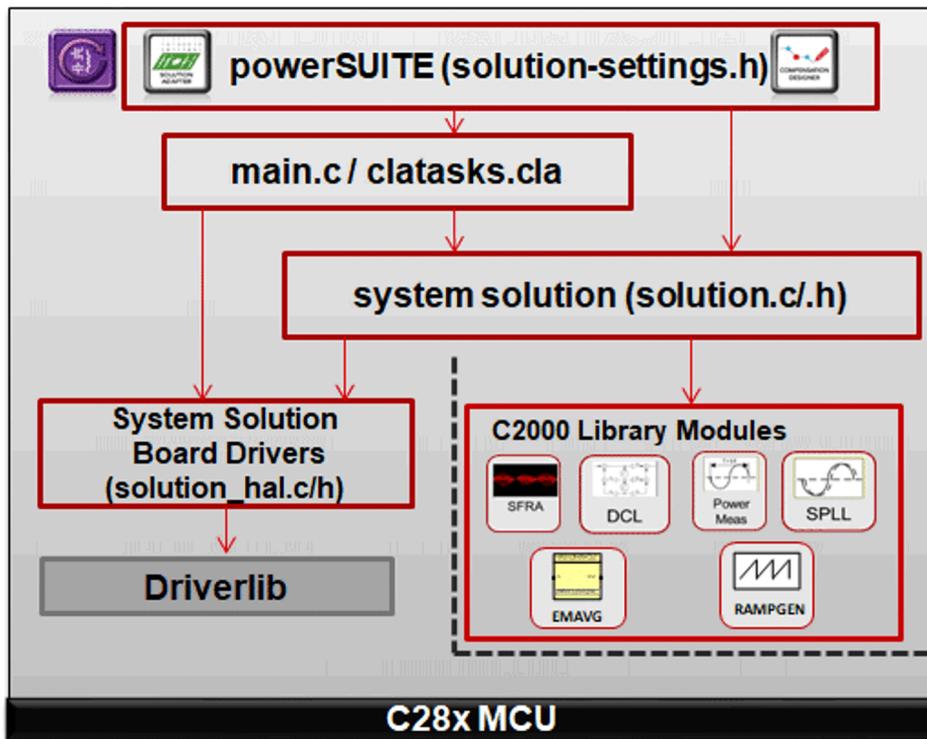


図 3-3. プロジェクト構造の概要

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

基板別でデバイス別のファイルは `_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_gan` 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して開くことができます。`powerSUITE` 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。

`_user_settings.h` は `_settings.h` ファイルに含まれており、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。

`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。

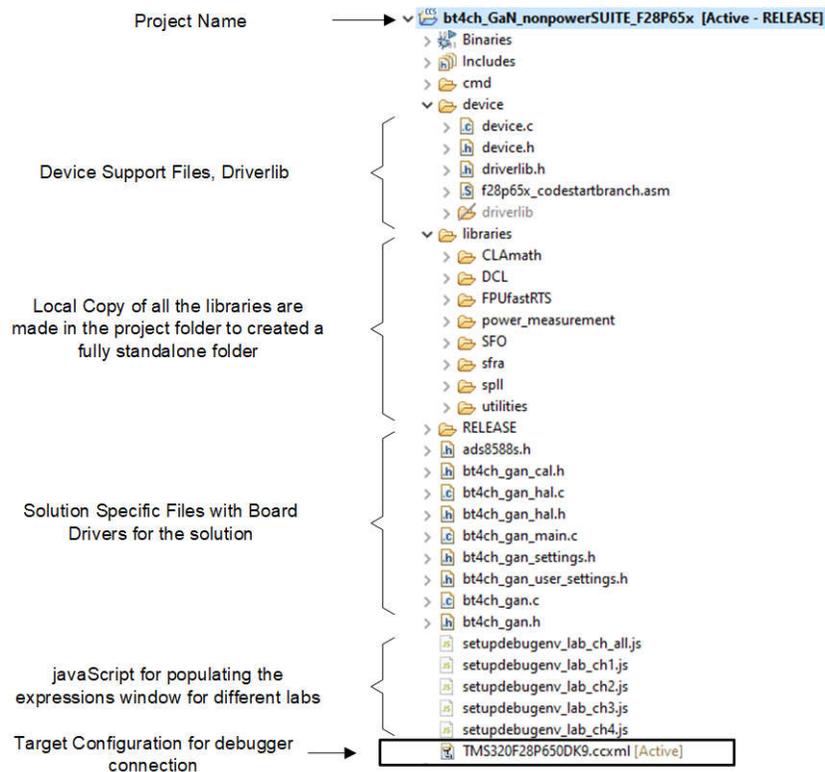


図 3-4. BT4PH プロジェクトの [Project Explorer] ビュー

bt4ch_gan 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、ADS8588S の BUSY 信号によってトリガされます。外部 ADC は 50kSPS のサンプルレートにプログラムされ、これによって ISR 周波数が設定されます。

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。

図 3-5 に、4 つのチャン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 6 μ s 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 3-6 および図 3-7 に、1 つのチャンネルのみがオンですべてのチャンネルがオフのときの ISR 時間を示します。

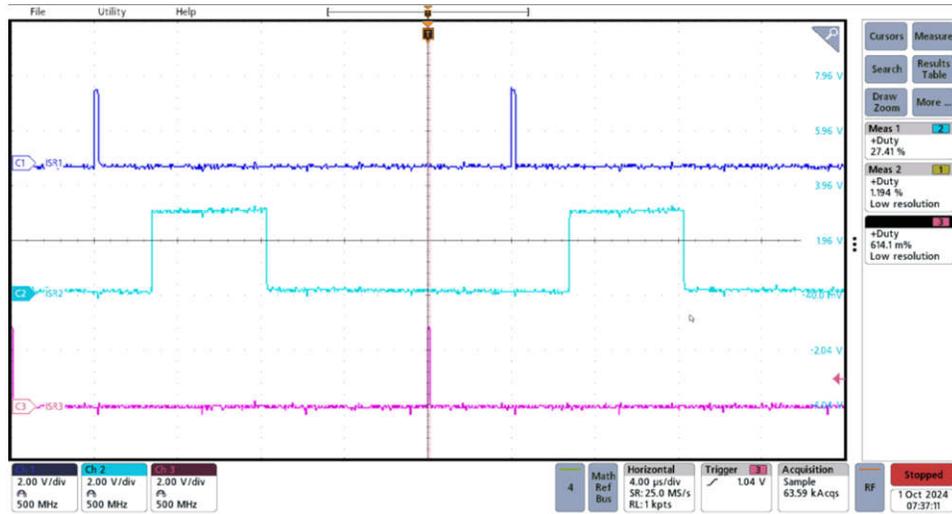


図 3-5. 4 チャンルの ISR 実行時間

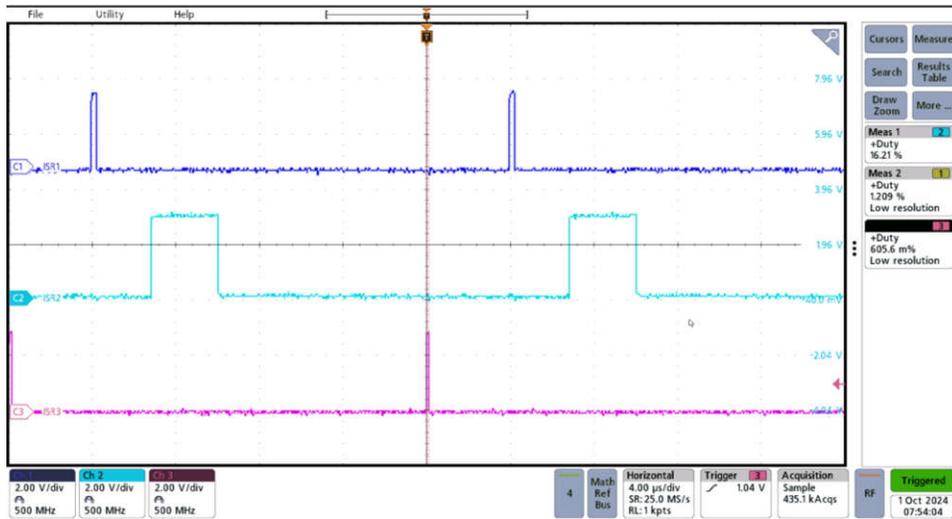


図 3-6. 1 チャンルの ISR 実行時間

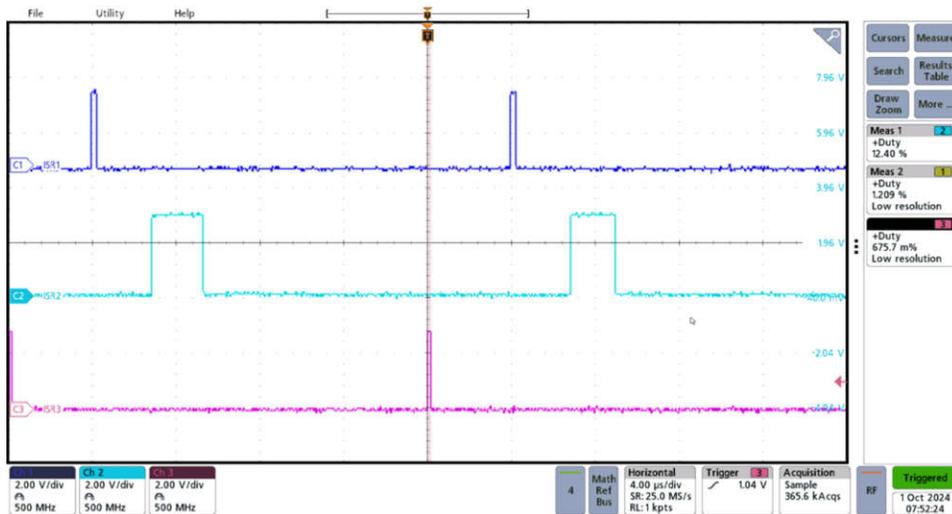


図 3-7. すべてのチャンネルがオフのときの ISR 実行時間

3.2.3 ソフトウェアフロー図

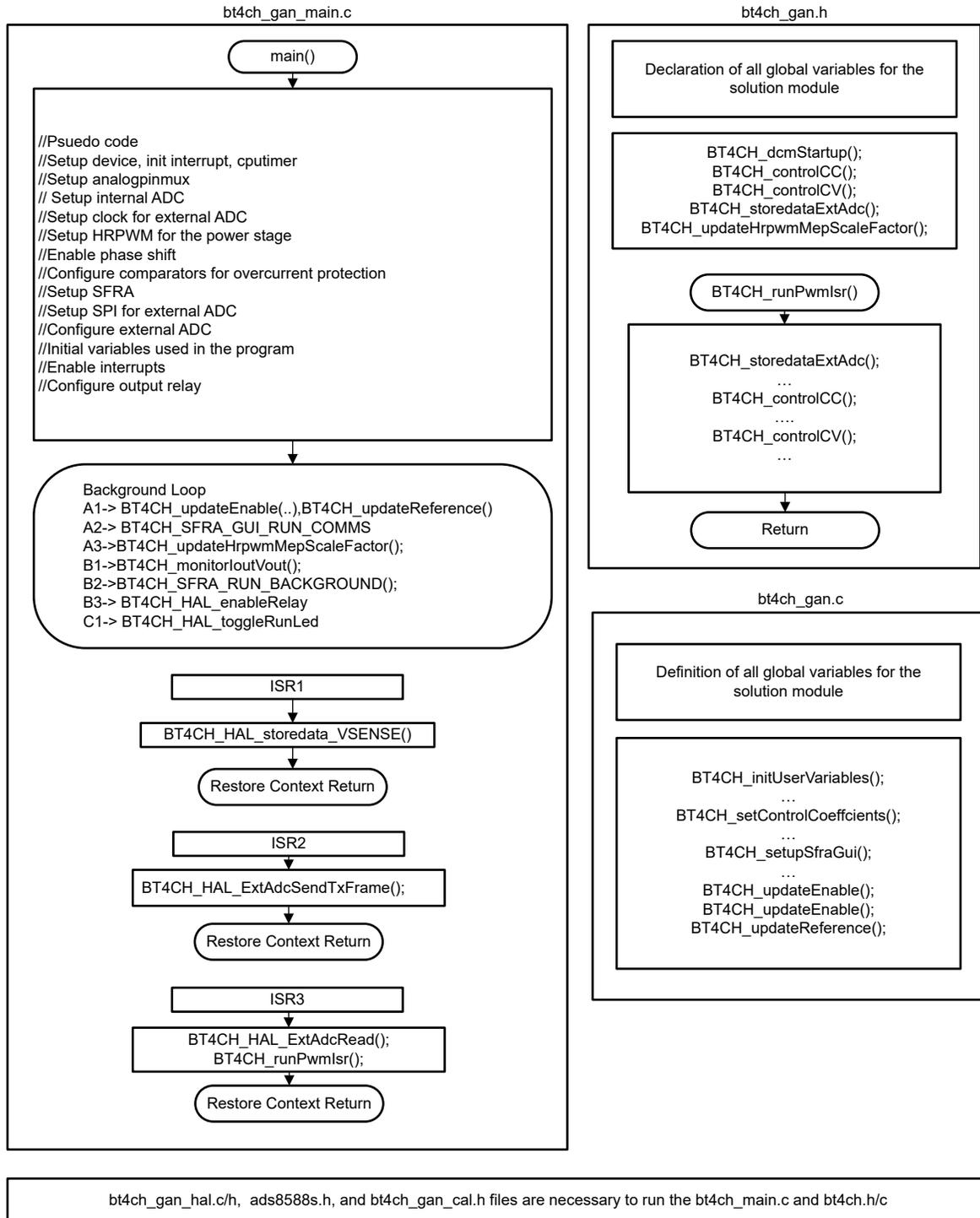


図 3-8. ソフトウェアフロー図

3.3 テスト構成

3.3.1 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

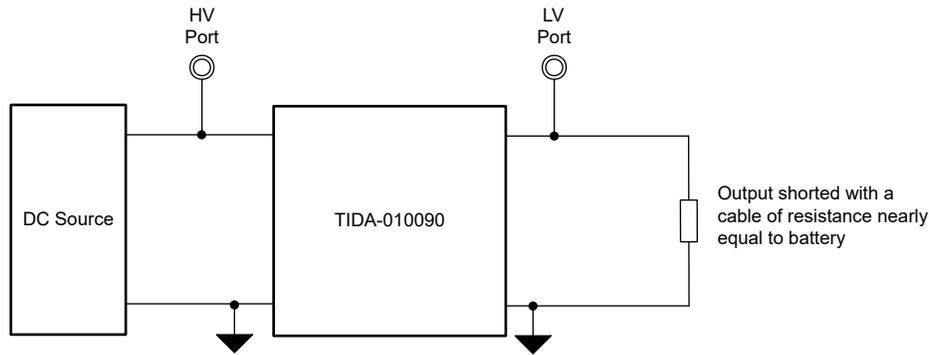


図 3-9. 電流および電圧ループをチューニングするためのハードウェア設定

3.3.2 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

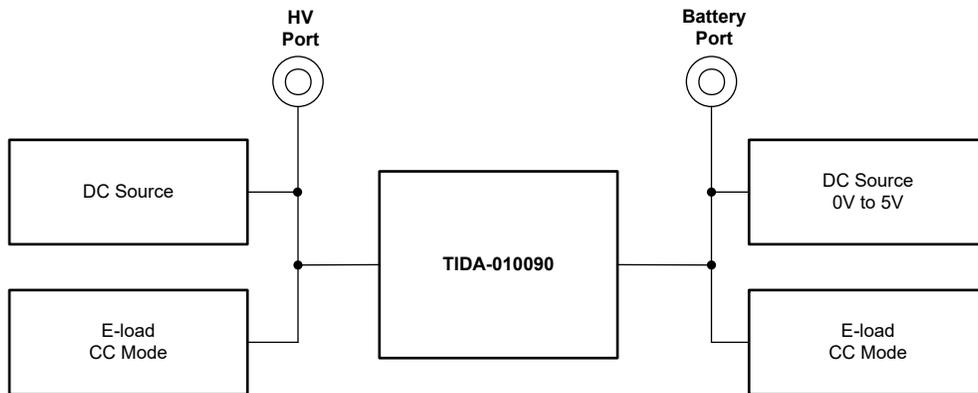


図 3-10. 双方向の電力フローをテストするためのハードウェア設定

3.3.3 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

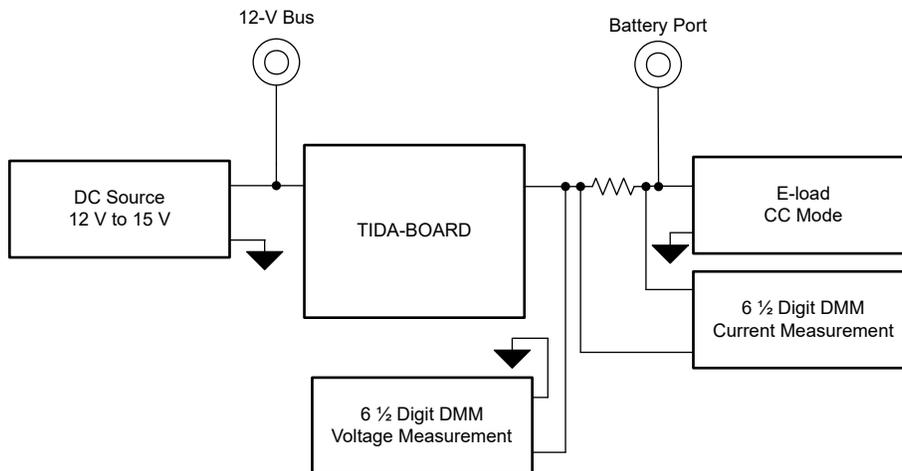


図 3-11. 電流および電圧キャリブレーションのハードウェア設定

3.4 テスト方法

3.4.1 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します。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, 100]
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
En_bool	符号なし整数	チャン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
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
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、バッテリー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、バッテリー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、バッテリー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、バッテリー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
VbatGain_pu	フローティング	変数には、バッテリー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、バッテリー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、バッテリー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、バッテリー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

3.4.2 ラボ 1. 開ループ電流制御単相

3.4.2.1 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 つのチャン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)
```

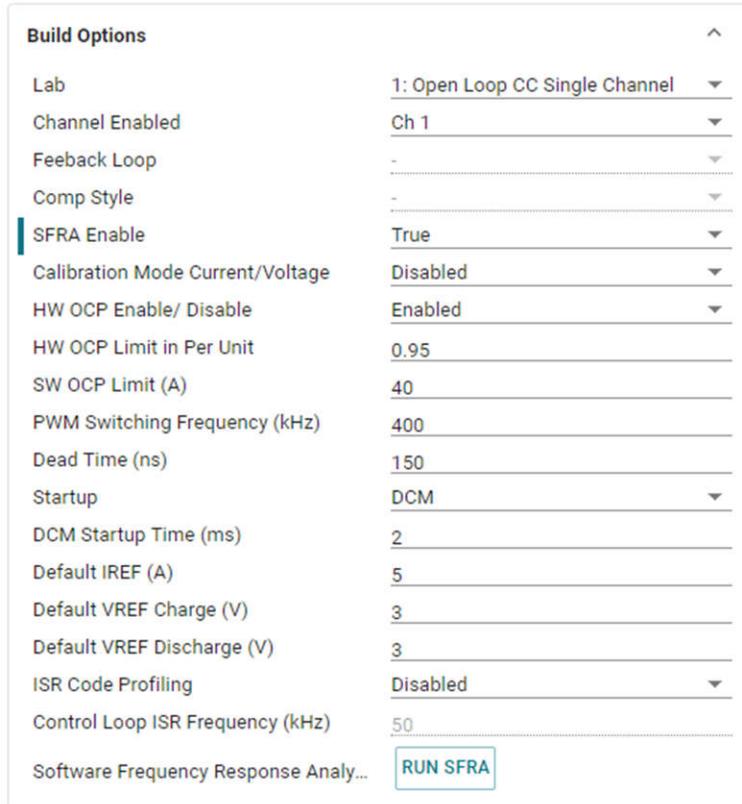


図 3-12. ラボ 1 のビルド オプション

3.4.2.2 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

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。

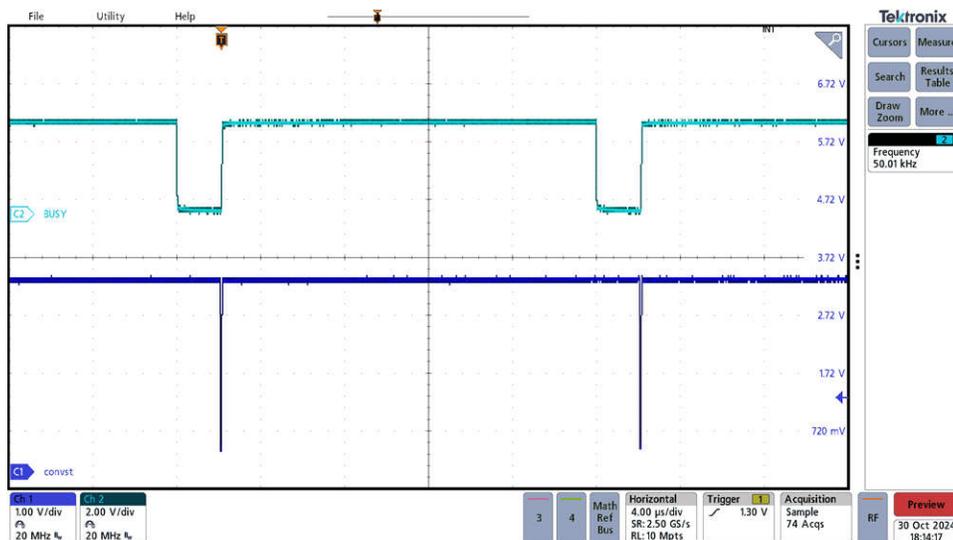
1. プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
2. プロジェクトが正常にビルドされます。
3. Project Explorer で、targetConfigs 下で適切な目標構成ファイルが有効になっていることを確認します。
4. **[Run]** → **[Debug]** をクリックしてデバッグ セッションを開始します。
5. するとプロジェクトがデバイスにロードされ、CCS デバッグ ビューが有効になります。メイン ルーチンの開始時にコードは停止します。
6. **[Watch]** および **[Expressions]** ウィ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

3.4.2.3 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. 図 3-9 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 BUSY 信号をチェックします。MCU が動作しているときの ADS8588S の BUSY 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 を 0.06 に設定します
 - BT4CH_userParam_chX->en_bool を 1 に設定します。
 - 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。
 - [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 15A 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。


図 3-13. ADS8588S の CONVST および BUSY 信号

Expression	Type	Value	Address
BT4CH_calibrationMode	enum <unnamed>	Calibration_CV	0x0000B75D@Data
BT4CH_startupMode	enum <unnamed>	DCM_Startup	0x0000B75E@Data
BT4CH_HAL_InputVoltageSense_V	float	14.9707031	0x0000B610@Data
BT4CH_ISR2_Loading	float	0.0	0x0000B5D8@Data
BT4CH_ISR2_LoadingMax	float	0.0	0x0000B5DA@Data
BT4CH_userParam_ch4	struct <unnamed>	{Iref_A=5.0,VrefCharge_V=4.0,VrefDischarge_V=2.0,dir_bool...	0x0000AA80@Data
Iref_A	float	5.0	0x0000AA80@Data
VrefCharge_V	float	4.0	0x0000AA82@Data
VrefDischarge_V	float	2.0	0x0000AA84@Data
dir_bool	unsigned int	1	0x0000AA86@Data
Relay_ON	unsigned int	1	0x0000AA87@Data
Start_bool	unsigned int	1	0x0000AA88@Data
remote_sense	unsigned int	0	0x0000AA89@Data
DutyRef_pu	float	0.0599999987	0x0000AA8A@Data
IbatCal_pu	float	0.0500000007	0x0000AA8C@Data
VbatCal_pu	float	0.200000003	0x0000AA8E@Data
IoutGain_pu	float	0.0200256333	0x0000AA90@Data
IoutOffset_pu	float	-0.000424541533	0x0000AA92@Data
IoutGain_A	float	49.935997	0x0000AA94@Data
IoutOffset_A	float	0.0211999062	0x0000AA96@Data
VoutGain_pu	float	0.165272892	0x0000AA98@Data
VoutOffset_pu	float	3.30209732e-05	0x0000AA9A@Data
VoutGain_V	float	6.0505991	0x0000AA9C@Data
VoutOffset_V	float	-0.000199796676	0x0000AA9E@Data
VbatGain_pu	float	0.165272892	0x0000AAA0@Data
VbatOffset_pu	float	3.30209732e-05	0x0000AAA2@Data
VbatGain_V	float	6.0505991	0x0000AAA4@Data
VbatOffset_V	float	-0.000199796676	0x0000AAA6@Data
BT4CH_ExtAdc1	struct <unnamed>	{channel0=4,channel1=65529,channel2=65523,channel3=7...	0x0000B614@Data
BT4CH_measureV1_ch4	struct <unnamed>	{Ibat_A=10.725647,Vout_V=0.284806907,Vbat_V=0.28292578...	0x0000B000@Data
Ibat_A	float	10.7257776	0x0000B000@Data
Vout_V	float	0.284806907	0x0000B002@Data
Vbat_V	float	0.283039749	0x0000B004@Data
Ibat_16b	int[128]	[7030,7023,7043,7028,7010...]	0x0000B006@Data
Vout_16b	int[128]	[195,192,192,193,193...]	0x0000B086@Data
Vbat_16b	int[128]	[1538,1512,1539,1517,1532...]	0x0000B106@Data
Index	unsigned int	66	0x0000B186@Data

図 3-14. ラボ 1 [Expression] ウィンドウ、開ループ

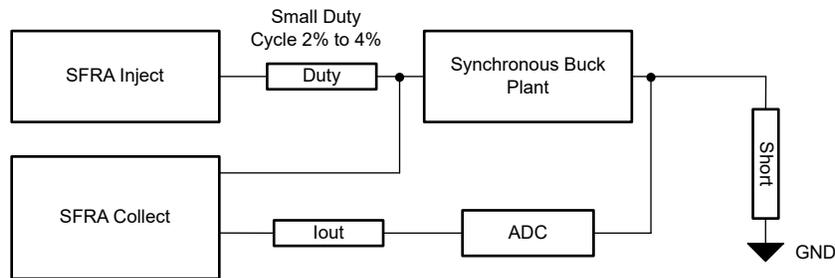
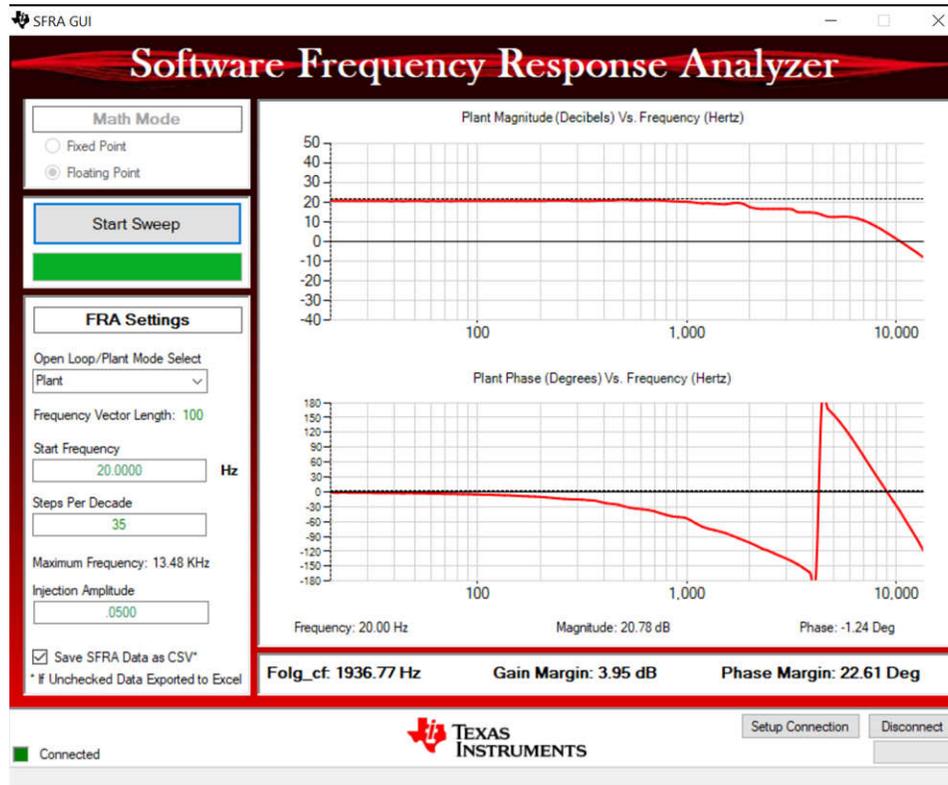


図 3-15. 開ループ電流制御用の SFRA 設定


図 3-16. 電流制御の開ループ周波数応答

3.4.3 ラボ 2. 閉ループ電流制御単相

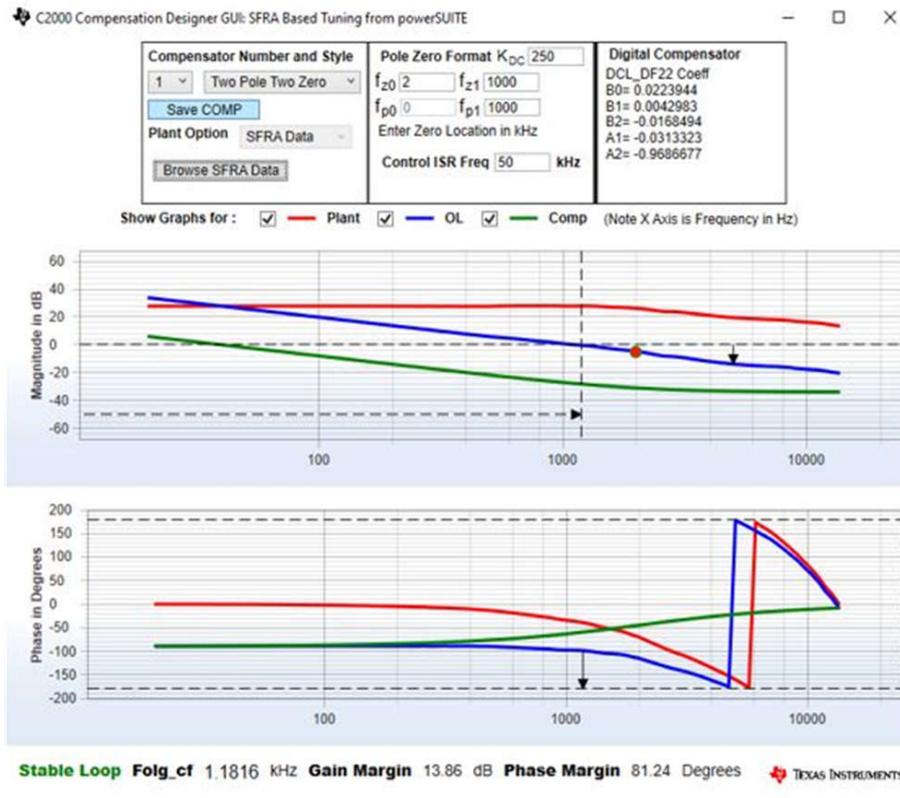
3.4.3.1 ラボ 2 のソフトウェア オプションの設定

- このラボを実行するには、前のセクション、[図 3-9](#) 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。
- [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。
- 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。
 - 補償デザイナーが起動し、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ーに 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。
 - 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。
- 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 に配置されています。

```
#define LAB_NUMBER (2)
#define CHANNEL_NUMBER (1)
#define SFRA_ENABLED (true)
```

Build Options		^
Lab	2: Closed Loop CC Single Chan...	▼
Channel Enabled	Ch 1	▼
Feedback Loop	CC_Loop	▼
Comp Style	DCL_DF22	▼
SFRA Enable	True	▼
Calibration Mode Current/Voltage	Disabled	▼
HW OCP Enable/ Disable	Enabled	▼
HW OCP Limit in Per Unit	0.95	
SW OCP Limit (A)	40	
PWM Switching Frequency (kHz)	400	
Dead Time (ns)	150	
Startup	DCM	▼
DCM Startup Time (ms)	2	
Default IREF (A)	5	
Default VREF Charge (V)	3	
Default VREF Discharge (V)	3	
ISR Code Profiling	Disabled	▼
Control Loop ISR Frequency (kHz)	50	
Software Frequency Response Analy...	<input type="button" value="RUN SFRA"/>	
Compensation Designer	<input type="button" value="RUN COMPENSATION DESIGNER"/>	

図 3-17. ラボ 2 のビルド オプション


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

3.4.3.2 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

- プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
- プロジェクトが正常にビルドされます。
- 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。
- [Run] → [Debug]** をクリックしてデバッグ セッションを開始します。
- 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。
- [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。
- [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。

3.4.3.3 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。

- このラボを実行するには、**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。
- 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。
- [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。
- [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->i_ref_A** を 15.0 に設定します。
 - BT4CH_userParam_chX->en_bool** を 1 に設定します。

- [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。
- 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 $\pm 1\text{mA}$ です。
 - 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。
 - 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。
 - 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。
 - 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。

Expression	Type	Value
BT4CH_lab	enum <unnamed>	Lab2_singleChannelClosedLoopCC
BT4CH_sfraStatus	enum <unnamed>	SFRA_Enabled
BT4CH_calibrationStatus	enum <unnamed>	Calibration_Disabled
BT4CH_calibrationMode	enum <unnamed>	Calibration_CV
BT4CH_startupMode	enum <unnamed>	DCM_Startup
BT4CH_HAL_InputVoltageSense_V	float	14.9578857
BT4CH_ISR2_Loading	float	0.0
BT4CH_ISR2_LoadingMax	float	0.0
BT4CH_userParam_ch3	struct <unnamed>	{Iref_A=15.0,VrefCharge_V=4.0,VrefDischarge_V=2.0,dir_boo...
Iref_A	float	15.0
VrefCharge_V	float	4.0
VrefDischarge_V	float	2.0
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
lbatCal_pu	float	0.0500000007
VbatCal_pu	float	0.200000003
loutGain_pu	float	0.01999999996
loutOffset_pu	float	3.7252903e-09
loutGain_A	float	50.0
loutOffset_A	float	-1.86264515e-07
VoutGain_pu	float	0.160000011
VoutOffset_pu	float	-1.49011612e-08
VoutGain_V	float	6.24999952
VoutOffset_V	float	9.31322504e-08
VbatGain_pu	float	0.164176673
VbatOffset_pu	float	3.28123569e-05
VbatGain_V	float	6.09099913
VbatOffset_V	float	-0.000199860049
BT4CH_ExtAdc1	struct <unnamed>	{channel0=65523,channel1=65533,channel2=9856,channel3...
BT4CH_measureVI_ch3	struct <unnamed>	{lbat_A=15.0009155,Vout_V=0.200462416,Vbat_V=0.1891389...
lbat_A	float	15.0013561
Vout_V	float	0.200581625
Vbat_V	float	0.189153463

図 3-19. ラボ 2 [Expression] ウィンドウ、閉ループ

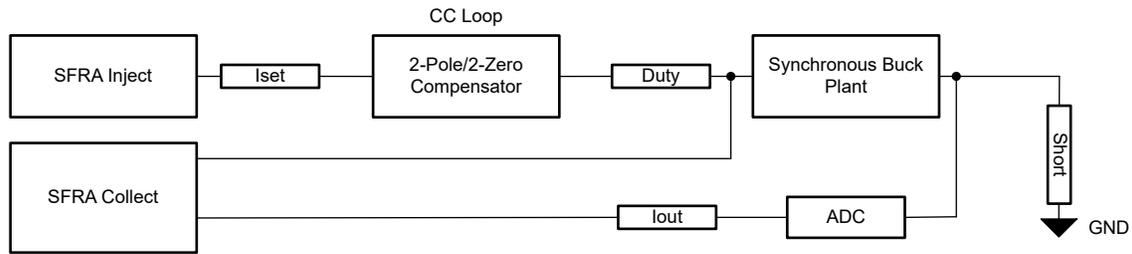


図 3-20. 閉ループ電流制御の SFRA 設定

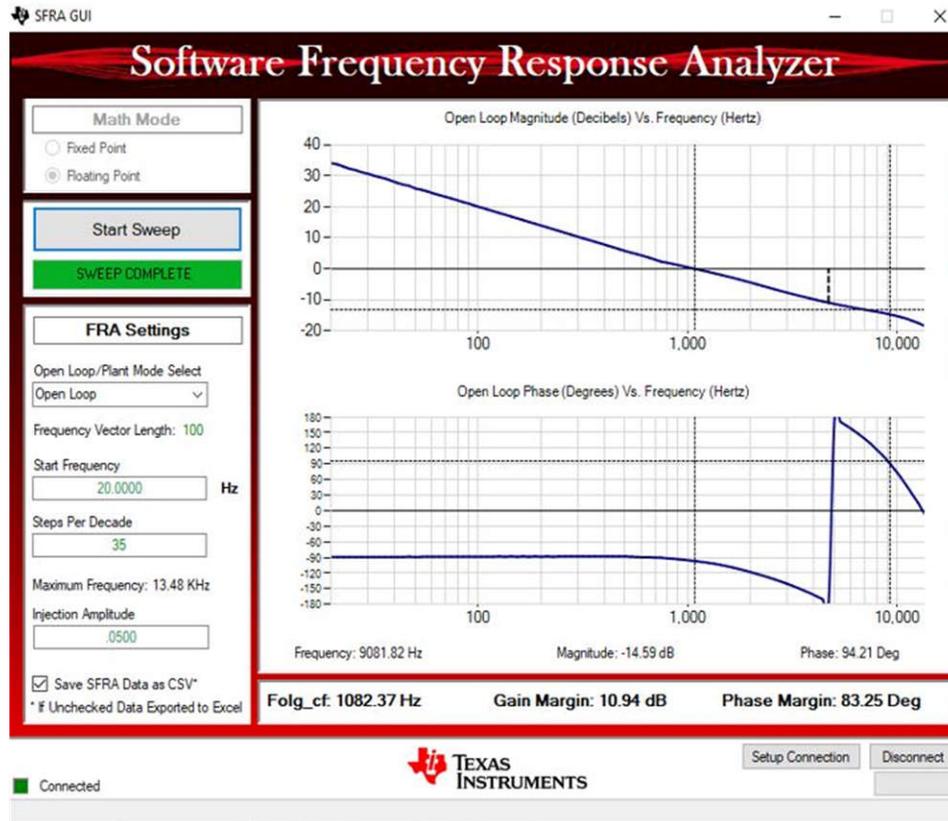


図 3-21. 電流制御の閉ループ周波数応答

3.4.4 ラボ 3. 開ループ電圧制御 1 チャンネル

3.4.4.1 ラボ 3 のソフトウェア オプションの設定

1. 図 3-9 に示すようにテスト設定を使用します。
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 3: Single Channel Open-Loop CV Control] 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。
 - ページを保存します。

4. powerSUITE 以外のバージョンのプロジェクトを使用する場合、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 (1)
#define SFRA_ENABLED (true)
```

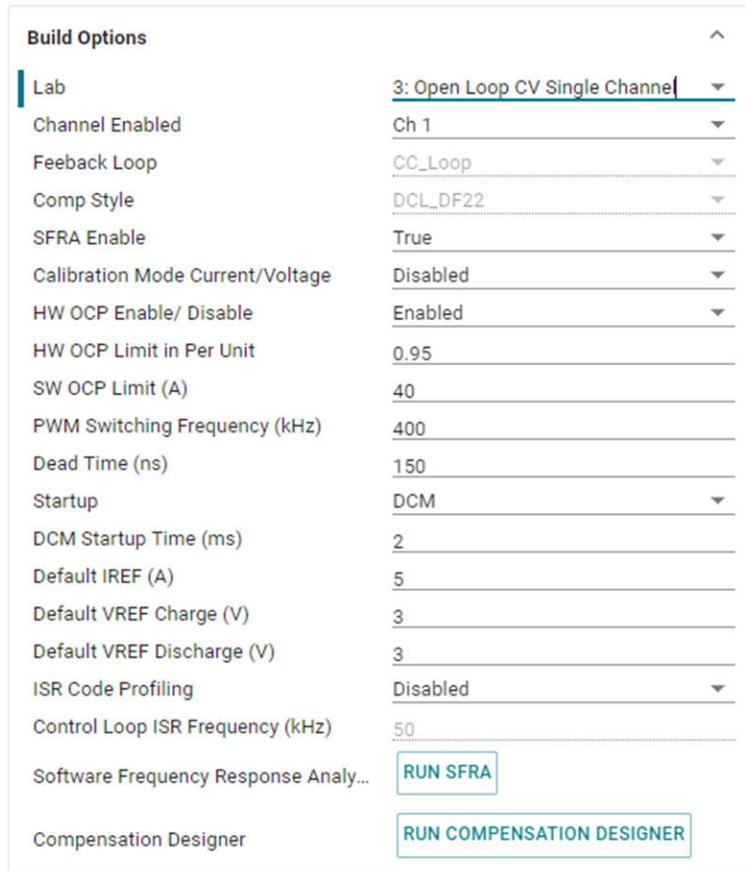


図 3-22. ラボ 3 のビルド オプション

3.4.4.2 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

- プロジェクト名を右クリックし、**[Rebuild Project]** をクリックします。
- プロジェクトが正常にビルドされます。
- 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。
- 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。
- 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。
- [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。
- [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。

3.4.4.3 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。

- 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 `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_V_I_chm->iref_A` を 15.0 に設定します。
 - `BT4CH_userParam_chX->en_bool` を 1 に設定します。
 - [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。`lbatsense_A` の表示値は `iref_A` 設定に近く、誤差は $\pm 1\text{mA}$ 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。
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。

Expression	Type	Value
(*) BT4CH_lab	enum <unnamed>	Lab3_singleChannelOpenLoopCV
(*) BT4CH_sfraStatus	enum <unnamed>	SFRA_Enabled
(*) BT4CH_calibrationStatus	enum <unnamed>	Calibration_Disabled
(*) BT4CH_calibrationMode	enum <unnamed>	Calibration_CV
(*) BT4CH_startupMode	enum <unnamed>	DCM_Startup
(*) BT4CH_HAL_InputVoltageSense_V	float	14.9578857
(*) BT4CH_ISR2_Loading	float	0.0
(*) BT4CH_ISR2_LoadingMax	float	0.0
BT4CH_userParam_ch3	struct <unnamed>	{Iref_A=15.0,VrefCharge_V=4.0,VrefDischarge_V=2.0,dir_boo...
(*) Iref_A	float	15.0
(*) VrefCharge_V	float	4.0
(*) VrefDischarge_V	float	2.0
(*) 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	0.0500000007
(*) VbatCal_pu	float	0.200000003
(*) IoutGain_pu	float	0.01999999996
(*) IoutOffset_pu	float	3.7252903e-09
(*) IoutGain_A	float	50.0
(*) IoutOffset_A	float	-1.86264515e-07
(*) VoutGain_pu	float	0.160000011
(*) VoutOffset_pu	float	-1.49011612e-08
(*) VoutGain_V	float	6.24999952
(*) VoutOffset_V	float	9.31322504e-08
(*) VbatGain_pu	float	0.164176673
(*) VbatOffset_pu	float	3.28123569e-05
(*) VbatGain_V	float	6.09099913
(*) VbatOffset_V	float	-0.000199860049
BT4CH_ExtAdc1	struct <unnamed>	{channel0=65528,channel1=65534,channel2=9830,channel3...
BT4CH_measureVl_ch3	struct <unnamed>	{Ibat_A=14.9994259,Vout_V=0.205826834,Vbat_V=0.1944830...
(*) Ibat_A	float	15.0003071
(*) Vout_V	float	0.205862597
(*) Vbat_V	float	0.19449468

図 3-23. ラボ 3 [Expression] ウィンドウ、閉ループ

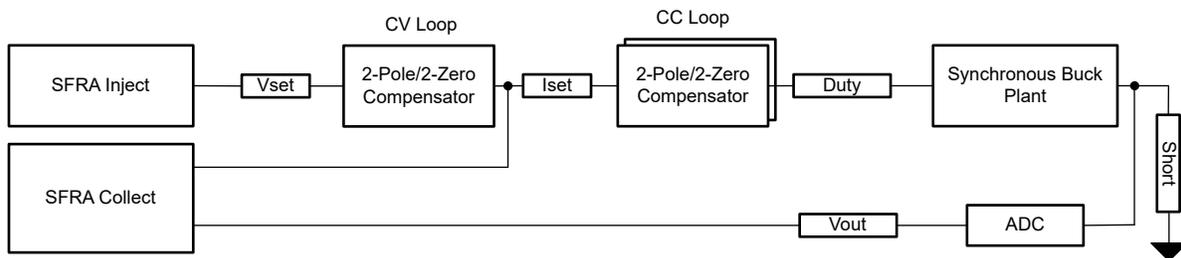


図 3-24. 開ループ電流制御の SFRA 設定

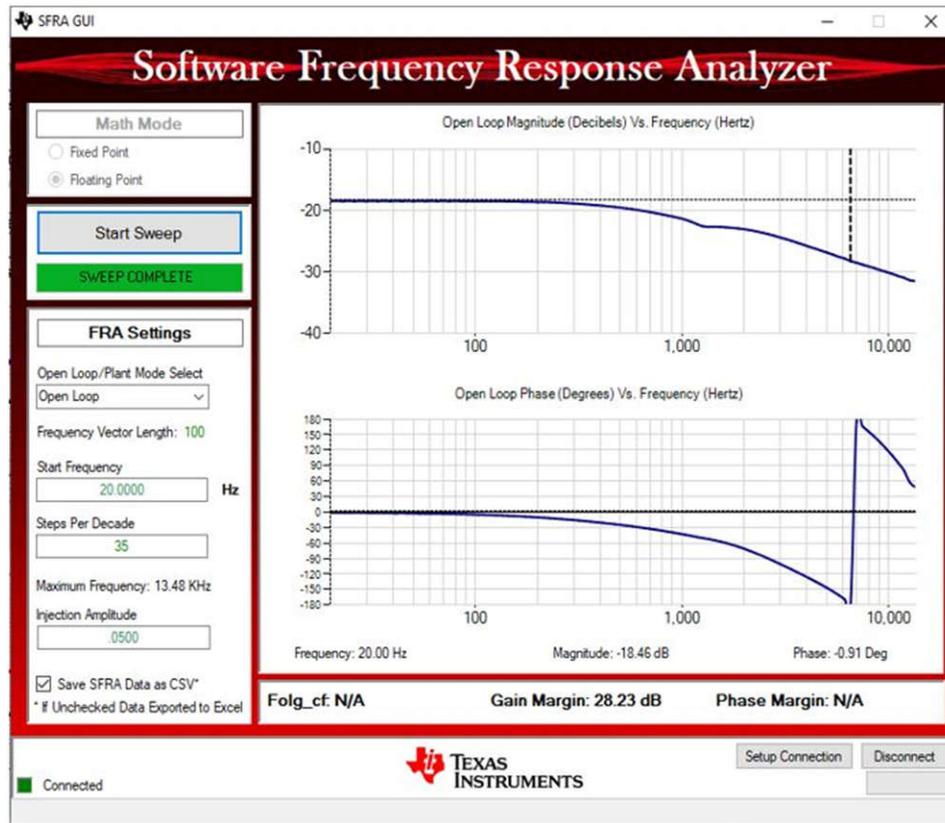


図 3-25. 電圧制御の開ループ周波数応答

3.4.5 ラボ 4. 閉ループ電流および電圧制御 1 チャンネル

3.4.5.1 ラボ 4 のソフトウェア オプションの設定

- 図 3-9 に示すようにテスト設定を使用します。
- 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。
- 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: Single Channel Closed-Loop CV] 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。
 - ページを保存します。
- 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 (4)
#define CHANNEL_NUMBER (1)
#define SFRA_ENABLED (true)
```

Build Options ^	
Lab	4: Closed Loop CCCV Single Chan ▾
Channel Enabled	Ch 1 ▾
Feedback Loop	CCCV_Loop ▾
Comp Style	DCL_DF22 ▾
SFRA Enable	True ▾
Calibration Mode Current/Voltage	Disabled ▾
HW OCP Enable/ Disable	Enabled ▾
HW OCP Limit in Per Unit	0.95
SW OCP Limit (A)	40
PWM Switching Frequency (kHz)	400
Dead Time (ns)	150
Startup	DCM ▾
DCM Startup Time (ms)	2
Default IREF (A)	5
Default VREF Charge (V)	3
Default VREF Discharge (V)	3
ISR Code Profiling	Disabled ▾
Control Loop ISR Frequency (kHz)	50
Software Frequency Response Analy...	<input type="button" value="RUN SFRA"/>
Compensation Designer	<input type="button" value="RUN COMPENSATION DESIGNER"/>

図 3-26. ラボ 4 のビルド オプション

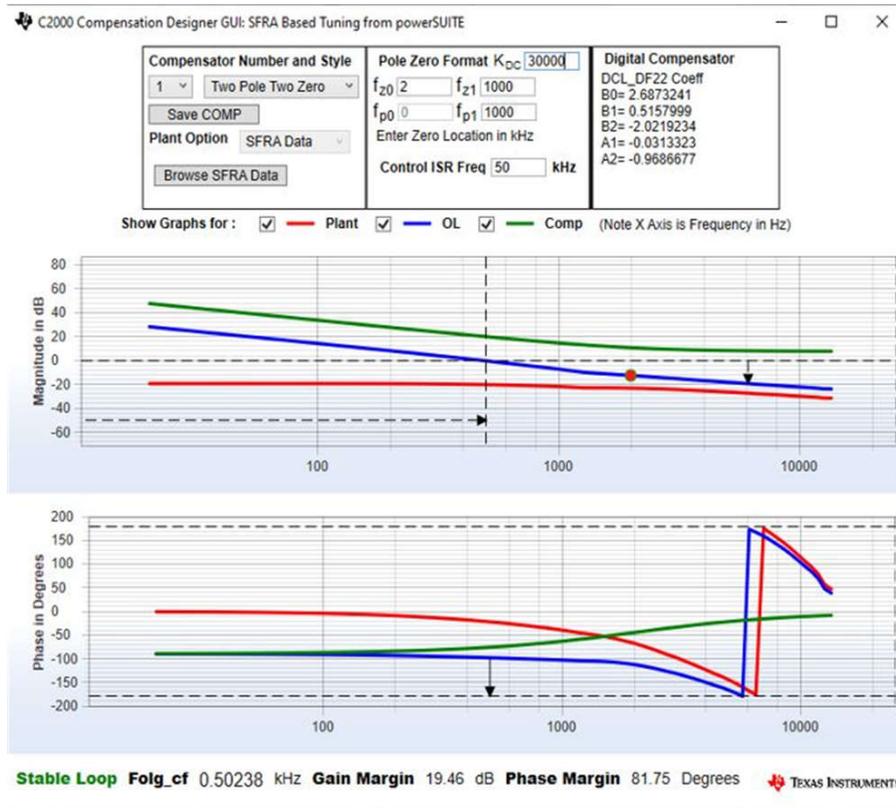


図 3-27. Compensation Designer を使用した電圧ループのチューニング

3.4.5.2 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

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。
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。

3.4.5.3 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 3.3.2 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 **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->i ref_A** を 25.0 に設定します。
 - **BT4CH_userParam_chX->vrefCharge_V** を 0.12 に設定します。

- BT4CH_userParam_chX->en_bool を 1 に設定します。
 - [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。
- 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 $\pm 0.5\text{mV}$ です。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。

Expression	Type	Value
(x)- BT4CH_lab	enum <unnamed>	Lab4_singleChannelClosedLoopCV
(x)- BT4CH_sfraStatus	enum <unnamed>	SFRA_Enabled
(x)- BT4CH_calibrationStatus	enum <unnamed>	Calibration_Disabled
(x)- BT4CH_calibrationMode	enum <unnamed>	Calibration_CV
(x)- BT4CH_startupMode	enum <unnamed>	DCM_Startup
(x)- BT4CH_HAL_InputVoltageSense_V	float	14.9578857
(x)- BT4CH_ISR2_Loading	float	0.0
(x)- BT4CH_ISR2_LoadingMax	float	0.0
BT4CH_userParam_ch3	struct <unnamed>	{lref_A=30.0,VrefCharge_V=0.189999998,VrefDischarge_V=2....
(x)- lref_A	float	30.0
(x)- VrefCharge_V	float	0.189999998
(x)- VrefDischarge_V	float	2.0
(x)- dir_bool	unsigned int	1
(x)- Relay_ON	unsigned int	1
(x)- Start_bool	unsigned int	1
(x)- remote_sense	unsigned int	1
(x)- DutyRef_pu	float	0.0
(x)- lbatCal_pu	float	0.0500000007
(x)- VbatCal_pu	float	0.200000003
(x)- loutGain_pu	float	0.0199999996
(x)- loutOffset_pu	float	3.7252903e-09
(x)- loutGain_A	float	50.0
(x)- loutOffset_A	float	-1.86264515e-07
(x)- VoutGain_pu	float	0.160000011
(x)- VoutOffset_pu	float	-1.49011612e-08
(x)- VoutGain_V	float	6.24999952
(x)- VoutOffset_V	float	9.31322504e-08
(x)- VbatGain_pu	float	0.164176673
(x)- VbatOffset_pu	float	3.28123569e-05
(x)- VbatGain_V	float	6.09099913
(x)- VbatOffset_V	float	-0.000199860049
BT4CH_ExtAdc1	struct <unnamed>	{channel0=0,channel1=65531,channel2=9644,channel3=65...
BT4CH_measureVI_ch3	struct <unnamed>	{lbat_A=14.7399902,Vout_V=0.201237276,Vbat_V=0.1900407...
(x)- lbat_A	float	14.7366285
(x)- Vout_V	float	0.201094225
(x)- Vbat_V	float	0.189928949

図 3-28. ラボ 4 [Expression] ウィンドウ、閉ループ

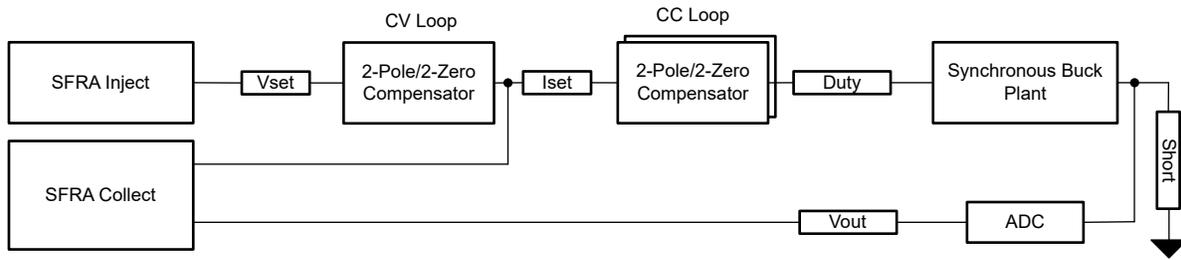


図 3-29. 閉ループ電流制御の SFRA 設定

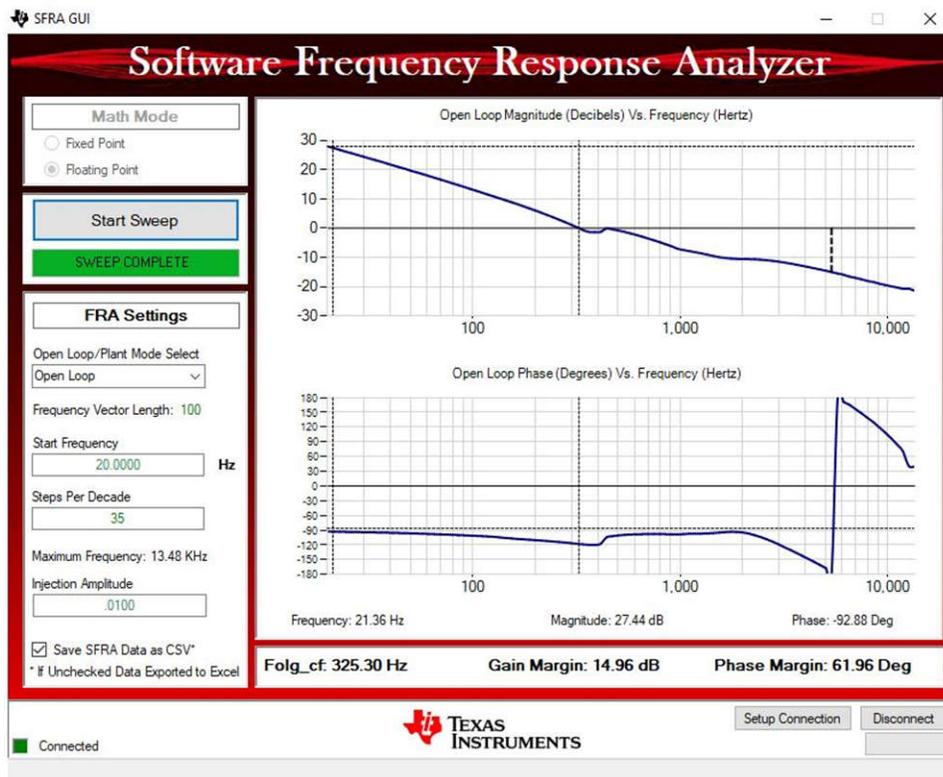


図 3-30. 電圧制御の閉ループ周波数応答

3.4.6 ラボ 5. 閉ループ電流および電圧制御 4 チャンネル

3.4.6.1 ラボ 5 のソフトウェア オプションの設定

- 図 3-11 に示すようにテスト設定を使用します。
- 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。
- 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。
 - すべてのチャンネルをイネーブルにします。
 - [SFRA Enable/Disable] を [1] に設定します。
 - ページを保存します。
- 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 (5) #define
CHANNEL_NUMBER (5) #define SFRA_ENABLED (true)
```

Build Options	
Lab	5: Closed Loop CCCV All Chann...
Channel Enabled	Ch 1
Feedback Loop	-
Comp Style	-
SFRA Enable	False
Calibration Mode Current/Voltage	Disabled
HW OCP Enable/ Disable	Enabled
HW OCP Limit in Per Unit	0.95
SW OCP Limit (A)	40
PWM Switching Frequency (kHz)	400
Dead Time (ns)	150
Startup	DCM
DCM Startup Time (ms)	2
Default IREF (A)	5
Default VREF Charge (V)	3
Default VREF Discharge (V)	3
ISR Code Profiling	Disabled
Control Loop ISR Frequency (kHz)	50

図 3-31. ラボ 5 のビルド オプション

3.4.6.2 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。プロジェクトが正常にビルドされます。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。
2. 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。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。
3. **[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_chAll.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。
4. **[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。

3.4.6.3 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 3.3.2 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 **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->i ref_A** を 5.0 に設定します。
 - **BT4CH_userParam_chX->vrefCharge_V** を 4.2 に設定します。
 - **BT4CH_userParam_chX->en_bool** を 1 に設定します。
 - **[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。
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. 電流現在の方向を変更するには、**BT4CH_userParam_chX -> Dir_bool** を切り替えます。

3.4.7 校正

- このラボを実行するには、[図 3-11](#) 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キャリブレーションには、2 点較正法を使用します。
- 電流を測定するには、外付けの高精度抵抗と DMM を使用するか、E-Load 電流測定値を使用します。または、TIDA-010090 基板上のセンス抵抗の両端の電圧を使用して、出力電流を測定できます。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 センス接続の間に DMM を使用します。
- 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 3-32](#) 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。
- BT4CH_userParam_chX->en_bool = 1 を設定します。
- BT4CH_userParam_chX->ibatCal_pu を「0.05」および「0.3」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)2.5)
#define BT4CH_IBAT_ACTUAL_CH1_P2_A ((float32_t)15.00)
```

- チャンネル 2、3、4 について、この手順を繰り返します。
- 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 3-33](#) 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)1.195)
#define BT4CH_VBAT_ACTUAL_CH1_P2_V ((float32_t)3.589)
```

- チャンネル 2、3、4 について、この手順を繰り返します。
- SYSCONFIG ページを開き、キャリブレーション モードをディセーブルにします。
 - 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 3-32. 電流キャリブレーションのビルド オプション

図 3-33. 電圧キャリブレーションのビルド オプション

3.5 テスト結果

3.5.1 電流ループ負荷レギュレーション

表 3-2. 充電モード負荷レギュレーション

FSR (A)	50							
出力モード	充電				放電中			
TIDA_010090 I _{SET} (A)	0.1	1	10	25	0.1	1	10	25
ELOAD CV モード	電流センス抵抗で測定される電流							
V _{SET} (V)	I _{actual} (A)							
1.0	0.101	1.000	9.999	24.998	0.099	1.000	9.996	24.994
2.0	0.101	1.000	9.9984	24.997	0.099	0.998	9.996	24.993
3.0	0.1	1.001	9.9984	24.9966	0.098	0.998	9.997	24.994
4.0	0.101	1.001	10.000	24.997	0.1	0.999	9.996	24.993
誤差 (mA)	1	1	1.6	3.4	2	2	4	7
誤差 (% FSR)	0.002	0.002	0.0032	0.0068	0.004	0.004	0.008	0.014

3.5.2 電流ループ直線性テスト

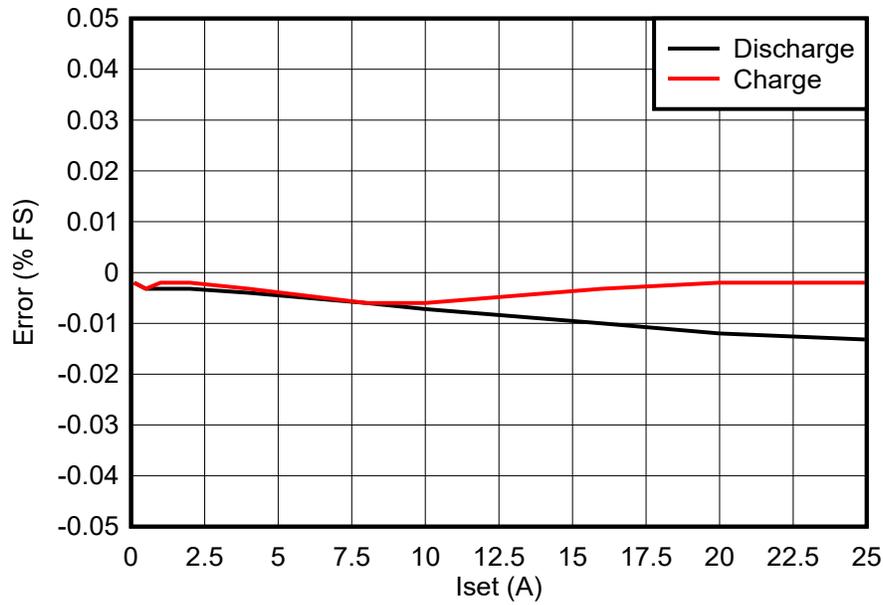


図 3-34. 電流ループ直線性テスト

3.5.3 電圧ループ直線性テスト

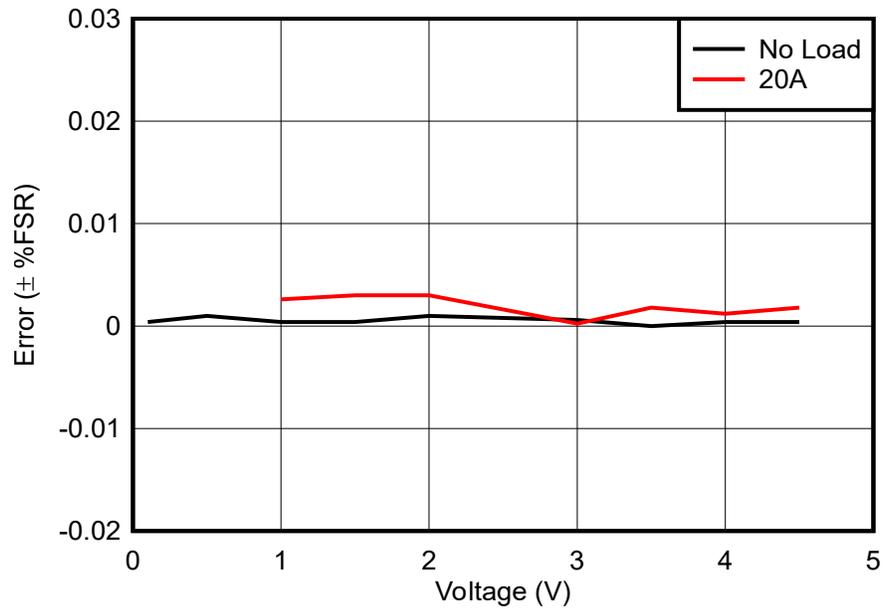


図 3-35. 電圧ループ直線性テスト

3.5.4 DCM のスタートアップ

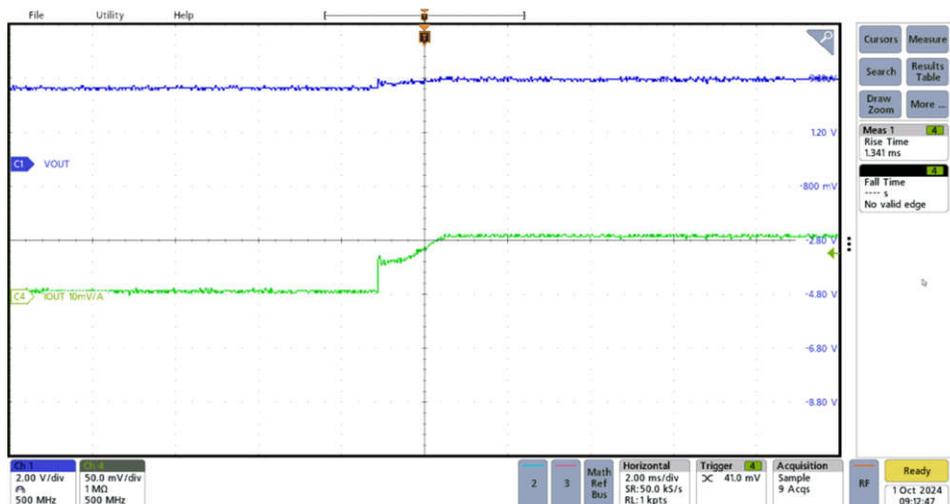


図 3-36. 5A スタートアップ時の過渡応答

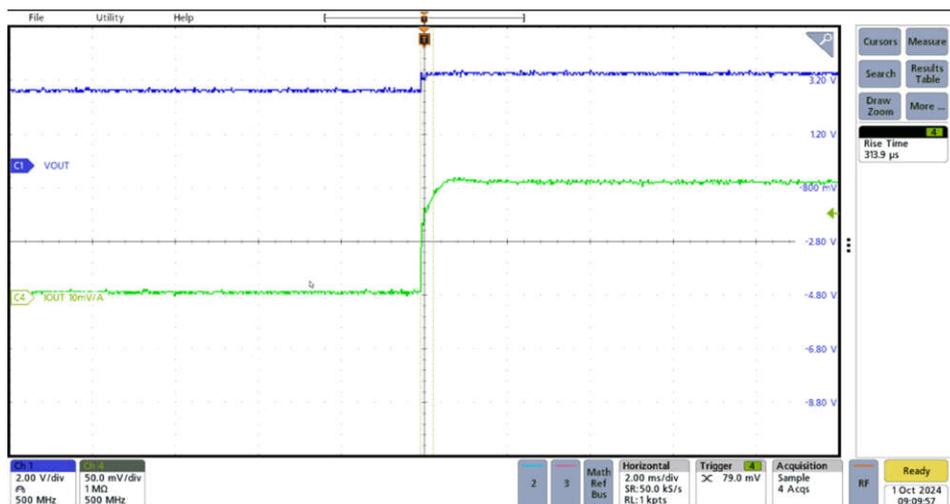


図 3-37. 10A スタートアップ時の過渡応答

3.5.5 双方向電流スイッチング時間

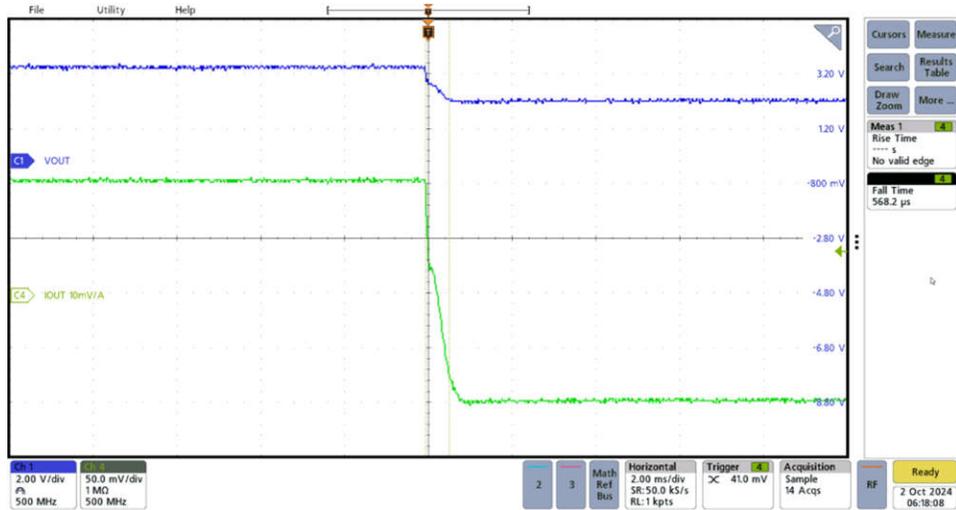


図 3-38. 充電から放電の過渡応答

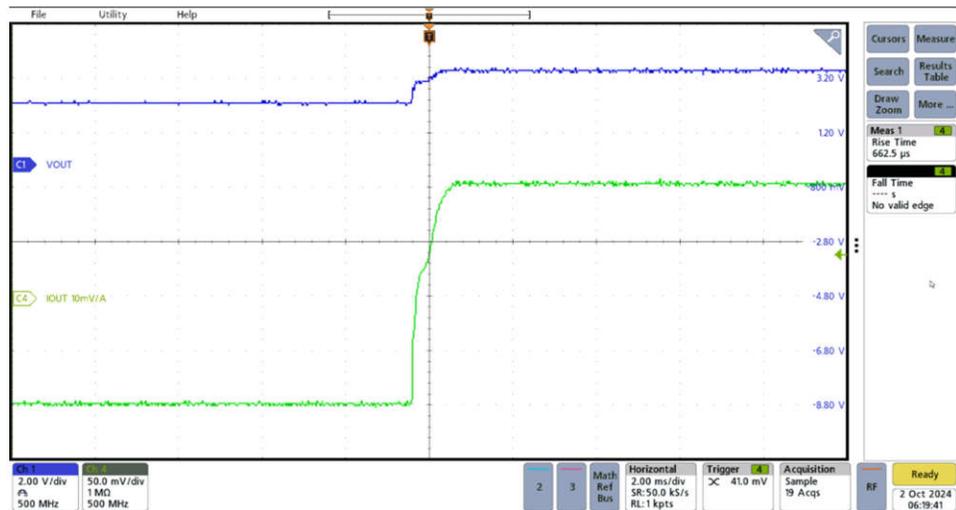


図 3-39. 放電から充電の過渡応答

3.5.6 熱性能

図 3-40 に、2 つのチャンネルが 20A をソースし、他の 2 チャンネルが 20A をシンクする場合のリファレンス デザインの熱画像を示します。この測定では、チャンネル 1 をチャンネル 2 と、チャンネル 3 をチャンネル 4 と短絡しています。

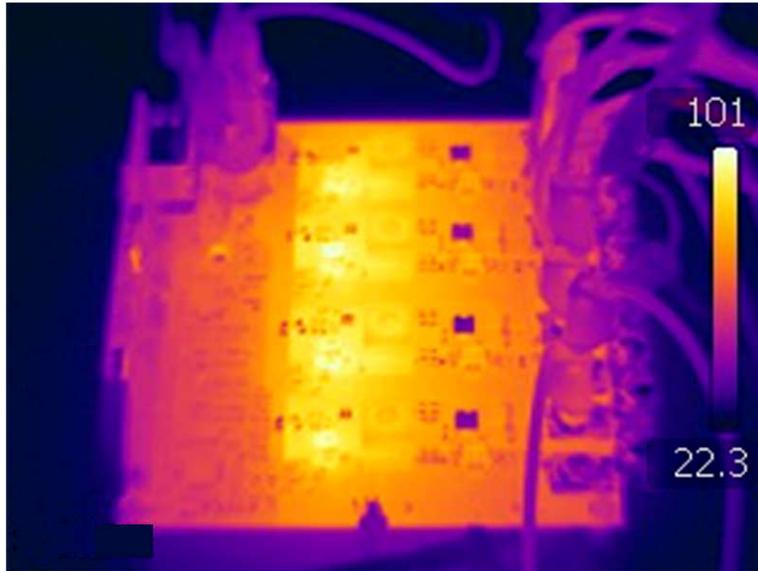


図 3-40. 熱画像

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-010090](#) の設計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-010090](#) 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](#)

TSM320F28P65x 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](#)』
2. テキサス・インスツルメンツ、『[ADS8588S 16 チャネル、高速、8 チャネル、同時サンプリング ADC、単一電源によるバイポーラ入力](#)』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、設計に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

C2000™, テキサス・インスツルメンツの™, 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

SHAURY ANAND 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。インド工科大学ローキー校で電気工学の学士号 (B.Tech) を取得しています。

このリファレンス デザインへのサポートについて、JINGQUAN ZHU、ETHAN YU、SEN WANG、OZINO ODHARO、TIM PRICE に感謝します。

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 © 2024, 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](#)、または [ti.com](#) やかかる TI 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。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。

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、TI 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 © 2024, Texas Instruments Incorporated