

## Application Note

# PMP41140 - 280W デジタル非対称ハーフブリッジ DC/DC コンバータソフトウェアガイド



Shashank Madineni, Xingqi Chen

## 概要

このドキュメントは、デジタル制御型非対称ハーフブリッジ (AHB) コンバータのソフトウェアについて説明します。このコンバータは、フライバック由来の共振トポロジであり、高効率かつ広範な出力電力変換を目的として設計されています。ファームウェアは、高速 PWM のオン時間計算と適応型ゼロ電圧検出 (ZVD) 技術を用いたゼロ電圧スイッチング (ZVS) 制御を実装しており、入力条件や負荷条件が変化しても境界 ZVS を維持します。また、プログラマブルな ZCD カウント制御、適応型ソフトスタート、およびインダクタンス補償アルゴリズムも含まれており、製造のばらつきがあっても安定した性能を維持します。

このソフトウェアは、280W AHB ハードウェア プロトタイプで検証済みであり、28V/9A 時に 97.8% のピーク効率を達成しています。このドキュメントでは、USB PD 3.1 アダプタ、産業用チャージャ、電動工具などの大電力アプリケーション (100W 超) を対象とした、AHB コンバータの堅牢なデジタル制御を可能にするソフトウェア構造、制御アルゴリズム、実装方法について説明します。

## 目次

1 概要.....	3
2 電力段の概要.....	4
3 ソフトウェアの概要.....	6
3.1 ソフトウェア アーキテクチャ.....	6
3.2 PowerSuite の使用方法.....	25
4 ラボの構造.....	29
4.1 ハードウェア設定.....	29
4.2 Lab1.....	32
4.3 Lab2.....	33
4.4 Lab3.....	35
4.5 Lab4.....	37
5 まとめ.....	40

## 図の一覧

図 1-1. 非対称ハーフブリッジ (AHB) DC/DC コンバータ.....	3
図 2-1. PMP41140 回路図の概要.....	4
図 2-2. C2000™ マイコンを使用した制御の実装.....	5
図 3-1. プロジェクト構造の概要.....	6
図 3-2. ASYSCTL ペリフェラルの初期化.....	8
図 3-3. ADC クロック初期化.....	9
図 3-4. ADC SOC の初期化.....	9
図 3-5. ADC を使用したアナログ信号マッピング.....	10
図 3-6. CMPSS を使用したアナログ信号マッピング.....	10
図 3-7. シングルエンド用の CMPSS の構成.....	11
図 3-8. 重負荷 PWM の波形.....	12
図 3-9. 軽負荷の PWM 波形.....	13
図 3-10. EPWM 時間ベースおよびカウンタ比較サブモジュール.....	15
図 3-11. アクション クオリファイア サブモジュール.....	16

☒ 3-12. デッドバンド サブモジュール.....	17
☒ 3-13. トリップゾーン サブモジュール.....	17
☒ 3-14. デジタル比較サブモジュール.....	18
☒ 3-15. イベントトリガおよび割り込みサブモジュール.....	18
☒ 3-16. AHB_CLB の構成.....	20
☒ 3-17. ZVS_CLB の構成.....	21
☒ 3-18. CPU タイマの構成.....	22
☒ 3-19. 割り込み構成.....	22
☒ 3-20. XBAR の構成.....	23
☒ 3-21. GPIO の構成.....	23
☒ 3-22. 割り込みフロー図.....	24
☒ 3-23. PMP41140 PowerSuite (アプリケーション UI).....	25
☒ 3-24. 電力段ハードウェア.....	26
☒ 3-25. 出力電圧と保護.....	27
☒ 3-26. 起動、スイッチングおよびタイミング.....	27
☒ 3-27. アンチワインドアップ付き PI コントローラ.....	28
☒ 4-1. AHB ハードウェア ボードのセットアップ.....	30
☒ 4-2. ハードウェア ボードのテスト ポイント.....	31
☒ 4-3. Lab1 の [Expressions] ウィンドウ.....	33
☒ 4-4. Lab1 の波形。Ch1 が HS PWM、Ch2 が LS PWM.....	33
☒ 4-5. Lab2 の式.....	35
☒ 4-6. Lab2 の波形。Ch1 が HS PWM、Ch2 が LS PWM、Ch3 が補助ノード、Ch4 が 1 次電流.....	35
☒ 4-7. Lab3 の式.....	37
☒ 4-8. Lab3 の波形。Ch1 が HS PWM、Ch2 が LS PWM、Ch3 が補助ノード、Ch4 が 1 次電流.....	37
☒ 4-9. 出力電圧調整 (9V->15V->20V->28V->20V->15V->9V、370V 入力、9A CC 負荷).....	39
☒ 4-10. ソフトスタートおよび低電圧保護動作、320V 入力、15V@9A CC 負荷条件.....	39

## 表の一覧

表 2-1. 電力段の仕様.....	4
表 3-1. サポートされるソース ファイルとヘッダー ファイル.....	7
表 3-2. 重負荷 PWM のイベントとアクション.....	12
表 3-3. 軽負荷 PWM のイベントとアクション.....	13
表 4-1. ランタイム変数.....	30

## 商標

すべての商標は、それぞれの所有者に帰属します。

## 1 概要

非対称型ハーフブリッジ (AHB) コンバータは、共振絶縁型 DC/DC トポロジであり、フライバック コンバータのシンプルさと共振スイッチング設計の性能を融合します。図 1-1 に示すように、AHB はハイサイド (HS) MOSFET とローサイド (LS) MOSFET が DC バス電圧  $V_{BUS}$  と接地の間に直列に接続され、ハーフブリッジ ノード  $V_{HB}$  を構成しています。共振インダクタ ( $L_r$ ) と共振コンデンサ ( $C_r$ ) はスイッチング ノードの電圧波形を形成し、スムーズな遷移とゼロ電圧スイッチング (ZVS) を実現できます。トランスは 1 次側と 2 次側の間で絶縁とエネルギー転送を行い、リーケージ インダクタンス ( $L_r$ ) と励磁インダクタンス ( $L_m$ ) は共振特性を定義します。2 次側では、同期整流 (SR) と出力コンデンサが、安定化された出力電圧  $V_o$  と電流  $I_o$  を供給します。

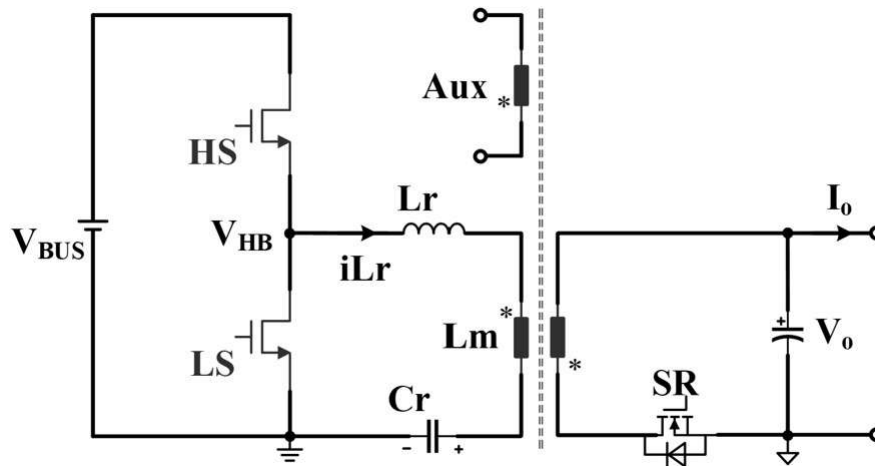


図 1-1. 非対称ハーフブリッジ (AHB) DC/DC コンバータ

インダクタ  $L_r$  と  $L_m$  は、トランスの等価モデルのリーケージ インダクタンスと励磁インダクタンスを表し、コンデンサ  $C_r$  は、共振電流と電圧を形成するために追加された外部共振コンデンサを示します。各スイッチング サイクルで、ローサイド MOSFET のオン期間中にトランスの励磁インダクタンスにエネルギーが蓄積され、オフ期間中は 2 次巻線を経由して出力に解放されます。 $L_r$  と  $C_r$  の間の共振相互作用により、ドレイン電圧がゼロに近づくときハイサイド MOSFET をオンに切り替えることができるため、ZVS を実現し、スイッチング損失を最小限に抑えることができます。AHB トポロジは小型サイズ、高電力密度、高効率を必要とする、中電力から大電力 (>100W) の絶縁型アプリケーションに最適です。従来型のフライバックや LLC コンバータとは異なり、幅広い出力電圧範囲に対応し、50% を超えるデューティサイクル動作を実現しています。これらの特長から、USB PD 3.1 アダプタ、産業用電動工具、オールインワン PC、スマートチャージャに最適です。

幅広い動作範囲で ZVS を維持することは、依然として重要な課題です。次のターンオン イベントの前にハーフブリッジ ノード  $V_{HB}$  が自動的に  $V_{BUS}$  まで共振すると、ハイサイド スイッチ (HS) の ZVS 条件が満たされます。この条件の実現は、負荷電流、トランスのパラメータ、スイッチング周波数に依存します。一貫性のある ZVS を確保するため、コントローラはシステム フィードバックに基づいて PWM オン時間とスイッチング周波数を動的に調整します。従来型のアナログ制御アプローチでは、柔軟性やパラメータの可視性は限られていますが、デジタル実装では精密なタイミング制御、プログラマビリティ、およびリアルタイムでの適応能力が実現されます。このドキュメントで説明するデジタル コントローラでは、ZVS 検出のための補助巻線フィードバックを使用して、適応型ソフトスタートおよびセーフストップ ルーチンを実装し、ZCD 数と遅延調整に関するプログラマブルなパラメータを使用することができます。これらの機能により、コンバータの性能が向上し、設計の反復を迅速化し、製造の堅牢性が向上します。このドキュメントでは、デジタル AHB コンバータ ソフトウェアの概要を説明し、ファームウェア アーキテクチャ、制御アルゴリズム、ステート マシン フロー、構成設定、診断機能について解説します。高効率の DC/DC 電源システム向けにデジタル AHB 制御ソリューションを実装またはカスタマイズする開発者の技術リファレンスとして役立つことを目的としています。

## 2 電力段の概要

非対称ハーフブリッジ (AHB) ハードウェア プロトタイプは、PMP41140 リファレンス デザインから派生したもので、C2000™ デジタル電源 SDK を使用して開発されています。このプラットフォームは、提案された高速 ZVS 計算ループと適応型ソフトスタート制御の検証環境として機能します。入出力定格、主要なパラメータを含め、このプロトタイプの電氣的仕様の全体を以下の表 2-1 にまとめます。

表 2-1. 電力段の仕様

パラメータ	仕様
入力電圧	120-420V
出力電圧	9V、15V、20V、28V
出力電流	15A、12A、9A
出力電力	150-250W
スイッチング周波数	20KHz~200KHz

図 2-1 のように、AHB コンバータは LMG2650 GaN ドライバを使用してハイサイド スイッチとローサイド スイッチを効率よく制御します。これにより、 $L_r$ 、 $C_r$ 、トランスの励磁インダクタンス ( $L_m$ ) で形成される共振タンクを通じて低損失のスイッチングを実現できます。補助トランス巻線は、ZVS 検出専用で使用されているため、負荷と入力さまざまな条件で、ゼロ電圧遷移を正確に検出できます。

### Asymmetric Half bridge DC/DC Converter

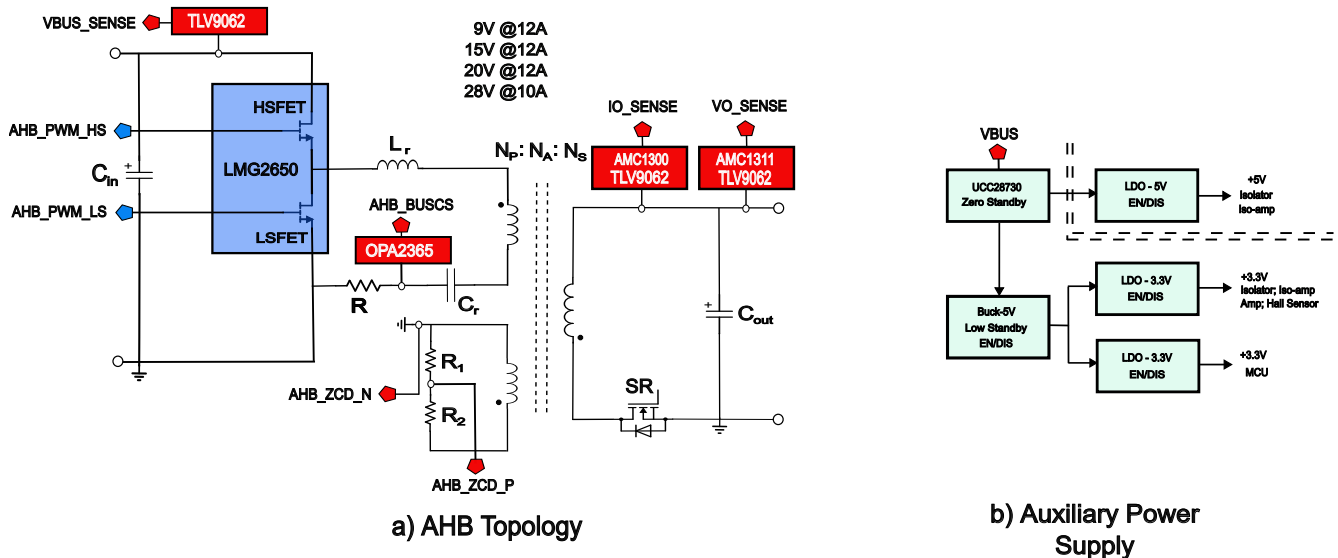


図 2-1. PMP41140 回路図の概要

#### シグナル コンディショニングおよび帰還回路

帰還回路とセンシング回路は、高精度アンプを使用して、シグナル インテグリティ、絶縁、高精度のリアルタイム測定を保証します。

- AMC1300/AMC1311 - 1 次側の電流および電圧検出用の絶縁型アンプで、優れた同相信号除去を実現します。
- TLV9062 - 電圧帰還のスケールリングおよびバッファリング用オペアンプ
- OPA2365 - 共振信号増幅と位相検出向けの高帯域幅アンプ

これらのアナログ フロントエンド コンポーネントは、制御の計算と診断のために、高精度でノイズのない入力を C2000™ ADC サブシステムに提供します。

## 補助電源

補助電源セクションは、UCC28730 およびローパワー降圧コンバータを使用して、安定化された +12V、+5V、+3.3V レールを生成します。これらのレールは、アナログ、デジタル、ゲートドライバの各セクションに対して安定したバイアスを供給し、適切なスタートアップシーケンシングとフォルト耐性を実現します。

## デジタル制御の実装

デジタル制御プラットフォームは、AHB コンバータの中央制御ユニットとして機能する C2000™ F28P55x マイコンをベースにしています。図 2-2 に、AHB ハードウェアと制御ロジック間の信号のやり取りを示します。

使用されるコア パリフェラルは次のとおりです。

- ADC - 電圧および電流帰還のリアルタイム サンプリング用 (VBUS、VO、共振電流)
- CMPSS - 高速保護応答および ZVS イベント検出用
- CLB - カスタム デジタル ロジックとタイミング クリティカルな機能の実装
- PWM – 高分解能ゲート駆動信号を高精度で生成する機能

これらのパシフェラルは緊密に同期された方法で動作し、適応型オン時間計算、ZVS 境界検出、スムーズなソフトスタートシーケンシングを実行します。C2000™ マイコンのフレキシブルなリアルタイム制御機能により、動作条件に基づいてスイッチング動作を動的に調整し、効率と安定性を最適化することができます。

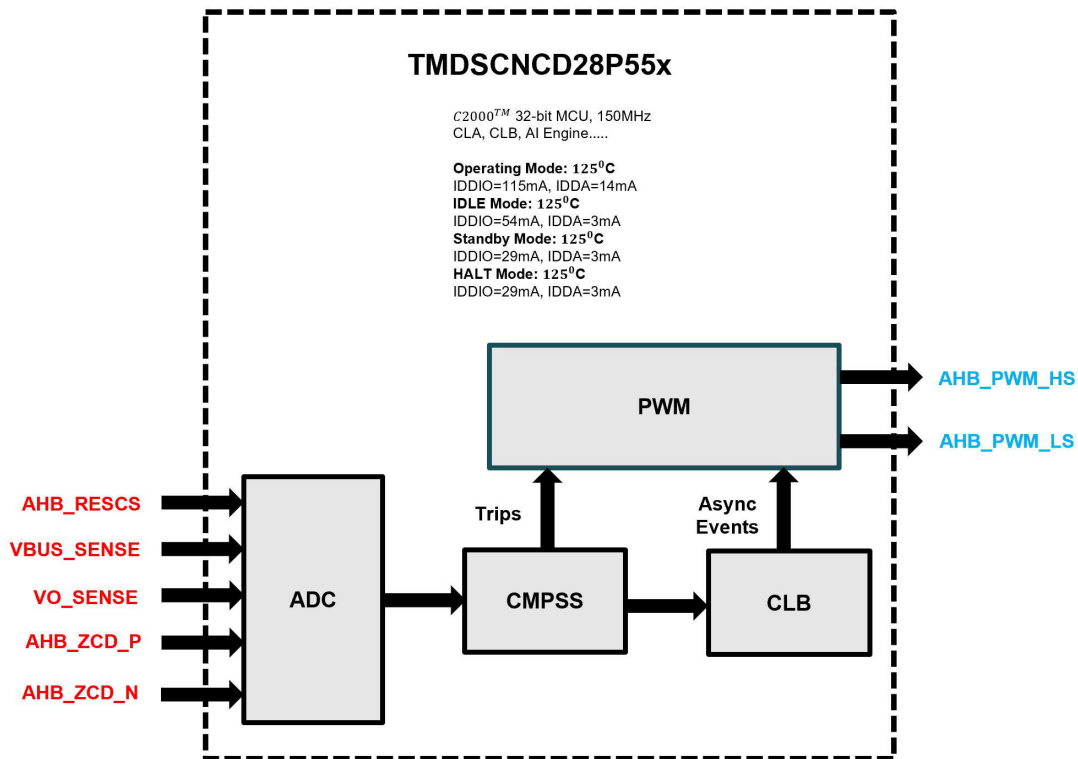


図 2-2. C2000™ マイコンを使用した制御の実装

図に、デジタル制御アーキテクチャの概要を示します。この図には、AHB ハードウェアから制御ロジックへの信号フローが示されています。各パシフェラルの具体的な機能とタイミングの調整については、このドキュメントの以降のセクションで説明します。

### 3 ソフトウェアの概要

関連するソフトウェア情報については、次のリンクを参照してください。 [C2000WARE-DIGITAL-POWER-SDK](#)

#### CCS プロジェクトを開く:

このリファレンス デザインのソフトウェアは C2000Ware\_DigitalPower\_SDK で利用可能であり、初期化用としてペリフェラル SYSCFG フレームワークでサポートされています。プロジェクトを開くには、次の手順に従います。

1. CCS (バージョン 12.5 以降) をインストールします。
2. [ツール ページ](#)から C2000Ware DigitalPower SDK をインストールします
3. CCS を開き、新しいワークスペースを作成します。
4. CCS 内で、[View] -> [Resource Explorer] を選択します。Resource Explorer で、[Software] → [C2000Ware DigitalPower SDK - <version>] → [solutions] の順に選択し、ソリューション PMP41140 を選択して [import project] をクリックします。このコードは、F28P55x デバイスで利用できます。

#### 注

CCS は、インポートされたプロジェクトに関連する特定バージョンのコンパイラをインストールすることを推奨します。必要に応じて、TI.com でコンパイラを見つけて、ダウンロードとインストールを行ってください。コンパイラのインストール後、プロジェクトのプロパティ メニューでコンパイラのバージョンを構成します。CCS プロジェクトツールの検出パスに、コンパイラのインストールのパスが含まれていることを確認してください。

#### 3.1 ソフトウェア アーキテクチャ

PMP41140 プロジェクトが Code Composer Studio (CCS) に正常にインポートされると、[図 3-1](#) に示すように、Project Explorer に完全なディレクトリ構造が表示されます。

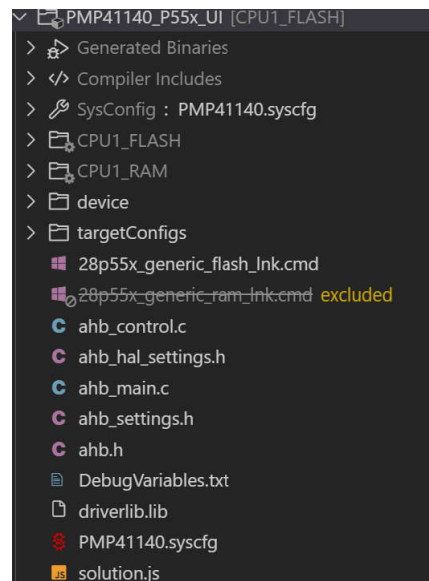


図 3-1. プロジェクト構造の概要

このプロジェクトは、SysConfig を活用して C2000™ F28P55x マイコン ファミリのデバイスとペリフェラルの初期化を行います。この設計では、使用する特定のデバイスは TMS320F28P550SGE-64PM を使用して、非対称ハーフブリッジ (AHB) DC/DC コンバータの高周波デジタル制御に必要な計算性能とペリフェラル セットを実現します。

表 3-1 の次のセクションでは、プロジェクトに含まれる主要なファイルの構造と機能について説明します。

表 3-1. サポートされるソース ファイルとヘッダー ファイル

ファイル	説明
ahb_main.c	ファームウェアのエントリ ポイント。このファイルは、システムの初期化、起動シーケンス、およびメイン制御ループの実行を管理します。また、ラボ固有のフォルト状態を監視し、適切なフォルト診断および回復メカニズムをトリガします。
PMP41140.syscfg	TI SysConfig ツールによって生成される構成ファイル。クロック設定、GPIO 割り当て、ペリフェラルのマッピング、割り込みソース、その他の低レベルの初期化パラメータを定義します。整合性を維持し、レジスタの競合を防止するために、システム構成の変更はこのファイルで行う必要があります。
ahb_control.c	割り込みサービス ルーチン (ISR) とリアルタイム デジタル制御アルゴリズムを含むコア制御ファイル。これには、電圧/電流のフィードバック処理、PWM デューティサイクルの更新、AHB コンバータのレギュレーションに適した制御ループの実行が含まれます。
ahb.h	デバイス固有およびプロジェクトレベルのすべての宣言を含む、コアとなるヘッダー ファイル。制御モジュールと診断モジュールで使用されるグローバル変数、マクロ、データ構造を定義します。また、すべての ISR のプロトタイプも宣言しています。
ahb_settings.h	ユーザーが調整可能なパラメータの構成ヘッダー。制御定数、保護スレッショルド、およびスケール係数の定義が含まれます。このファイルを使用すると、コア アルゴリズムを変更せずに、システム パラメータを迅速に調整できます。
ahb_hal_settings.h	物理的な ADC チャンネルから論理制御変数へのマッピングを含む、ハードウェア抽象化レイヤ ヘッダー。これにより、ハードウェアの移植性が保証され、類似のデバイスやボード間でのファームウェアの移行が簡素化されます。

PMP41140 ファームウェアは、開発、デバッグ、再利用性を簡素化するためにモジュール化された形式で構成されています。各ソース モジュールは、特定のサブシステムを処理します。

- **初期化レイヤ:** SysConfig と AHB\_main.c で管理され、確定的なブートとペリフェラルの設定を実装します。
- **制御レイヤ:** AHB\_control.c に実装されており、メインの閉ループ制御アルゴリズムをリアルタイムで実行します。
- **ハードウェア抽象化レイヤ (HAL):** AHB\_hal\_settings.h で実装され、ハードウェア構成とアルゴリズム ロジックが明確に分離されます。
- **ユーザー構成レイヤ:** AHB\_user\_settings.h で公開され、フレキシブルなシステム チューニングとラボ評価が可能です。

この構造化されたアプローチにより、ハードウェア構成、リアルタイム制御、特定用途向けのチューニングを明確に分離できるため、デバッグの高速化、拡張性の強化、他の C2000™ F28x デバイスへの容易な移植が可能になります。

### 3.1.1 デバイスの初期化

このプロジェクトでは、TI SysConfig ツールを使用して、すべてのデバイス ペリフェラルの初期化と構成を行います。

AHB\_main.c ファイル内で呼び出される `board_init()` 関数は、アプリケーション ソフトウェアを `c2000.syscfg` ファイルで定義されている構成パラメータにリンクします。

SysConfig ファイルは、F28P55x ファミリーに固有のデバイスであり、ペリフェラルとピンの割り当てを一元管理する中央リポジトリとして機能します。この構成により、アナログ ペリフェラルと制御ペリフェラルの両方を、一貫性のある保守可能な方法で初期化します。

具体的には、SysConfig を使用して以下のモジュールを構成します。

- **アナログ ペリフェラル:** ADC、ASYSCTL、CMPSS

- 制御ペリフェラル: ePWM、CLB
- システム ペリフェラル: GPIO、X-BAR、割り込み、CPU タイマ

このアプローチにより、起動時の確定的な初期化が保証され、手動によるレジスタ構成エラーが低減されます。また、ペリフェラル マッピングを簡単に変更できるグラフィカル インターフェイスが提供されます。構成された各ペリフェラルと、制御アーキテクチャにおける役割の詳細については、以降のセクションで説明します。

### 3.1.1.1 アナログ ペリフェラルの初期化

このプロジェクトのアナログ ペリフェラルは、主に、アプリケーション ハードウェアからのアナログ フィードバック信号とマイコンのインターフェイスとなります。これらのペリフェラルは、電圧と電流のリアルタイム情報をキャプチャするため、デジタル制御アルゴリズムにより高精度で確定的な決定を行うことができます。

#### ASYSCTL の初期化:

ASYSCTL (アナログ システム制御) ペリフェラルは、ADC 変換の電圧リファレンスとして機能するアナログ リファレンス (VREF) の構成に使用されます。VREF を適切に設定することにより、異なる温度および動作条件にわたって測定の一貫性と精度が確保されます。

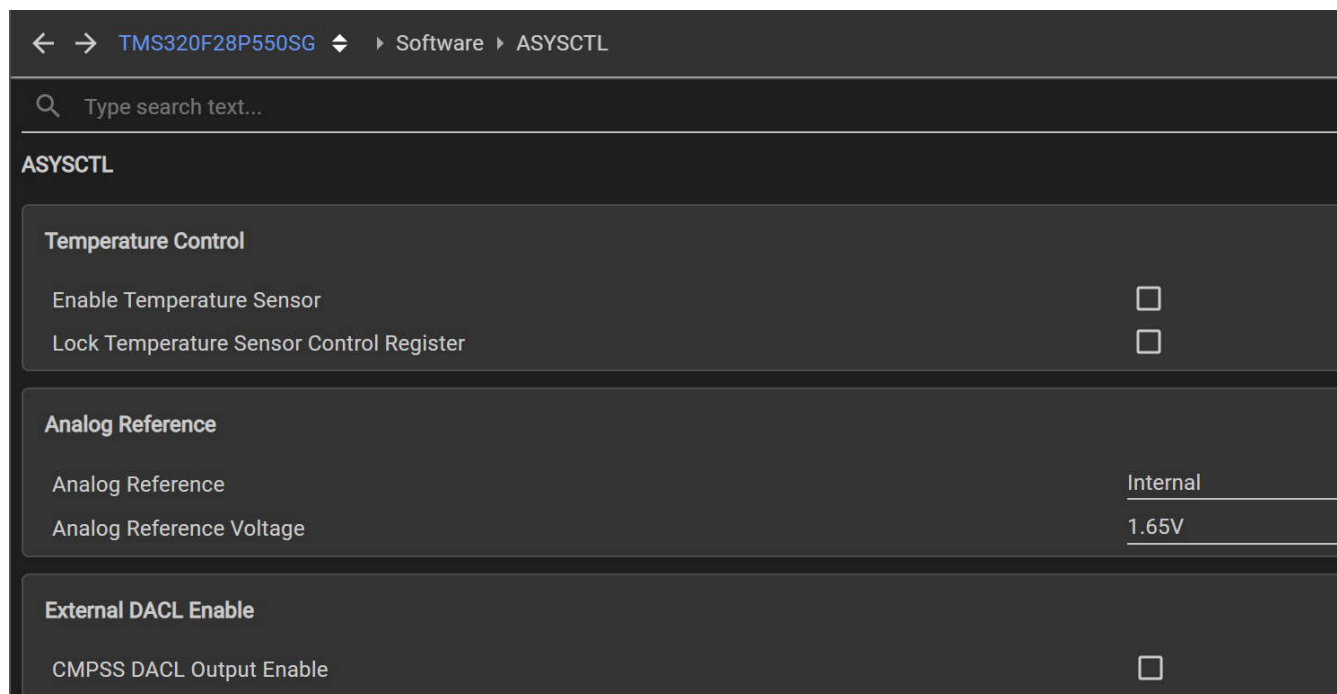


図 3-2. ASYSCTL ペリフェラルの初期化

図に示すように、このプロジェクトのアナログ リファレンスは 1.65V のアナログ リファレンス電圧で内部構成されています。内部リファレンスは内部ゲイン係数が  $\times 2$  でさらにスケールされるため、0V ~ 3.3V の実効 ADC 入力範囲が得られます。この構成により、ADC は外部リファレンス回路を必要とせずに、コンバータのフィードバック範囲に対応するフルスケール アナログ信号を正確にサンプリングできます。

#### ADC の初期化:

A/D コンバータ (ADC) モジュールは、出力電圧、入力電圧、インダクタ電流など、各種アナログ フィードバック信号のサンプリングに使用します。SysConfig では、アナログ入力チャネル、サンプリング ウィンドウの構成、変換速度、トリガ ソー

ス、ADC クロック周波数の選択が可能です。これらのパラメータにより、PWM スイッチング サイクルにおけるフィードバック信号のサンプリング精度と同期が決定されます。

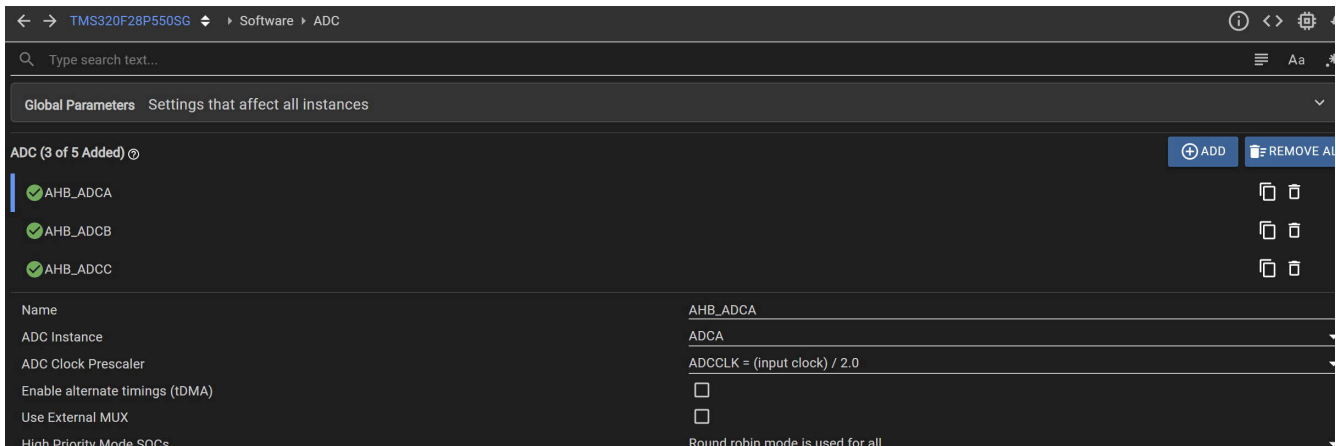


図 3-3. ADC クロック初期化

このプロジェクトでは、3 つの ADC モジュール (AHB\_ADCA、AHB\_ADCB、AHB\_ADCC) を使用して、電力段からアナログフィードバック信号を捕捉します。すべての ADC は 75MHz の ADC クロック周波数で動作するように構成されます。これらの ADC は、図 3-3 に示すように、各 ADC の SysConfig に適切な ADC クロックプリスケアラを設定することで実現されます。

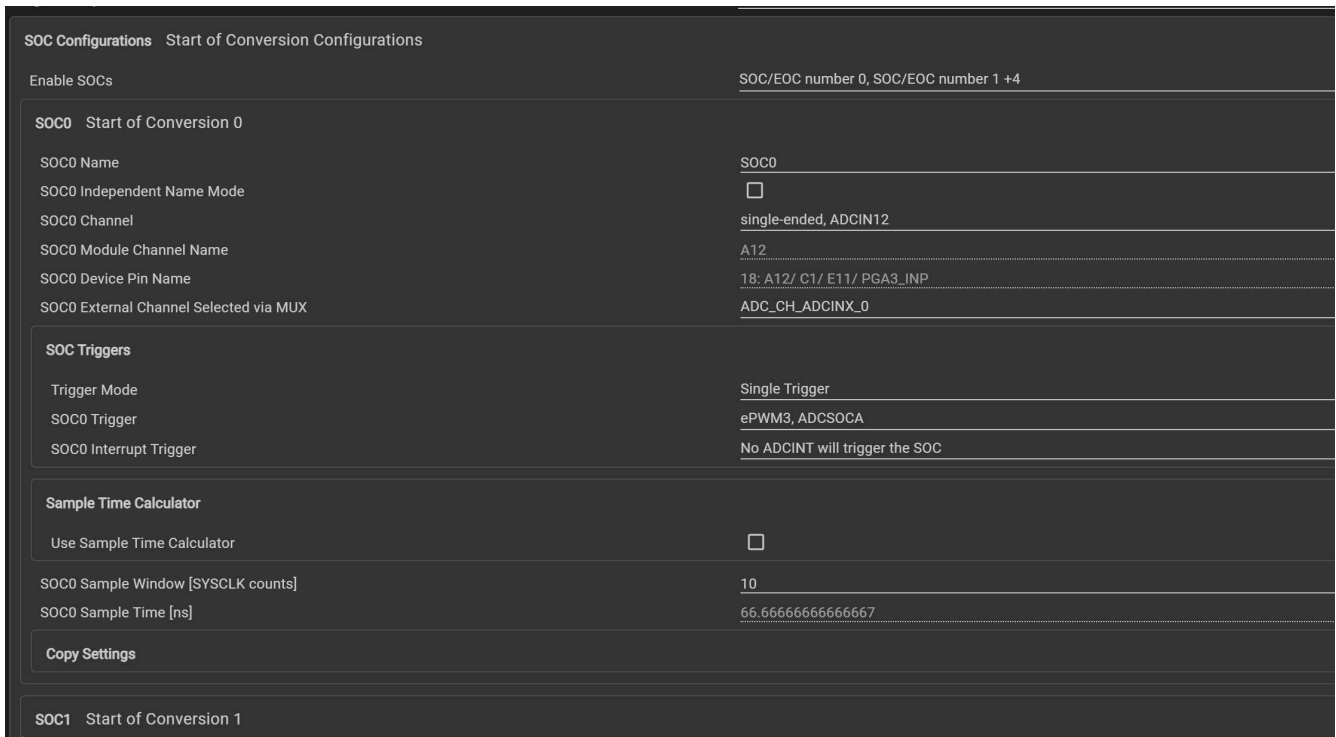


図 3-4. ADC SOC の初期化

各 ADC モジュールには、SOC0 ~ SOC15 というラベルの付いた 16 個の変換開始 (SOC) ユニットが含まれています。ファームウェアでは、デフォルトのラウンドロビン優先方式を使用しています。SOC0 は最高の変換優先度、SOC15 は最低の優先度となります。このスケジューリングにより、複数のフィードバックチャンネル間での確定的なサンプリングが保証されると同時に、平衡な変換タイミングが維持されます。

各 SOC は、次のように個別に構成できます。

- 入力チャンネル選択: ADC 入力マルチプレクサに接続するアナログ入力ピン (A0 ~ A15, B0 ~ B15 など) を指定します。
- トリガ ソース: 変換を開始するイベント (ePWM トリガなど) を決定します。
- サンプル アンド ホールド ウィンドウ: 変換開始前の安定した信号サンプリングのアクイジション期間を定義します。

図 3-5 に示すように、構成の 1 つの例は ADCA-SOC0 であり、入力にアナログ チャンネル A12 を使用し、ePWM3.ADCSOCA をトリガ ソースとして、10 サイクルのサンプリング ウィンドウを使用します。この設定では、ADC サンプリングと PWM スイッチング イベントを同期し、制御と測定の間で一貫したタイミングを確保します。

Priority	ADC-A	ADC-B	ADC-C
High Priority Frequency with PWM (20KHz - 200KHz)	AHB_ZCD_N (A12) SOC0 -> ADC_TRIGGER_EPWM3_SOCA SOC1 -> ADC_TRIGGER_EPWM3_SOCA		
	AHB_RESCS (A1) SOC3 -> ADC_TRIGGER_EPWM3_SOCA		
	VO_SENSE (A3) SOC2 -> ADC_TRIGGER_EPWM3_SOCA SOC4 -> ADC_TRIGGER_EPWM3_SOCA SOC5 -> ADC_TRIGGER_EPWM3_SOCA		
low Priority Frequency with Timer (50KHz)		VBUS_SENSE (B15) SOC2 -> ADC_TRIGGER_CPU1_TINT1 SOC3 -> ADC_TRIGGER_CPU1_TINT1 SOC4 -> ADC_TRIGGER_CPU1_TINT1	AHB_TEMP (C14) SOC5 -> ADC_TRIGGER_CPU1_TINT1 SOC6 -> ADC_TRIGGER_CPU1_TINT1

図 3-5. ADC を使用したアナログ信号マッピング

図 3-5 に、アナログ ピンから ADC チャンネルへのマッピング、関連する SOC 構成、それぞれのトリガ ソースを示します。このマッピングは、リアルタイム制御機能のために、出力電圧、入力電圧、インダクタ電流などの主要なフィードバック変数をシステムがどのようにキャプチャするかを定義します。

### CMPSS の初期化:

CMPSS (コンパレータ サブシステム) ペリフェラルは、ハードウェアベースのフォルト監視と保護に使用されます。各 CMPSS モジュールは、検出されたアナログ入力がプログラムされたスレッシュホールドを上回るか下回ると、トリップ信号を生成できます。これらのトリップ イベントは一般に、過電流、過電圧、またはその他の異常動作条件でシステムを保護するために、PWM 出力のディスエーブルやフォルト割り込みのトリガなど、すぐに保護動作を開始するために使用されます。

Module	Comparator	Analog Pins	Default DAC Value	Description
CMPSS1	High	AHB_RESCS (A1)	3000	Inductor Current Peak Trip
	Low	AHB_RESCS (A1)	3120	Over Current Trip
CMPSS2	High	AHB_ZCD_N (A12)	AHB_ZCD_P (A10)	Zero Crossing Detection
	Low	AHB_ZCD_P (A10)	1490	ZVS Diagnostics
CMPSS3	High	VO_SENSE (A3)	4040	Over Voltage Trip

図 3-6. CMPSS を使用したアナログ信号マッピング

このプロジェクトでは、図 3-5 に示すように、保護、センシング、制御診断のために 3 つの CMPSS (コンパレータ サブシステム) モジュールを使用します。各 CMPSS モジュールは、独立した DAC リファレンスと柔軟な入力構成を備えた、ハイコンパレータとローコンパレータの両方を提供します。

#### 1. CMPSS1:

- ハイコンパレータとローコンパレータは、どちらも **AHB\_RESCS** から同じアナログ入力を受信します。
- ハイコンパレータは、ピーク電流検出用に構成されています。この **DAC** リファレンス値は、コントローラ出力に基づいてすべての **ISR2** で動的に更新されるため、適応型の電流制御が可能になります。
- ローコンパレータは、過電流保護メカニズムとして機能します。この **DAC** リファレンスは、定義された電流制限に基づいて事前に計算され、ハードウェアレベルのフォルト応答を実現します。

## 2. CMPSS2:

- ハイコンパレータはゼロ交差検出 (**ZCD**) 用に構成され、入力 **AHB\_ZCD\_P (A10)** および **AHB\_ZCD\_N (A12)** による差動モードで動作します。これにより、共振またはソフトスイッチング動作におけるゼロ電流ポイントまたはゼロ電圧ポイントを高精度で検出できます。
- ローコンパレータは、**AHB\_ZCD\_P (A10)** からのシングルエンド入力を使用して、**ZVS (ゼロ電圧スイッチング)** 診断用に構成されます。**DAC** シャドウ値は、**PWM** サイクルごとに更新され、スイッチング条件を動的に監視します。

## 3. CMPSS3:

- ハイコンパレータは、過電圧保護 (**OVP**) のために使用されます。アナログ入力 **VO\_Sense (A3)** は、設定された電圧制限に対応する **DAC** スレッショルドと比較されます。

過電流保護トリップと過電圧保護トリップの両方が **EPWM X-BAR** を介してルーティングされ、すべてのパワースイッチを即座に安全なオフ状態に強制するワンショットトリップ信号が生成されます。

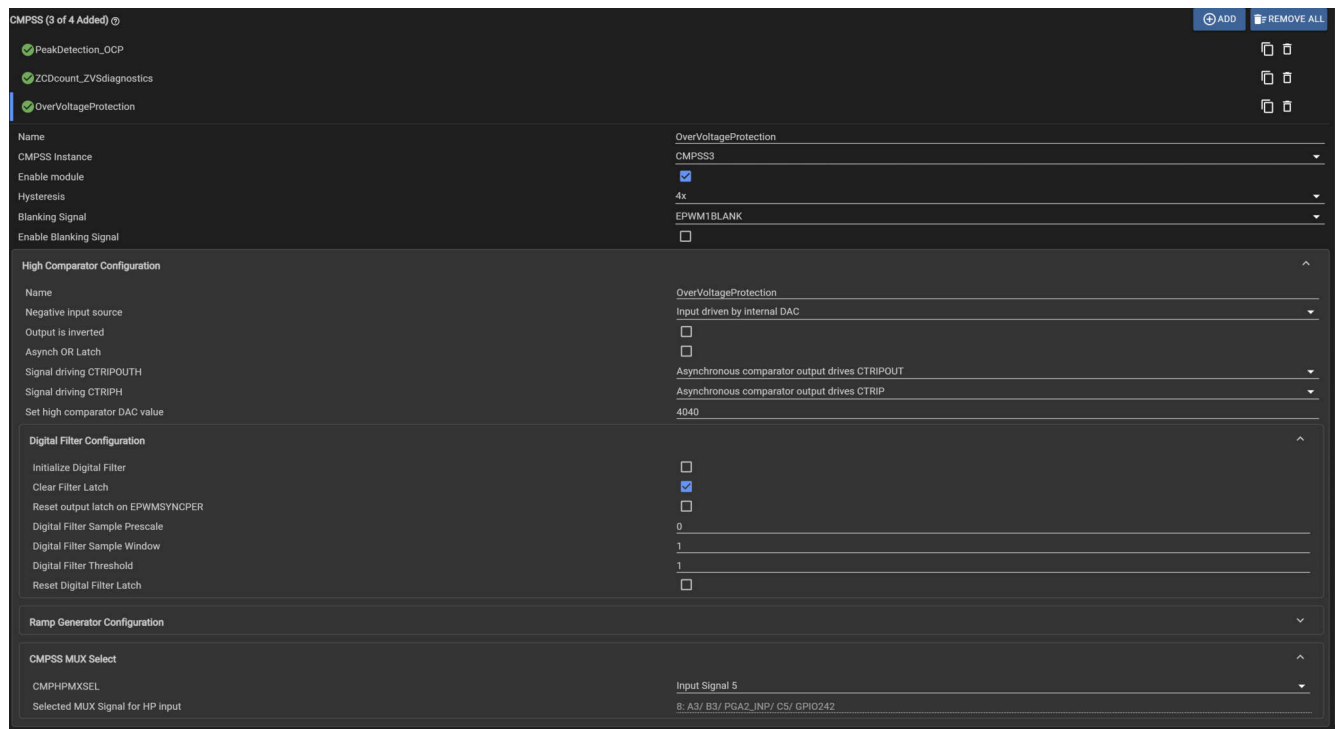



図 3-7. シングルエンド用の CMPSS の構成

以下の  に CMPSS3 ハイコンパレータの構成例を示します。

- 始めに、**SysConfig** の **[Enable Module]** オプションを選択して、**CMPSS3** インスタンスをイネーブルにする必要があります。コンパレータのヒステリシスは **x4** に設定されているため、入力でのノイズ耐性が向上します。
- ハイコンパレータの場合、負入力源に **[Input driven by internal DAC]** を選択し、コンパレータをシングルエンドモードに構成します(差動動作の場合、負入力源は **[Input driven by external pin]** に設定します)。コンパレータ出力 (**CTRIPOUTH/CTRIPH**) は非同期として構成されているため、システムクロックを待たずに即座にフォルト応答が可能です。過電圧比較のための **DAC** リファレンス値は **4040** として構成されています。これは、システム電圧スケールから得られた目標の過電圧トリップスレッショルドに対応しています。

- 最後に、CMPHMSEL レジスタを使用してコンパレータ入力ピンを選択します。この場合、入力信号 5 を選択します。これは、アナログ入力 A3 にマッピングされます(CMPSS 入力信号のアナログピンへのマッピングについては、デバイスの TRM を参照してください)。

要約すると、アナログペリフェラルは総合的に、電力段とデジタル制御コアをブリッジ接続する重要なセンシングおよび保護フレームワークを提供し、コンバータの安定かつ安全で効率的な動作を保証します。

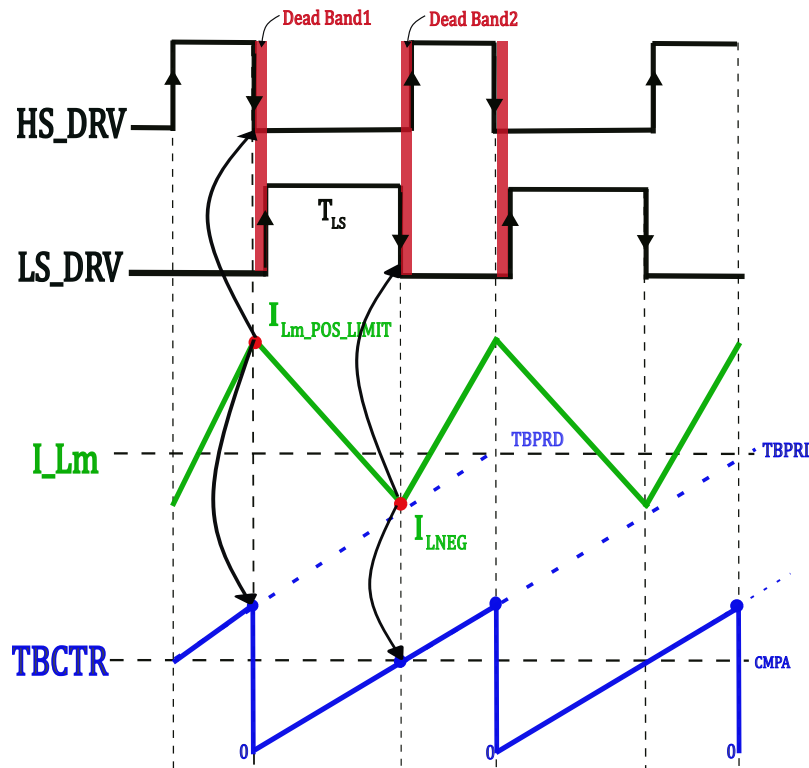
### 3.1.1.2 制御ペリフェラルの初期化

#### 負荷固有の PWM の実装:

PWM スwitching サイクルごとに、負荷条件が継続的に監視され、事前に定義された負荷スレッショルドレベルに基づいて、次の Switching サイクルの PWM 構成が動的に更新されます。このリアルタイム適応により、さまざまな負荷条件に効果的に変調手法を対応させることが可能になり、軽負荷と重負荷の両方のシナリオにおいて、高効率で信頼性の高い ZVS (ゼロ電圧 Switching) 動作と安定したコンバータ性能を維持できます。

#### 重負荷 PWM 波形

図に示すように、重負荷状況では、PWM 信号は、適切に挿入されたデッドバンドによる相補波形として構成されます。時間ベース周期 (TBPRD) は大きな値にプログラムされますが、ピークトリップ信号によってトリガされる同期イベントにより、時間ベースカウンタ (TBCTR) が TBPRD に達する前に強制的に再起動され、PWM Switching 周期が効果的かつ動的に制御されます。各 Switching サイクルで、比較レジスタ (CMPA) は、制御ループで実行される  $T_{LS}$  の計算に基づいて更新されます。



$$T_{LS} = [L_m * (I_{Lm\_POS\_LIMIT} - I_{LNEG})] / (N * V_o)$$

図 3-8. 重負荷 PWM の波形

表 3-2. 重負荷 PWM のイベントとアクション

PWM イベント	HS_DRV	LS_DRV
TBCTR = 0	立ち下がりエッジ	立ち上がりエッジ

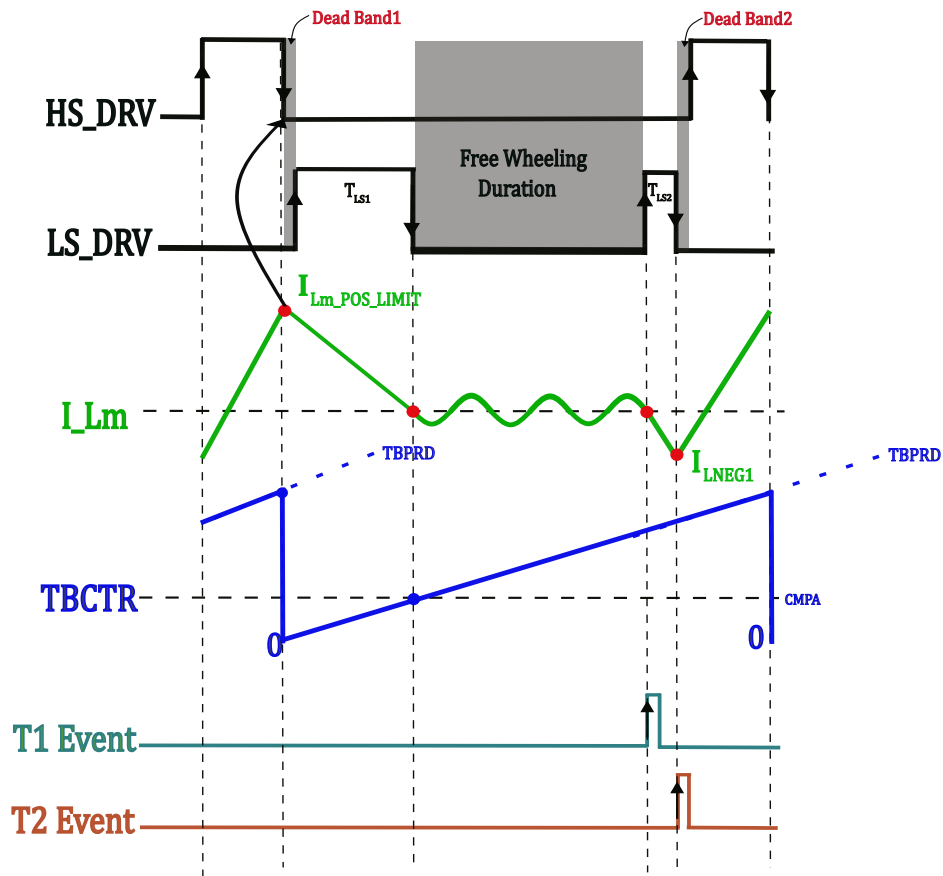
表 3-2. 重負荷 PWM のイベントとアクション (続き)

PWM イベント	HS_DRV	LS_DRV
TBCTR = CMPA	立ち上がりエッジ	立ち下がりエッジ

表に示されているように、HS\_DRV と LS\_DRV の PWM アクションは、PWM モジュールの時間ベース カウンタ (TBCTR) に基づいて実行されます。

軽負荷の PWM 波形

図に示すように、軽負荷状況では、PWM 信号は相補型ではありません。重負荷動作と同様に、TBPRD は大きな値で設定されますが、ピークトリップ信号によって発生した同期イン イベントにより、プログラムされた期間に達する前に TBCTR カウンタが再起動され、オン時間動作を変化させることができます。各スイッチング サイクルで、 $T_{LS1}$  の計算に基づいて CMPA が更新され、フリーホイール間隔が導入されてスイッチング損失を最小限に抑えます。このフェーズ中、システムはゼロ交差 (ZCD) イベントを監視し、正から負への遷移をカウントします。ターゲットのゼロ交差カウントが達成されると、T1 イベントが生成され、T2 イベントが生成されます。これは、 $T_{LS2}$  タイミング パラメータで決定される T1 の遅延です。



$$T_{LS1} = [L_m * (I_{Lm\_POS\_LIMIT} - 0)] / (N * V_o)$$

$$T_{LS2} = [L_m * (0 - I_{LNEG1})] / (N * V_o)$$

図 3-9. 軽負荷の PWM 波形

表 3-3. 軽負荷 PWM のイベントとアクション

PWM イベント	HS_DRV	LS_DRV
TBCTR = 0	立ち下がりエッジ	立ち上がりエッジ
TBCTR = CMPA	変更なし	立ち下がりエッジ

表 3-3. 軽負荷 PWM のイベントとアクション (続き)

PWM イベント	HS_DRV	LS_DRV
T1 イベント	変更なし	立ち上がりエッジ
T2 イベント	立ち上がりエッジ	立ち下がりエッジ

これらの負荷固有の PWM 波形は、EPWM および CLB (構成可能なロジック ブロック) ペリフェラルとの連携により実装されています。EPWM は高精度の時間ベース制御および同期を実現し、CLB はカスタム ロジックで T1、T2、ゼロ交差検出などの適応型タイミング イベントを生成できます。これらのペリフェラルの詳細な構成と機能的な相互作用については、以降のセクションで説明します。

### 3.1.1.2.1 EPWM の初期化

AHB\_EPWM モジュールは、電力段スイッチ、つまりハイサイド FET (HSFET) とローサイド FET (LSFET) を駆動するための 1 次制御ブロックとして機能します。このプロジェクトでは、EPWM3 が AHB\_EPWM モジュールとして指定され、GPIO4 が EPWM3A に割り当てられ (HSFET を制御)、GPIO5 が EPWM3B に割り当てられています (LSFET を制御)。

時間ベース サブモジュール -

- PWM クロック プリスケアラは、時間ベース クロック (TBCLK) がデバイスのシステム クロックと等しくなるように構成されます (すなわち、TBCLK = SYSCLK/1)。時間ベース クロック分周器と高速クロック分周器の両方が 1 に設定されています。
- カウンタ モードはアップカウントとして設定され、初期時間ベース周期 (TBPRD) は 4000 です。これは、コンバータの動作条件に基づいて、実行時に ISR 内で動的に更新されます。
- PWM 同期は、入力 XBAR5 によってピークトリップ イベントによってトリガされる EPWM 同期パルスによって実現されます。位相ロードが有効で、位相シフト値が 0 のため、同期中に PWM カウンタの整合性が確保されます。

カウンタ比較レジスタ (CMPA/CMPB) は初期化後に設定され、確定的な更新のため、TBCTR = 0 イベント時にシャドウロードされます。

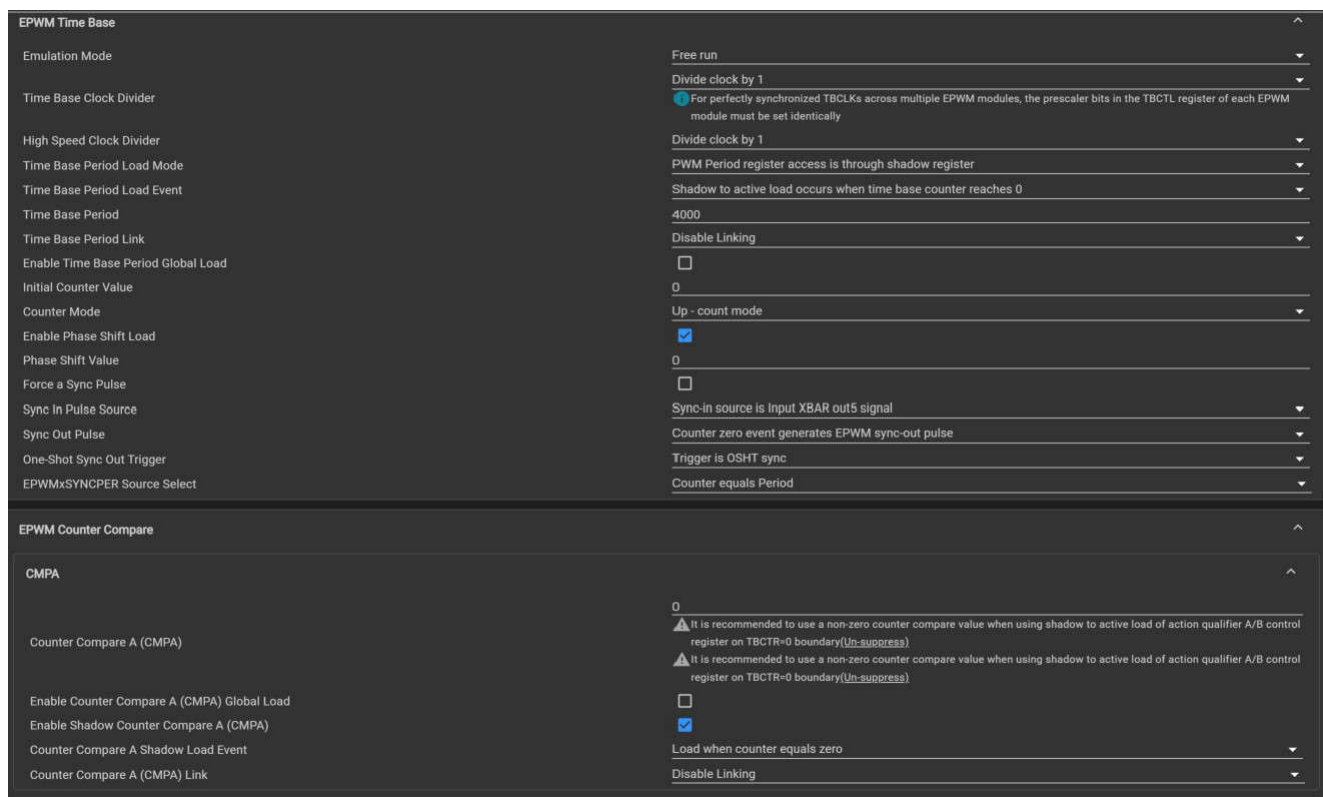


図 3-10. EPWM 時間ベースおよびカウンタ比較サブモジュール

アクション クオリファイア サブモジュール -

- この構成では、トリガ 1 (T1) ソースとトリガ 2 (T2) ソースがそれぞれ DCAEVT2 と DCBEVT2 にマッピングされます。
- EPWM A/B 出力は、同期時、またはカウンタがゼロまたは周期値となった場合にシャドウロードされます (参照図を参照)。

- 最初に、アクション クオリファイア (AQ) イベントは、両方の PWM 出力が Low 状態を維持するように設定されます。実行時、ISR 内では、AHB コンバータの電圧スレッショルド レベルと負荷条件に基づいて AQ 動作が動的に更新されます。

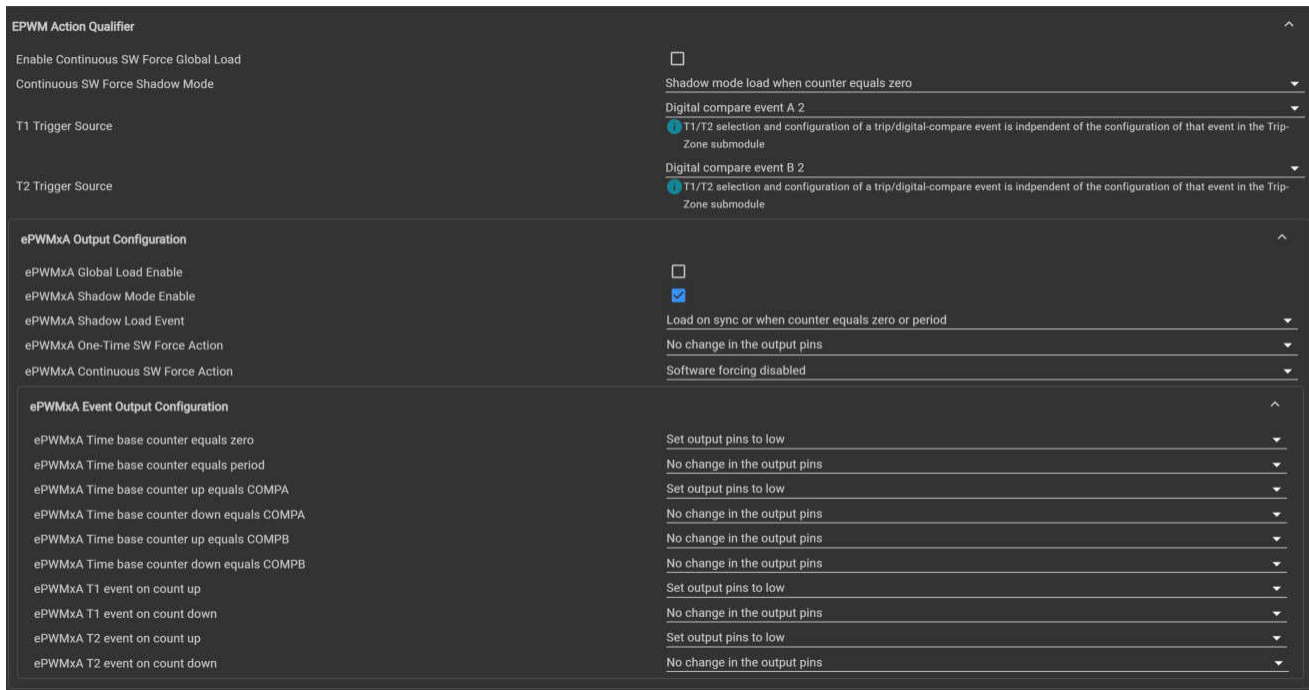
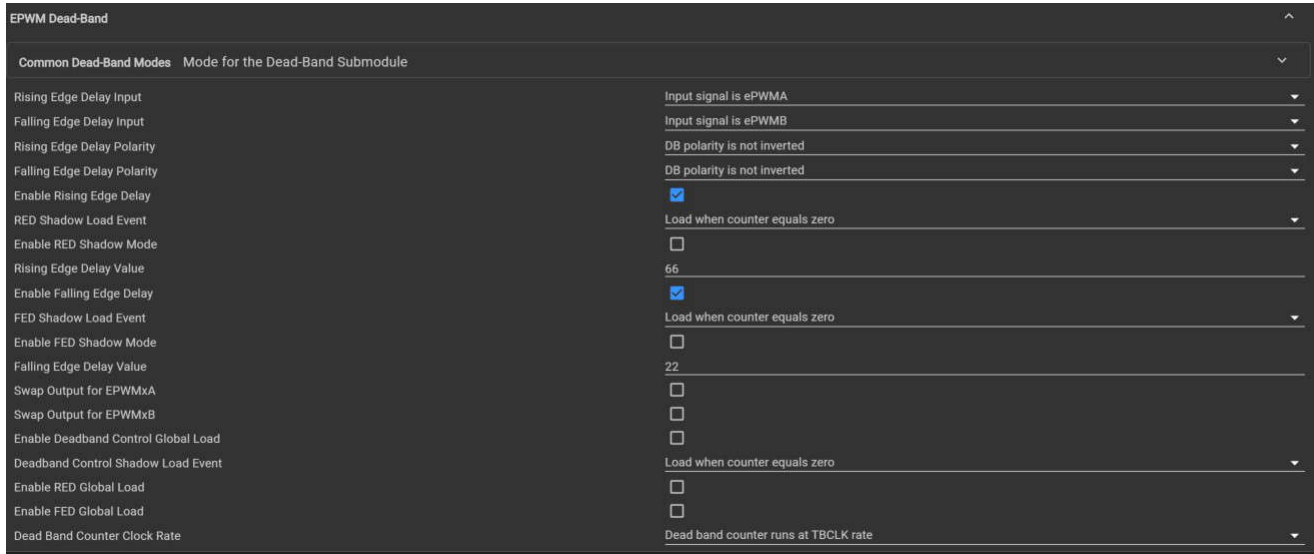


図 3-11. アクション クオリファイア サブモジュール

#### デッドバンド サブモジュール -

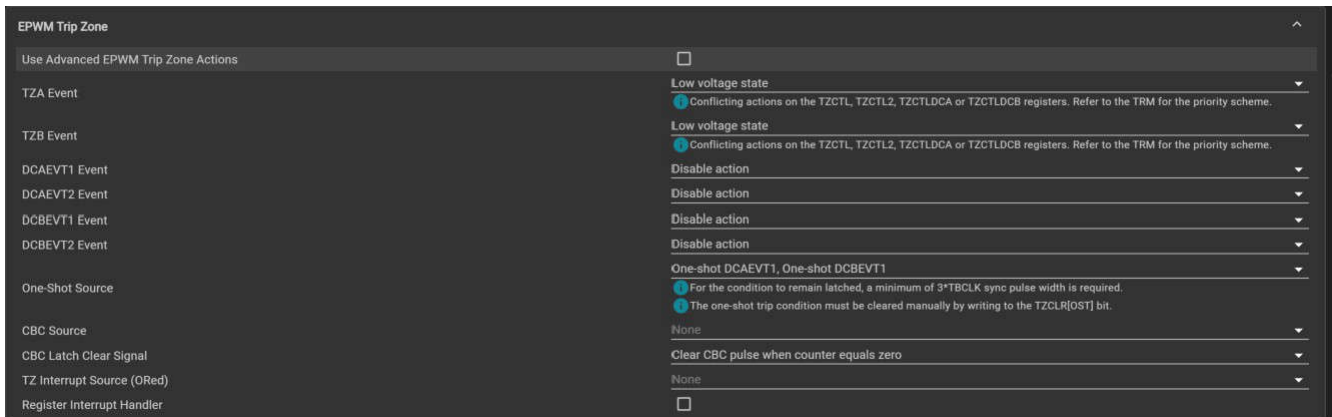
- デッドバンド ジェネレータは図のように構成されており、立ち上がりエッジ遅延 (RED) 入力と立ち下がりエッジ遅延 (FED) 入力はそれぞれ EPWM A と EPWM B によって駆動されます。初期化中は、RED と FED の両方の極性は反転しません。
- ただし、main.c での基板初期化後、FED 極性は反転され、正しい相補駆動動作が保証されます。この反転には、AQ サブモジュールで対応する調整が必要です。ここで、出力を強制的に Low にすることを目的としたイベントを、反転のために High にプログラムする必要があります。

- RED と FED はどちらもシャドウ モードに設定され、TBCTR = 0 でロード イベントが発生し、それらの遅延値はそれぞれ 66 と 22 に設定されます。


**図 3-12. デッドバンド サブモジュール**

トリップゾーン サブモジュール -

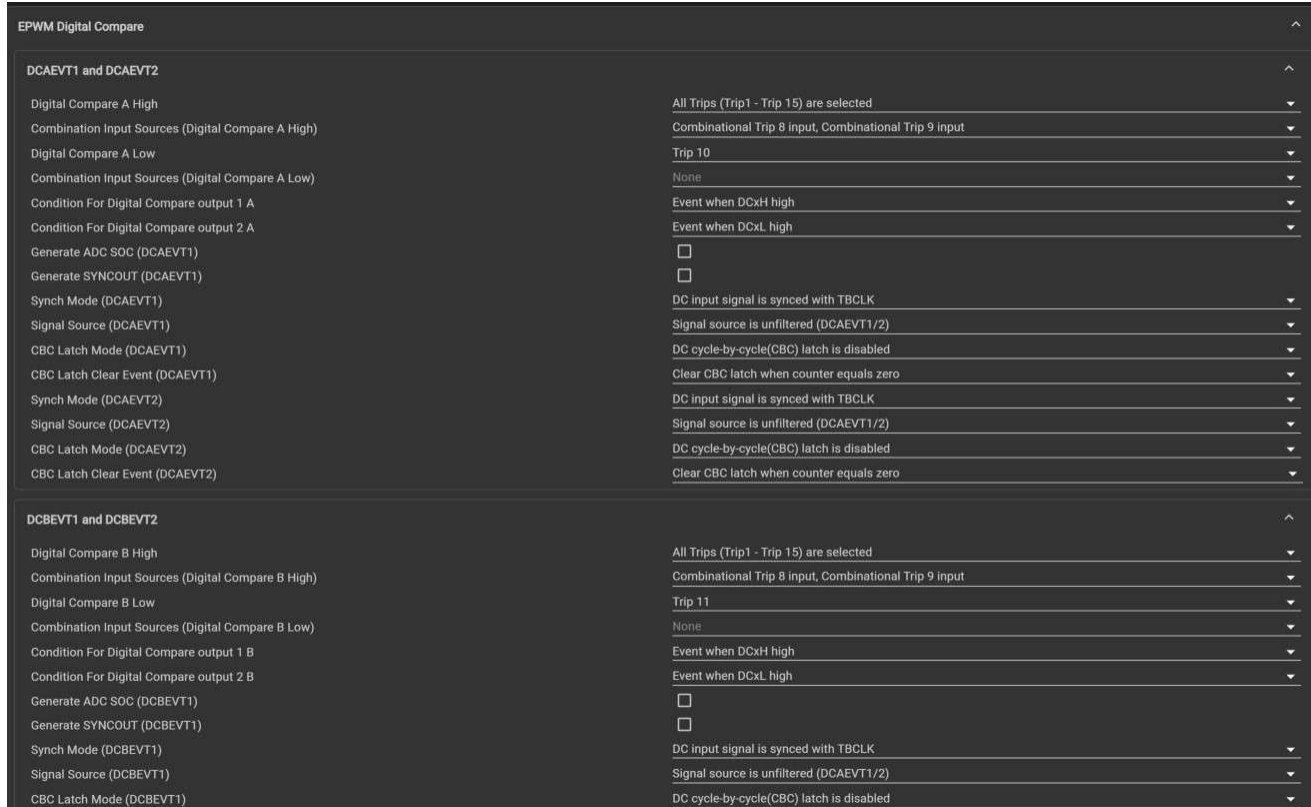
- トリップゾーン (TZ) サブモジュールは、過電流または過電圧フォルトが発生した場合にワンショット保護を提供するように設定されています。
- ワンショット ソースは DCAEVT1 または DCBEVT1 として選択され、直接デジタル比較アクションを無効になっています。トリガ時に TZA と TZB の両方のイベントがそれぞれの PWM 出力を Low 状態に駆動し、直ちにスイッチをオフにしてハードウェアを保護します。


**図 3-13. トリップゾーン サブモジュール**

デジタル比較サブモジュール -

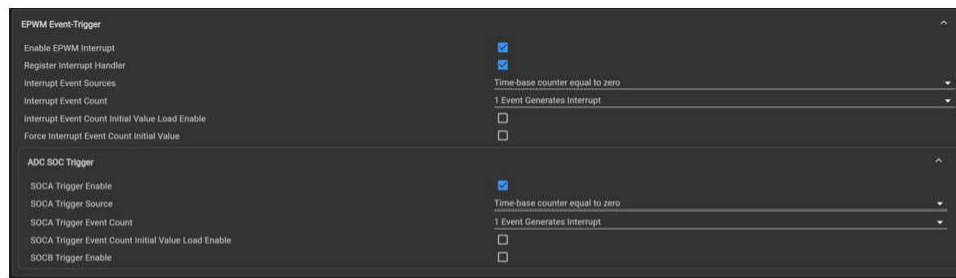
- デジタル比較 (DC) サブモジュールは、トリップおよび診断条件のための複数のイベントソースを実装しています。
  - DCAEVT1 は、EPWM XBAR の Trip8 と Trip9 の論理的な組み合わせによって生成されます。
  - DCAEVT2 は、Trip10 をソースとしています。
  - DCBEVT1 は、DCAEVT1 に類似した Trip8 と Trip9 の組み合わせから派生しています。
  - DCBEVT2 は、EPWM XBAR の Trip11 により構成されます。

- ソース信号がロジック High (DCxH/L = 1) のとき、DCA/BEVT1 および DCA/BEVT2 イベントは両方もアクティブになり、フォルト条件発生時の即座のトリップ応答が保証されます。


**図 3-14. デジタル比較サブモジュール**

#### イベントトリガ サブモジュール -

- イベントトリガ (ET) サブモジュールは、割り込みと ADC 変換開始 (SOC) 信号を生成します。
- EPWM 割り込みは有効化され、時間ベース カウンタがゼロ (TBCTR = 0) になると生成されるように設定されます。このイベントで ADC SOC もトリガされ、ADC サンプリングが PWM スイッチング サイクルと同期されます。
- 割り込みサービス ルーチンは、PWM 更新タスクを実行するために INT\_AHB\_EPWM\_ISR (ISR1) ハンドラにリンクされています。


**図 3-15. イベントトリガおよび割り込みサブモジュール**

メインの AHB\_EPWM モジュールに加えて、このプロジェクトでは 2 つの補助 EPWM モジュール (ZCD\_EPWM および CMPSS\_BLANK\_EPWM) を使用して、シグナル コンディショニングおよび診断機能をサポートしています。これらのモ

ジュールはどちらも同じ時間ベース サブモジュール構成を共有しているため、システム全体での同期と一貫したタイミング動作を実現しています。

**ZCD\_EPWM** モジュールは、**DCAEVT1** がアクティブになるたびに **PWMA** 出力を **High** にアサートするように設定されています。**DCAEVT1** イベントソースは **EPWM XBAR Trip7** から生成されます。これは通常、ゼロ交差検出 (**ZCD**) イベントを示します。この **PWM** パルスは、制御アルゴリズム向けの高精度デジタル タイミング リファレンスとして機能するため、ファームウェアはスイッチング ノードまたは電流波形のゼロ交差遷移を正確に検出および処理できます。

**CMPSS\_BLANK\_EPWM** モジュールは、**CMPSS** コンパレータ用のブランキング ウィンドウを生成するように構成されています。このブランキング信号は、スイッチング遷移の直後に発生する可能性のある不要な過渡信号やスイッチング ノイズをマスクし、誤トリップやスプリアス コンパレータトリガを防止します。**EPWM** モジュールを介したブランキングパルスのタイミングを制御することで、コンパレータの応答速度と信頼性の両方を微調整できます。

これらの補助 **EPWM** モジュールはコンバータが動作するうえで必須ではありませんが、システムの安定性を向上させ、測定信頼性を高め、高度な診断および保護メカニズムの実装の簡素化を図るために重要な追加機能となります。

### 3.1.1.2 CLB の初期化

このアプリケーションでは、構成可能ロジック ブロック (**CLB**) モジュールを使用して、正確な **PWM** 制御と診断に必要な非同期イベントを生成します。2 つの **CLB** インスタンスが **AHB\_CLB** および **ZVS\_CLB** として使用され、それぞれがシステム内で異なる機能目的を果たしています。

#### AHB\_CLB

**AHB\_CLB** モジュールは主に **T1** および **T2** イベントを生成します。これは、軽負荷の **PWM** 波形の生成に重要です。これらのイベントは各 **PWM** サイクル内のタイミング境界を定義し、制御されたフリーホイール間隔と適応型スイッチング動作を実現します。このモジュールは、図 **X** のように **TILE0** で設定された **CLB1** インスタンスを使用します。7 つの入力信号を利用し、**CLB** ファブリック内に 3 つのカウンタ モジュールを実装して、**T1** および **T2** 生成の高精度のタイミング イベント

ントを合成します。CLB 出力は、EPWM X-BAR とその他の CLB モジュールの両方に接続されており、PWM エッジ制御やタイミング同期などの対応する動作のトリガとして機能します。

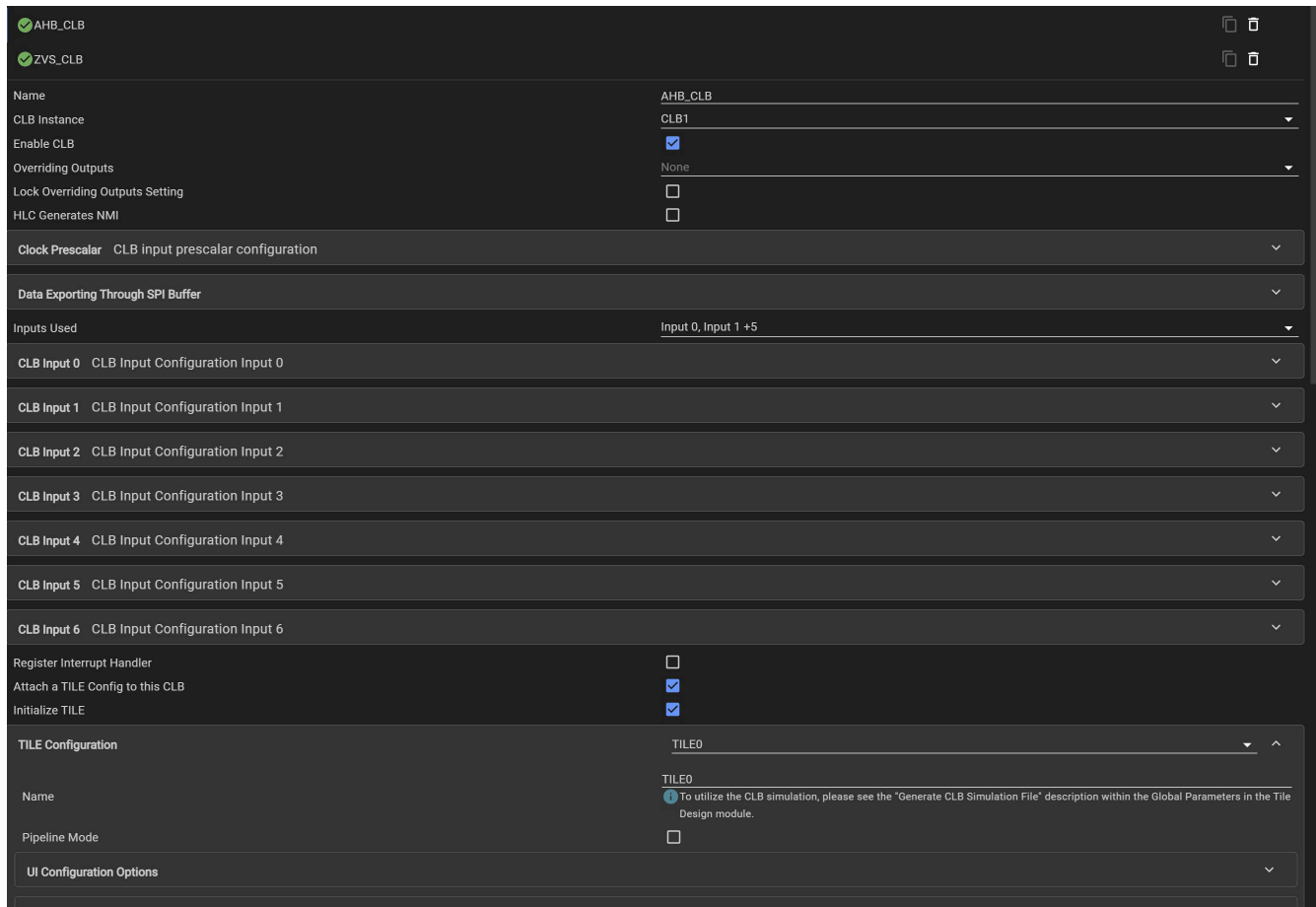


図 3-16. AHB\_CLB の構成

## ZVS\_CLB

ZVS\_CLB モジュールは、ZVS 動作を監視し、診断イベントの生成を支援するために使用されます。これにより、ゼロ電圧スイッチング条件で PWM 遷移が確実に発生するようになります。偏差の発生時には是正動作を実行できるように、制御ループにフィードバックを渡します。このロジックは、図 Y のように TILE1 上に構成された CLB2 インスタンスを使用して実装されます。6 つの入力信号を使用し、プログラマブルな組み合わせ論理ブロックとして動作する 2 つのルックアップテーブル (LUT) を使用します。これらの LUT 出力はカウンタ モジュールを介してさらに処理され、最終的な CLB 出力

は CLBOUTPUT X-BAR および他の CLB インスタンスを介して接続され、タイミング イベントを生成して必要に応じて診断をトリガします。

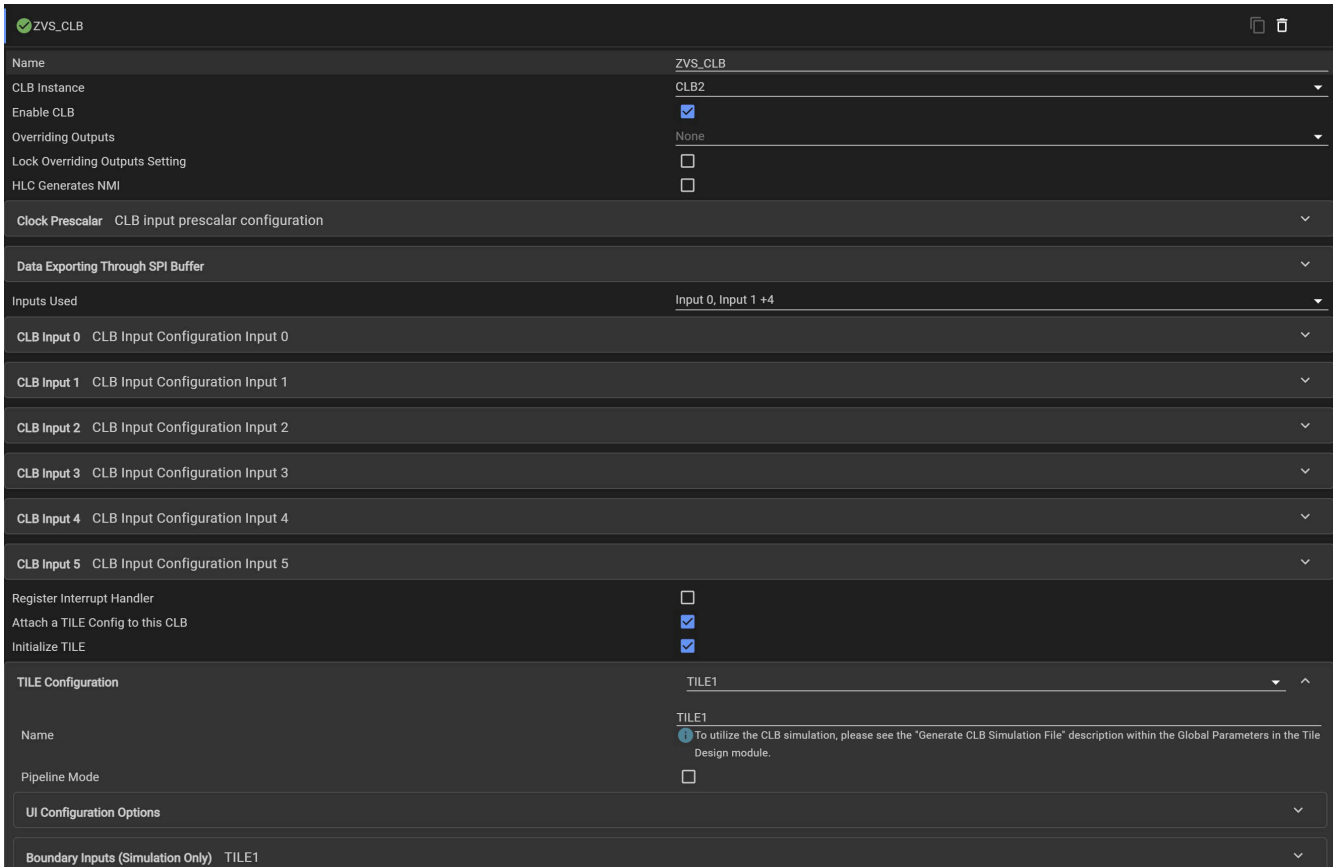


図 3-17. ZVS\_CLB の構成

AHB\_CLB と ZVS\_CLB を組み合わせて使用することにより、ファームウェアはハードウェアで非同期イベントを柔軟に生成および管理できるため、CPU の負荷を大幅に軽減できます。EPWM および X-BAR サブシステムと統合されることで、これらの CLB は高効率の適応型 PWM 制御の実現に不可欠な確定的タイミング、低レイテンシ イベント生成、リアルタイム応答性の向上を実現します。

### 3.1.1.3 システム ペリフェラルの初期化

GPIO、CPU タイマ、割り込み、クロスバー (X-BAR) などのシステム ペリフェラルは、システムレベルの統合と観測において重要な役割を果たします。これらのペリフェラルは、モジュールを相互接続するためのバックボーンとして機能し、ファームウェアの診断を可能にするとともに、リアルタイム実行制御を管理します。

このアプリケーションでは、これらはさまざまな重要な目的で使用されます。たとえば、デバッグと監視のために内部イベントを外部ピンにルーティングする、機能調整のためにペリフェラルの出力を別の入力に接続する、割り込みを生成してファームウェアで対応する ISR ルーチンとリンクする、コード実行時間のプロファイリングや特定のコード セクションの性能測定を行う、などです。これらのシステム ペリフェラルを組み合わせることで、アナログ、制御、デジタルの各サブシステムを、統合された応答性の高い組込み制御プラットフォームに接続する重要なグルー ロジックを実現できます。

#### CPU タイマの初期化:

このアプリケーションでは、次の 2 つの CPU タイマを別々の目的で使用しています。Timer1 は周期的な割り込みの生成に、ProfilingTimer は実行時間の測定に使用しています。

Timer1 は、CPUTIMER1 インスタンスを使用して構成され、タイマ周期値は 3030 に設定されています。対応する割り込みがイネーブルになり、INT\_Timer1\_ISR サービス ルーチンにリンクされます。設定されたパラメータに基づき、タイマ オ

オーバーフローは 50kHz の周波数で発生し、このレートで ISR がトリガされ、スケジュールされた制御または監視タスクを実行します。

**ProfilingTimer** は、**CPUTIMER2** インスタンスを使用して設定します。**SYSCCLK** をクロックソースとして、プリスケアラ値 1 として動作し、最高のタイミング分解能を実現します。タイマ周期は最大値 (65535) に設定され、割り込みはディスエーブルになります。このタイマは、ファームウェア内のプロファイリングとベンチマーク測定にのみ使用され、さまざまなユーザー定義セクションでコードの実行時間を高精度で測定できます。

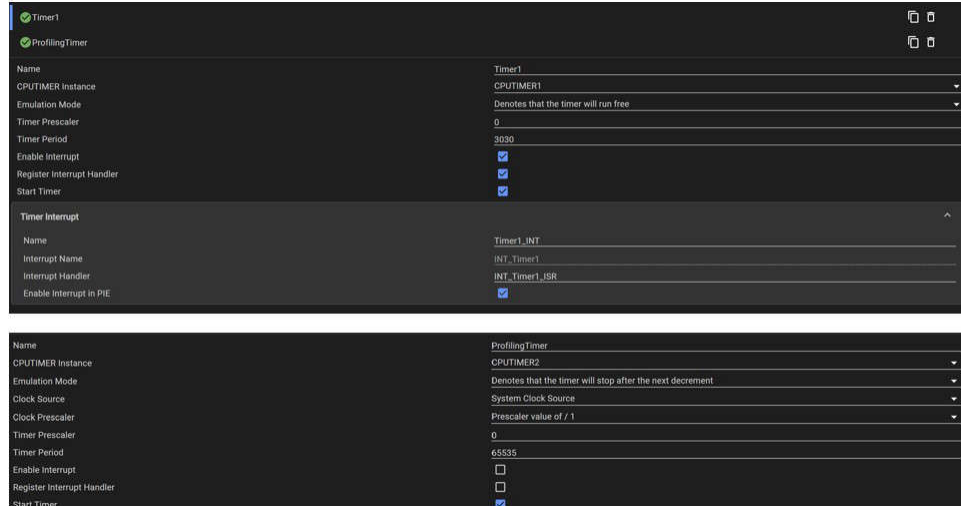


図 3-18. CPU タイマの構成

#### 割り込みの初期化:

これらの割り込みは、割り込み構成セクションで個別に構成されません。代わりに、タイマまたは PWM モジュールで対応する割り込みが有効になると、自動的に設定されます。この自動化により、追加の手動設定を必要とせずに、適切な割り込みマッピングと、それぞれの割り込みサービスルーチン (ISR) へのリンクが保証されます。

図のように、リンクされた ISR が割り込み構成ビューで正しく表示され、検証されていることを確認します。

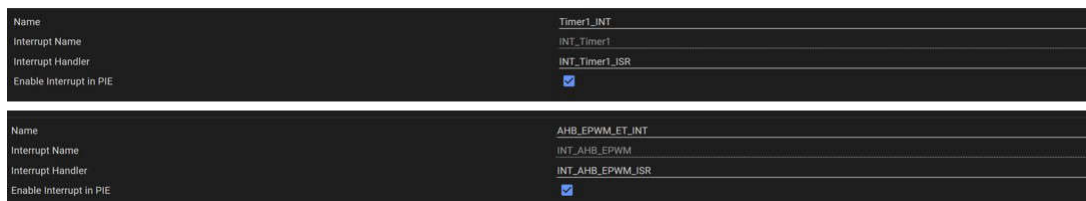


図 3-19. 割り込み構成

#### XBAR の初期化:

クロスバー モジュール (X-BAR) は、さまざまなオンチップ ペリフェラルをリンクする内部相互接続として機能し、外部領域への信号ルーティングを容易にします。

- **EPWMXBAR** および **CLBXBAR** は、それぞれ **EPWM** および **CLB** ペリフェラルへの入力接続を提供するため、デバイス内での柔軟なイベントルーティングが可能になります。
- **OUTPUTXBAR** と **CLBOUTPUTXBAR** は、内部信号を外部 **GPIO** ピンにルーティングするために使用され、デバッグや信号監視を可視化できます。

- INPUTXBAR を使用すると、イベントトリガまたは制御ロジックのために、外部 GPIO 入力を内部ペリフェラルにルーティングできます。

XBAR	Name	Source	Output
EPWMXBAR	ZCD Event	CMPSS2 TRIPH	Trip7
	PeakDetect	CMPSS1 TRIPH	Trip4
	T1 Event	CLB1 OUT4	Trip10
	T2 Event	CLB1 OUT5	Trip11
	OC_Protection	CMPSS1 TRIPL	Trip8
	OV_Protection	CMPSS3 TRIPH	Trip9
OUTPUTXBAR	PeakDetection_Outputxbar	CMPSS1 CTRIPOUTH	GPIO2
	ZeroCrossingDetect	CMPSS2 CTRIPOUTH	GPIO3
	ZVSDiagnosticsOutput	CLB2 OUT5	GPIO29
INPUTXBAR	Peakdetect_inputxbar5	GPIO2	XBAR_INPUT5
CLBXBAR	PeakTrip_CLBXBAR	CMPSS1 CTRIPH	AUXSIG0
	ZCDTrip_CLBXBAR	CMPSS2 CTRIPH	AUXSIG1
	ZVS_Threshold	CMPSS2 CTRIPL	AUXSIG2
CLBOUTPUT XBAR	T1_Event	CLB1 OUT4	GPIO22
	T2_EVENT	CLB1 OUT5	GPIO23
	ZCD_Count_MatchEvent	CLB1 OUT3	GPIO10

図 3-20. XBAR の構成

このアプリケーションでは、XBAR は図のように構成されており、EPWM、CLB、および外部インターフェイス間の信号レベルの調整が可能になります。

#### GPIO の初期化:

図のように、GPIO の 1 つはアプリケーションからの出力信号として構成され、同期整流 (SR) コントローラへの入力として使用できます。この信号により、メイン コンバータと SR 段の間の調整が容易になり、効率的な同期動作が可能になります。

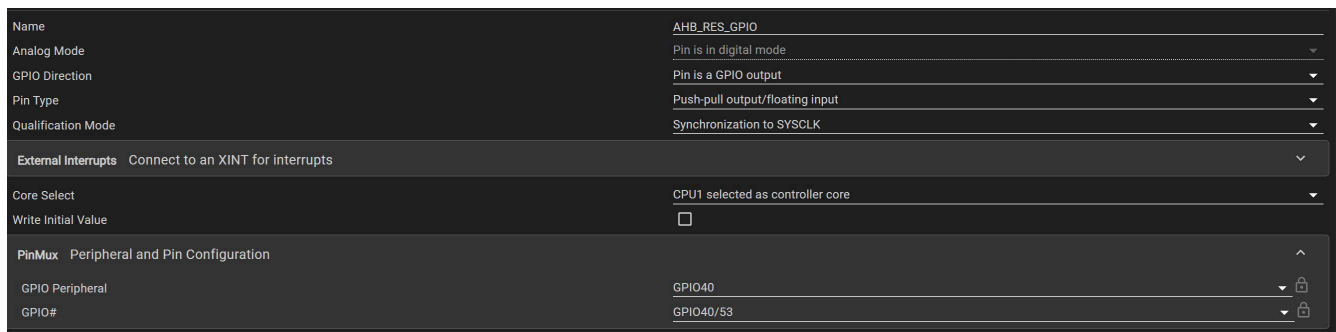


図 3-21. GPIO の構成

### 3.1.2 割り込み構造

#### 割り込み構造

このプロジェクトは、リアルタイム制御、PWM 更新、およびシステム メンテナンス タスクを分割する 2 つの主要な割り込みサービス ルーチン (ISR) を使用して動作します。

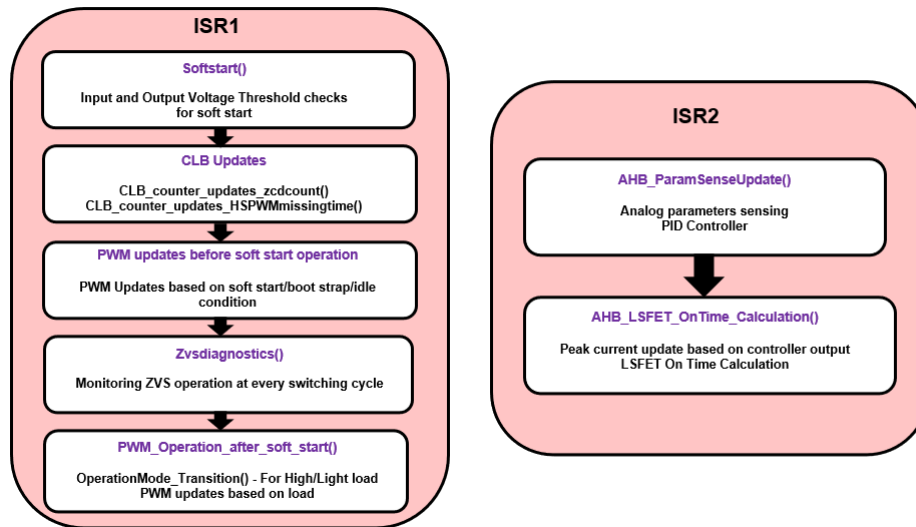


図 3-22. 割り込みフロー図

図に示すように、このデュアル割り込み構造により、高いスイッチング周波数で確定的な PWM 動作と安定した閉ループ制御が可能になります。

### 1. ISR1 — 可変周波数 ISR

- 関数名: INT\_AHB\_EPWM\_ISR()
- トリガ ソース: 時間ベース カウンタ (TBCTR) がゼロになったときの AHB EPWM モジュール。
- 動作周波数: 可変。コンバータのスイッチング周波数と同期、20kHz ~ 200kHz の範囲内。
- 機能:
  - すべての PWM 更新アクティビティと相補ゲート信号の同期を処理します。
  - ゼロ電圧スイッチング (ZVS) 動作を実現するために、ハイサイドとローサイドの遷移の高精度タイミングを保証します。
  - 最新の制御ループ計算から得られた変調パラメータを更新します。
  - 最小限のレイテンシで動作し、可変周波数全体で一貫性のあるスイッチング動作を維持します。
  - この ISR はタイムクリティカルであり、高い動作周波数であっても確定的な PWM 生成が保証されるように厳格に最適化されています。

### 2. ISR2 – 固定周波数制御とハウスキーピング ISR

- 関数名: INT\_Timer1\_ISR()
- トリガ ソース: CPU タイマ割り込み。タイマ オーバーフロー イベントによって生成。
- 動作周波数: 50kHz に固定。コンバータのスイッチング周波数に依存しない。
- 機能:
  - メイン デジタル制御ループを実行し、電圧と電流のフィードバック処理を実行します。
  - スwitching ノイズを抑制する目的で、測定した信号にデジタル フィルタリングと移動平均を適用します。
  - コマンド指定の電圧リファレンスに基づいて、スイッチのオン時間とデューティ サイクルを計算します。
  - スルーレート制限を実行して、スムーズなリファレンス遷移とソフトスタート処理を行います。
  - ISR1 はコンバータのスイッチング周波数に動的に適応し、ISR2 は固定実行レートを維持することで制御アルゴリズムに一貫した計算ウィンドウを提供します。

この割り込み階層は、効率的でスケーラブルなアプローチを提供します。ISR1 は高速のハードウェア同期イベントを処理し、ISR2 は制御計算とバックグラウンド タスクを管理し、高精度なタイミング、安定した動作、および CPU 負荷の低減を保証します。

### 3.2 PowerSuite の使用方法

PMP41140 アプリケーション ファームウェアには、PMP41140.syscfg ファイルが付属しています。このファイルは、Code Composer Studio™ (CCS) の SysConfig エディタで開くことができます。SysConfig インターフェイスには、専用の PowerSUITE カテゴリが用意されており、ファームウェアの構成とカスタマイズを簡素化できます。

PowerSUITE インターフェイスにより、ユーザーはファームウェア ソース ファイルを手動で変更するのではなく、グラフィカル ユーザー インターフェイスで主要なアプリケーション パラメータを変更できます。この結果、評価を迅速に行い、派生するハードウェア設計のカスタマイズや制御パラメータのチューニングを簡単に行うことができます。

The screenshot displays the PowerSuite application interface for the PMP41140. The main window shows a circuit diagram titled "Asymmetric Half Bridge DC/DC Converter". The diagram includes an LMG2650 controller, an input capacitor  $C_{in}$ , a transformer with inductance  $L_r$  and turns ratio  $N_p : N_A : N_s$ , a resistor  $R$ , a capacitor  $C_r$ , and an output capacitor  $C_{out}$ . It also shows sense resistors  $R_1$  and  $R_2$  for zero-current detection (ZCD) and current sensing (IO\_SENSE). The output voltage is sensed by a TLV9062 (VO\_SENSE). The application is configured for a TMS320F28P550SG processor.

Output specifications listed in the diagram:

- 9V @ 12A
- 15V @ 12A
- 20V @ 12A
- 28V @ 10A

Below the diagram, a configuration panel is visible with the following sections:

- Lab Selection:** Selects the firmware build · Automatically updates all lab-dependent parameters on selection
- Active Lab:** Lab 4 · Closed-Loop Voltage Regulation – Adaptive Soft-Start & ZVS
- Power Stage Hardware:** Board-specific hardware constants – update when customizing the power stage for your design
- Output Voltage:** Target output voltage and programmable range · Configure for USB PD profiles (5 V · 9 V · 12 V · 15 V · 20 V) or fixed output
- Protection:** Input supply qualification and output fault thresholds – safeguards the converter and connected load
- Startup & Power Delivery:** Soft-start sequence and peak current limits – controls inrush behaviour and maximum output power capability
- Switching & Timing:** LS FET on-time control · ZVS valley currents · Lab-dependent values auto-set on lab selection
- Voltage Controller:** Outer voltage loop PI controller · Tustin (trapezoidal) discretization with back-calculation anti-windup · Visible in Lab 4 only

図 3-23. PMP41140 PowerSuite (アプリケーション UI)

使用可能な構成カテゴリは次のとおりです。

ラボの選択

ファームウェアは 4 つのラボ構成をサポートしており、デフォルト オプションとして **Lab 4** が選択されています。ラボ選択カテゴリでは、目的の動作モードまたは評価目的に応じて、使用可能なラボを切り替えることができます。各ラボの詳細については、このドキュメントの以降のセクションで説明します。

## 電力段ハードウェア

電力段ハードウェア カテゴリには、物理的なパワー コンバータの設計に関連するパラメータが含まれています。

- トランス構成により、ファームウェアをカスタム ハードウェア設計に適合させるときに、トランスの巻線比を更新できます。
- 電気的パラメータを利用すると、励磁インダクタンス、共振インダクタンス、共振容量など、電力段の主な部品をカスタマイズできます。
- センシング キャリブレーションは、ADC の測定値を実際の電流量に変換するために使用するスケーリング構成を提供します。ユーザーは、電力基板に実装されている検出回路に基づいて、測定可能な最大電圧および電流範囲を定義できます。

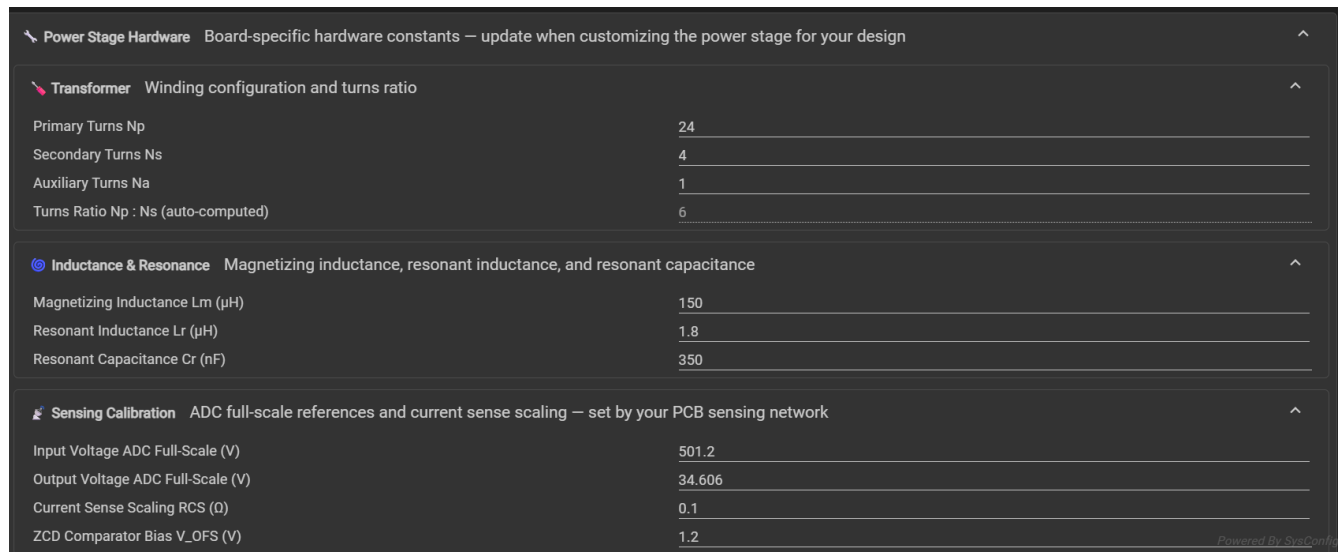


図 3-24. 電力段ハードウェア

## 出力電圧と保護

このカテゴリでは、出力電圧レギュレーションとフォルト保護の構成オプションを提供します。

- ターゲット出力電圧と、許容される出力電圧範囲を構成できます。PMP41140 リファレンス デザインでは、サポートされている出力電圧範囲を **9V ~ 28V**、デフォルトのターゲット電圧を **28V** に構成しています。
- 保護設定には次のものが含まれます。
  - 開ループ動作と閉ループ動作の両方の入力低電圧スレッショルド。

- 異常動作条件時にコンバータと負荷を保護するために使用される出力過電圧保護スレッショルド。

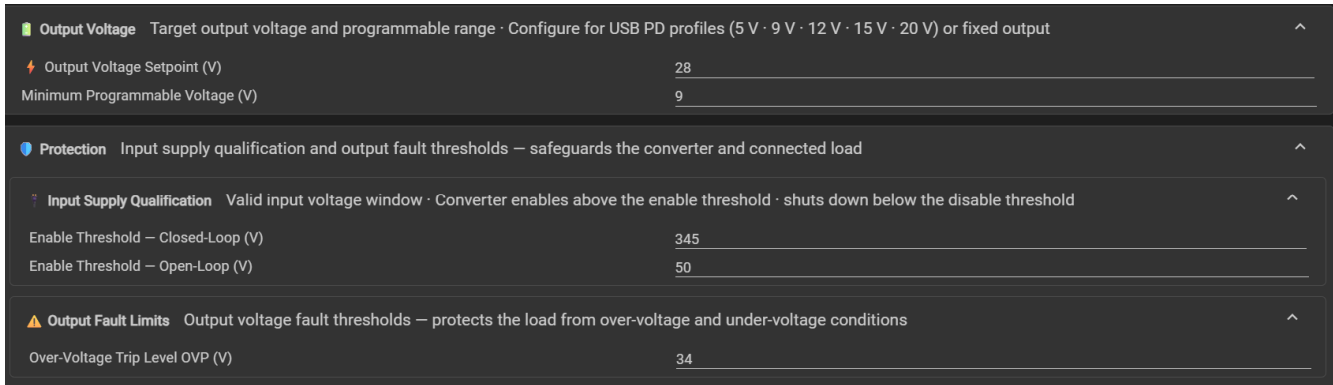


図 3-25. 出力電圧と保護

### 起動、スイッチングおよびタイミング

このカテゴリには、起動動作とスイッチング動作に影響するパラメータが含まれています。

- 適応型スタートアップは、アプリケーションの要件に応じて有効または無効にできます。
- ローサイド スイッチのタイミング動作を設定するために、LSFET オン時間制御パラメータが用意されています。
- ZVS バレー電流ターゲット設定を使用すると、ゼロ電圧スイッチング (ZVS) を実現するために使用されるターゲット電流を調整できます。
- ZVS 制御アルゴリズムで使用される ZVS 補正インダクタンスの制限、診断比、適応型補正境界には、その他の設定を利用できます。

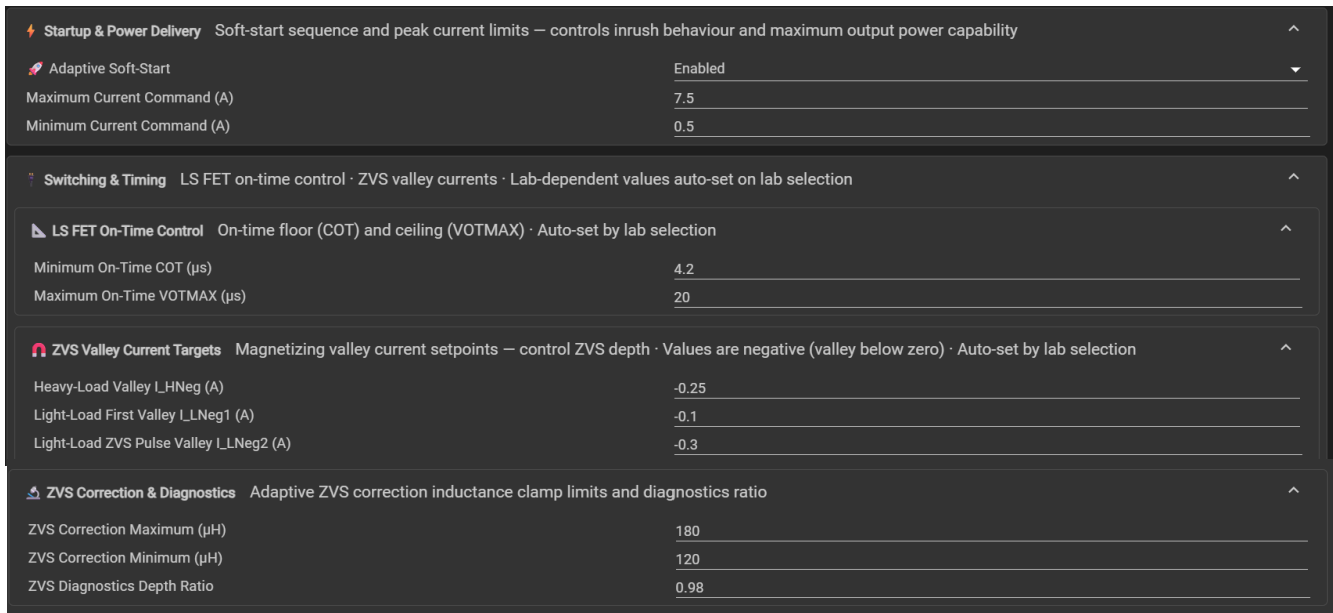


図 3-26. 起動、スイッチングおよびタイミング

### 電圧制御

電圧コントローラ カテゴリには、デジタル電圧ループ補償パラメータが含まれています。

- 目的の制御ループ応答を実現するために、PI コントローラの係数とアンチwind アップパラメータを設定できます。

- サンプルング周期も調整可能です。この値は、電圧制御ループが実行するタイム割り込み周波数に対応し、制御ループの帯域幅と動的応答に直接影響します。

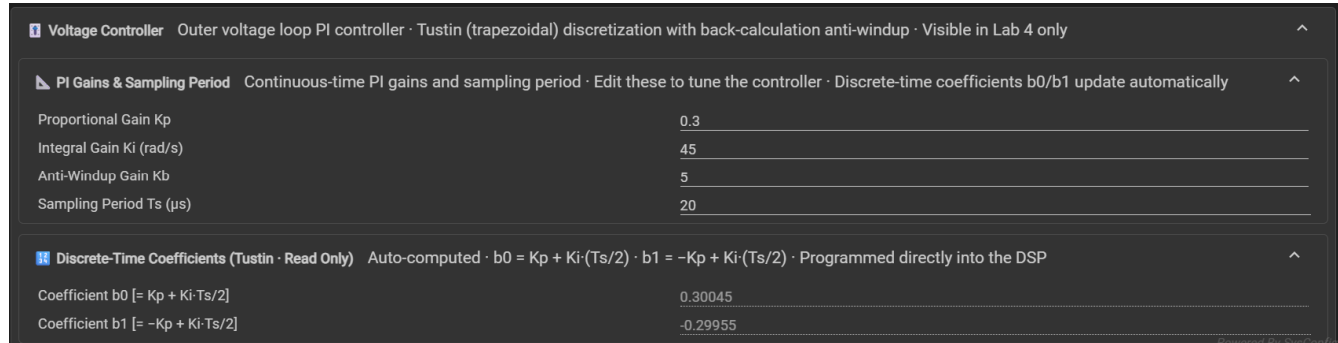


図 3-27. アンチwindアップ付き PI コントローラ

PowerSUITE インターフェイスは集中型の構成環境を実現し、ユーザーは PMP41140 ファームウェアをさまざまなハードウェア実装に適応させ、ソースコードへの直接の変更を最小限に抑えることができます。

## 4 ラボの構造

プロジェクトをビルドするには、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。

プロジェクトをロードするには、まず Project Explorer の targetConfigs (\*.ccxml ファイル) の下で正しいターゲット構成ファイルがアクティブに設定され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。

システムをデバッグするには、[Watch]/[Expressions] ウィンドウで変数を監視します。このウィンドウに正しい変数を入力するには、[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\_lab.js script 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] ボタンをオンにして、コントローラからの値が連続的に更新されるようにします。

このセクションのすべてのラボは、TMS320F28P550 制御カードを使用して実行できます。

アプリケーション パラメータを変更したり、別のラボに切り替える場合は、AHB\_user\_settings.h ファイルを直接更新する必要があります。特に、AHB\_LAB マクロによって、どのラボがアクティブになるかが決まります (デフォルトでは、Lab4 に対応する 4 に設定されています)。

このアプリケーションは、以下に説明する 4 つの構造化ラボで構成されています。

- Lab1: 重負荷 PWM 動作、アナログ信号センシング、フォルト保護に対する制御カードの検証。
- Lab2: 開ループ重負荷動作。
- Lab3: 開ループ軽負荷動作。
- Lab4: 負荷遷移による閉ループ動作、ZVS 診断、適応型ソフトスタート。

### 4.1 ハードウェア設定

ファームウェア検証および閉ループ テストの前に、このセクションで説明する推奨設定に従ってハードウェア プラットフォームを組み立て、接続する必要があります。

この手順の目的:

1. 制御カードと電源ボード間の通信を確立します。
2. 電力段が正常に動作していることを確認します。
3. コンバータの重要な信号を安全な方法で監視できるようにします。
4. ラボ試験中の測定ノイズを最小限に抑えます。

#### ハードウェア接続

入力電力の印加前:

1. TMDSCNCD28P55x コントローラ カードを PMP41140 インターフェイス コネクタに接続します。
2. 向きが正しく、コネクタが装着されていることを確認します。
3. 入力電源をコンバータの入力端子に接続します。
4. 電子負荷または抵抗性負荷を出力端子に接続します。
5. デバッガ インターフェイスを接続して、ファームウェアの読み込みとリアルタイム監視を行います。

次の図に推奨設定を示します。

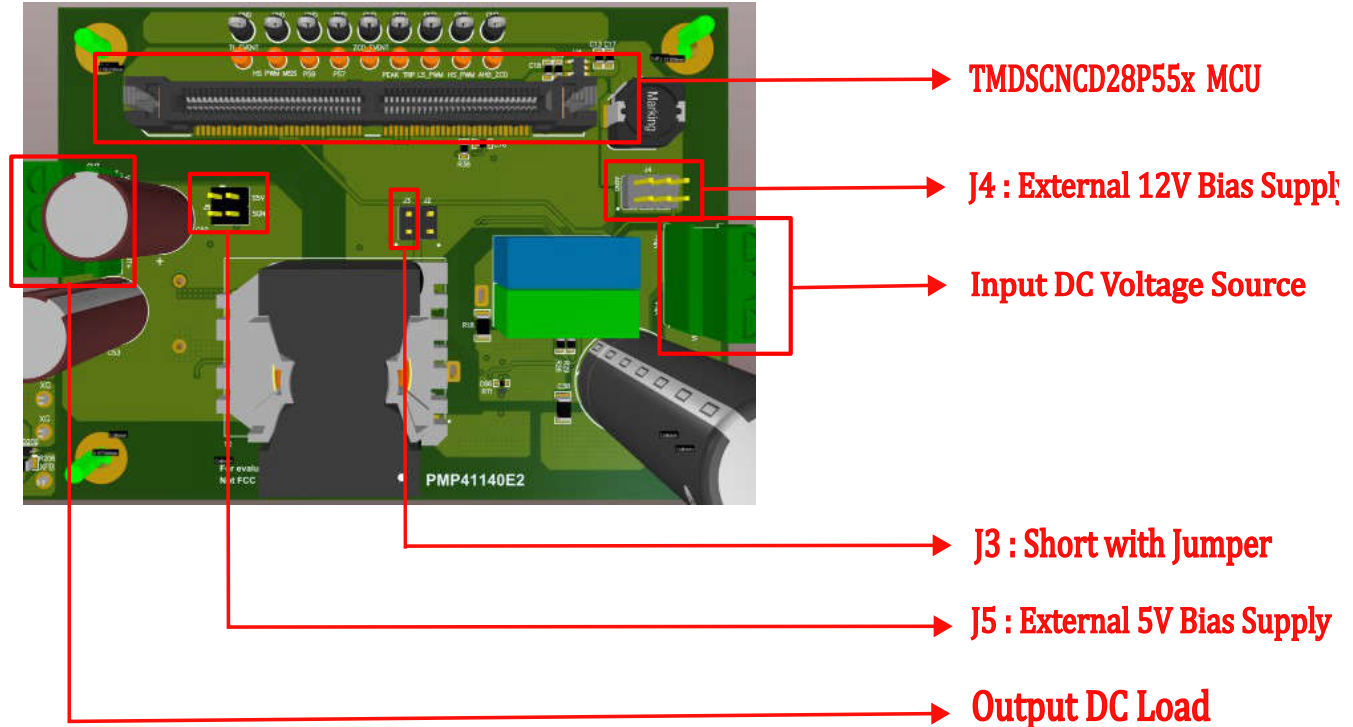


図 4-1. AHB ハードウェア ボードのセットアップ

#### 必要な機器:

1. TMDSCNCD28P55x 制御カード
2. DC 電源: 420V DC の範囲内で動作可能な入力電圧源
3. 電子負荷: 抵抗性負荷または定電流負荷として動作可能な負荷
4. 信号を監視するためのオシロスコープ

**安全上の警告:** 高電圧により感電の危険性があるため、基板に電源が投入されている間は、基板や電気回路に触れないでください。基板を取り扱う前に、高電圧が完全に放電されていることを確認してください。

#### テスト設定

ハードウェア接続の完了後:

1. Code Composer Studio (CCS) を起動します。
2. ファームウェア プロジェクトをインポートし、PMP41140.syscfg powersuite で必要なラボを選択します。
3. プロジェクトをビルドし、ターゲット マイコンに接続して、デバイスのフラッシュまたは RAM をプログラムします (プロジェクト プロパティからビルド構成を選択します)。
4. ウォッチ ウィンドウで正常に実行されたことを確認します。

#### 主なモニタリング変数:

ハードウェア検証中に、次の表に示す変数を監視する必要があります。

表 4-1. ランタイム変数

バリエーション	説明
AHB_VprimSensed_volts	入力電圧のリアルタイム センシング
AHB_aux_ZCD_N_volts	オフセット電圧のリアルタイム センシング。約 1.2V にする必要があります。

表 4-1. ランタイム変数 (続き)

バリエーション	説明
Vout_Sensed_Volts	出力電圧のリアルタイム センシング
AHB_VsecRef_volts	ターゲット出力電圧コマンド (9 ~ 28V)
Kp	制御ループ係数
Ki	制御ループ係数
Kd	制御ループ係数
Kb	制御ループ係数
AHB_MAG_INDUCTANCE_SET_uH	初期励磁インダクタンス設定値
ZVS_correction_factor	適応型補正済み励磁インダクタンス値
AHB_ZCDP_Compare	適応型 ZVS 検出スレッシュホールド
ILm_PeakCurLimit_Amps	リアルタイム更新の 1 次側ピーク電流
Adaptive_ILNEG_AMPS	重負荷状況での適応型ターゲット負電流
Adaptive_ILNEG1_AMPS	軽負荷状況での適応型ターゲット負電流 1
Adaptive_ILNEG2_AMPS	軽負荷状況での適応型ターゲット負電流 2
LS_ONTIME_HL_us	重負荷条件での LS オン時間のリアルタイム計算
LS_ONTIME1_LL_us	軽負荷条件での LS オン時間 1 のリアルタイム計算
LS_ONTIME2_LL_us	軽負荷条件での LS オン時間 2 のリアルタイム計算
light_load_enable	軽負荷モード イネーブル フラグ。1 = イネーブル、0 = ディスエーブル
zcd_count_update_new	軽負荷状態での適応型ゼロ交差カウント数

AHB\_VsecRef\_volts を、9V ~ 28V の範囲で予想される出力電圧に変更できます。

ハードウェア ボードには、ハイサイドおよびローサイド PWM 信号など、複数のテストポイントがあります。HS PWM、LS PWM、1 次側 LMG2650 のスイッチング ノード電圧、1 次側共振電流をプローブして、AHB ボードの動作を確認することができます。

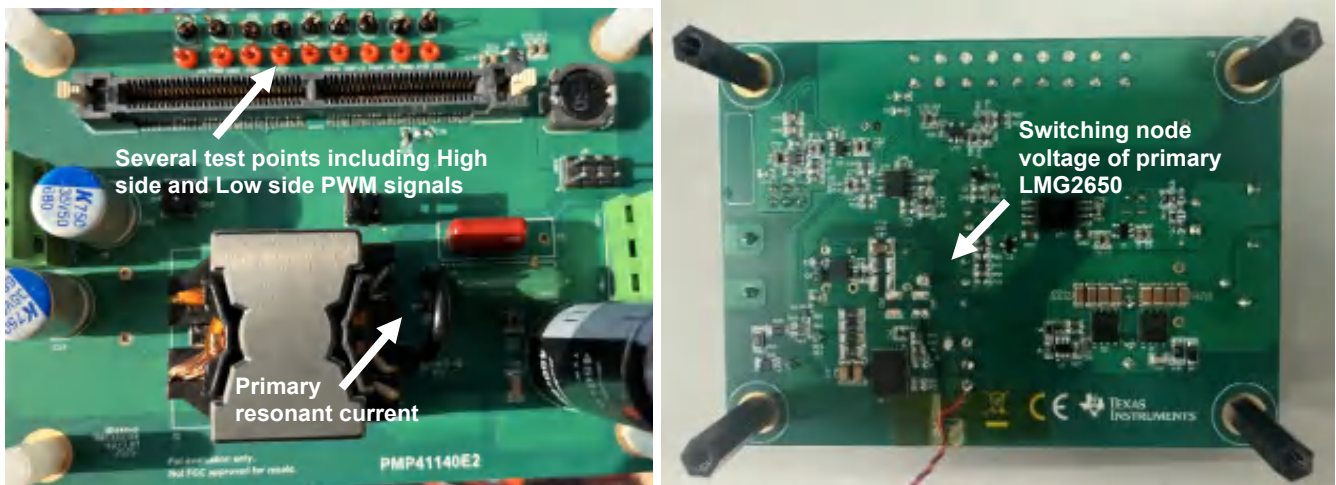


図 4-2. ハードウェア ボードのテストポイント

**推奨事項:** 外部ノイズを回避するために、上の図に示す最小外部テスト グランド ループ。

## 4.2 Lab1

このラボは、制御カードとドッキングステーションで実行することを意図としています。

AHB\_user\_settings.h ファイルのラボ番号を変更して、プロジェクトを **Lab 1** に設定します。他のすべての構成パラメータは、このステージのデフォルト値のままにします。

### 注

ほとんどのユーザーは、**Lab 2** に直接進み、機能テストフローを開始できます。

このラボの主な目的は、PWM 動作とアナログ信号センシングを検証することで、特に制御ループの精度に重要な入力および出力電圧チャンネルを検証することです。ユーザーは、ラボのステータスとフォルトステータス変数を観測して、システムが適切に動作するようにする必要があります。重要なチェックポイントは、対応するフォルトピンの過電流または過電圧イベントがフォルトステータスと PWM 波形に正しく反映されているかどうかを確認することです。

### 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 下で適切なターゲット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. 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。コードはメインルーチンの開始で停止します。
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。[Expressions] ウィンドウで右クリックしてインポートオプションを選択し、ガイドに従ってワークスペースにプロジェクトフォルダをインポートして、AHB\_Lab1Expressions.txt を開きます。このファイルにより、[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。図に示すように [Watch] ウィンドウが表示されます。

### コードの実行

1. CCS デバッグウィンドウで [Run] ボタンをクリックして、プロジェクトを実行します。
2. ウォッチビューで、PWMISR\_count と TimerISR\_count が連続的にインクリメントしていることを確認します。
3. LabStatus.type に「OffBoard\_Lab\_Sensing」と表示されていることを確認します。
4. オシロスコープの波形に示されているように、予測される PWM 波形を観測するために、AHB\_PWM\_HS と AHB\_PWM\_LS (HSEC 50 および 52) を接続します。
5. LS\_ONTIME\_HL\_us が、PWM 波形で測定された LS-FET のオン時間と一致していることを確認します。
6. VBUS\_SENSE (HSEC 9 → AHB\_VprimSensed\_volts) をプローブして入力電圧検出を検証し、VO\_SENSE (HSEC 21 → AHB\_VsecSensed\_volts) をプローブして出力電圧検出を検証します。ピンがフローティングのときに検出されるデフォルト電圧を確認します (図を参照)。
7. AHB\_RESCS (HSEC 13) に 3.3V を印加して、過電流フォルトをシミュレートします。
8. VO\_SENSE (HSEC 21) に 3.3V を印加して、過電圧フォルトをシミュレートします。
9. 適用された条件に応じて、faultStatus.type が次に更新されることを確認します。
  - a. OVERCURRENT\_FAULT、
  - b. OVERVOLTAGE\_FAULT、または
  - c. OV\_OVC\_COMBINED\_FAULT
10. faultClear = 1 に設定すると、PWM フォルト動作がクリアされ、PWM 出力が再度イネーブルになります。

11. `faultStatusClear = 1` に設定すると、CMPSS フォルト ステータス フラグがクリアされます。

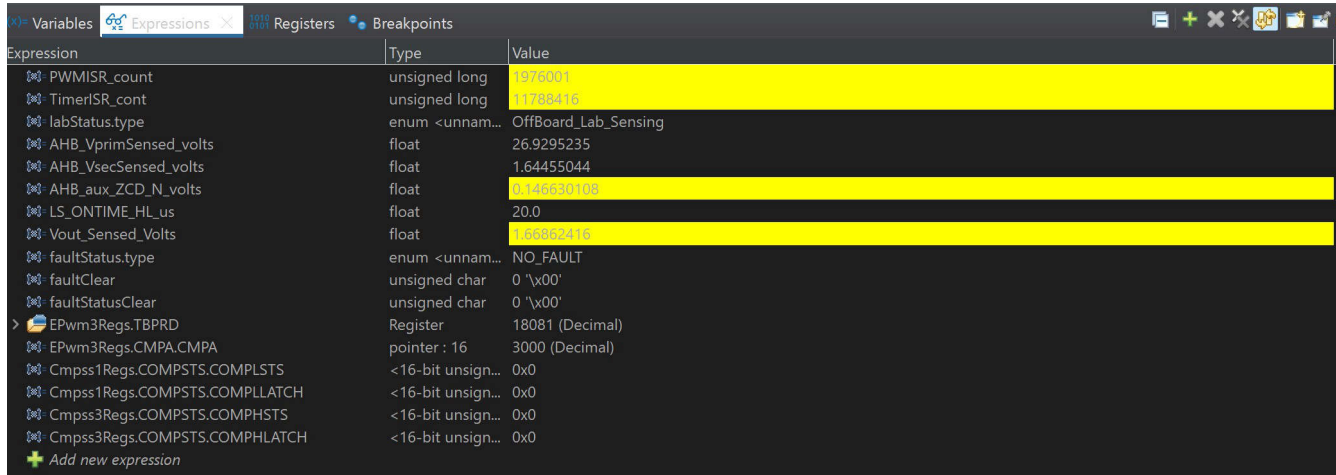


図 4-3. Lab1 の [Expressions] ウィンドウ

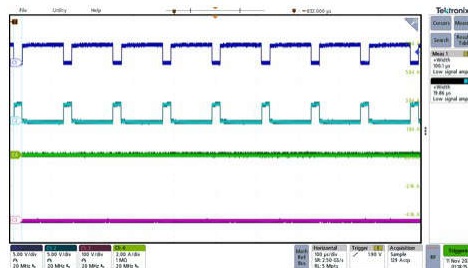


図 4-4. Lab1 の波形。Ch1 が HS PWM、Ch2 が LS PWM

### 4.3 Lab2

このラボはボード上で実行し、重負荷時の開ループ動作を検証することを目的としています。

`AHB_user_settings.h` ファイルのラボ番号を変更して、プロジェクトを **Lab 2** に設定します。他のすべての構成パラメータは、このステージのデフォルト値のままにします。

このラボの主な目的は、重負荷に基づいて PWM 波形を検証し、対応する変数を変更して `AHB_RESCS` のピーク電流を観測することです。

#### ハードウェア接続:

- 12V の外部バイアス電源をボードの J4 コネクタに、5V を J5 コネクタに接続します。ジャンパを使用して J3 を短絡します。
- 入力電流を 0.2A に制限して、入力電圧 60V の DC 電源を接続します。
- 150Ω の CR 負荷を接続します。


#### 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` 下で適切なターゲット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。
- 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。コードはメイン ルーチンの開始で停止します。

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。[Expressions] ウィンドウで右クリックしてインポート オプションを選択し、ガイドに従ってワークスペースにプロジェクト フォルダをインポートして、AHB\_Lab2/3Expressions.txt を開きます。このファイルにより、[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。図に示すように [Watch] ウィンドウが表示されます。

### コードの実行

1. CCS デバッグウィンドウで [Run] ボタンをクリックして、プロジェクトを実行します。
2. ウォッチ ビューで、PWMISR\_count と TimerISR\_count が連続的にインクリメントしていることを確認します。
3. LabStatus.type に「OpenLoop\_HighLoad」と表示されていることを確認します。
4. オシロスコープの波形に示されているように、予測される PWM 波形を観測するために、AHB\_PWM\_HS と AHB\_PWM\_LS (HSEC 50 および 52) を接続します。波形の Ch3 および Ch4 は、それぞれ AHB\_ZCD\_P および AHB\_RESCS です。
5. LS\_ONTIME\_HL\_us が、PWM 波形で測定された LS-FET のオン時間と一致していることを確認します。
6. VBUS\_SENSE (HSEC 9 → AHB\_VprimSensed\_volts) をプローブして入力電圧検出を検証し、VO\_SENSE (HSEC 21 → AHB\_VsecSensed\_volts) をプローブして出力電圧検出を検証します。
7. デフォルトの ILm\_PeakCurTemp = 1.0A の場合：
  - a. 1 次側電流ピーク (CH4/AHB\_RESCS)  $\approx +1.0A$ 、負のピーク  $\approx -0.5A$  (Adaptive\_IHNEG\_AMPS に一致) になることを確認します。
  - b. 出力電圧  $\approx 3.25V$  であることを確認します。
  - c. LS\_ONTIME\_HL\_us  $\approx 11.74 \mu s$  になっていること (LS FET の測定されたオン時間が一致していること) を確認します。
8. ILm\_PeakCurTemp を 1.1A に変更したとき：
  - a. 1 次側電流ピーク (CH4/AHB\_RESCS)  $\approx +1.1A$ 、負のピーク  $\approx -0.5A$  (Adaptive\_IHNEG\_AMPS に一致) になることを確認します。
  - b. 出力電圧  $\approx 3.89V$  であることを確認します。
  - c. LS\_ONTIME\_HL\_us  $\approx 10.65 \mu s$  になっていること (LS FET の測定されたオン時間が一致していること) を確認します。

9. [Expression] ウィンドウを観測して、すべての変数がリアルタイムで更新されることを確認し、のオシロスコープの波形と比較します。

Expression	Value	Expression	Value
> ahb_opStatus	{type=AHB_SteadyState,desc...	> ahb_opStatus	{type=AHB_SteadyState,desc...
☞ AHB_aux_ZCD_N_volts	1.23226321	☞ AHB_aux_ZCD_N_volts	1.23347163
☞ AHB_VprimSensed_volts	60.1660271	☞ AHB_VprimSensed_volts	60.1256485
☞ AHB_VsecRef_volts	9.0	☞ AHB_VsecRef_volts	9.0
☞ Vout_Sensed_Volts	3.25785089	☞ Vout_Sensed_Volts	3.89781642
☞ AHB_MAG_INDUCTANCE_SET_uH	150.0	☞ AHB_MAG_INDUCTANCE_SET_uH	150.0
☞ ZVS_correction_factor	150.0	☞ ZVS_correction_factor	150.0
☞ light_load_enable	0 '\x00'	☞ light_load_enable	0 '\x00'
☞ zcd_count_update_new	5	☞ zcd_count_update_new	5
☞ ILm_PeakCurLimit_Amps	1.0	☞ ILm_PeakCurLimit_Amps	1.10000002
☞ ILm_PeakCurTemp	1.0	☞ ILm_PeakCurTemp	1.10000002
☞ LS_ONTIME_HL_us	11.7458363	☞ LS_ONTIME_HL_us	10.6534576
☞ LS_ONTIME1_LL_us	8.47186089	☞ LS_ONTIME1_LL_us	7.88961792
☞ LS_ONTIME2_LL_us	3.0	☞ LS_ONTIME2_LL_us	3.0
☞ Adaptive_IHNEG_AMPS	-0.518034279	☞ Adaptive_IHNEG_AMPS	-0.518018901
☞ Adaptive_ILNEG2_AMPS	-0.718032241	☞ Adaptive_ILNEG2_AMPS	-0.718022287
☞ Kp	0.300000012	☞ Kp	0.300000012
☞ Ki	45.0	☞ Ki	45.0
☞ Kd	0.0	☞ Kd	0.0
☞ Kb	5.0	☞ Kb	5.0

図 4-5. Lab2 の式

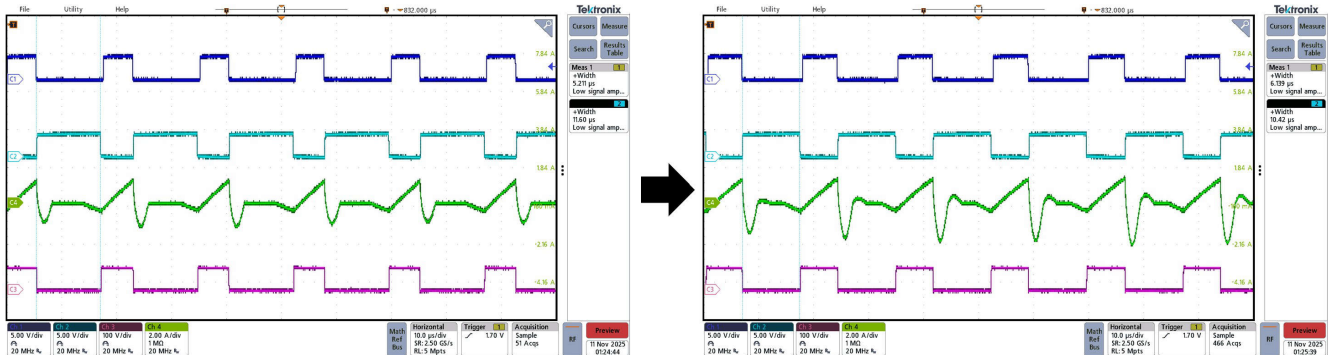


図 4-6. Lab2 の波形。Ch1 が HS PWM、Ch2 が LS PWM、Ch3 が補助ノード、Ch4 が 1 次電流

#### 4.4 Lab3

このラボはボード上で実行し、軽負荷時の開ループ動作を検証することを目的としています。

AHB\_user\_settings.h ファイルのラボ番号を変更して、プロジェクトを Lab 3 に設定します。他のすべての構成パラメータは、このステージのデフォルト値のままにします。

このラボの主な目的は、軽負荷に基づいて PWM 波形を検証し、対応する変数を変更して AHB\_RESCS のピーク電流を観測することです。

ハードウェア接続:

- 12V の外部バイアス電源をボードの J4 コネクタに、5V を J5 コネクタに接続します。ジャンパを使用して J3 を短絡します。
- 入力電流を 0.2A に制限して、入力電圧 60V の DC 電源を接続します。
- 150Ω の CR 負荷を接続します。

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 下で適切なターゲット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. 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。コードはメイン ルーチンの開始で停止します。
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。[Expressions] ウィンドウで右クリックしてインポート オプションを選択し、ガイドに従ってワークスペースにプロジェクト フォルダをインポートして、AHB\_Lab2/3Expressions.txt を開きます。このファイルにより、[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。図に示すように [Watch] ウィンドウが表示されます。

### コードの実行

1. CCS デバッグウィンドウで [Run] ボタンをクリックして、プロジェクトを実行します。
2. ウォッチ ビューで、PWMISR\_count と TimerISR\_count が連続的にインクリメントしていることを確認します。
3. LabStatus.type に「OpenLoop\_LightLoad」と表示されていることを確認します。
4. オシロスコープの波形に示されているように、予測される PWM 波形を観測するために、AHB\_PWM\_HS と AHB\_PWM\_LS (HSEC 50 および 52) を接続します。波形の Ch3 および Ch4 は、それぞれ AHB\_ZCD\_P および AHB\_RESCS です。
5. 開始後約 10 秒間は、オン時間が 10 $\mu$ s の LS PWM のみが観測され、その後 HS PWM が出力されます。
6. LS\_ONTIME\_HL\_us が、PWM 波形で測定された LS-FET のオン時間と一致していることを確認します。
7. VBUS\_SENSE (HSEC 9 → AHB\_VprimSensed\_volts) をプローブして入力電圧検出を検証し、VO\_SENSE (HSEC 21 → AHB\_VsecSensed\_volts) をプローブして出力電圧検出を検証します。
8. デフォルトの ILm\_PeakCurTemp = 1.0A の場合：
  - a. 1 次側電流ピーク (CH4/AHB\_RESCS)  $\approx +1.0A$ 、負のピーク  $\approx -0.71A$  (Adaptive\_ILNEG2\_AMPS に一致) になることを確認します。
  - b. 出力電圧  $\approx 3.1V$  であることを確認します。
  - c. LS\_ONTIME\_1LL\_us  $\approx 9.09\mu s$  と LS\_ONTIME2\_LL\_us  $\approx 3\mu s$  になっていること (LS FET の測定されたオン時間が一致していること) を確認します。
  - d. zcd\_count\_update\_new 変数が 3 であることを確認できます。補助ノード (SW 電圧) には 2 回のゼロ交差イベントが発生します。

9. [Expression] ウィンドウを確認し、すべての変数がリアルタイムで更新されることを確認します。図に示すように、これらの値をオシロスコープの波形と比較します。この図には、波形および [Expression] ウィンドウのスナップショット (HS PWM がアクティブになる前と後) が表示されています。

Expression	Value
> ahb_opStatus	(type=AHB_SteadyState,desc...
☺ AHB_aux_ZCD_N_volts	1.23508298
☺ AHB_VprimSensed_volts	60.246788
☺ AHB_VsecRef_volts	9.0
☺ Vout_Sensed_Volts	0.0591411106
☺ AHB_MAG_INDUCTANCE_SET_uH	150.0
☺ ZVS_correction_factor	150.0
☺ light_load_enable	1 "\x01"
☺ zcd_count_update_new	3
☺ ILm_PeakCurLimit_Amps	1.0
☺ ILm_PeakCurTemp	1.0
☺ LS_ONTIME_HL_us	20.0
☺ LS_ONTIME1_LL_us	10.0
☺ LS_ONTIME2_LL_us	3.0
☺ Adaptive_IHNEG_AMPS	-0.518047631
☺ Adaptive_ILNEG2_AMPS	-0.718050659
☺ Kp	0.300000012
☺ Ki	45.0
☺ Kd	0.0
☺ Kb	5.0

➔

Expression	Value
> ahb_opStatus	(type=AHB_SteadyState,desc...
☺ AHB_aux_ZCD_N_volts	1.23145747
☺ AHB_VprimSensed_volts	60.2871704
☺ AHB_VsecRef_volts	9.0
☺ Vout_Sensed_Volts	3.10022545
☺ AHB_MAG_INDUCTANCE_SET_uH	150.0
☺ ZVS_correction_factor	150.0
☺ light load enable	1 "\x01"
☺ zcd_count_update_new	3
☺ ILm_PeakCurLimit_Amps	1.0
☺ ILm_PeakCurTemp	1.0
☺ LS_ONTIME_HL us	12.5800095
☺ LS_ONTIME1_LL_us	9.09319401
☺ LS_ONTIME2_LL_us	3.0
☺ Adaptive_IHNEG_AMPS	-0.518040836
☺ Adaptive_ILNEG2_AMPS	-0.718044102
☺ Kp	0.300000012
☺ Ki	45.0
☺ Kd	0.0
☺ Kb	5.0

図 4-7. Lab3 の式

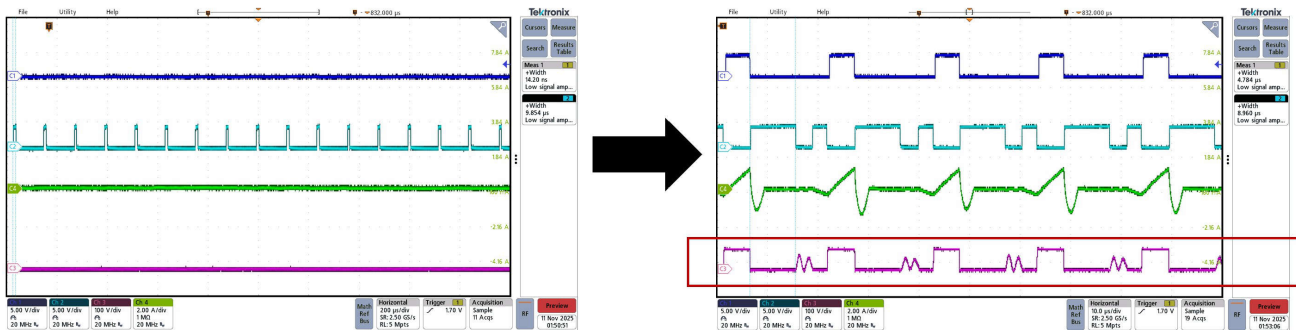


図 4-8. Lab3 の波形。Ch1 が HS PWM、Ch2 が LS PWM、Ch3 が補助ノード、Ch4 が 1 次電流

#### 4.5 Lab4

これはデフォルトのラボであり、完全なエンド ツー エンド ソリューションを検証するためにハードウェア上で実行することを意図しています。これには、幅広い入出力動作範囲にわたる閉ループレギュレーション、適応型ソフトスタート、高速負荷遷移応答、ZVS 診断機能が含まれ、信頼性の高い ZVS 動作を保証します。このラボを有効にするには、AHB\_user\_settings.h のラボ番号を更新して、プロジェクトを Lab 4 に設定します。他のすべての構成パラメータは、このステージのデフォルト値のままにします。

このラボの主な目的は、さまざまな負荷条件で PWM の動作を検証し、出力基準変数を調整して閉ループ電圧レギュレーションを観測することです。このラボでは、実用的な負荷および電圧シナリオでのコンバータのリアルタイム応答、安定性、保護動作を検証できます。

#### ハードウェア接続:

- 12V の外部バイアス電源をボードの J4 コネクタに、5V を J5 コネクタに接続します。ジャンパを使用して J3 を短絡します。


2. 接続する DC ソース入力電圧は、300 ~ 400V の範囲内にする必要があります。
3. 電流制限を 12A にして CC 負荷 (可能であれば電子負荷) を接続します。想定される出力電圧は 9 ~ 28V の範囲になります。

### 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 下で適切なターゲット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. 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。コードはメイン ルーチンの開始で停止します。
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。[Expressions] ウィンドウで右クリックしてインポート オプションを選択し、ガイドに従ってワークスペースにプロジェクト フォルダをインポートして、AHB\_Lab4Expressions.txt を開きます。このファイルにより、[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。図に示すように [Watch] ウィンドウが表示されます。

### コードの実行

1. CCS デバッグ ウィンドウの [Run] ボタンを使用してプロジェクトを実行します
2. ウォッチ ビューで、PWMISR\_count と TimerISR\_count が連続的にインクリメントしていることを確認します。
3. LabStatus.type に「Closedloop」と表示されていることを確認します。
4. DC 入力電源を接続し、320V 入力でボードに電源を投入します。デフォルトのリファレンス出力電圧は 9V です。
5. AHB\_VsecRef\_volts 変数を 9V ~ 28V の範囲内に変更し、高速で安定した閉ループ電圧レギュレーションを観測します。
6. 波形を表示して、リファレンス変数の変化に対応する出力電圧の動的調整を確認します。(Ch1: 出力電圧、Ch3: 補助ノード、Ch4: 1 次側電流)
7. 軽負荷で動作する場合、ファームウェアは出力電圧に基づいてターゲット ZCD 数を自動的に調整し、効率を向上させます。詳細な効率データはテストレポートに記載されています。

8. また、には、適応型ソフトスタート動作と低電圧ロックアウト (UVLO) 応答も示されています。

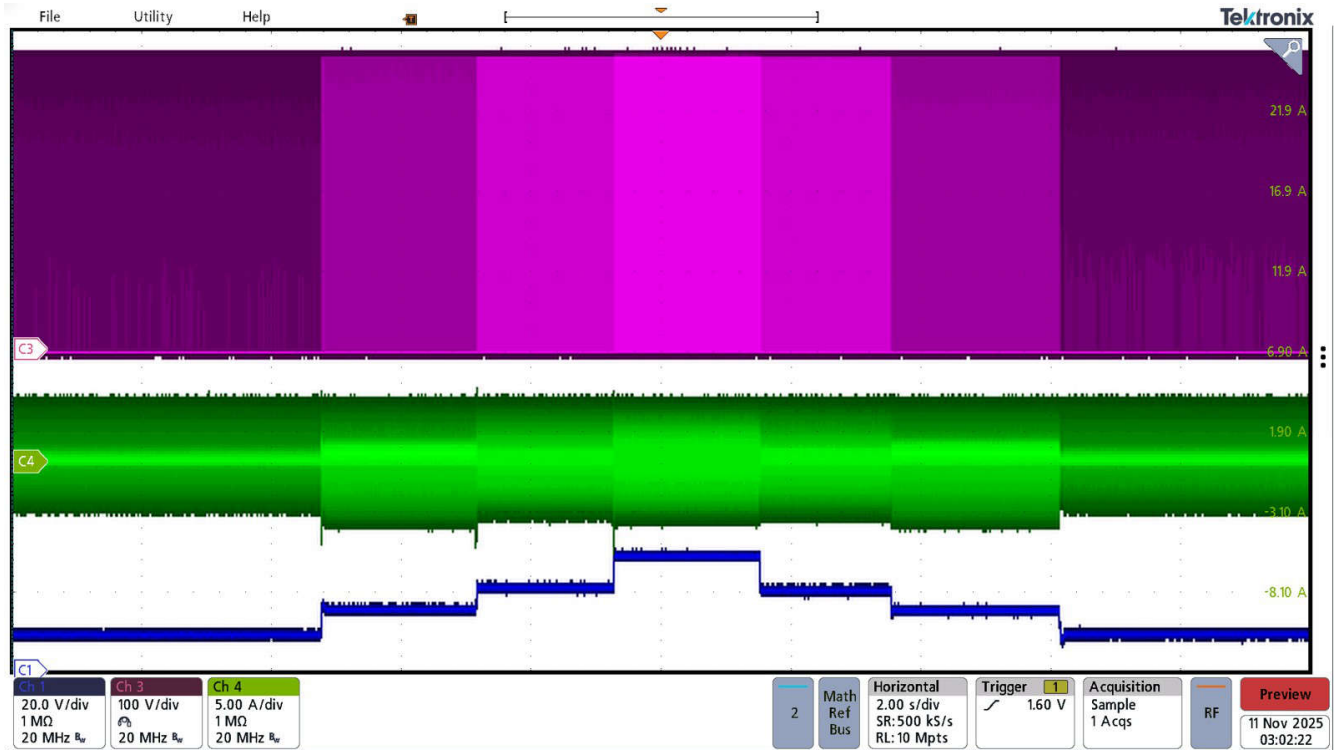
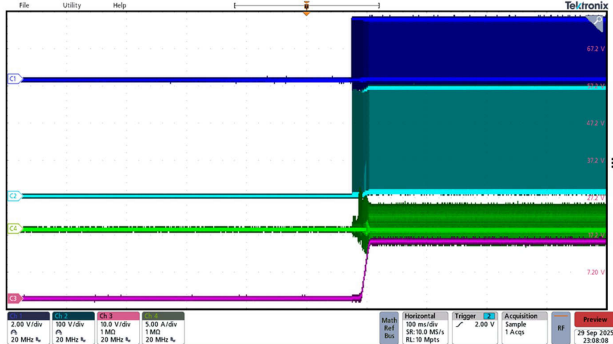
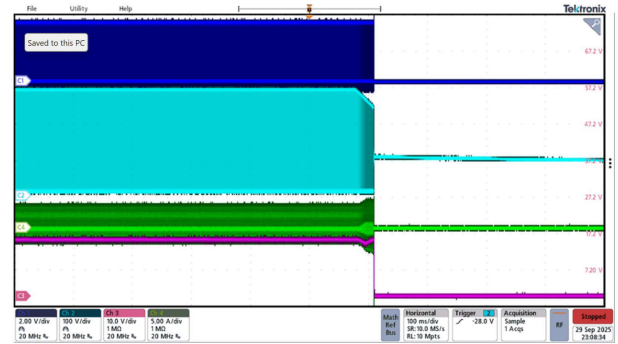


図 4-9. 出力電圧調整 (9V->15V->20V->28V->20V->15V->9V、370V 入力、9A CC 負荷)



a) Soft Start Operation



b) Under-Voltage Protection

図 4-10. ソフトスタートおよび低電圧保護動作、320V 入力、15V@9A CC 負荷条件

## 5 まとめ

このドキュメントでは、TMS320F28P550 マイコンに実装されている AHB コンバータリファレンス デザインの包括的な概要を説明します。全体的なソフトウェア構造、TI SysConfig を使用したシステム初期化フロー、ISR 実行モデル、クロスバの配線、信頼性の高いコンバータ動作を実現するための制御およびセンシング フレームワークの編成方法について説明します。このアーキテクチャは 2 つの ISR を使用します。1 つは PWM 更新のための可変周波数で動作し、もう 1 つは制御ループとハウスキーピング タスク用の固定周波数 ISR です。また、このドキュメントでは、EPWM および CLB モジュールを使用した負荷依存 PWM パターンの生成、保護イベントの管理、および軽負荷 T1/T2 イベント、ZVS 診断などのタイミング要件のサポートについても説明します。

このシステムは、センシングと基本的な PWM チェック、開ループでの重負荷、開ループでの軽負荷、負荷遷移と適応型ソフトスタートによる全閉ループ動作、の 4 つのラボから構成され、テストと検証の方法を説明します。各ラボでは、ハードウェアでシステムが正常に動作することを検証するために、セットアップ手順、予測される信号、観測結果の概要を説明します。さらに、効率の結果、ZVS 動作、ソフトスタート特性、負荷過渡応答、フォルト応答タイミングなどの追加の実験波形が [テストレポート](#) に示されており、文書化された実装との相関を確認できます。

## 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、

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