

1 システムの説明

3 相電源は、産業用アプリケーションにおいて大電力で動作する機器に使用されます。順方向負荷の多くが DC であるため、グリッドの電力品質を向上させ、消費される高調波電流を低減するには、力率補正が必要になります。たとえば、業務用エアコンや EV チャージャなどです。

アクティブな 3 相力率変換には多くのトポロジが存在しますが、Vienna 整流器は、連続導通モード (CCM) での動作、固有のマルチレベルスイッチング (3 レベル)、パワー デバイスへの電圧ストレスの低減により、広く採用されています。従来、Vienna 整流器は、ヒステリシス ベースのコントローラが使用されてきました。正弦波三角波ベースの PWM が Vienna 整流器制御に有効であることが示されたのは、最近のことです。この制御は、設計が非常に難しいものです。Vienna 整流器にはいくつかのバリエーションがあり、図 1-1 に、このデザインで選択されている Vienna 整流器のバリエーションと、検出対象の主要な電圧と電流を示します。

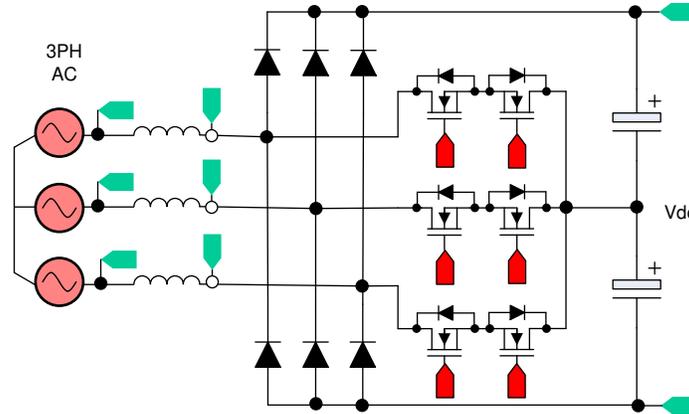


図 1-1. 実装されている Vienna 整流器のバリエーション

このデザイン ガイドでは、Y 接続 Vienna 整流器が実装されています。このデザインでは、Vienna 整流器の制御方法と、C2000 MCU を使用した異なるループの調整方法の例を示すことを目的としています。

1.1 用語

PWM	パルス幅変調
FET、MOSFET	金属酸化膜半導体電界効果トランジスタ
IGBT	絶縁型ゲート バイポーラトランジスタ
HVAC	暖房、換気、空調
RMS	実効値
PLL	位相ロック ループ

1.2 主なシステム仕様

表 1-1 に、3 相 Vienna 整流器の主な電源仕様を示します。

表 1-1. 主なシステム仕様

パラメータ	仕様
入力電圧 (V_{IN})	<ul style="list-style-type: none"> AC 208V_{RMS} V_{L-L} または 120V_{RMS} L-N、60Hz または AC 380V_{RMS} V_{L-L} または 220V_{RMS} L-N、50Hz
入力電流 (I_{IN})	<ul style="list-style-type: none"> 最大 16A RMS
出力電圧 (V_{OUT})	<ul style="list-style-type: none"> 650V DC バス (公称値)
出力電流 (I_{OUT})	<ul style="list-style-type: none"> 絶対 RMS 最大値 16A、パルス最大値 29A
定格電力	<ul style="list-style-type: none"> 3 相 380V_{RMS} で 10kW
電流 THD	<ul style="list-style-type: none"> 10kW 負荷で < 1.5% 2.5kW 負荷で < 5%
効率	ピーク 98%
1 次フィルタインダクタ	31A バイアス電流で 355 μ H
出力容量	940 μ F
PWM スイッチング周波数	40 kHz

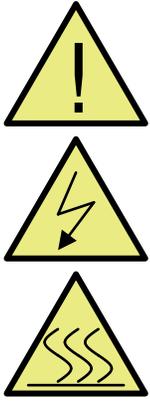


警告

テキサス・インスツルメンツは、このリファレンス デザインをラボ環境のみで使用するものとし、この基板を一般消費者向けの完成品とはみなしていません。

テキサス・インスツルメンツは、このリファレンス デザインを高電圧電気 機械部品、システム、およびサブシステムの取り扱いに伴うリスクを熟知した有資格のエンジニアおよび技術者のみが使用するものとしています。

基板上は高電圧状態になっており、接触するおそれがあります。基板は、不適切に取り扱ったり適用したりした場合に感電、火災、負傷の原因となる電圧および電流で動作します。負傷や物品の破損を避けるために、必要な注意と適切な対策をもって機器を使用してください。



注意

電源を入れたままその場を離れないでください。

高電圧！ 基板上は高電圧状態になっており、接触するおそれがあります。感電する可能性があります。基板は、不適切に取り扱った場合に感電、火災、負傷の原因となる電圧および電流で動作します。負傷や物品の破損を避けるために、必要な注意と適切な対策をもって機器を使用してください。安全のため、過電圧および過電流保護機能付きの絶縁された試験装置の使用を強く推奨します。

テキサス・インスツルメンツは、基板への電源投入やシミュレーション実行の前に、電圧要件および絶縁要件を確認し理解することがユーザーの責任であると考えます。電源投入中は、リファレンス デザインおよびその接続部品には触れないでください。

表面は高温！ 触れるとやけどの原因になることがあります。触れないでください！

基板の電源を入れると、一部の部品は > 55°C の高温に達することがあります。動作中は常に、また動作直後も高温の状態が続く可能性があるため、基板に触れてはいけません。

2 システム概要

2.1 ブロック図

図 2-1 に、このデザインで選択されている Vienna 整流器のブロック図と、検出対象の主要な電圧と電流を示します。

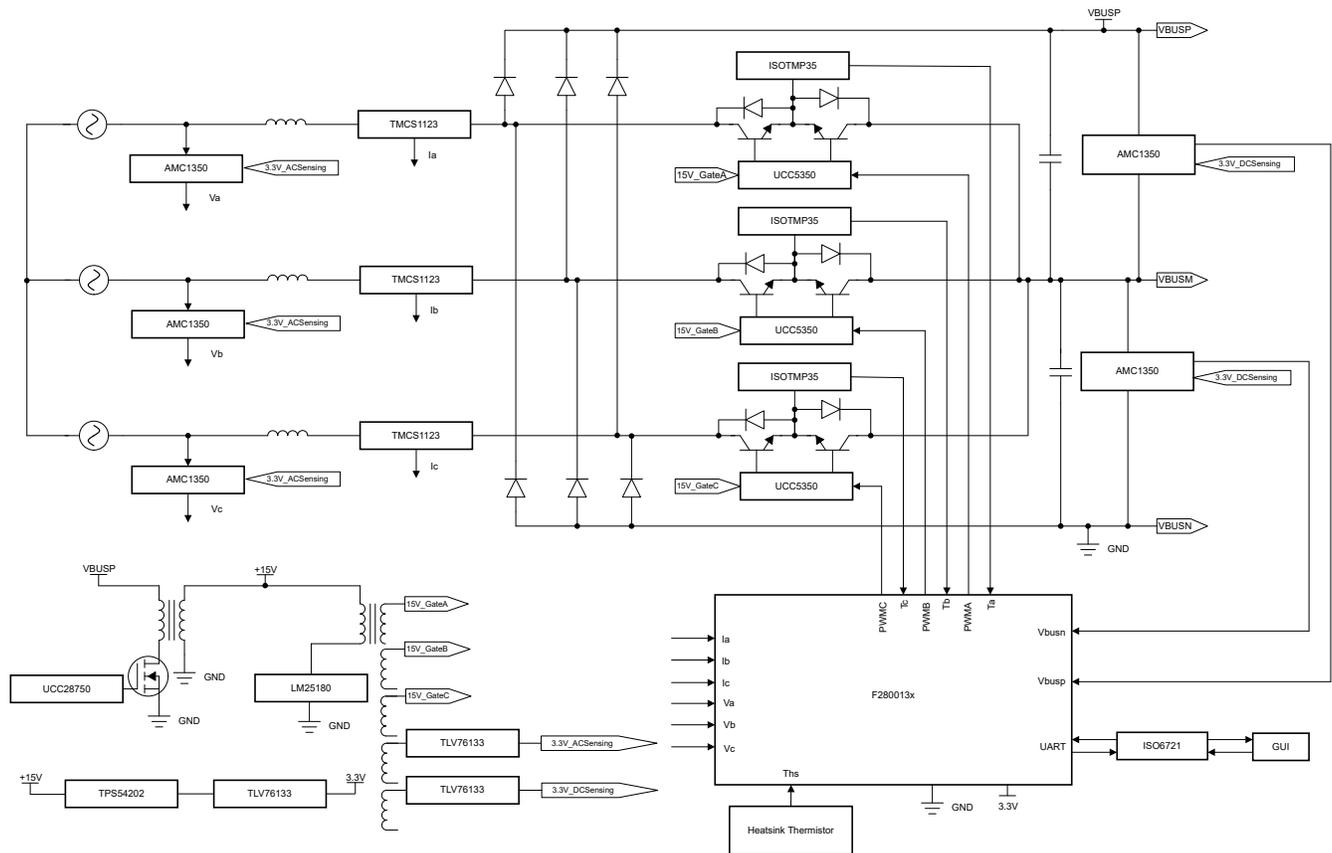


図 2-1. ブロック図

2.2 設計上の考慮事項

2.2.1 制御システム設計理論

このセクションでは、制御システムの設計理論について説明します。

2.2.1.1 PWM 変調

図 2-2 に、Vienna 整流器の単相概略図を示します。この整流器を制御するには、デューティサイクルを制御して、電圧 v_{xiN} を直接調整します。つまり、ソフトウェア変数 **Duty** が 1 に設定されている場合、 v_{xiN} は、Q1 と Q2 のスイッチをオンにせず、ブリッジダイオードを介してインダクタを DC バスに接続することで、可能な限り高い電圧になります。同様に、**Duty** が 0 に設定されている場合、Q1 と Q2 が常に導通するように PWM が変調され、 v_{xiN} は DC バスの midpoint (ゼロ) に接続され、スイッチングサイクルにおいて可能な限り低い電圧が生成されます。

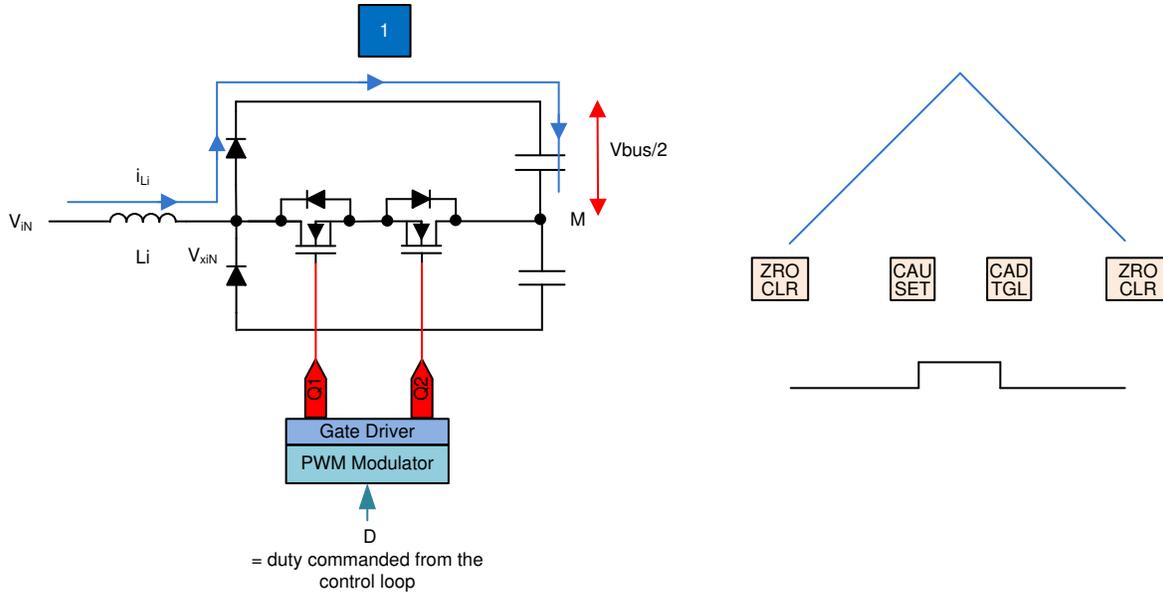


図 2-2. Vienna 整流器の単相概略図

図 2-3 に、PWM の詳細な構成を示します。

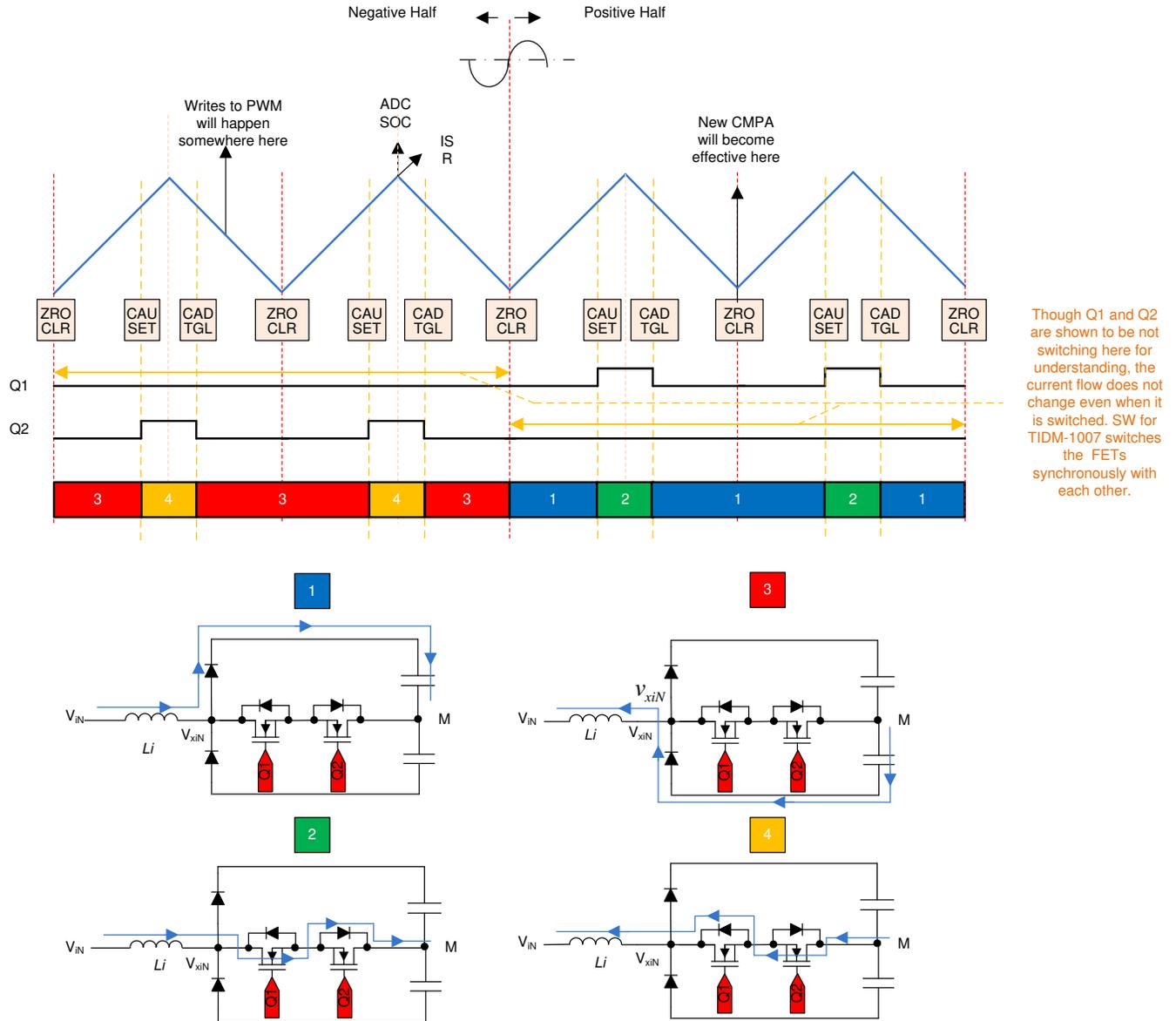


図 2-3. Vienna 整流器の詳細な PWM 変調方式

2.2.1.2 電流ループモデル

電流ループモデルを理解するには、まずインダクタ電流に注目します。図 2-4 では、スイッチ Q1 と Q2 に接続された PWM 変調器にデューティ サイクル D が設定されます。これを踏まえて、式 1 を参照してください。

$$V_{xiN} = D \times \frac{V_{bus}}{2} \tag{1}$$

注

D が 1 に設定されると、すべてのスイッチがオフになり、 D が 0 に設定されると、すべてのスイッチがオンになり、インダクタが M 点に接続されます。

インダクタを流れる電流を変調するために、スイッチ Q1 と Q2 のデューティ サイクル制御を用いて電圧 v_{xiN} が制御されます。電流の方向を AC ラインから整流器への方向に正と仮定し、DC バスのフィードフォワード、入力 AC 電圧のフィードフォワード、およびグリッドが十分に剛性であるという仮定を基にします。電流ループは図 2-4 に示すように簡略化でき、電流ループプラントモデルは式 2 のように記述できます。

$$H_{p_i} = \frac{i_{Li}^*}{D} = \frac{1}{K_{v_gain}} \times K_{i_gain} \times K_{i_fltr} \times G_d \times \frac{1}{Z_i} \tag{2}$$

ここで、

- K_{v_gain} は、バスと AC 入力で検出される最大電圧の逆数 $1/V_{max_sense}$ です。AC 電圧の最大検出値と DC バス電圧の最大検出値が等しいと想定されています。
- K_{i_gain} は検出された最大 AC 電流の逆数です。
- K_{i_fltr} は、電流センサから ADC ピンに接続された RC フィルタの応答です。
- G_d は、PWM 更新とデジタル制御に関連するデジタル遅延です。
- i_{Li}^* は電流指令で、 i_{Li} は実際のインダクタ電流です。
- $V_{bus}/2$ は、出力バスコンデンサの 1 つにかかる電圧です。
- Z_i はインダクタのインピーダンスで、インダクタンス L_i と抵抗 R_i が含まれます。
- H_{p_i} は、デジタルコントローラ G_i によって示される電流ループプラントです。
- v_{iN} は入力の瞬時 AC 電圧です。

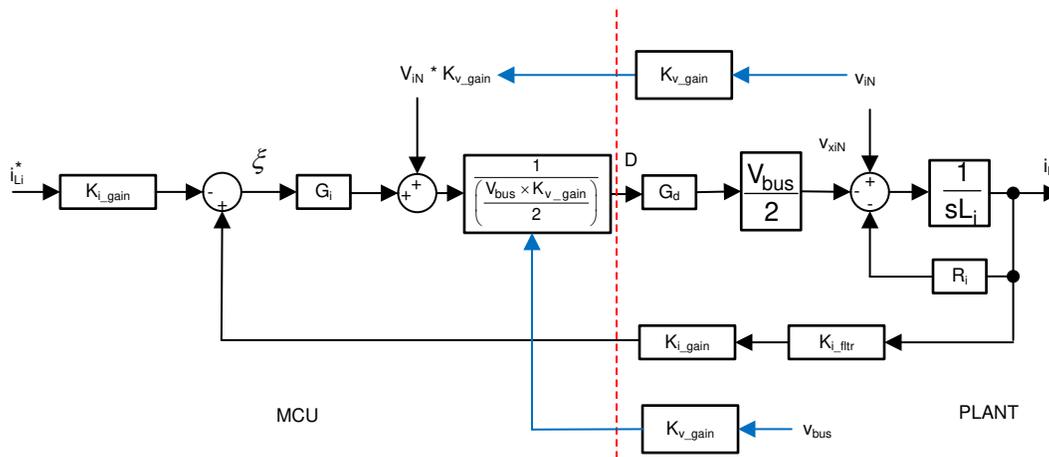


図 2-4. 電流ループ制御モデル

注

電流ループが電圧 v_{xiN} を制御していると考えられるため、基準に負記号が付けられています。電流を上げるには、 v_{xiN} を下げる必要があります。このため、図 2-4 の基準と帰還では記号が逆になっています。この電流ループモデルは、電流補償器の調整に使用されます。電流ループには、単純な比例コントローラが使用されます。比例ゲインのゲインは、システムが安定するように調整されます。

2.2.1.3 DC バス電圧制御ループ

DC バス電圧制御ループは、電力リファレンスを提供するものと仮定されています。このループを入力電圧 RMS の 2 乗で割って導電率を出し、さらに入力電圧を乗じて瞬時電流指令を求めます。

DC バス電圧制御ループの小信号モデルは、式 3 を動作点近傍で線形化して作成します。

$$i_{DC} v_{bus} = 3n v_{Nrms} i_{Nrms} \Rightarrow \hat{i}_{DC} = 3n \frac{\bar{V}_{Nrms}}{\bar{V}_{bus}} i_{Li} \quad (3)$$

抵抗性負荷の場合、バス電圧と電流は式 4 に示すような関係にあります。

$$\hat{V}_{bus} = \frac{R_L}{1 + sR_L C_o} \hat{i}_{DC} \quad (4)$$

DC バス電圧制御ループの制御モデルは、図 2-5 のように描くことができます。 V_{bus} フィードフォワードを追加して、制御ループをバス電圧に依存しないようにすると、バス制御のプラントモデルは式 5 のように表せます。

$$H_{p_bus} = H_{load} \times N \times K_{i_gain} \times K_{v_gain} \times K_{v_flt} \quad (5)$$

ここで、

- H_{p_bus} はデジタル コントローラ G_v によって示される電圧ループ プラントです。
- G_v の出力は、電力リファレンス P_o^* になります。
- v_{bus}^* は電圧コマンドと電圧リファレンス、 v_{bus} は実際のバス電圧です。
- C_o は出力コンデンサ、 R_L は負荷抵抗です。

図 2-5 を用いて、電圧ループ用に比例積分(PI)補償器を設計します。このループの帯域幅は、ループが定常状態で THD と衝突するため、低く抑えられています。

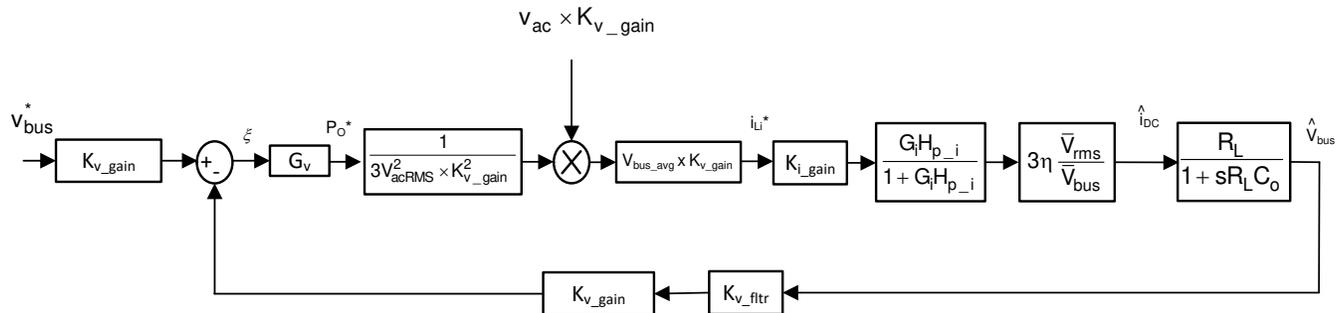


図 2-5. DC 電圧ループ制御モデル

さらに、負荷が段階的に変化する場合の過渡現象を低減するために、非線形 PI ループが使用されています。図 2-6 に、このデザインに実装されている非線形 PI ループの構造を示します。

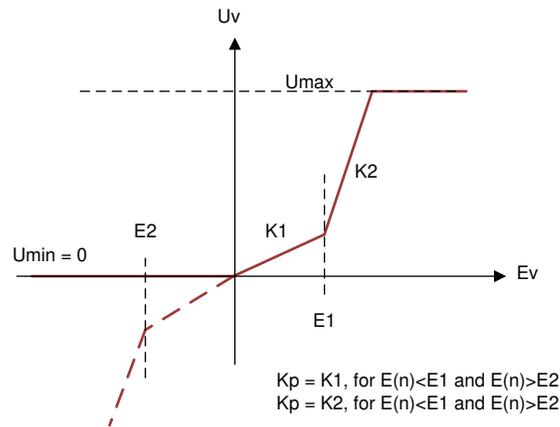


図 2-6. 電圧コントローラの非線形 PI ループ

2.2.1.4 DC 電圧のバランスコントローラ

Vienna 整流器の出力電圧バスには、分割コンデンサを使用しています。これらのコンデンサにかかる電圧は必ずしも自然にバランスが取れるわけではないため、DC バランスコントローラのループが追加されます。このループは、デューティサイクルにオフセットを加えて変調し、分割コンデンサの電圧を平衡させるために中点を流れる電流を調節します。

DC バス バランス コントローラには単純な比例ゲインが使用され、バランス ループの出力は式 6 のとおりです。

$$G_{s_out} = (V_{bus_PM} - V_{bus_MN}) \times G_{s_gain_Kp} \quad (6)$$

2.3 主な使用製品

2.3.1 TMS320F280013x

TMS320F280013x (F280013x) は、パワー エレクトロニクスの効率を高めるために設計されたスケーラブルな超低レイテンシ デバイスである C2000™ リアルタイム マイクロコントローラ ファミリの製品です。リアルタイム制御サブシステムは、テキサス・インスツルメンツの 32 ビット C28x DSP コアをベースにしており、オンチップ フラッシュまたは SRAM から実行される浮動小数点または固定小数点コードに対して 120 MHz の信号処理性能を発揮します。C28x CPU は、三角関数 算術演算ユニット (TMU) によってさらに強化され、リアルタイム制御システムでよく使われる重要なアルゴリズムを高速化します。F280013x は最大 256KB (128KW) のフラッシュ メモリをサポートしています。最大 36KB (18KW) のオンチップ SRAM も利用でき、フラッシュ メモリを補完できます。F280013x リアルタイム マイクロコントローラ (MCU) に内蔵された高性能アナログ ブロックは、クラス最高のリアルタイム シグナル チェーン性能を提供するために、処理ユニットや PWM ユニットと密結合されています。14 の PWM チャンネルにより、3 相インバータから力率補正とその他の先進マルチレベル電源トポロジまで、各種出力段を制御できます。

2.3.2 UCC5350

UCC53x0 は、MOSFET、IGBT、SiC MOSFET、GaN FET (UCC5350SBD) を駆動するように設計されたシングル チャネル絶縁型ゲートドライバのファミリーです。UCC53x0S には分割出力があり、立ち上がりと立ち下りの時間を別々に制御できます。UCC53x0M は、トランジスタのゲートを内部的なクランプへ接続することで、ミラー電流により誤って電源オンが発生することを防止します。UCC53x0E には、GND2 を基準とする UVLO2 があり、真の UVLO 読み取り値を得られます。UCC53x0 は、4mm SOIC-8 (D) または 8.5mm SOIC-8 (DWV) パッケージで供給され、それぞれ最大 3kV_{RMS} および 5kV_{RMS} の絶縁電圧をサポートできます。これらの各種のオプションを取りそろえた UCC53x0 ファミリーは、モータドライブや産業用電源に適しています。フォトカプラと比較して、UCC53x0 ファミリーは部品間スキューが小さく、伝搬遅延時間が短く、より高い温度でも動作し、CMTI が高いという特長があります。

2.3.3 AMC1350

AMC1350 は高精度の絶縁型アンプで、磁気干渉に対して高い耐性のある絶縁バリアにより、入力側と出力側の回路が分離されています。この絶縁バリアは、VDE V 0884-11 および UL1577 に従って最大 5kV_{RMS} の強化ガルバニック絶縁を達成していることが認証されており、最大 1.5kV_{RMS} の使用電圧に対応しています。この絶縁バリアは、異なる同相電圧レベルで動作するシステム領域を分離し、潜在的に危険な電圧と損傷から低電圧側を保護します。AMC1350 の高インピーダンス入力は、高インピーダンスの抵抗分圧器や、出力抵抗の高い他の電圧信号ソースと接続するよう最適化されています。優れた精度と低い温度ドリフトにより、拡張産業用温度範囲の -40 °C~+125°C にわたって、DC/DC コンバータ、周波数インバータ、AC モータ、サーボドライブ アプリケーションで高精度の AC および DC 電圧センシングをサポートします。

2.3.4 TMCS1123

TMCS1123 は、業界をリードする絶縁性と精度を備えたガルバニック絶縁ホール効果電流センサです。入力電流に比例する出力電圧により、優れた直線性と、あらゆる感度オプションで低ドリフトを実現しています。ドリフト補償を内蔵した高精度のシグナル コンディショニング回路は、システム レベルのキャリブレーションなしで、温度範囲と寿命全体にわたり 1.4% 未満の最大感度誤差を達成しており、室温キャリブレーションを 1 回のみ実行した場合には、寿命および温度ドリフトを考慮しても 1% 未満の最大感度誤差を達成します。

AC または DC 入力電流は内部導体を通して磁界を生成し、内蔵のオンチップ ホール効果センサにより測定します。コアレス構造のため、磁気コンセントレータは不要です。差動ホール センサは、外部の浮遊磁界による干渉を排除します。導体抵抗が小さいので、測定可能な電流範囲が最大 ±96A まで拡大すると同時に、電力損失を最小化し、放熱要件を緩和できます。5kV_{RMS} に耐える絶縁と、最小 8.1mm の沿面距離および空間距離により、最大 1.3kV_{DC} の信頼性の高い寿命の強化動作電圧を実現します。内蔵シールドにより、優れた同相除去と過渡耐性を実現しています。

固定感度とすることで、デバイスは 3V~5.5V の単一電源で動作でき、レシオメトリック誤差をなくし、電源ノイズ除去を向上させています。

2.3.5 UCC28750

UCC28750 は、フォトカプラを使用する、高性能、低スタンバイ電力、コスト効率の優れたオフライン フライバック コンバータ アプリケーション向けに最適化された、高集積電流モード、連続導通対応の PWM コントローラです。軽負荷の状況では周波数フォールドバックおよびバースト モードに移行し、軽負荷時の効率を向上させます。UCC28750 で使用されて

いるバーストモードアルゴリズムは、最小実効スイッチング周波数を制御して、軽負荷状況における可聴ノイズを防止します。周波数ディザリングにより EMI 性能が向上します。これは、通常動作、周波数フォールドバック動作、および電力昇圧動作でアクティブになります。

UCC28750 は保護機能を搭載しており、最小限の外付け部品で堅牢なコンバータ設計を実現できます。出力過電力保護 (OPP) とサイクル単位の過電流制限により、負荷段および電力段の部品を電氣的ストレスから保護します。過電圧および低電圧誤動作防止 (OVLO および UVLO) により、望ましくない入力状態でのスイッチングを防止します。FLT ピンは、デバイスバリエーションに応じて、ラインブラウナウトの検出および保護機能、または外部過熱および過電圧保護機能を備えています。FLT ピンは、デバイスバリエーションに関係なく、ピンをグランドにプルダウンして外部制御でデバイスをディセーブルにする目的にも使用されます。

2.3.6 LM25180

LM25180 は、4.5V~42V の幅広い入力電圧範囲にわたって、高い効率を実現する 1 次側レギュレーション (PSR) フライバックコンバータです。1 次側フライバック電圧から絶縁型出力電圧をサンプリングし、フォトカプラ、電圧リファレンス、トランスの第 3 (補助) 巻線をいづれも必要とせずに出力電圧の安定化を実現します。

高レベルの統合により、シンプルで信頼性が高く、絶縁バリアと交差する部品が 1 つのみの高密度な設計を実現しています。境界導通モード (BCM) スwitchingにより、小型の磁氣的設計と、 $\pm 1.5\%$ 以内の負荷およびラインレギュレーション性能を実現できます。内蔵の 65V パワー MOSFET は最大 7W の出力電力能力を持ち、ライン過渡に対する余裕が拡大されています。

2.3.7 ISOTMP35

テキサス・インスツルメンツの ISOTMP35 は、業界初の絶縁型温度センサ IC であり、最大 3000V_{RMS} の耐電圧の内蔵絶縁バリアと、 $-40^{\circ}\text{C}\sim 150^{\circ}\text{C}$ で 10mV/ $^{\circ}\text{C}$ の勾配を特長とするアナログ温度センサを組み合わせています。この統合により、高価な絶縁回路を必要とせずに、センサを高電圧熱源 (たとえば HV FET、IGBT、HV コンタクト) と同じ場所に設置することができます。また、高電圧熱源に直接接触することで、絶縁要件を満たすためにセンサを遠くに配置するアプローチに比べ、より高い精度と高速な熱応答が得られます。ISOTMP35 は非絶縁型の 2.3V~5.5V 電源で動作するため、高電圧プレーンでサブレギュレートされた電源を利用できないアプリケーションに簡単に統合できます。内蔵絶縁バリアは UL 1577 の要件を満たしています。表面実装パッケージ (7 ピン SOIC) は、熱源から組み込み熱センサへの優れた熱流を提供し、熱質量を最小限に抑え、より正確な熱源測定を実現します。これにより、時間のかかる熱モデリングの必要性が減り、製造や組み立てによる機械的なばらつきが減少するため、システム設計のマージンが向上します。ISOTMP35 の Class-AB 出力ドライバは、最大出力が 500 μA と強力で、最大 1000pF の容量性負荷を駆動でき、A/D コンバータ (ADC) のサンプルホールド入力と直接インターフェイスするように設計されています。

2.3.8 TLV76133

TLV761 は、より厳密な出力精度と、スタンバイ消費電力を低減させる小さい静止電流 (I_Q) とによって、従来の x1117 レギュレータ (TLV1117 または LM1117) の機能を向上させるリニア電圧レギュレータです。TLV761 はその他の固定 SOT-223、TO-252 レギュレータとピン互換です。TLV761 の入力電圧範囲は 2.5V~16V であり、0.8V~13V の出力電圧範囲を持っているため、幅広いアプリケーションに対応できます。TLV761 の広帯域の PSRR 特性 (標準値) は、1kHz で 60dB、1MHz で 40dB を超え、上流の DC/DC コンバータのスイッチング周波数を減衰して、レギュレータ後のフィルタ処理を最小化できます。さらに、TLV761 は、起動時の突入電流を減らすために内部ソフトスタート機能を備えているため、入力容量を最小化することで設計のスペースとコストを節約できます。TLV761 は、過負荷電流フォルトまたは短絡発生時のデバイスの消費電力を制限するフォールドバック電流制限機能を備えています。

2.3.9 TLV9062

TLV9061 (シングル)、TLV9062 (デュアル)、TLV9064 (クワッド) は、レールツーレールの入力および出力スイング機能を備えたシングル/デュアル/クワッド低電圧 (1.8V~5.5V) オペアンプです。これらのデバイスは、低電圧での動作、小さな占有面積、大きな容量性負荷の駆動が必要なアプリケーション向けの、コスト効率の優れた選択肢です。TLV906x の容量性負荷駆動能力は 100pF ですが、開ループ出力インピーダンスは抵抗性なので、大きい容量性の負荷でも簡単に安定できます。これらのオペアンプは低電圧 (1.8V~5.5V) で動作し、OPAx316 および TLVx316 デバイスと同様の性能仕様を満たすよう、特別に設計されています。

TLV906xS デバイスにはシャットダウンモードが備わっており、標準消費電流 1 μA 未満で、アンプをスタンバイモードに切り替えることができます。TLV906xS ファミリーはユニティゲイン安定で、RFI および EMI 除去フィルタが内蔵され、オー

バードライブ状況で位相反転が発生しないため、システム的设计を簡素化するため役立ちます。すべてのチャンネルバリエーション (シングル、デュアル、クワッド) が、X2SON や X2QFN などのマイクロサイズパッケージと、SOIC、MSOP、SOT-23、TSSOP などの業界標準パッケージで供給されます。

2.4 ハードウェア設計

2.4.1 インダクタの設計

入力インダクタ (L_i) は、スイッチング周波数の高調波をフィルタリングします。インダクタの設計は、他の要因とともに、電流リップルの計算と、計算された電流リップルに耐えられるコア材質の選択によって決まります。図 2-7 に、インダクタ電流に対するインバータ出力電圧 v_i のスイッチング サイクル波形を示します。

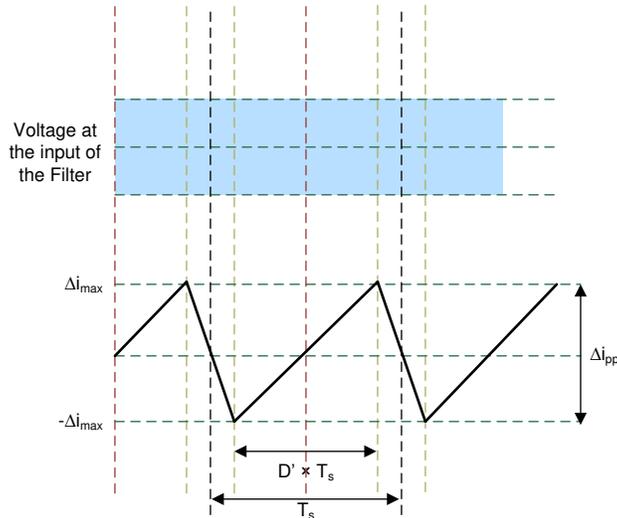


図 2-7. 電流リップルの計算

インダクタにかかる電圧は、 $V = L_i(di/dt)$ で示されます。Vienna 整流器については、式 7 を参照してください。

$$\left(\frac{V_{bus}}{2} - V_{in}\right) = L_i \times \frac{\Delta i_{pp}}{D' \times T_s} \quad (7)$$

ここで、

- $T_s = 1/F_{sw}$ はスイッチング周期です。
- D' は、スイッチが ON のデューティサイクルです。

制御設計では、 D はインダクタのもう一方の端子の電圧と仮定され、 $D' = 1 - D$ によって D' と関連付けられます。AC 波形の任意の瞬間における電流リップルを再配置すると、式 8 のようになります。

$$\Delta i_{pp} = \frac{D' \times T_s \times \left(\frac{V_{bus}}{2} - V_{in}\right)}{L_i} \quad (8)$$

ここで、変調インデックスを m_a とすると、デューティサイクルは $D' = m_a \sin(\omega t)$ となり、 $V_{in} = D' \times (V_{bus}/2)$ と仮定すると、式 9 が導かれます。

$$\Delta i_{pp} = \frac{\frac{V_{bus}}{2} \times T_s \times m_a \times \sin(\omega t) \times (1 - m_a \sin(\omega t))}{L_i} \quad (9)$$

式 9 から、ピークリップルは、入力 AC が正弦波形であることが要因であることは明らかです。

時間に関して式を微分して最大値を求めるには、式 10 を使用します。

$$\frac{d(\Delta i_{pp})}{dt} = K\{\cos(\omega t)(1 - m_a \sin(\omega t)) - m_a \sin(\omega t) \times \cos(\omega t)\} = 0 \quad (10)$$

最大リップルは $\sin(\omega t)=1/(2 \times m_a)$ の場合に発生し、この値を代入すると、式 11 が導かれます。

$$\Delta i_{pp\max} = \frac{V_{bus}}{2} \times T_s \Rightarrow L_i = \frac{V_{bus}}{4 \times F_{sw} \times \Delta i_{pp\max}} \quad (11)$$

これらの値を踏まえて、このインダクタンス値を満たすように設計されたインダクタとともに、適切なコアを選択できます。

2.4.2 バス コンデンサの選択

バス コンデンサは、正弦波電流の流入によって発生する DC 電圧のリップルを除去する役割を果たします。コンデンサの値と DC バスリップルは、式 12 によって関連付けられています。

$$C = \left(\frac{1}{3}\right) \frac{P_{ac}}{4 \times f \times (V^2 - (V - \Delta V)^2)} \quad (12)$$

この式を使用して、DC バス容量の最小値を選択します。

注

式 12 の計算では、コンデンサの過剰設計になる可能性があります。コンデンサのサイズは、負荷および流入する電流の性質により左右されます。3 相 PFC の場合、入力が常に出力へのパスを持つため、電力リップルは比較的小さくなります。式 12 は参考程度に使用し、これが過剰設計につながることに注意してください。

2.4.3 入力 AC 電圧検出

まず、Y 接続された抵抗ネットワークと、安定性を確保するための容量を使用して、仮想中性点が構築されます。このデザインでは、コントローラをコールド側に維持するため、図 2-8 に示すように、絶縁型アンプ AMC1350 を使用して VL-N' の電圧を処理します。AMC1350 は電圧検出アプリケーション向けに低インピーダンスソースを考慮して設計されているため、入力差動抵抗が総ゲインの計算で非線形的な役割を果たします。したがって、ビルド レベル 1 の間に最終キャリブレーションを行い、最大 AC 電圧範囲をキャリブレーションに従って調整する必要があります。

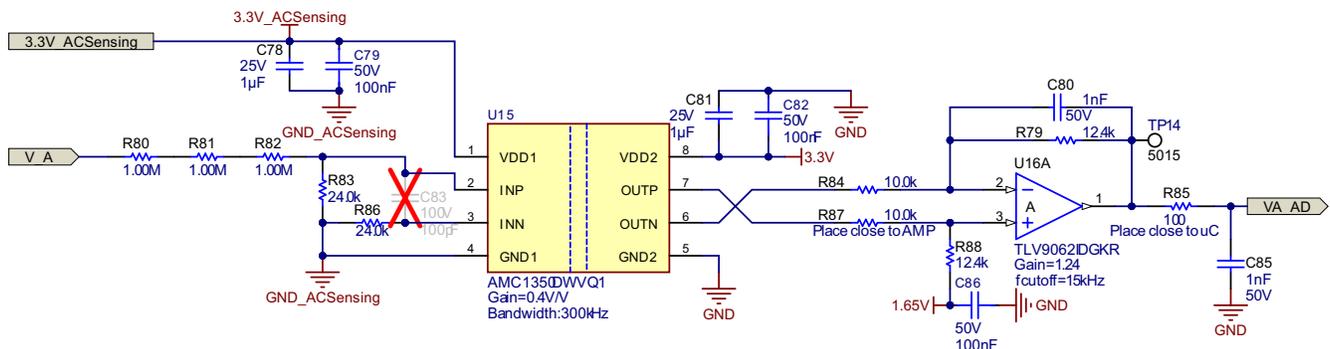


図 2-8. 入力 AC 電圧検出

2.4.4 出力 DCBUS 電圧検出

図 2-9 に示すように、同様に、2 つのコンデンサに分割された DCBUS 電圧は、AMC1350 と TLV9062 を使用して検出されます。DCBUS 電圧検出では精度が重要であるため、オフセットとゲインの調整によるキャリブレーションをさらに行う必要があります。

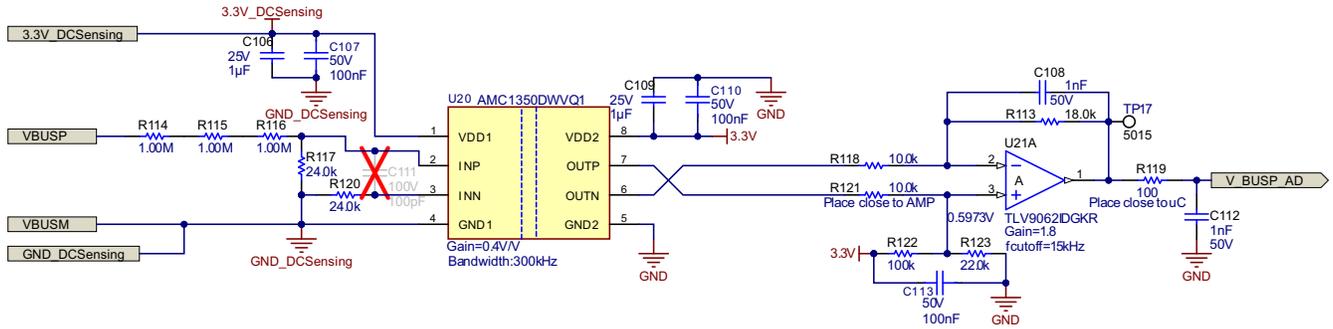


図 2-9. バス電圧検出

2.4.5 補助電源

補助電源は、15V の電圧レールを供給するように設計されています。この電源を制御するのは、U7 UCC28750 と高電圧 MOSFET Q8 です。3.3V の電圧レールは、LDO U11 TLV76133 に続いて DC/DC 降圧電源で生成されます。この設定により、LDO の電圧降下と発熱が抑えられます。また、LDO はクリーンで安定した 3.3V を供給し、ADC の電圧リファレンスとして使用できます。この回路を 図 2-10 に示します。

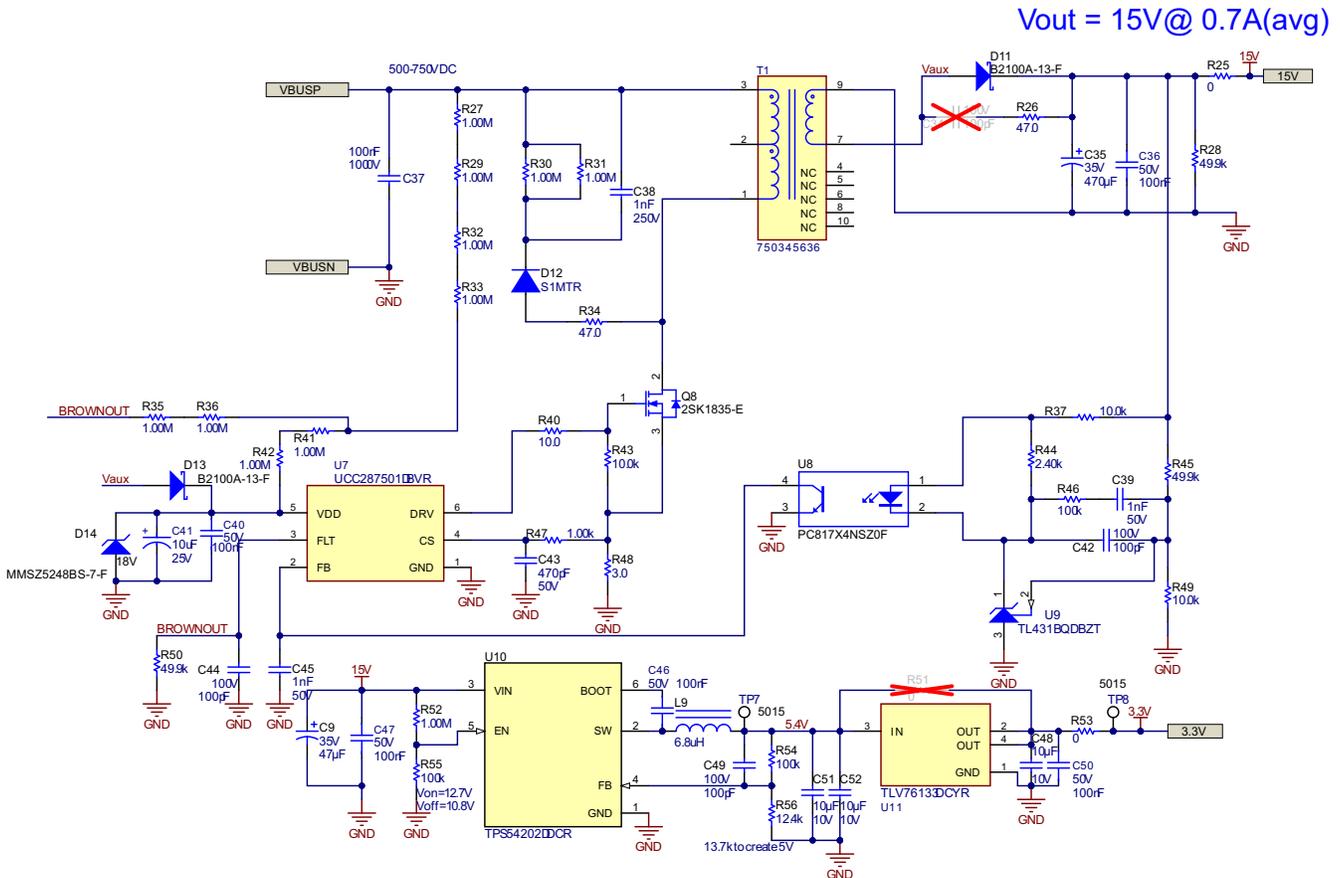


図 2-10. 補助電源

2.4.6 絶縁型電源

このリファレンス デザインでは、AC 入力電圧検出、DC 出力電圧検出、IGBT ゲートドライバの動作をサポートするために、5 つの絶縁型低電圧レールが必要です。LM25180 をベースとするシンプルな低電圧フライバック電源は、これらの絶縁型電圧レールを生成するように設計されています。この回路を 図 2-11 に示します。

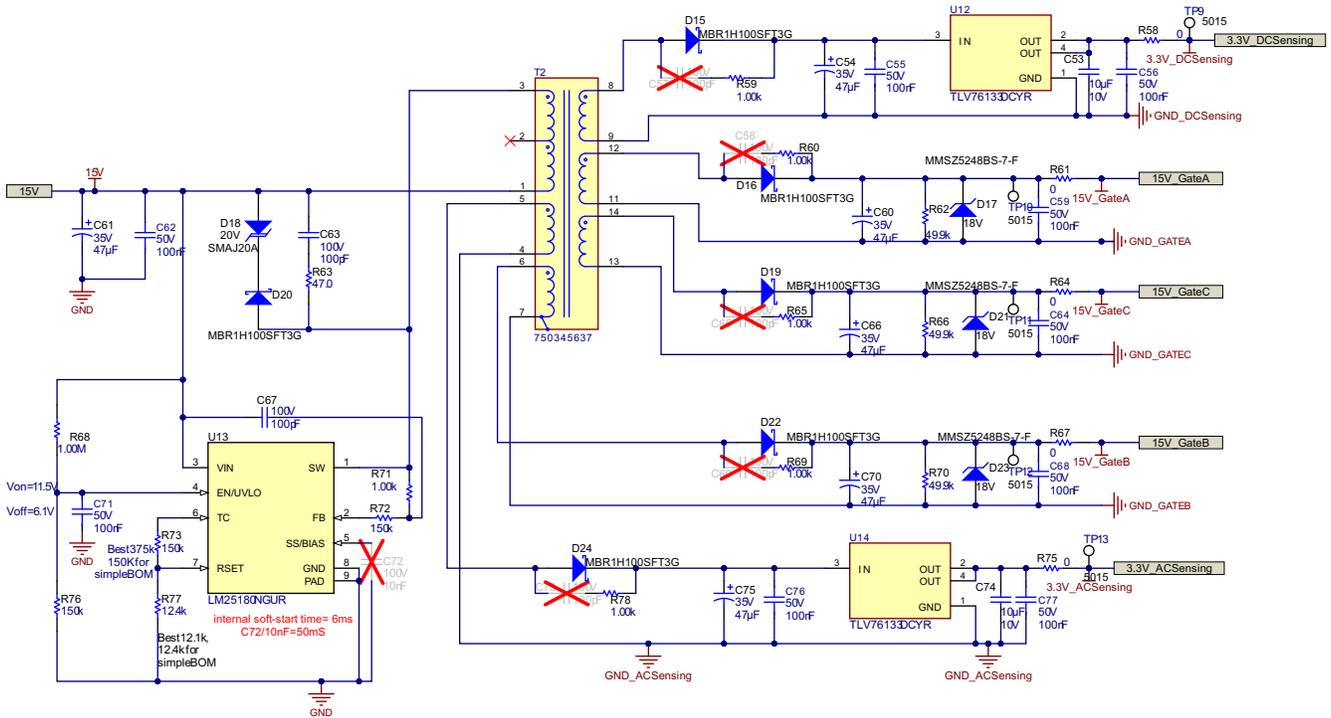


図 2-11. 絶縁型電源

2.4.7 インダクタ電流検出

TMCS1123 ホール効果センサは、インダクタを流れる電流を検出するために使用されます。このホール効果センサには 1.65V のオフセットが内蔵されており、ADC で直接測定できます。図 2-12 に回路を示します。

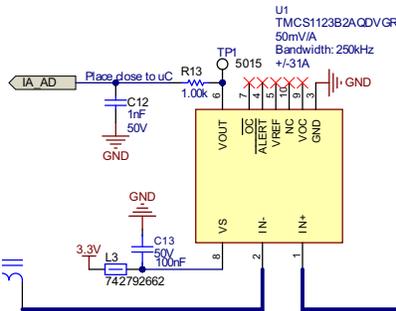


図 2-12. ホール効果センサを使用した電流検出

2.4.8 ゲートドライバ

図 2-13 に、このリファレンス デザインで IGBT を駆動するために使用されている UCC5350MC デバイスを示します。このゲートドライバは、10A のソースおよびシンク電流、およびミラー クランプを搭載しています。大電流駆動を使用することで、IGBT の立ち上がり時間と立ち下がり時間を短縮し、オン/オフ遷移時の電力損失を低減できます。

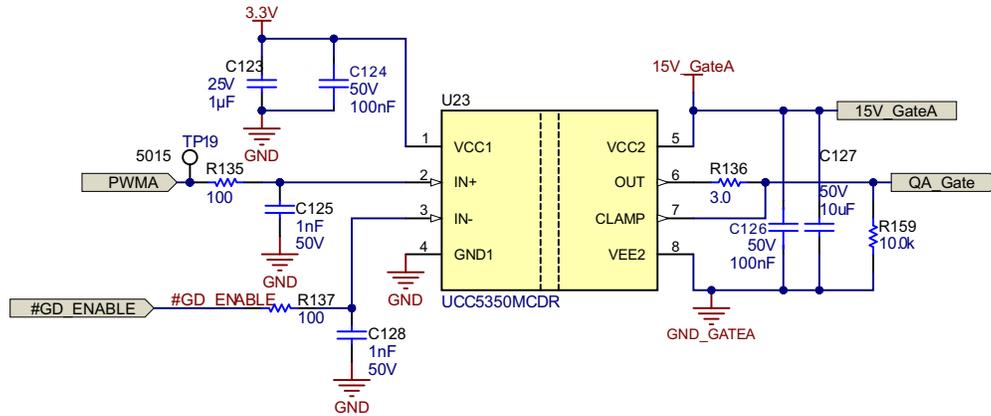


図 2-13. ゲートドライバ

2.4.9 絶縁型温度センシング

図 2-14 に、パワー デバイスの温度検出に使用される ISOTMP35 絶縁型温度センサを示します。TSENSE パッドは、パッドが絶縁されているため、パワー デバイスと同じ銅箔上に配置可能です。周囲温度は、出力電圧が線形であるため、出力電圧を使用して簡単に計算できます。

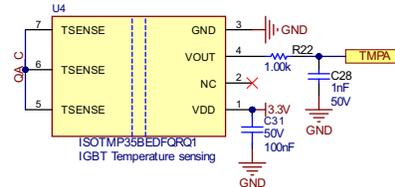


図 2-14. 絶縁型温度センシング

2.4.10 過電流および過電圧保護 (CMPSS)

大半のパワー エレクトロニクス コンバータは、過電流と過電圧イベントから保護する必要があります。図 2-15 に示すように、このデザインでは複数のコンパレータが必要で、トリップの基準を生成する必要があります。

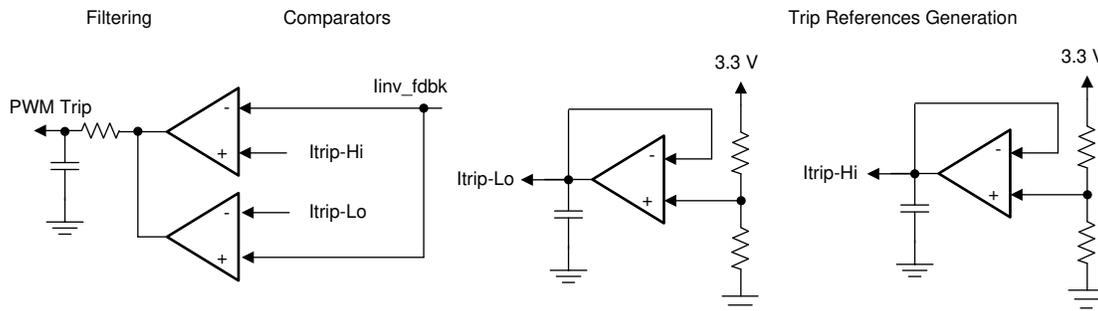


図 2-15. コンパレータと基準発生器を使用した PWM の検出信号生成

TMS320F280013x などの C2000 MCU を使用することで、CMPSS の一部としてオンチップ ウィンドウ コンパレータが搭載され、PWM モジュールに内部接続されるため、PWM の高速トリップが可能となり、このような回路はすべて不要となります。図 2-16 に示すように、オンチップ リソースを使用することで追加部品が不要となり、最終アプリケーションにおいて基板面積を節約し、コスト効率を高めることができます。

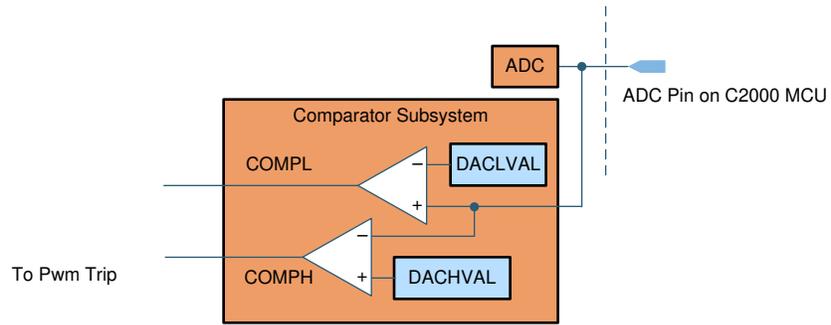


図 2-16. 過電流および過電圧保護用のコンパレータ サブシステム (CMPSS)

3 ハードウェア、ソフトウェア、テスト要件、テスト結果

このセクションでは、ハードウェア、ソフトウェア、およびテスト結果について説明します。

3.1 ハードウェア要件

3.1.1 ハードウェアの概要

このセクションでは、リファレンス デザインの基板とソフトウェアのテストと検証に必要な機器、テスト設定、手順について詳しく説明します。

3.1.1.1 基板の概要

リファレンス ボードには、完全な 3 相 Vienna PFC システムを実現する機能グループがあります。以下のリストに、基板上の機能ブロックを示します。図 3-1 に、基板の上面図とさまざまなブロックを示します。表 3-1 に、主要なコネクタと機能を示します。

- AC 入力ライン フィルタおよびリレー
- 昇圧インダクタ
- ブリッジおよび IGBT
- 補助電源および絶縁型電源
- DSP、F2800137
- 電圧検出およびゲートドライバ
- IGBT ゲートドライバ

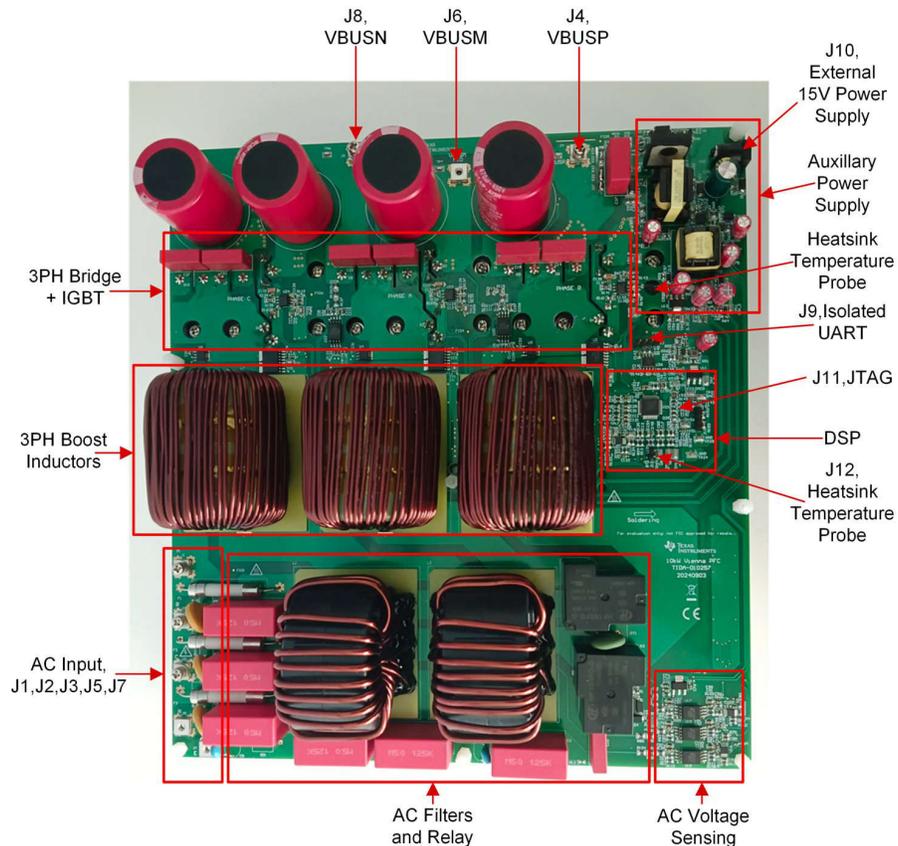


図 3-1. 基板の概要

表 3-1. 主要なコネクタと機能

コネクタ名	機能
J1、J2、J3	3 相 AC 入力ライン
J5	3 相 AC 入力のニュートラル
J7	3 相 AC 入力のグラウンド
J4	出力 DCBUS 正
J6	出力 DCBUS 中間
J8	出力 DCBUS 負
J9	GUI 用絶縁型ユニバーサル非同期レシーバ/トランスミッタ (UART)
J10	15V、1A 外部電源
J11	デバッグ用 cJTAG
J12	ヒートシンク サーミスタ

3.1.1.2 試験装置

以下の機器がテストができる状態であることを確認します。

1. 0VAC~400VAC L-L の 10kW AC 電源、または 208VAC~400VAC L-L のグリッド電力
2. 10kW DC 負荷、負荷は変更可能。
3. 15V、1A 電源アダプタ
4. USB - UART アダプタとケーブル
5. マルチメータ
6. 3 相電力アナライザ
7. デジタル オシロスコープ

3.2 ソフトウェア要件

3.2.1 GUI の概要

セクション 3.2.2 で説明しているように、このリファレンス デザインのソースコードは設計者がファームウェアを直接デバッグできるように提供されています。ただし、ソフトウェアのデバッグには時間がかかります。開発を加速するために、UART ベースの GUI ソフトウェアが提供され、迅速な基板の制御と動作状況の観察が可能です。このセクションでは、UART GUI の使用方法を紹介します。

ホスト PC を J9 でこのリファレンス ボードに接続する場合、J9 は絶縁型 UART ポートであるため、UART ポートには外部の 5V または 3.3V 電圧レールで電力を供給する必要があります。

3.2.1.1 テスト設定

図 3-2 に、GUI を使用してテストを行うためのハードウェア接続を示します。次の手順に従ってハードウェアの設定を行ってください。

1. USB-UART アダプタを介して、J9 の GND、TX、RX、VCC_ISO (3.3V または 5V) をホスト PC に接続します。
2. AC 入力ケーブルを J1、J2、J3 に接続します。J5 と J7 は接続しても、接続しなくてもどちらでも構いません。
3. DC 出力ケーブルを J4 と J8 に接続し、無負荷に設定します。
4. マルチメータ、オシロスコープのプロープ、その他の測定機器を接続して、さまざまな信号やパラメータを調べたり、分析したりします。
5. ノート PC で GUI ソフトウェアを実行します。

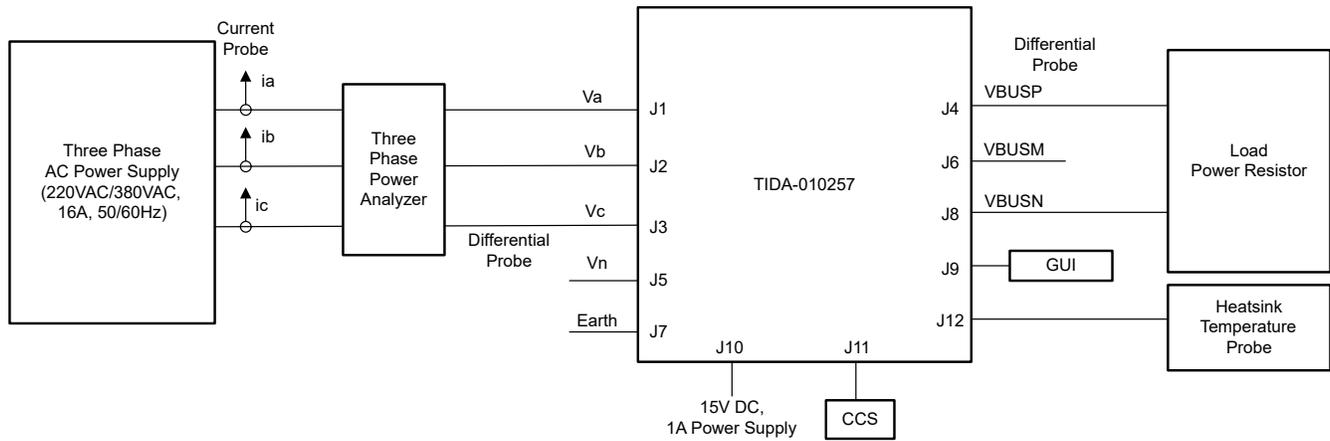


図 3-2. GUI 用ハードウェア設定

3.2.1.2 GUI ソフトウェアの概要

ホスト PC とリファレンス デザインの基板との間の通信には、どのような UART 端末 GUI ソフトウェアでも使用できます。図 3-3 に、GUI ソフトウェアを示します。このソフトウェアには、リファレンス デザインの基板から報告される動作ステータスを示す表示ウィンドウがあり、GUI ソフトウェアは HEX モードで基板にコマンドを送信することもできます。サポートされているボーレートは 115200 です。

報告される基板の動作ステータスには、AC 入力電圧、DC 出力電圧、AC 入力電流、AC 入力力率、パワー デバイスの温度、ヒート シンク温度、パワーオン時間が含まれます。これらのデータは ASCII モードです。

コマンドは次のとおりです。

- 0x11 は PFC を開始するコマンドです。
- 0x22 は PFC を停止するコマンドです。
- 0x33 はエラーをクリアするコマンドです。



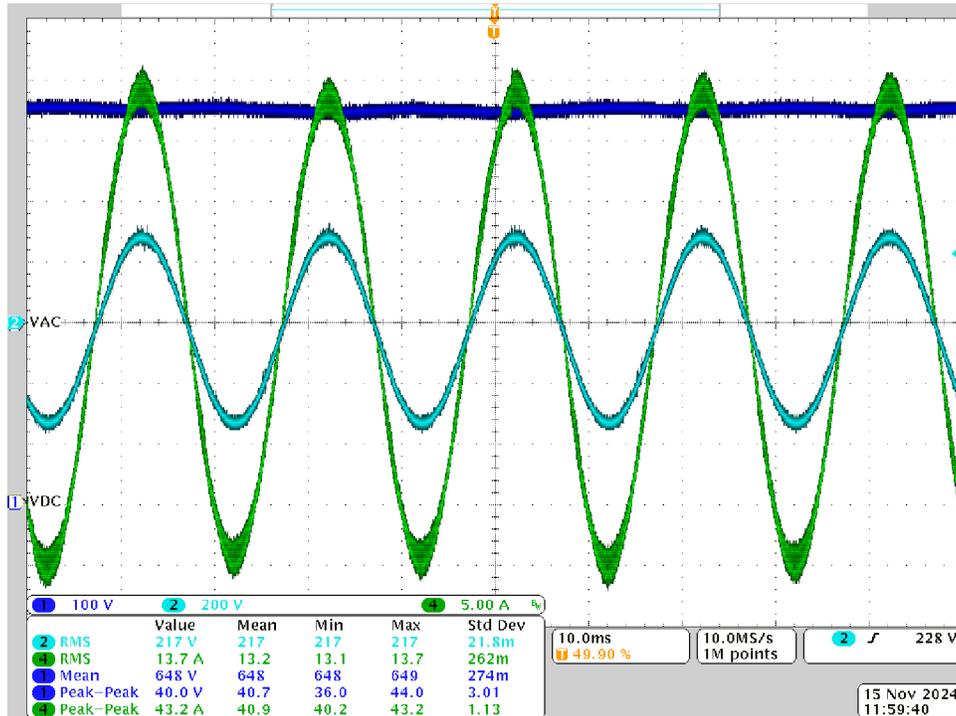
図 3-3. GUI ソフトウェアの概要

3.2.1.3 GUI によるテスト手順

以下の手順で、GUI を使用してリファレンス デザインをテストします。

1. ノート PC から USB - UART アダプタを J9 に接続します。

2. AC 電源 (208VAC~400VAC_{L-L}) を、J1、J2、J3 に接続します。J5 と J7 は接続しても、接続しなくてもどちらでも構いません。
3. DC 負荷を J4 と J8 に接続し、無負荷に設定します。
4. GUI ソフトウェアを実行し、適切な UART ポートを選択して、ボーレートを 115200bps に設定します。
5. 基板に電力供給し、D25 LED が点滅するのを待ちます。
6. リレーが閉じるのを待ちます。
7. GUI を確認します。PFC が始まっていないため、DC 出力電圧は 380VAC 入力時に約 530VDC です。
8. 無負荷時の DC 出力電圧が約 680VDC になるまで、0x11 コマンドを送信して PFC を開始します。
9. DC 出力電圧が約 650VDC になる (負荷が > 500W の場合) まで、負荷を段階的に上げます。負荷時の AC 入力電流、DC 出力電圧、温度を確認します。図 3-4 に、380VAC、650VDC、9kW 時の波形を示します。



注

- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-4. 380VAC、650VDC、9kW 時の波形

10. 無負荷になるまで、基板の負荷を段階的に減らします。
11. 0x22 コマンドを送信して PFC を停止します。
12. システムを安全に停止させるには、入力 AC 電圧をゼロまで下げます。

注意

このリファレンス デザインには電解コンデンサがあります。コンデンサの放電レートは、外部負荷がない場合、非常に遅いです。DCBUS 電圧には常に注意してください。

DCBUS に負荷がかかると放電が速く進みます。そうでない場合は、DCBUS がゼロになるまで長時間待機してください。

3.2.2 ファームウェアの概要

このデザインのソフトウェアは、C2000™ マイクロコントローラ向け DigitalPower ソフトウェア開発キット (SDK) に同梱済みです。

注

このデザインのファームウェアは、TMS320F2800137 リアルタイム マイクロコントローラに対応しています。

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

ソフトウェアを使用して開始するには:

1. Code Composer Studio™ (CCS) 統合開発環境 (IDE) ツール フォルダから Code Composer Studio™ IDE をダウンロードしてインストールします。バージョン 20.0.0 またはそれ以降をお勧めします。
2. C2000 マイクロコントローラ向け DigitalPower ソフトウェア開発キット (SDK) をダウンロードしてインストールします。
3. インストールが完了したら、CCS を開き、[Project] → [Import CCS Project] をクリックして、プロジェクトをインポートするための新しいワークスペースを作成し、
<install_location>\C2000ware_DigitalPower_SDK_5_03_00_00\solutions\TIDA-010257\f280013x を参照し、プロジェクト名をクリックしてプロジェクトをインポートします。

3.2.2.2 プロジェクト構造

図 3-5 に、プロジェクトがインポートされた後の CCS 内の [Project Explorer] を示します。

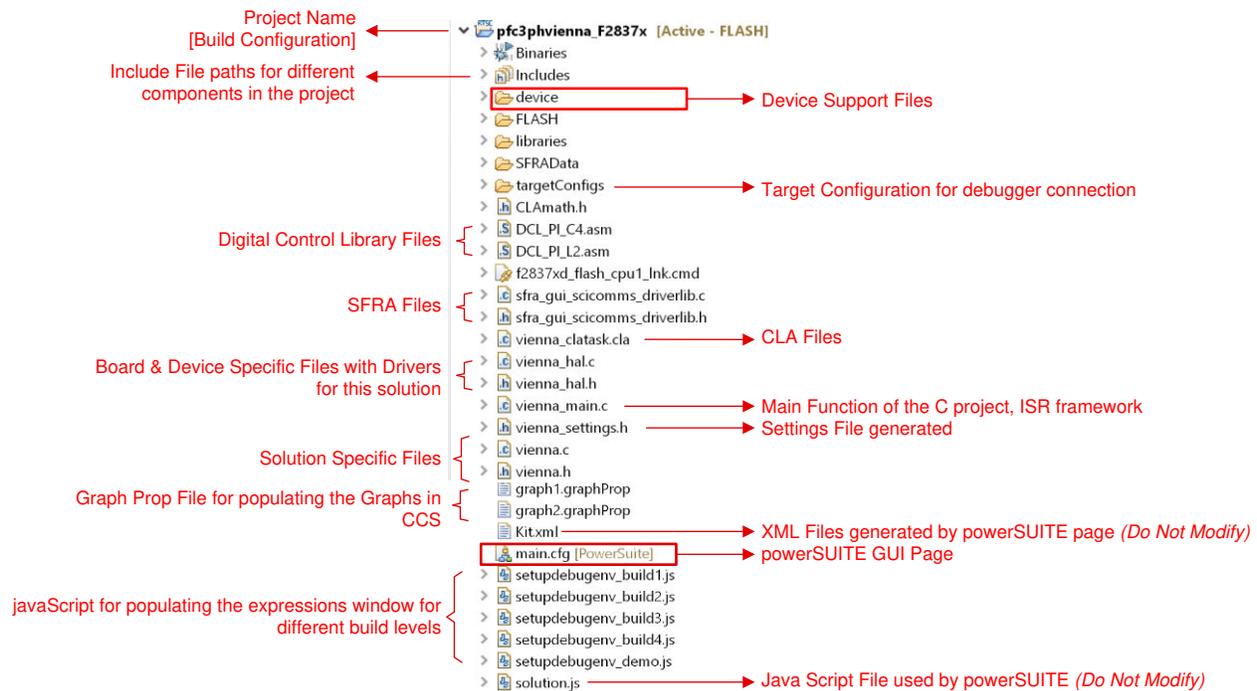


図 3-5. デザイン プロジェクトの [Project Explorer] ビュー

このプロジェクトは、制御アルゴリズムが実行される controlISR() という PWM サイクルごとに呼び出される割り込みサービス ルーチンで構成されています。これに加えて、バックグラウンド タスク A0-A3 および B0-B3 があり、これらはポーリング方式で呼び出され、絶対的なタイミング精度が要求されない低速タスクの実行に使用されます。低速の 10kHz ルーチン tenkHzISR() は計測のために呼び出され、タイミング精度が要求される低速のタスクを実行します。

このデザインガイドのソフトウェアは、4つの差分ビルド (INCR_BUILD) で構成されています。これらの差分ビルドプロセスによって、システムの立ち上げや設計が簡略化されます。

- INCR_BUILD 1: 開ループ チェック
- INCR_BUILD 2: 閉電流ループ
- INCR_BUILD 3: 閉電圧および電流ループ
- INCR_BUILD 4: 閉電圧、電流、バスコンデンサのバランスループ

これらのビルドレベルについては、セクション 3.2.2.4.1 からセクション 3.2.2.4.4 に詳述します。

3.2.2.3 テスト設定

図 3-6 に、ハードウェアブロックを示します。各ビルドレベルのテストを開始する前に、以下のテスト設定手順を実行してください。

1. エミュレータを J9 に接続します。
2. AC 電源を J1、J2、J3 に接続し、0V 出力に設定します。
3. DC 負荷を J4 と J8 に接続し、530Ω に設定します。
4. J9 を UART-USB アダプタを介してノート PC に接続します。
5. 15V、1A DC アダプタを J10 に接続します。

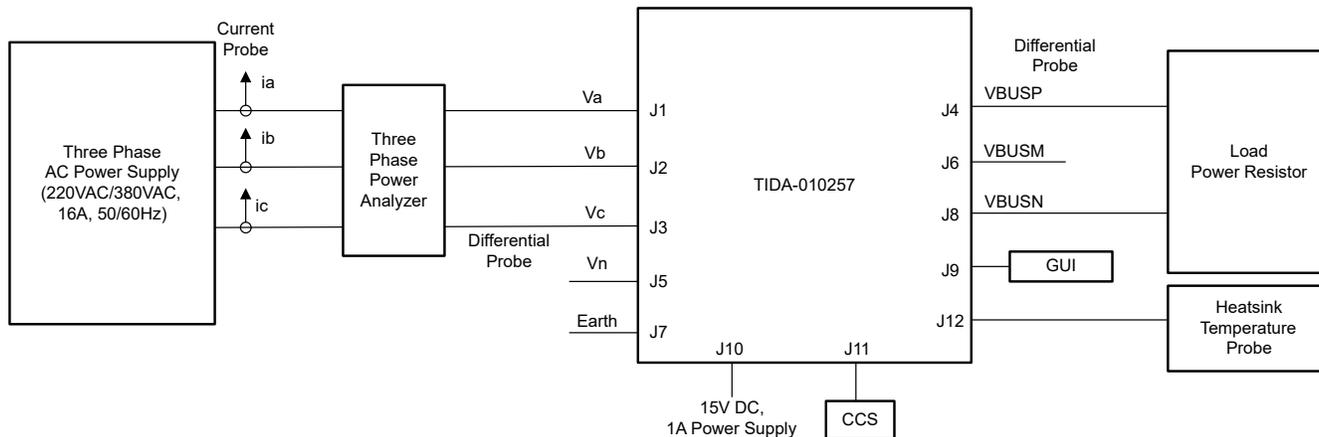


図 3-6. ハードウェアブロック

3.2.2.4 プロジェクトの実行

このセクションでは、ビルドレベル 1~4 のファームウェア デバッグについて説明します。ユーザー ガイドに記載されている変数名は、コード内の実際の名前よりも簡略化されています。たとえば、VIENNA_guiVbus_Volts は簡略化されて `guiVbus` となっています。

3.2.2.4.1 INCR_BUILD 1:開ループ

基板は、固定デューティ サイクルの開ループ モードで実行されます。デューティ サイクルは `dutyPU_DC` 変数で制御されます。このビルドでは、電力段からの帰還値の検出と PWM ゲートドライバの動作を検証し、ハードウェアに問題がないことを確認します。また、このビルドでは入出力電圧センシングの較正も実行できます。このビルドのソフトウェア構造を [図 3-7](#) に示します。低速 ISR で動作するブロックはマークされています。他のブロックは、高速の `controlISR` で動作します。

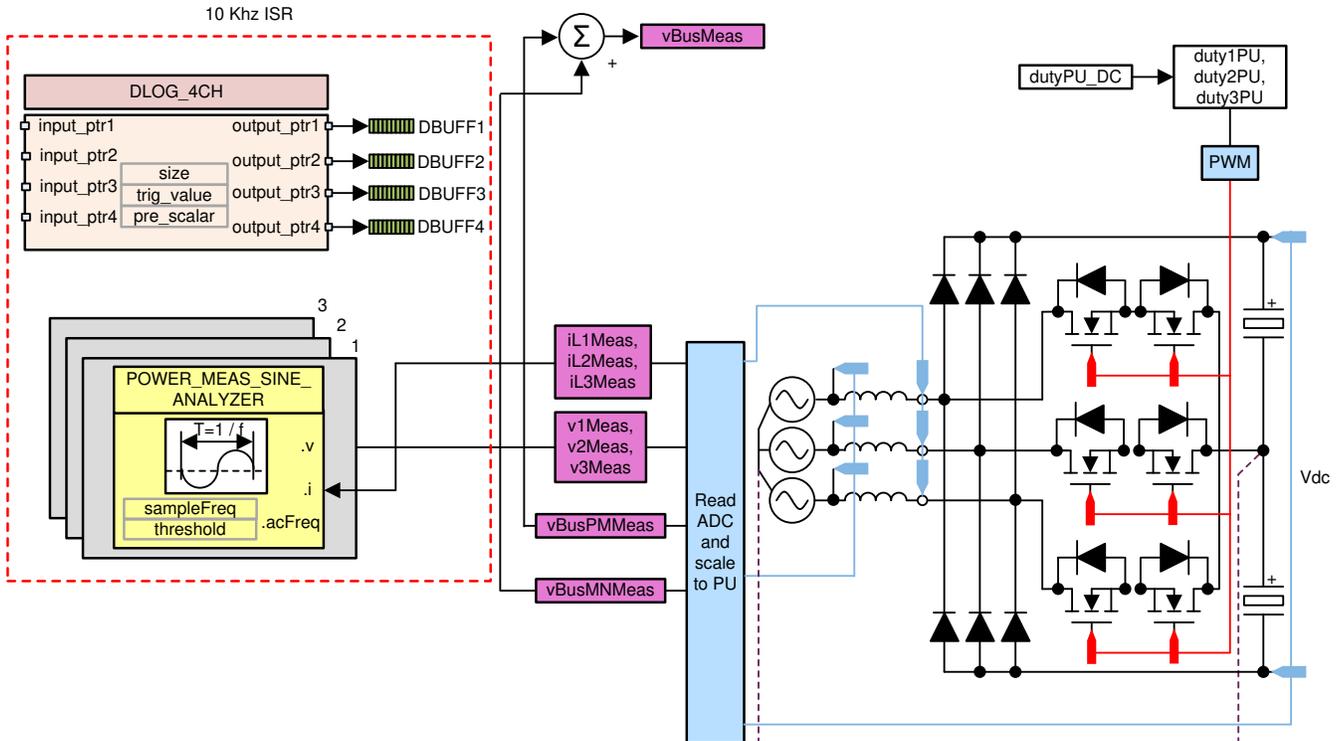


図 3-7. ビルドレベル 1 の制御ソフトウェア構成図:開ループ プロジェクト

3.2.2.4.1.1 プロジェクトの設定、ビルド、およびロード

1. vienna_settings.h ファイルを開き、VIENNA_INCR_BUILD を 1 に設定します。
2. [Project Explorer] で、targetConfigs の下で適切な目標構成ファイルが Active になっていることを確認します (図 3-5 を参照)。
3. プロジェクト名を右クリックし、[Rebuild Project] をクリックすると、プロジェクトが正常にビルドされます。
4. [Run] → [Debug] をクリックします。これによりデバッグ セッションが起動します。
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。メイン ルーチンの開始時にコードは停止します。

3.2.2.4.1.2 デバッグ環境設定ウィンドウ

1. [Watch] または [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_build1.js スクリプト ファイルを参照します。このスクリプト 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。[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 3-8 に、[Watch] ウィンドウを示します。

Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_1_OpenLoop
VIENNA_boardStatus	enum VIENNA_boardSt...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_dutyPU_DC	float	0.0
EPwm1Regs.TZFLG	Register	0x0004
EPwm2Regs.TZFLG	Register	0x0004
EPwm3Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	4.77491665
VIENNA_guiVbusPM_Volts	float	2.5455637
VIENNA_guiVbusMN_Volts	float	2.22988749
VIENNA_guiVrms1_Volts	float	0.0
VIENNA_guiVrms2_Volts	float	0.0
VIENNA_guiVrms3_Volts	float	0.0
VIENNA_guilrms1_Amps	float	0.0
VIENNA_guilrms2_Amps	float	0.0
VIENNA_guilrms3_Amps	float	0.0
VIENNA_guiPF1	float	0.0
VIENNA_guiPF2	float	0.0
VIENNA_guiPF3	float	0.0
Add new expression		

図 3-8. ビルドレベル 1 の Expressions ビュー

2. 電流と電圧の測定値は、グラフ ウィンドウにデータを表示して確認できます。これらの値は、低速の 10kHz ルーチンで記録されます。[Tools] → [Graph] → [DualTime] に進み、[Import] をクリックして、プロジェクトフォルダ内にある **graph1.GraphProp** ファイルを指定します。このファイルは、[Graph Properties] ウィンドウに入力されます。または、[図 3-9](#) に示すように値を入力します。入力内容が確認されたら、[OK] ボタンをクリックします。CCS に 2 つのグラフが表示されます。これらのグラフで [Continuous Refresh] ボタン () をクリックします。graph2.GraphProp ファイルをインポートして、グラフの 2 つ目のセットを追加することもできます。

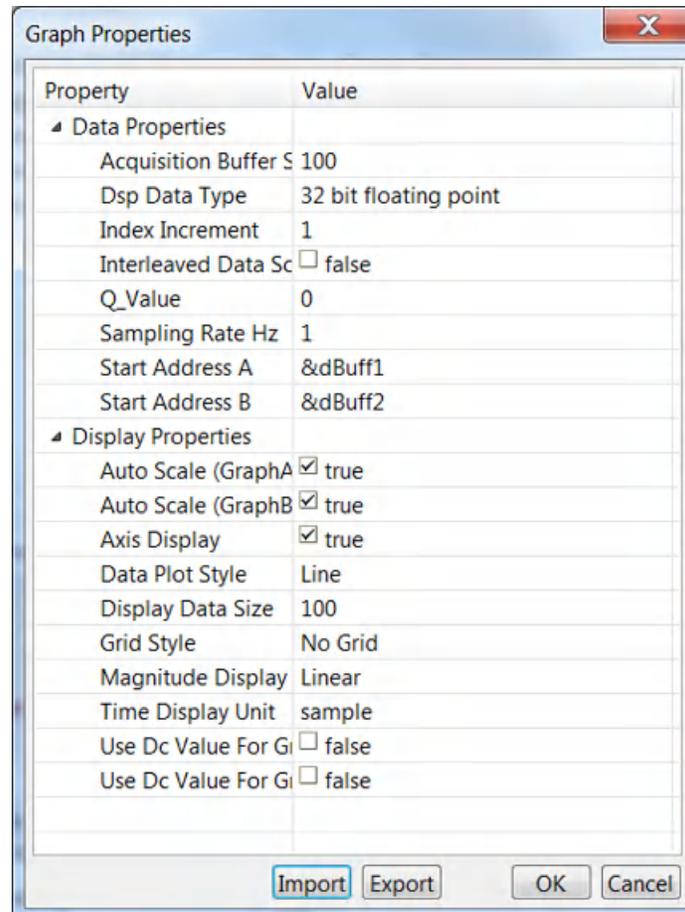


図 3-9. グラフの設定

3.2.2.4.1.3 リアルタイム エミュレーションの使用

リアルタイム エミュレーションは、マイクロコントローラ動作中に CCS 内のウィンドウを更新できる特別なエミュレーション機能です。この機能により、プロセッサを停止することなく、グラフおよび [Watch] ビューが更新可能になるだけでなく、ユーザーが [Watch] ウィンドウや [Memory] ウィンドウの値を変更して、その変更をシステムに反映できるようになります。

1. マウス ポインタを水平ツールバーのボタンの上に置き、  ボタンをクリックして、リアルタイム モードを有効にします。

Enable Silicon Real-time Mode (service critical interrupts when halted, allow debugger accesses while running)
2. メッセージ ボックスが表示されたら、[YES] を選択して、デバッグ イベントを有効にします。これにより、ステータスレジスタ 1 (ST1) のビット 1 (DGBM ビット) が 0 に設定されます。DGBM は、デバッグ イネーブル マスク ビットです。DGBM ビットが「0」に設定されると、メモリ値とレジスタ値がホスト プロセッサに渡されて、デバッガのウィンドウが更新できるようになります。

3.2.2.4.1.4 コードの実行 (ビルド 1)

-  ボタンをクリックしてプロジェクトを実行します。
- [Watch] ビューで、`guiVbus` ([Expression] ウィンドウの `VIENNA_guiVbus_Volts`) 変数が更新されているかどうかを定期的に確認します。値に変更がない場合は、リアルタイム モードが有効になっており、ハードウェアが正しく設定されていることを確認します。更新が確認されない限り、先に進まないでください。

注

現時点では電力が印加されていないため、この値はゼロに近くなっています。

- 入力 AC 電圧を $0V_{RMS}$ から $80V_{RMS}$ L-N まで徐々に上げます。
- 電圧検出の検証:`guiVbus`、`guiVbusPM`、`guiVbusMN` に正しい値が表示されていることを確認します。 $80V_{RMS}$ L-N の場合、`guiVbus` は $190V$ に近く、グラフ機能は [図 3-10](#) に示すような波形を表示できます。`guiVbusPM` と `guiVbusMN` 変数は、どちらも $85V$ に近くなっています。このコードは、正弦波アナライザ モジュールを実行し、電圧と電流の RMS 値を計算します。[図 3-11](#) は、`guiVrms1`、`guiVrms2`、`guiVrms3` の各値が入力値、つまり $80V_{RMS}$ に近くなっていることを示しています。これにより、基板の電圧センシングが検証されます。

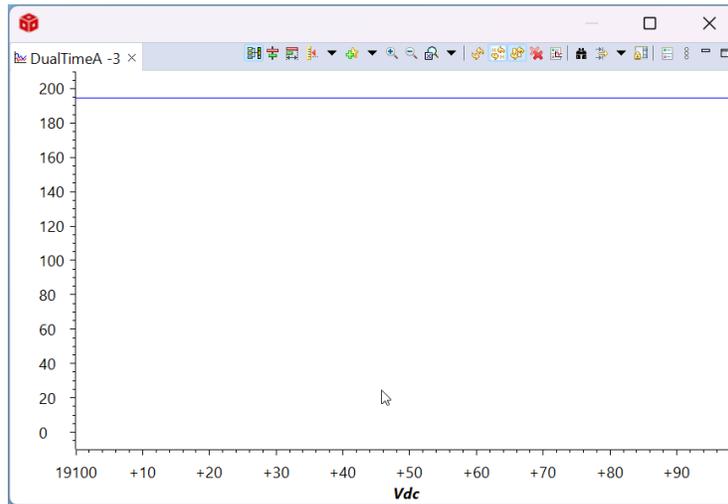


図 3-10. ビルド レベル 1: 出力電圧の測定値を示すグラフ

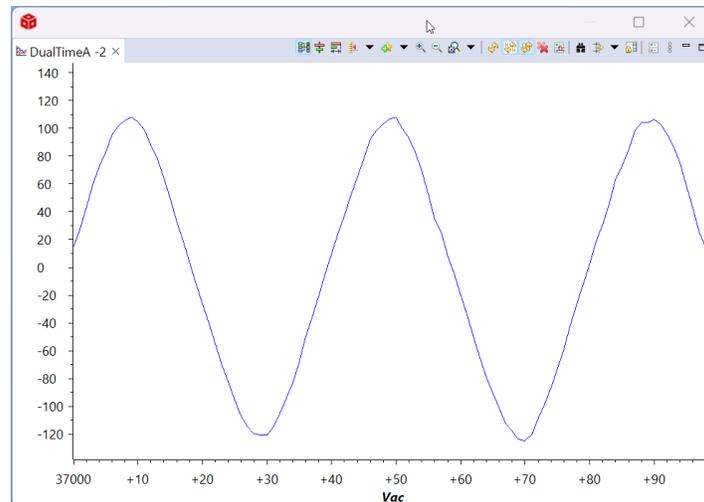


図 3-11. ビルド レベル 1: 入力電圧の測定値を示すグラフ

5. 電流検出の検証:指定されたテスト条件では、guiVrms1、guiVrms2、guiVrms3 の各変数が 0.5A に近くなっていることに注意してください。さらに、電流測定値を検証するにはグラフを確認する必要があります。図 3-12 に、グラフに表示された電流を示します。

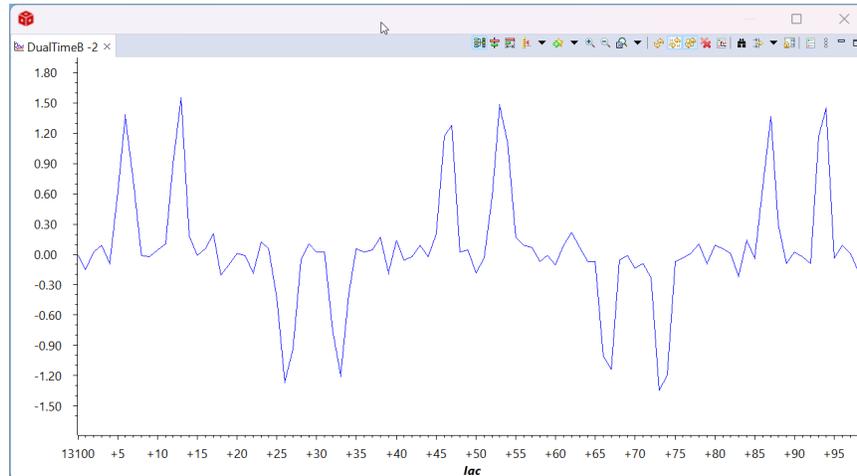
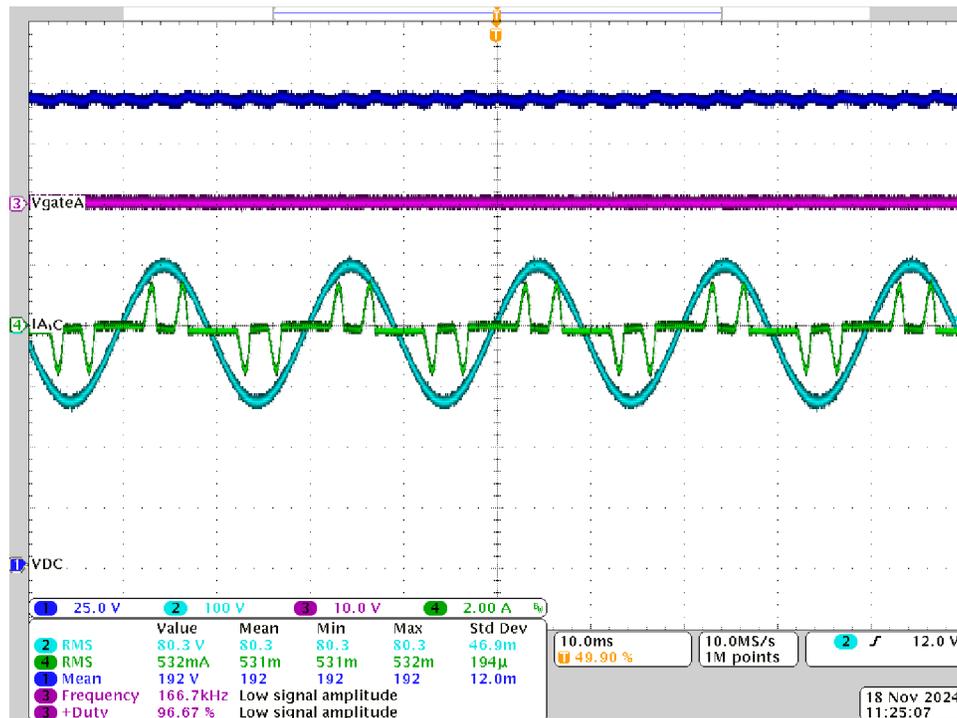


図 3-12. ビルドレベル 1:電流の測定値を示すグラフ

6. 図 3-13 に、入力電圧と入力電流のスコープ キャプチャを示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

図 3-13. ビルドレベル 1:PWM トリップ時のスコープ キャプチャ Ia および Va (80V_{RMS} L-N)

7. PWM 動作を確認するには、まず入力電圧をゼロに下げ、すべての電圧がゼロに下がるのを待ちます。
8. [Expressions] ビューで dutyPU_DC 変数を 0.5 に設定します。
9. clearTrip に 1 を書き込んで、PWM トリップをクリアします。
10. 入力電圧を徐々に上げ、入力電流の監視を続けます。デューティ サイクルによって昇圧動作が行われます。たとえば、スイッチングが有効でない状態で VAC が 80V_{RMS} の場合、guiVbus は約 190V ですが、スイッチングを行うと

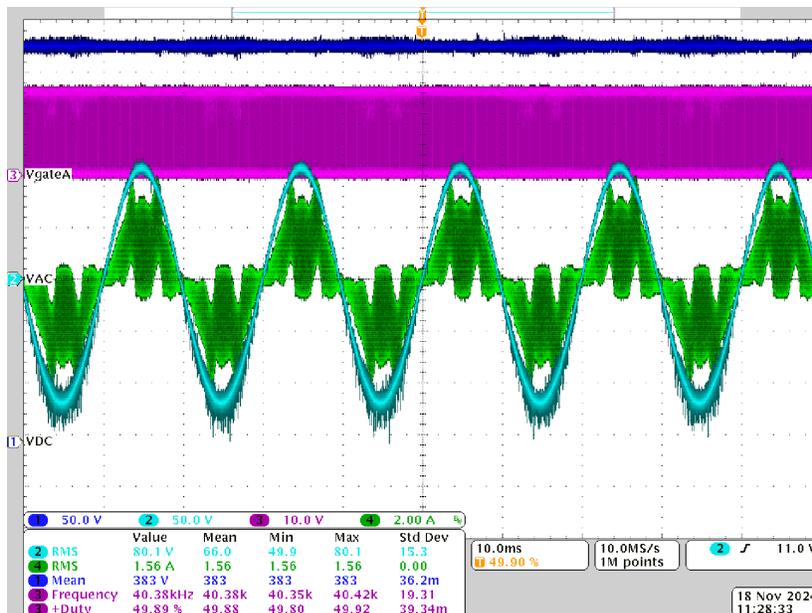
guiVbus は 380V まで上昇します。そのため、guiVbusPM と guiVbusMN は、どちらも最大入力電圧よりも高くなります。

11. このビルドで説明したテスト条件は以下のとおりです。guiVbus 変数は約 380V まで上昇し、guiVbusPM と guiVbusMN はそれぞれ 190V 近くになり、入力電圧が 80V_{RMS} L-N に達すると電流は 1.1A_{RMS} 近くになります。[Expressions] ビューはステップ 図 3-14 のように表示されます。すべての変数が正確であることを確認してください。つまり、guiVrms1、guiVrms2、guiVrms3、guiIrms1、guiIrms2、guiIrms3、guiPF1、guiPF2、guiPF3 です。正確ではない変数がある場合 (図 3-14 を参照)、検出回路にハードウェアの問題が発生していることを意味します。

Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_1_OpenLoop
VIENNA_boardStatus	enum VIENNA_boardST...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_dutyPU_DC	float	0.5
EPwm1Regs.TZFLG	Register	0x0010
EPwm2Regs.TZFLG	Register	0x0010
EPwm3Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	383.906738
VIENNA_guiVbusPM_Volts	float	227.420868
VIENNA_guiVbusMN_Volts	float	156.470276
VIENNA_guiVrms1_Volts	float	77.9399796
VIENNA_guiVrms2_Volts	float	79.6117783
VIENNA_guiVrms3_Volts	float	78.8453598
VIENNA_guiIrms1_Amps	float	1.12094522
VIENNA_guiIrms2_Amps	float	1.13230133
VIENNA_guiIrms3_Amps	float	1.11196923
VIENNA_guiPF1	float	0.940848231
VIENNA_guiPF2	float	0.940566897
VIENNA_guiPF3	float	0.953165889

図 3-14. ビルドレベル 1: 電力測定時の [Expressions] ビュー

12. 図 3-15 に、スコープ キャプチャを示します。



- CH1 (青): DCBUS 出力電圧
- CH2 (水色): AC 入力相 A 電圧
- CH3 (ピンク): IGBT ゲート電圧
- CH4 (緑): AC 入力相 A 電流

図 3-15. ビルドレベル 1: デューティサイクル 0.5 時のスコープ キャプチャ Ia および Va (80V_{RMS} L-N)

13. これにより、基本的なレベルで PWM ドライバとハードウェアの接続を検証できます。
14. 入力電圧をゼロまで下げ、バス電圧がゼロまで下がるのを確認します。
15. これでこのビルドのチェックは完了し、このビルドが正常に終了した時点で次の項目を検証します。
 - a. 電圧および電流の検出とスケーリングが適正であること
 - b. `controlISR` と `tenkHzISR()` の各変数におけるのビルド 1 コードの割り込み生成および実行
 - c. PWM ドライバおよびスイッチング

問題が確認された場合には、ハードウェアを慎重に点検し、ビルドの問題などを解消します。

16. これでコントローラを停止し、デバッグ接続を終了できます。
17. リアルタイム モードのマイクロコントローラを完全に停止するには、2 段階の手順を踏みます。まず、ツールバーの

[Halt] ボタン () を使用するか [Target] → [Halt] の順にクリックしてプロセッサを停止します。次に、  アイコンをクリックして、マイクロコントローラをリアルタイム モードから解除します。最後に、  ボタンをクリックして、マイクロコントローラをリセットします。

18. [Terminate Debug Session] ([Target] → [Terminate all]) をクリックして、CCS デバッグセッションを終了します。



3.2.2.4.2 INCR_BUILD 2: 閉電流ループ

BUILD 2 のビルドでは、内部電流ループが閉じているため、電流補償器 G_i を使用してインダクタ電流を制御します。式 13 に示すように、DC バスと出力電圧フィードフォワードの両方をこの電流補償器の出力に印加して、インバータのデューティサイクルを生成します。これにより、電流補償器の構成が簡略化され、比例 (P) コントローラを使用して内部電流のループを調整できます。電流ループのモデルは、セクション 2.2.1.2 で導き出されます。

$$\text{duty1PU} = \frac{(iL1\text{Meas} - iL1\text{Ref}) \times G_i \text{GainKp} + v1\text{Meas}}{v \text{BusHalfMeas}} \quad (13)$$

図 3-16 に、このビルドのソフトウェア構成図を示します。

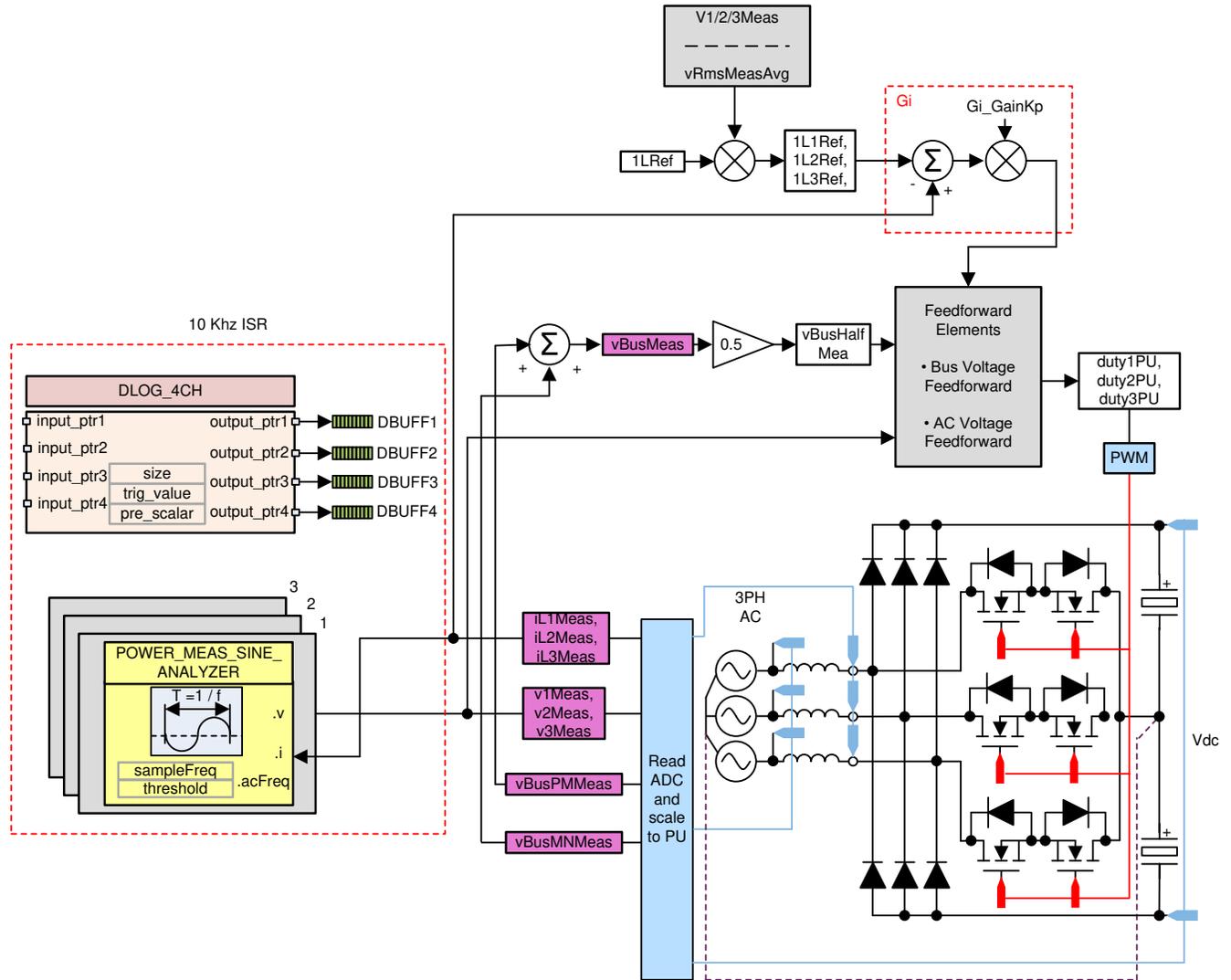
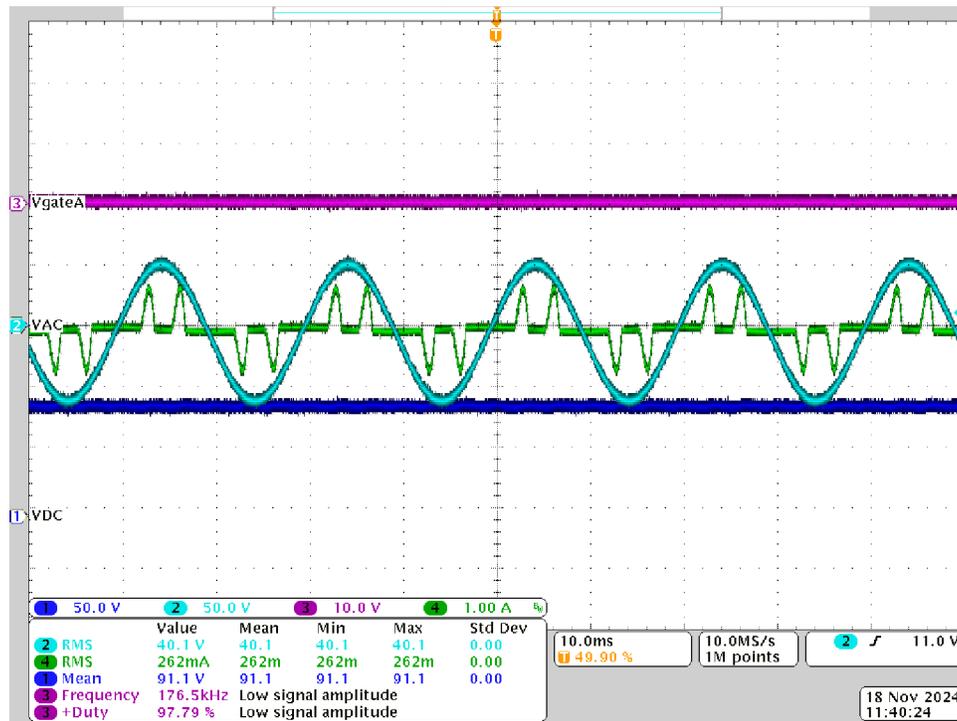


図 3-16. ビルドレベル 2 の制御ソフトウェア構成図: 閉電流ループ

3.2.2.4.2.1 コードの実行 (ビルド 2)

-  ボタンをクリックしてプロジェクトを実行します。
- まず低電圧でテストします。これにより、入力 AC 電圧は $40V_{RMS}$ 、50Hz までしか上がりません。
-  3-17 に、入力電流と入力電圧の波形を示します。

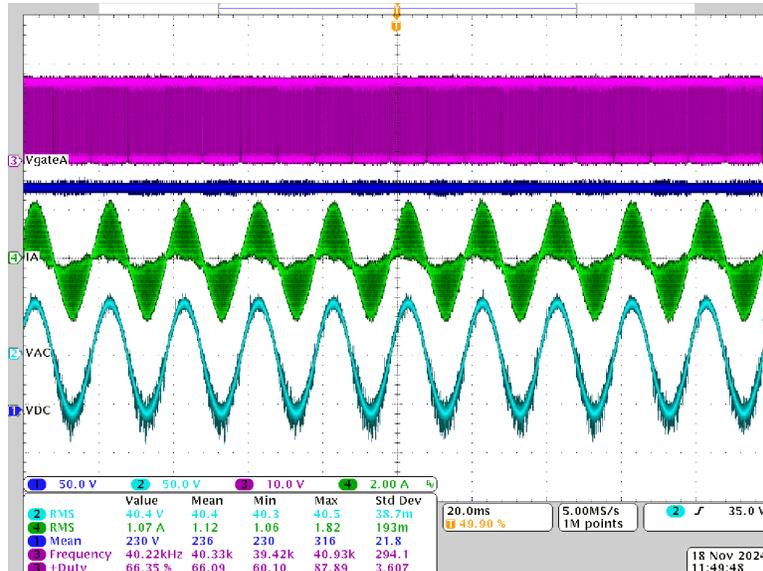


- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

図 3-17. ビルド レベル 2:PWM トリップ時のスコープ キャプチャ Ia および Va ($40V_{RMS}$ L-N)

- 電流リファレンスは、[Expressions] ビューの `iLRef` 変数を変更することで設定されます。この変数は 0.02 に設定されています。
- `clearTrip` 変数を 1 に設定して、トリップをクリアします。

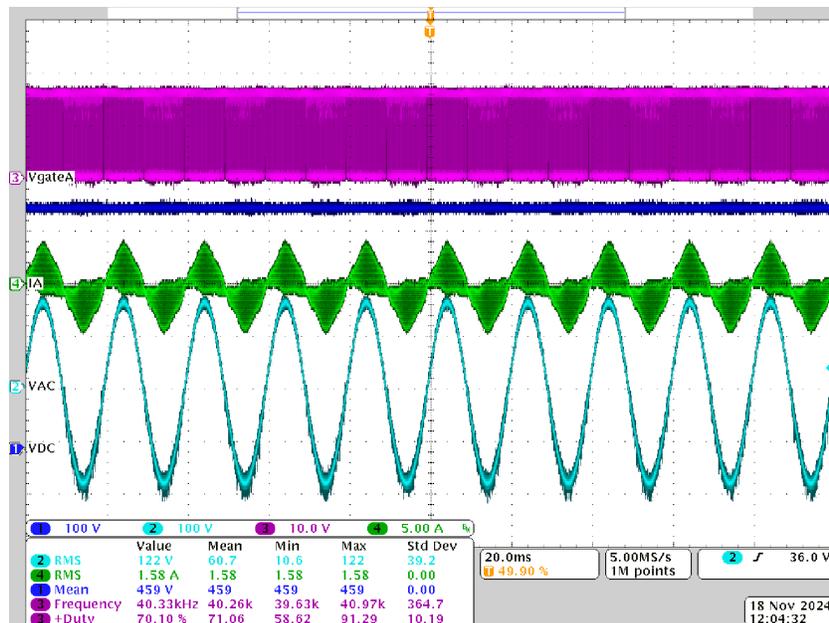
6. トリップがクリアされるとすぐに、入力から正弦波電流が引き込まれ、電流ループが正常に動作していることが確認されます。図 3-18 に波形を示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

図 3-18. ビルドレベル 2: $iL_{Ref} = 0.02$ 時のスコープ キャプチャ I_a および V_a ($40V_{RMS}$ L-N)

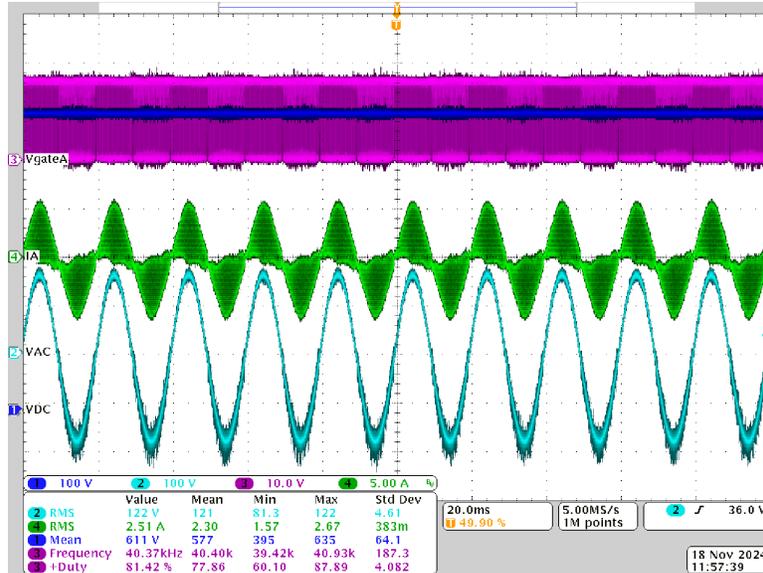
7. $guiV_{bus}$ 変数は 230V に近く、各相の入力 AC 電流は 1.07A に近いです。
8. 入力 AC 電圧を徐々に $120V_{RMS}$ まで上げます。この基板は、入力電圧が上昇しても、入力電流を一定に保ちます。出力電圧は、460V に上昇します。図 3-19 に、波形の様子を示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

図 3-19. ビルドレベル 2: $iL_{Ref} = 0.02$ 時のスコープ キャプチャ I_a および V_a ($120V_{RMS}$ L-N)

9. ここで、電流リファレンス $iLRef$ を 0.05 に上げます。バス電圧が 610V に、入力電流が約 2.5A になるのを確認してください。図 3-20 に波形を示します。



- CH1 (青): DCBUS 出力電圧
- CH2 (水色): AC 入力相 A 電圧
- CH3 (ピンク): IGBT ゲート電圧
- CH4 (緑): AC 入力相 A 電流

図 3-20. ビルドレベル 2: $iLRef = 0.05$ 時のスコープ キャプチャ i_a および V_a ($120V_{RMS}$ L-N)

- 補償回路では比例ゲインのみが使用されるため、電流リファレンスから帰還誤差を減算した値はゼロにはなりません。引き込まれる電流がリファレンスからわずかに逸脱していることに注意してください。
- システムを安全に停止させるには、入力 AC 電圧をゼロまで下げ、 $guiVBus$ もゼロに下がっていることを確認します。
- リアルタイム モードのマイクロコントローラを完全に停止するには、2 段階の手順を踏みます。まず、ツールバーの [Halt] ボタン () を使用するか、[Target] → [Halt] の順にクリックして、プロセッサを停止します。次に、  ボタンをクリックして、マイクロコントローラをリアルタイム モードから解除します。最後に、マイクロコントローラ () をリセットします。
- [Terminate Debug Session] ([Target] → [Terminate all]) をクリックして、CCS デバッグセッションを終了します。



3.2.2.4.2.2 プロジェクトのビルドおよびロードとデバッグの設定

1. `vienna_settings.h` ファイルを開き、`VIENNA_INCR_BUILD` を 2 に設定します。
2. 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。**[Run]** → **[Debug]** をクリックして、デバッグ セッションを起動します。デュアル CPU デバイスの場合、CPU を選択する必要がありますことを示すウィンドウが表示されたら、デバッグを実行します。ここでは、**CPU1** を選択します。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]** または **[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_build2.js` スクリプト ファイルを参照します。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。**[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 3-21 に **[Watch]** ウィンドウを示します。

Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_2_CurrentLoop
VIENNA_boardStatus	enum VIENNA_boardSt...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_iLRef_pu	float	0.0500000007
VIENNA_closeGilLoop	int	0
EPwm5Regs.TZFLG	Register	0x0000
EPwm4Regs.TZFLG	Register	0x0004
EPwm6Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	4.74759293
VIENNA_guiVbusPM_Volts	float	3.48382998
VIENNA_guiVbusMN_Volts	float	1.26423752
VIENNA_guiVrms1_Volts	float	0.0
VIENNA_guiVrms2_Volts	float	0.0
VIENNA_guiVrms3_Volts	float	0.0
VIENNA_guiIrms1_Amps	float	0.0
VIENNA_guiIrms2_Amps	float	0.0
VIENNA_guiIrms3_Amps	float	0.0
VIENNA_guiPF1	float	0.0
VIENNA_guiPF2	float	0.0
VIENNA_guiPF3	float	0.0

図 3-21. ビルドレベル 2: 閉電流ループ Expressions ビュー

4. マウス ポインタを水平ツールバーのボタンの上に置き、🔄 ボタンをクリックして、リアルタイム モードを有効にします。

3.2.2.4.3 INCR_BUILD 3: 閉電圧および電流ループ

このビルドでは、外部電圧ループも内部電流ループも閉じています (BUILD 2 で設計)。図 3-22 にこのビルドのソフトウェア構成図を示します。

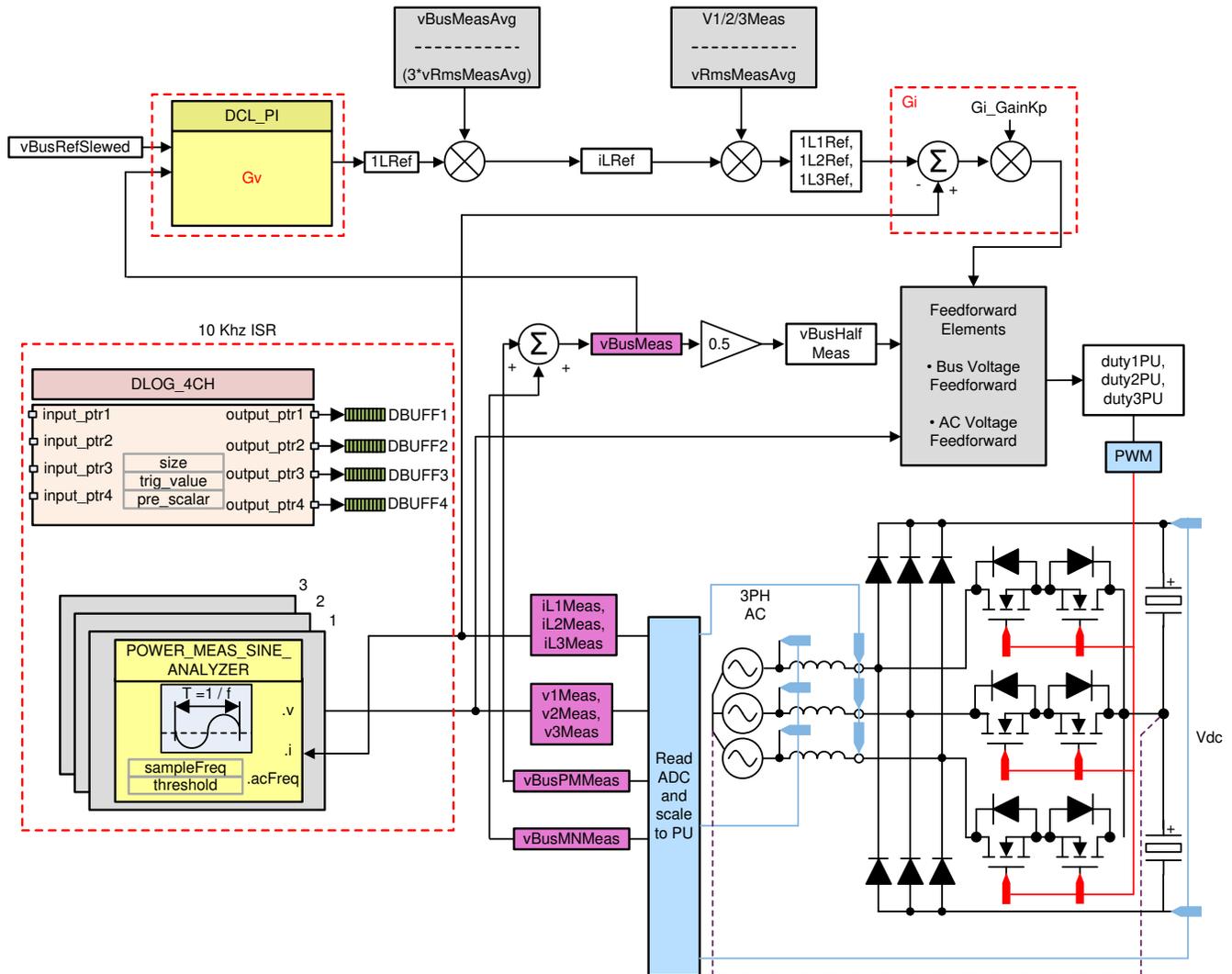


図 3-22. ビルドレベル 3 の制御図: 内部電流ループによる出力電圧制御

3.2.2.4.3.1 プロジェクトのビルドおよびロードとデバッグの設定

1. `vienna_settings.h` ファイルを開き、`VIENNA_INCR_BUILD` を 3 に設定します。
2. 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。**[Run]** → **[Debug]** をクリックして、デバッグ セッションを起動します。デュアル CPU デバイスの場合、CPU を選択する必要がありますことを示すウィンドウが表示されたら、デバッグを実行します。ここでは、**CPU1** を選択します。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]** または **[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_build3.js` スクリプト ファイルを参照します。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。**[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 3-23 に **[Watch]** ウィンドウを示します。

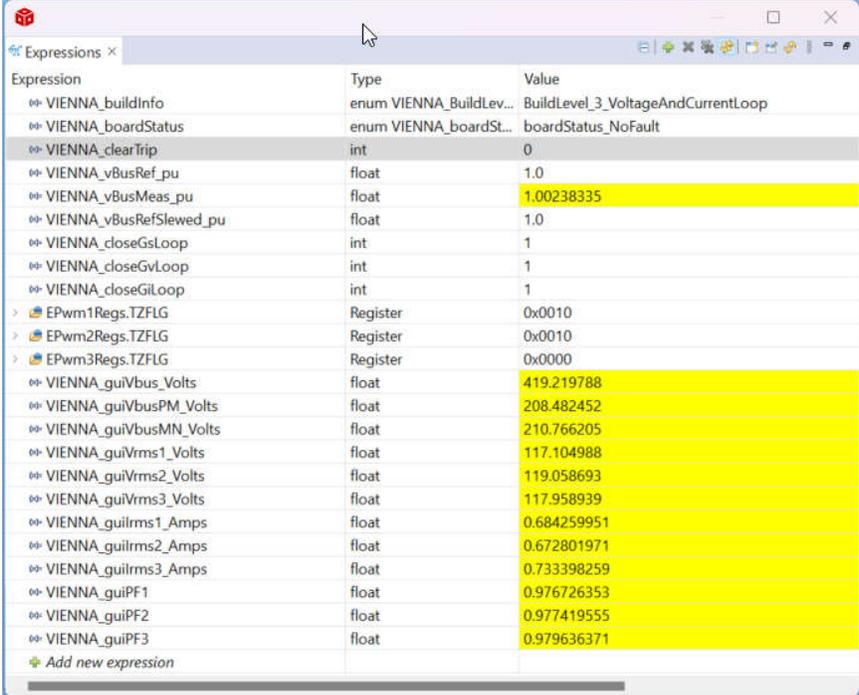
Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_3_VoltageAndCurrentl
VIENNA_boardStatus	enum VIENNA_boardSt...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_vBusRef_pu	float	1.55074549
VIENNA_vBusMeas_pu	float	0.0118097747
VIENNA_vBusRefSlewed_pu	float	0.0
VIENNA_closeGvLoop	int	0
VIENNA_closeGilLoop	int	0
EPwm1Regs.TZFLG	Register	0x0004
EPwm2Regs.TZFLG	Register	0x0004
EPwm3Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	4.84717178
VIENNA_guiVbusPM_Volts	float	3.48775887
VIENNA_guiVbusMN_Volts	float	1.35892749
VIENNA_guiVrms1_Volts	float	0.0
VIENNA_guiVrms2_Volts	float	0.0
VIENNA_guiVrms3_Volts	float	0.0
VIENNA_guiIrms1_Amps	float	0.0
VIENNA_guiIrms2_Amps	float	0.0
VIENNA_guiIrms3_Amps	float	0.0

図 3-23. ビルド レベル 3: Expressions ビュー

4. マウス ポインタを水平ツールバーのボタンの上に置き、🔄 ボタンをクリックして、リアルタイム モードを有効にします。

3.2.2.4.3.2 コードの実行 (ビルド 3)

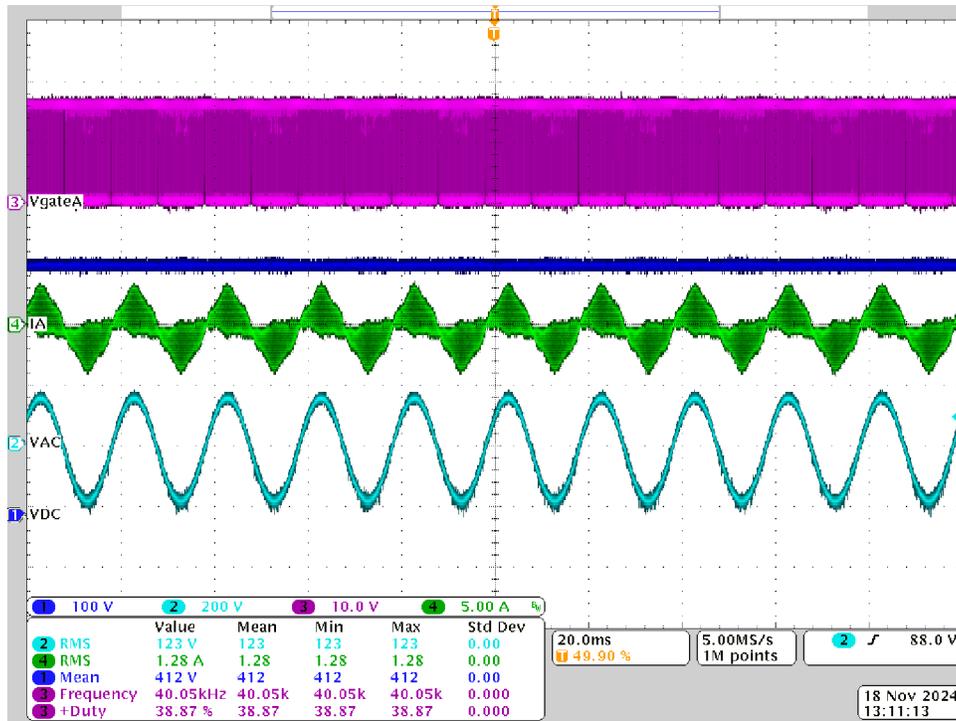
1.  ボタンをクリックしてプロジェクトを実行します。
2. 入力 AC 電圧を 120V_{RMS} VL-N または 208V_{RMS} VL-L、50Hz に上げます。
3. DC 電圧リファレンスは、vBusRef 変数によって設定されます。この値は 1.0 に設定され、このデザインでは 420V に相当します。
4. clearTrip 変数を 1 に設定して、トリップをクリアします。その後、バス電圧は 420V に上がります。
5. 閉ループ動作は、[図 3-24](#) に示すように、[Expressions] ウィンドウの vBusRef と vBusMeas を比較することで検証できます。



Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_3_VoltageAndCurrentLoop
VIENNA_boardStatus	enum VIENNA_boardSt...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_vBusRef_pu	float	1.0
VIENNA_vBusMeas_pu	float	1.00238335
VIENNA_vBusRefSlewed_pu	float	1.0
VIENNA_closeGsLoop	int	1
VIENNA_closeGvLoop	int	1
VIENNA_closeGiLoop	int	1
EPwm1Regs.TZFLG	Register	0x0010
EPwm2Regs.TZFLG	Register	0x0010
EPwm3Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	419.219788
VIENNA_guiVbusPM_Volts	float	208.482452
VIENNA_guiVbusMN_Volts	float	210.766205
VIENNA_guiVrms1_Volts	float	117.104988
VIENNA_guiVrms2_Volts	float	119.058693
VIENNA_guiVrms3_Volts	float	117.958939
VIENNA_guiIrms1_Amps	float	0.684259951
VIENNA_guiIrms2_Amps	float	0.672801971
VIENNA_guiIrms3_Amps	float	0.733398259
VIENNA_guiPF1	float	0.976726353
VIENNA_guiPF2	float	0.977419555
VIENNA_guiPF3	float	0.979636371
Add new expression		

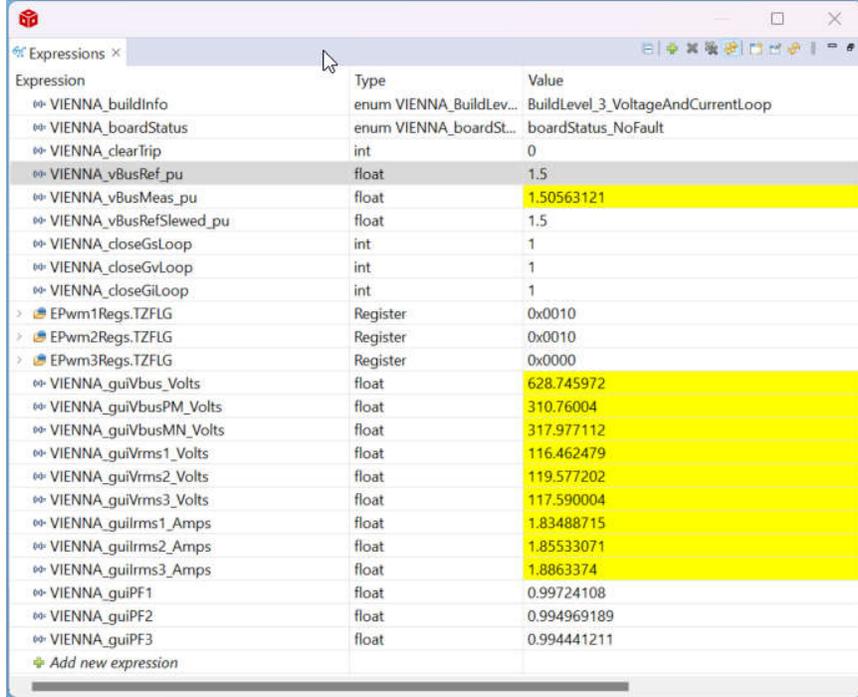
図 3-24. ビルドレベル 3: Vref = 1.0 の [Expressions] ウィンドウ

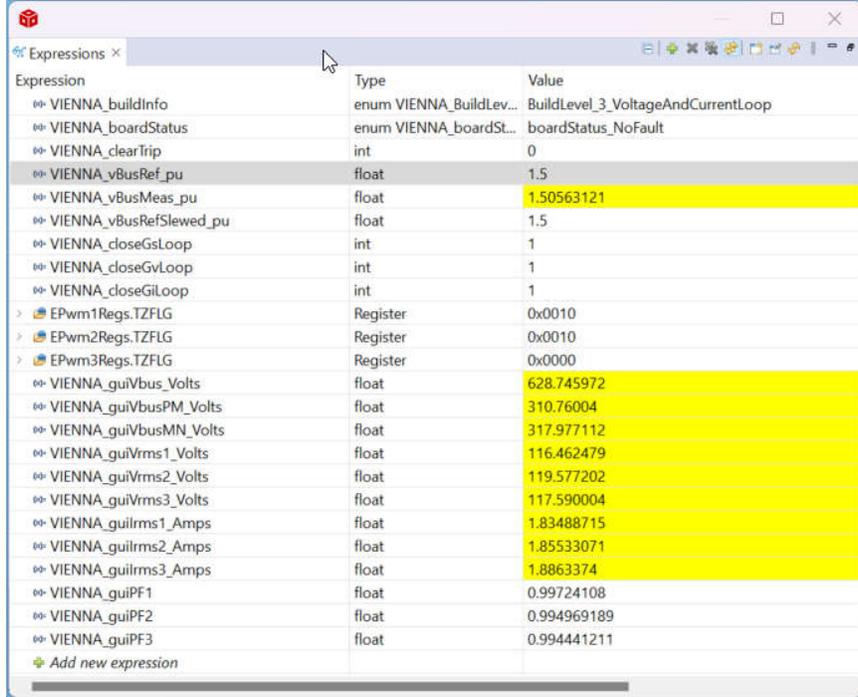
6. 図 3-25 に、入力電圧と入力電流の波形を示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

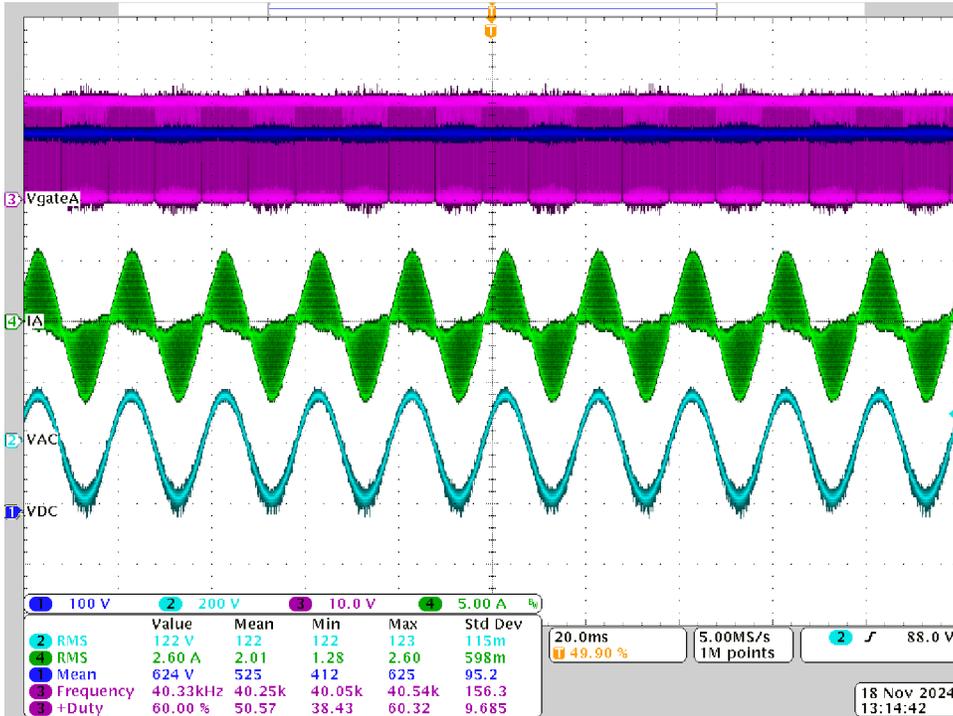
図 3-25. ビルド レベル 3:Vref = 1.0 時のスコープ キャプチャ Ia および Va (120V_{RMS} L-N)

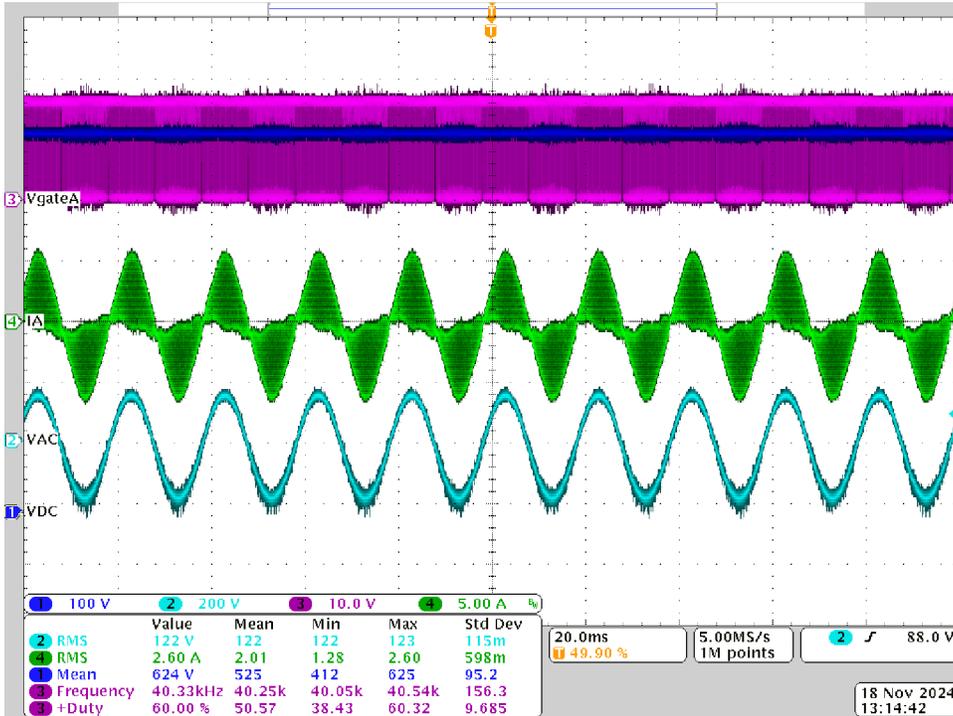
7. ここで、vBusRef を段階的に 1.5 まで上げると、バス電圧は 630V に上昇し、vBusRef と vBusMeas 変数が  のように [Expressions] ウィンドウに表示されます。



Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_3_VoltageAndCurrentLoop
VIENNA_boardStatus	enum VIENNA_boardSt...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_vBusRef_pu	float	1.5
VIENNA_vBusMeas_pu	float	1.50563121
VIENNA_vBusRefSlewed_pu	float	1.5
VIENNA_closeGsLoop	int	1
VIENNA_closeGvLoop	int	1
VIENNA_closeGILoop	int	1
EPwm1Regs.TZFLG	Register	0x0010
EPwm2Regs.TZFLG	Register	0x0010
EPwm3Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	628.745972
VIENNA_guiVbusPM_Volts	float	310.76004
VIENNA_guiVbusMN_Volts	float	317.977112
VIENNA_guiVrms1_Volts	float	116.462479
VIENNA_guiVrms2_Volts	float	119.577202
VIENNA_guiVrms3_Volts	float	117.590004
VIENNA_guiIrms1_Amps	float	1.83488715
VIENNA_guiIrms2_Amps	float	1.85533071
VIENNA_guiIrms3_Amps	float	1.8863374
VIENNA_guiPF1	float	0.99724108
VIENNA_guiPF2	float	0.994969189
VIENNA_guiPF3	float	0.994441211
Add new expression		

図 3-26. ビルドレベル 3:Vref = 1.5 の [Expressions] ウィンドウ

 3-25 に、入力電圧と入力電流の波形を示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

図 3-27. ビルドレベル 3:Vref = 1.5 時のスコープ キャプチャ Ia および Va (120V_{RMS} L-N)

8. システムを安全に停止させるには、入力 AC 電圧をゼロまで下げ、guiVBus 変数もゼロに下がっていることを確認します。
9. リアルタイム モードのマイクロコントローラを完全に停止するには、2 段階の手順を踏みます。まず、ツールバーの [Halt] ボタン () を使用するか、[Target] → [Halt] の順にクリックして、プロセッサを停止します。次に、  基板を上をクリックし、マイクロコントローラをリアルタイム モードから解除します。最後に、マイクロコントローラ () をリセットします。
10. [Terminate Debug Session] ([Target] → [Terminate all]) をクリックして、CCS デバッグセッションを終了します。 

3.2.2.4.4 INCR_BUILD 4: 閉ループ、電圧、および電流ループ

このビルドでは、ボードは 3 線式システムとして動作し、電源のニュートラルは出力の DC 中点に接続されていません。  3-28 に示すように、DC バス バランスを維持するため、制御構造に単純な比例ゲインのバランスループを追加しています。このビルドでは、3 次高調波注入も行われ、DC バスのバランス点を安定させるのに役立っています。

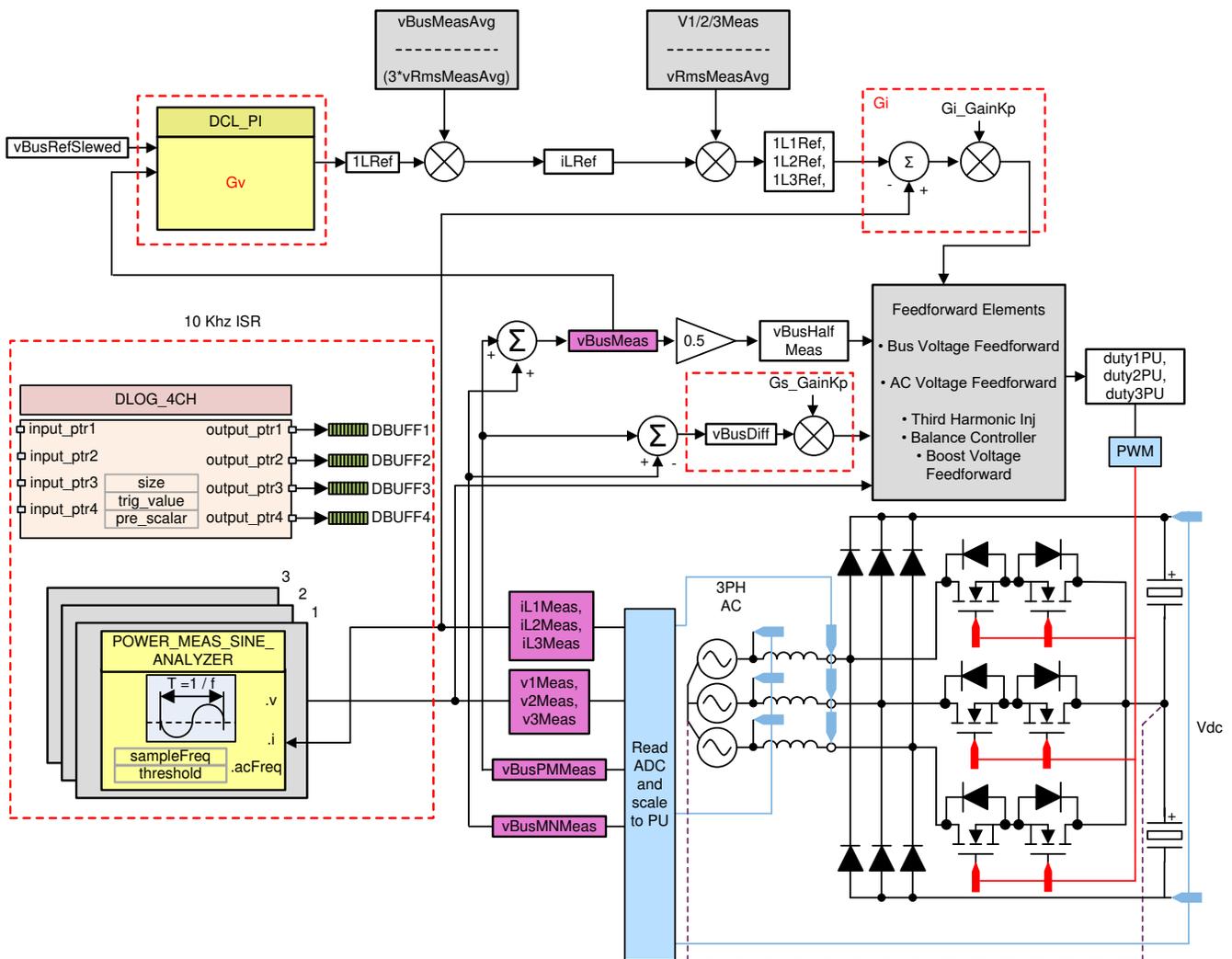
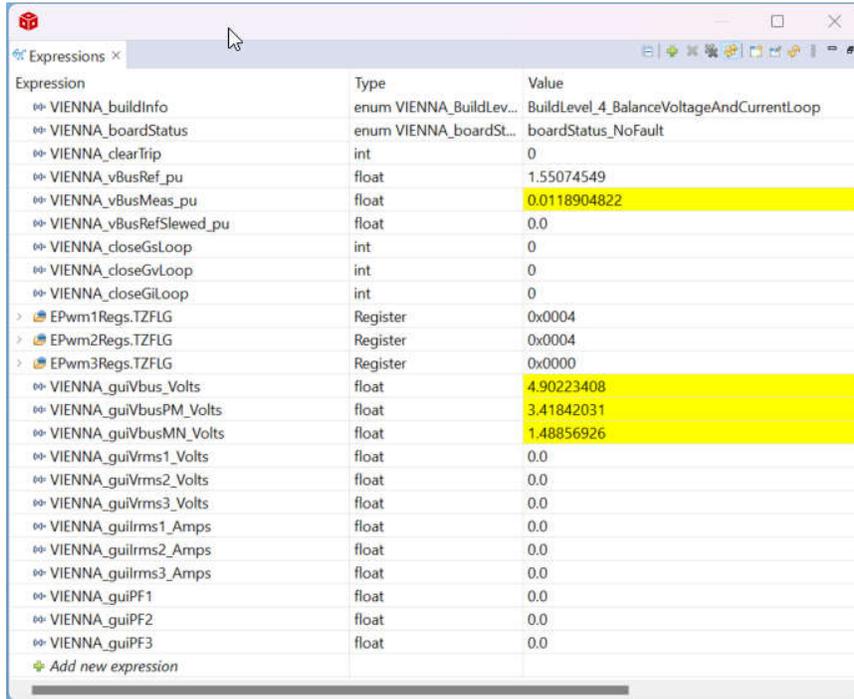


図 3-28. ビルド レベル 4 の制御図: 出力電圧、インダクタ電流、バス コンデンサのバランス ループ

3.2.2.4.4.1 プロジェクトのビルドおよびロードとデバッグの設定

1. `vienna_settings.h` ファイルを開き、`VIENNA_INCR_BUILD` を 4 に、`VIENNA_VBUS_REF_SET_VOLTS` を 650 に設定します。
2. 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。**[Run]** → **[Debug]** をクリックして、デバッグ セッションを開始します。デュアル CPU デバイスの場合、CPU を選択する必要がありますことを示すウィンドウが表示されたら、デバッグを実行します。ここでは、**CPU1** を選択します。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。メイン ルーチンの開始時にコードは停止します。
3. **[Watch]** または **[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_build4.js` スクリプト ファイルを参照します。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。**[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 3-29](#) に **[Watch]** ウィンドウを示します。



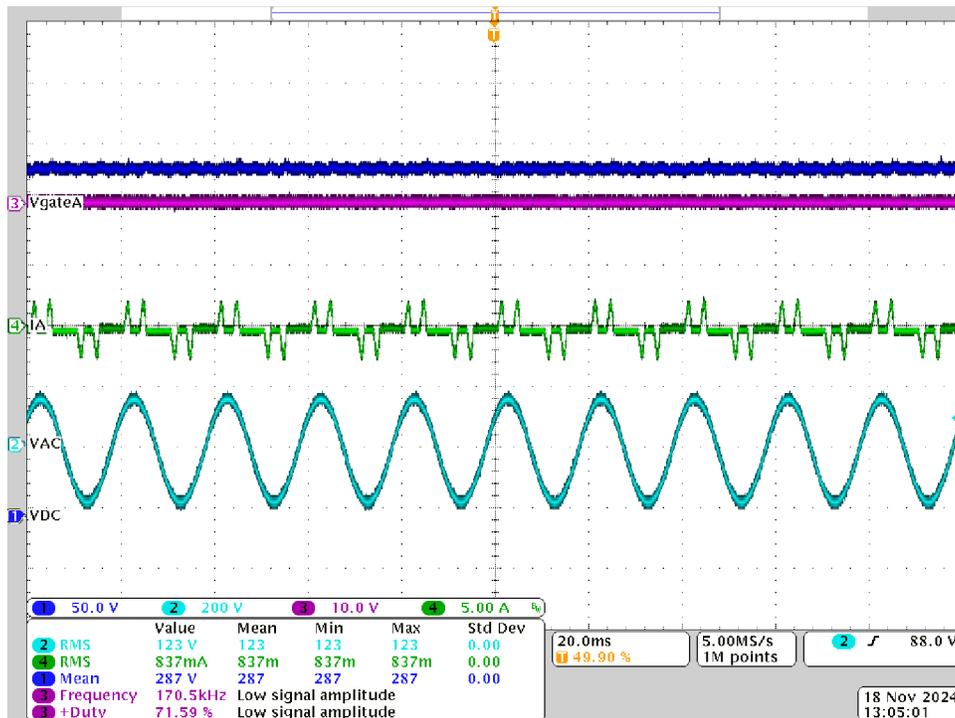
Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_4_BalanceVoltageAndCurrentLoop
VIENNA_boardStatus	enum VIENNA_boardSt...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_vBusRef_pu	float	1.55074549
VIENNA_vBusMeas_pu	float	0.0118904822
VIENNA_vBusRefSlewed_pu	float	0.0
VIENNA_closeGsLoop	int	0
VIENNA_closeGvLoop	int	0
VIENNA_closeGiLoop	int	0
EPwm1Regs.TZFLG	Register	0x0004
EPwm2Regs.TZFLG	Register	0x0004
EPwm3Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	4.90223408
VIENNA_guiVbusPM_Volts	float	3.41842031
VIENNA_guiVbusMN_Volts	float	1.48856926
VIENNA_guiVrms1_Volts	float	0.0
VIENNA_guiVrms2_Volts	float	0.0
VIENNA_guiVrms3_Volts	float	0.0
VIENNA_guiIrms1_Amps	float	0.0
VIENNA_guiIrms2_Amps	float	0.0
VIENNA_guiIrms3_Amps	float	0.0
VIENNA_guiPF1	float	0.0
VIENNA_guiPF2	float	0.0
VIENNA_guiPF3	float	0.0
Add new expression		

図 3-29. ビルドレベル 4: Expressions ビュー

4. マウス ポインタを水平ツールバーのボタンの上に置き、  ボタンをクリックして、リアルタイム モードを有効にします。

3.2.2.4.4.2 コードの実行 (ビルド 4)

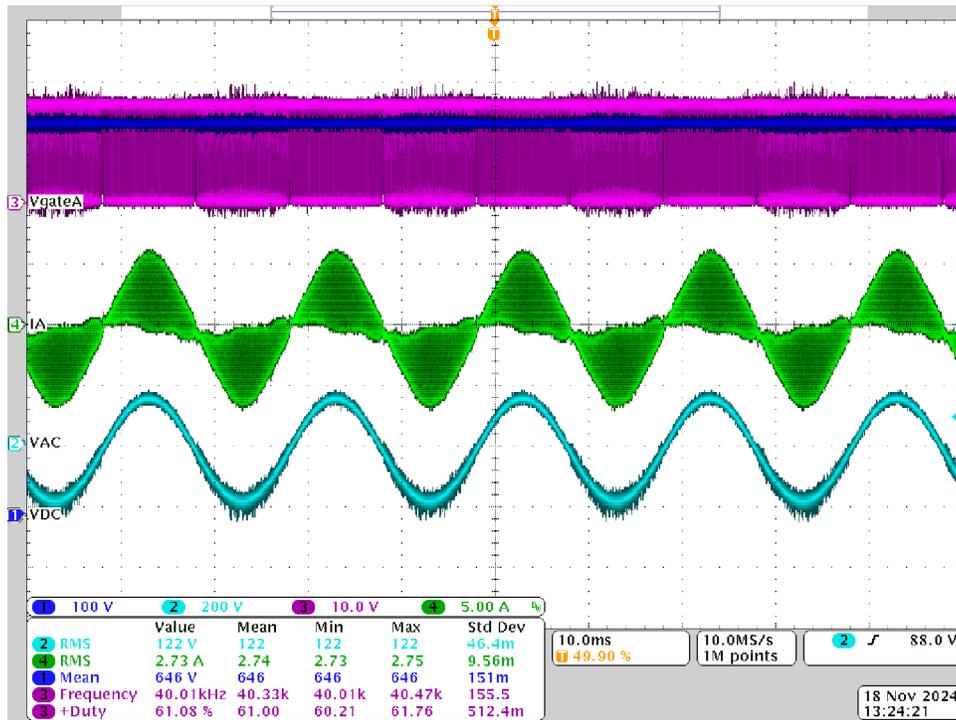
1.  ボタンをクリックして、プロジェクトを実行します。
2. AC 入力を $120V_{RMS}$ V_{L-L} と $208V_{RMS}$ V_{L-L} 、50/60Hz に上げます。図 3-30 に、入力から引き出される整流電流を示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

図 3-30. ビルドレベル 4:PWMトリップ時のスコープ キャプチャ Ia および Va ($120V_{RMS}$ L-N)

3. バス電圧は `vBusRef` 変数で設定されており、すでに約 1.55V で、このデザインでは 650V に相当します。
4. `clearTrip` 変数に 1 を書き込み、PFC 動作を開始します。
5. これで、基板は正弦波電流を流すようになります。図 3-31 に、スコープ キャプチャを示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH3 (ピンク):IGBT ゲート電圧
- CH4 (緑):AC 入力相 A 電流

図 3-31. ビルドレベル 4:フル PFC 時のスコープ キャプチャ Ia および Va (120VRMS L-N)

6. 図 3-32 に示す [Expressions] ウィンドウを確認します。DC バス電圧も平衡しています。つまり、guiVbusPM 変数と guiVbusMN 変数がほぼ等しく、これは閉ループ バランス コントローラが動作していることを示しています。

Expression	Type	Value
VIENNA_buildInfo	enum VIENNA_BuildLev...	BuildLevel_4_BalanceVoltageAndCurrentLoop
VIENNA_boardStatus	enum VIENNA_boardSt...	boardStatus_NoFault
VIENNA_clearTrip	int	0
VIENNA_vBusRef_pu	float	1.55074549
VIENNA_vBusMeas_pu	float	1.55194521
VIENNA_vBusRefSlewed_pu	float	1.55074549
VIENNA_closeGsLoop	int	1
VIENNA_closeGvLoop	int	1
VIENNA_closeGiLoop	int	1
EPwm1Regs.TZFLG	Register	0x0010
EPwm2Regs.TZFLG	Register	0x0010
EPwm3Regs.TZFLG	Register	0x0000
VIENNA_guiVbus_Volts	float	650.097412
VIENNA_guiVbusPM_Volts	float	324.564972
VIENNA_guiVbusMN_Volts	float	325.506866
VIENNA_guiVrms1_Volts	float	116.402916
VIENNA_guiVrms2_Volts	float	119.610641
VIENNA_guiVrms3_Volts	float	118.185844
VIENNA_guiIrms1_Amps	float	1.97930729
VIENNA_guiIrms2_Amps	float	2.00230312
VIENNA_guiIrms3_Amps	float	2.00848341
VIENNA_guiPF1	float	0.997370422
VIENNA_guiPF2	float	0.995169103
VIENNA_guiPF3	float	0.994325578

図 3-32. ビルドレベル 4:120VAC および 650VDC の [Expressions] ウィンドウ

7. バランス ループの開ループ ゲインは **GS_GainKp** 変数によって制御され、帯域幅が十分でない場合は調整できません。ただし、バランス ループの帯域幅は外部電圧ループよりも低くする必要があり、**1Hz~2Hz** 程度で十分です。
8. システムを安全に停止させるには、入力 AC 電圧をゼロまで下げ、**guiVBus** 変数もゼロに下がっていることを確認します。
9. リアルタイム モードのマイクロコントローラを完全に停止するには、**2 段階**の手順を踏みます。まず、ツールバーの [Halt] ボタン () を使用するか、[Target] → [Halt] の順にクリックして、プロセッサを停止します。次に、 ボタンをクリックして、マイクロコントローラをリアルタイム モードから解除します。最後に、 ボタンを選択して、マイクロコントローラをリセットします。
10. [Terminate Debug Session] ([Target] → [Terminate all]) をクリックして、CCS デバッグセッションを終了します。

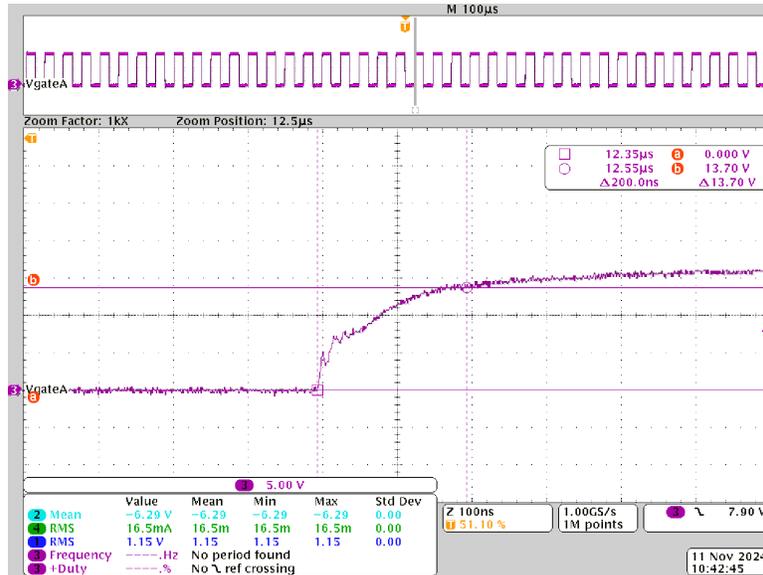


3.3 テスト結果

セクション 3.3.1 からセクション 3.3.11 に、電流と電圧の波形、熱性能などのテスト結果を示します。

3.3.1 IGBT ゲートの立ち上がり時間および立ち下がり時間

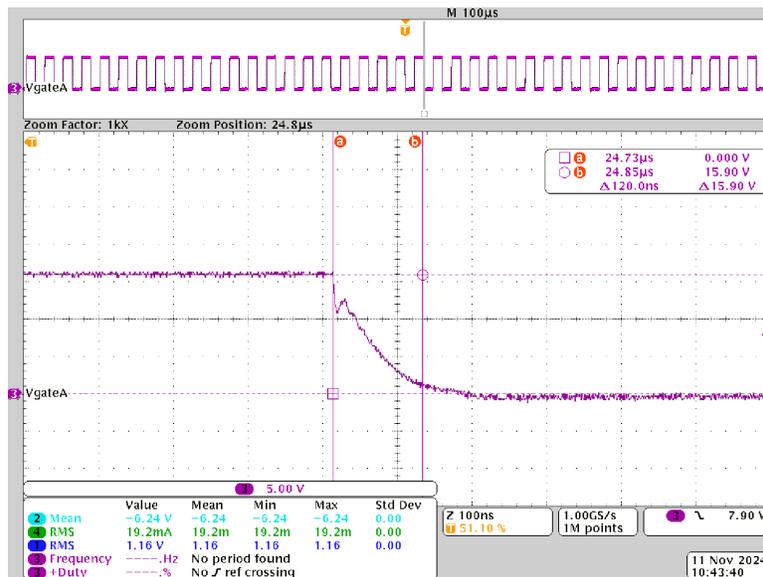
UCC5350MC は 10A ゲートドライバです。このデバイスは IGBT ゲートを非常に高速で駆動するため、IGBT の電力損失を改善するのに役立ちます。図 3-33 に、IGBT ゲート電圧の立ち上がり時間 200ns、立ち下がり時間 120ns を示します。



- CH3 (ピンク):IGBT ゲート電圧

図 3-33. IGBT ゲートの立ち上がりエッジ

図 3-34 に、IGBT ゲート電圧の立ち下がり時間の約 120ns を示します。

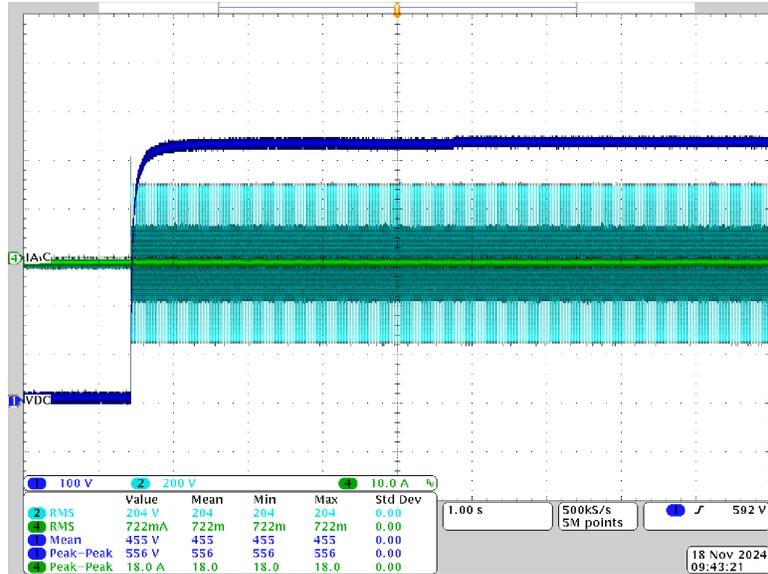


- CH3 (ピンク):IGBT ゲート電圧

図 3-34. IGBT ゲートの立ち下がりエッジ

3.3.2 パワーオンシーケンス

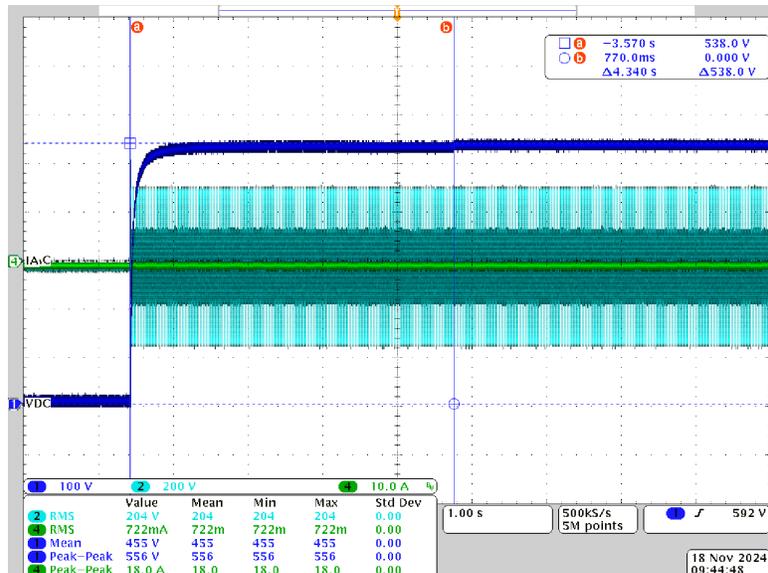
パワーオンの後、電解コンデンサ (C15、C16、C24、C25) の充電電流は PTC RT1 によって制限されます。DCBUS 電圧が十分に高くなると、チップが動作するための低電圧レールを供給するために、補助電源と絶縁型電源が動作を開始します。また、LED も点滅を開始し、遅延後にリレーが作動します。基板の電源を入れる際、DC 負荷が 800Ω 未満であることを確認してください。そうでない場合、大きな突入電流によって RT1 がトリガされます。図 3-35 に、380VAC で電源投入時の基板の波形を示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-35. パワーオンシーケンス

DCBUS 電解コンデンサの充電時間、補助電源の動作開始時間、MCU リセット、リレー閉動作の遅延を含む総パワーオン時間。図 3-36 に、380VAC 入力時の約 4 秒のパワーオン時間を示します。



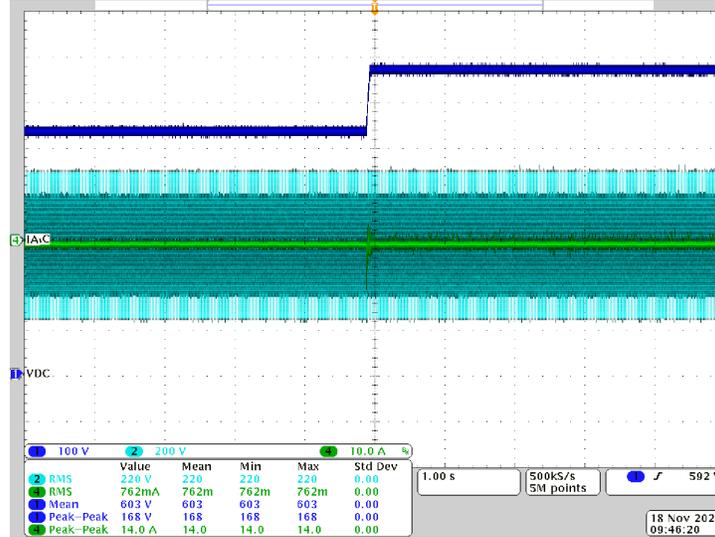
- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧

- CH4 (緑):AC 入力相 A 電流

図 3-36. パワーオン時間

3.3.3 GUI による PFC の開始

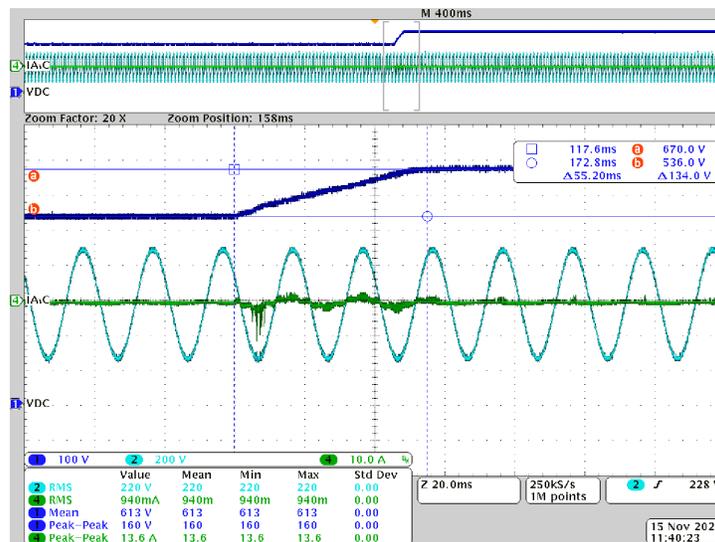
パワーオンの後、DCBUS 電圧は約 530VDC になります。PFC が開始されると、DCBUS 電圧は約 680VDC になります。図 3-37 に、GUI コマンド 0x11 を使用して PFC を開始したときの波形を示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-37. GUI による PFC の開始

PFC が開始されると、無負荷時に DCBUS は約 530V から約 680VDC に上昇します。図 3-38 に、立ち上がり時間が約 55ms であることを示します。

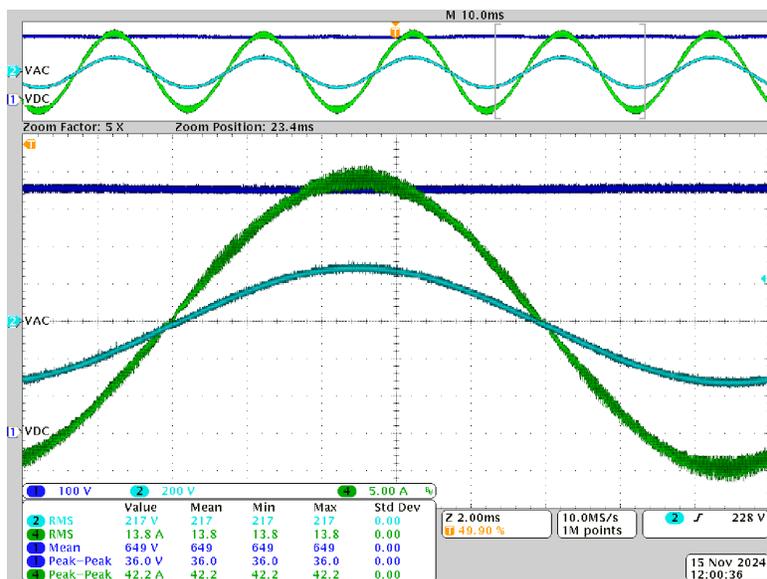


- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-38. PFC 開始時の DCBUS 立ち上がり時間

3.3.4 380VAC、9kW 時のゼロクロス

図 3-39 に、AC 入力電圧と電流のゼロクロス時間が一致していることを示します。

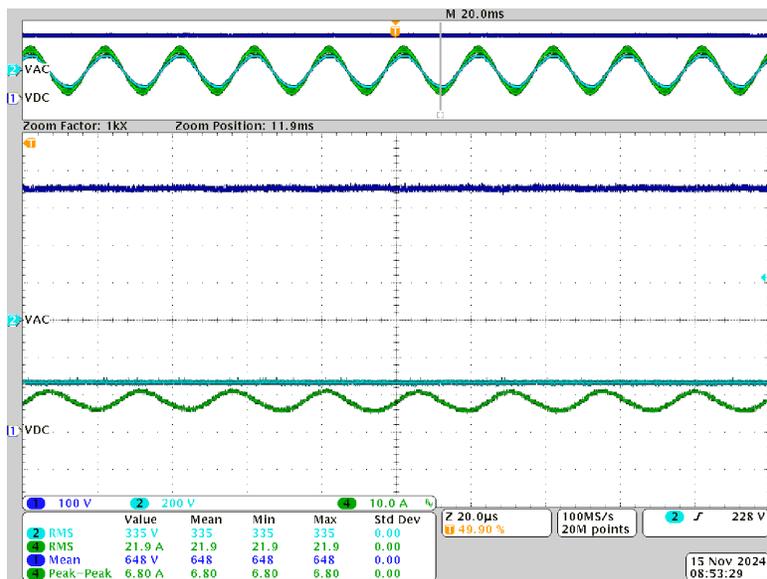


- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-39. 380VAC、9kW 時のゼロクロス時間

3.3.5 380VAC、10kW 時の電流リップル

図 3-40 に、380VAC、10kW における 6.8A の昇圧インダクタ電流リップルを示します。これは、ピーク AC 入力電流 21.9A の約 ±15.5% です。

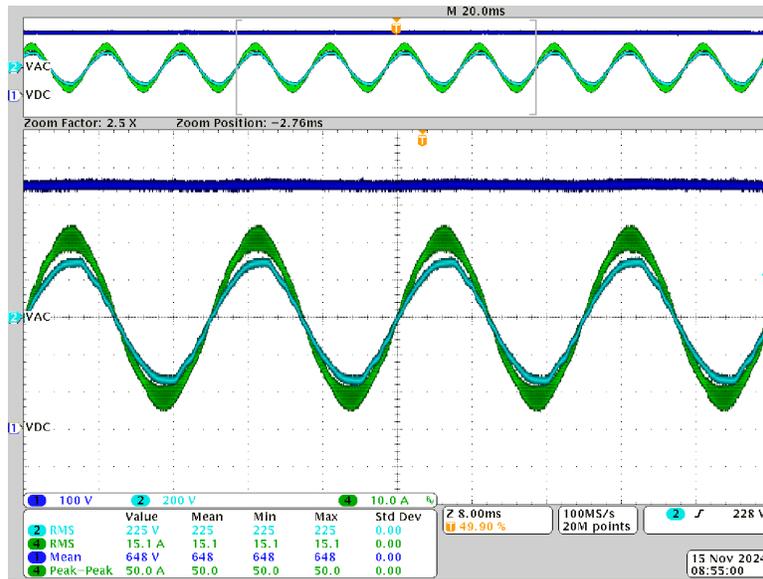


- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-40. 380VAC、10kW 時の電流リップル

3.3.6 グリッド電力での 10kW 負荷テスト

図 3-41 に、グリッド電力を使用した 10kW 時の負荷テスト波形を示します。

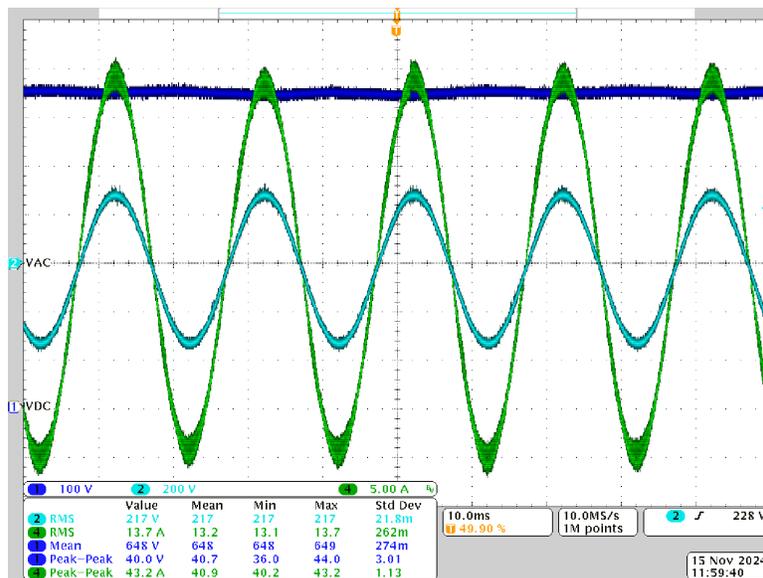


- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-41. グリッド電力での 10kW 負荷テスト

3.3.7 AC 電源での 9kW 負荷テスト

図 3-42 に、AC 電源を使用した 9kW 負荷テストを示します。



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-42. AC 電源での 9kW 負荷テスト

3.3.8 電力アナライザの結果

図 3-43 に、10kW のグリッド電力下での電力アナライザの結果を示します。

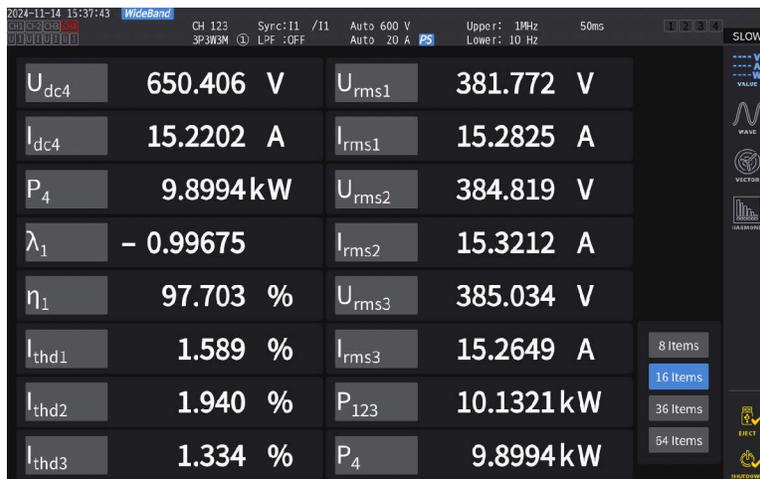


図 3-43. グリッド電力 10kW 時の電力アナライザの結果

図 3-44 に、9kW の AC 電源下での電力アナライザの結果を示します。

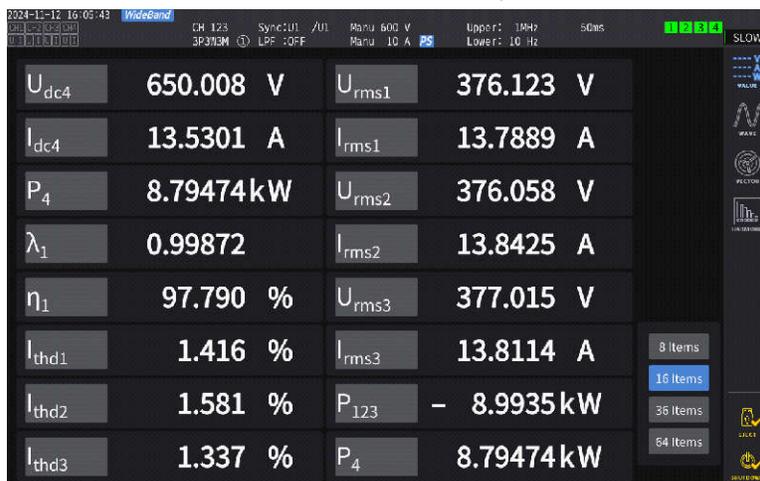


図 3-44. AC 電源 9kW 時の電力アナライザの結果

3.3.9 熱性能

図 3-45 に、冷却ファンなしの開放的な空間における、10kW 負荷、25°C 周囲温度での熱性能を示します。MOSFET Q8 の温度上昇が最も高く、約 55°C です。補助電源の場合、昇圧インダクタの温度上昇は約 23°C であり、IGBT の温度上昇は約 40 °C です。

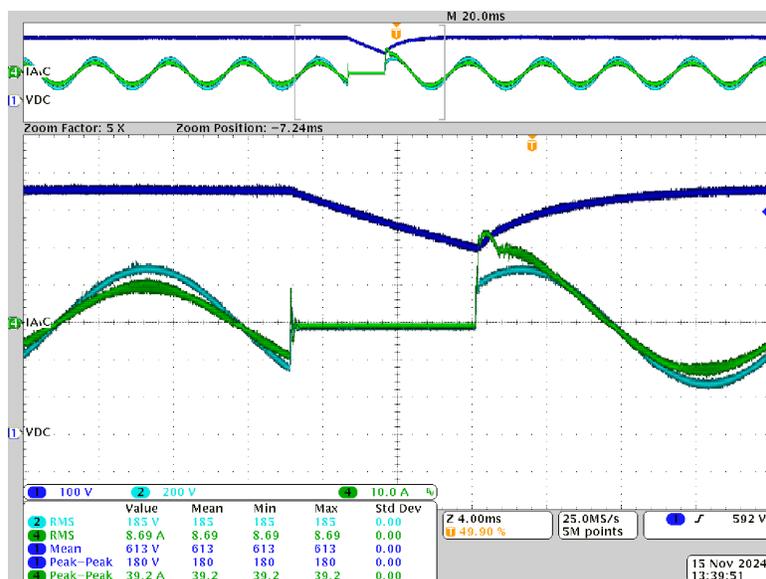
パワーデバイスの温度とヒートシンクの温度は、図 3-3 に示す GUI ソフトウェアからも読み取ることができます。



図 3-45. 熱性能

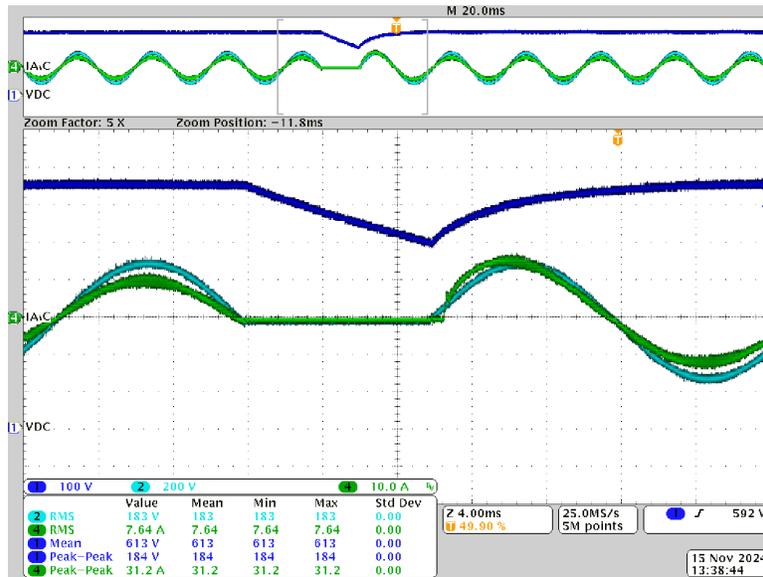
3.3.10 電圧短絡割り込みテスト

図 3-46、図 3-47、および図 3-48 に、90、0、45 の位相角における 5kW での電圧短絡割り込みテストを示します。昇圧インダクタの電流は、DSP 内部 CMPSS によって監視され、この電流が大きすぎる場合は、各 PWM サイクルで IGBT がオフになります。これは、サイクル単位の保護です。ただし、IGBT がオフになっている場合でもダイオードブリッジには制御されていない整流電流が存在するため、DSP はダイオードブリッジの電流を保護できません。



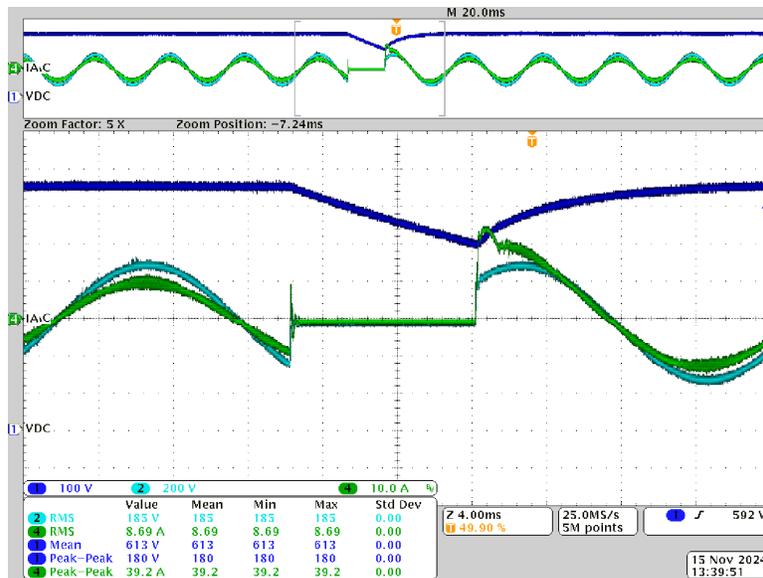
- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-46. 位相角 90 での電圧短絡割り込みテスト



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-47. 位相角 0 での電圧短絡割り込みテスト



- CH1 (青):DCBUS 出力電圧
- CH2 (水色):AC 入力相 A 電圧
- CH4 (緑):AC 入力相 A 電流

図 3-48. 位相角 45 での電圧短絡割り込みテスト

3.3.11 効率、iTHD、力率のテスト結果

図 3-49 に、325VAC、380VAC、400VAC における各電力レベルでの平均効率テスト結果を示します。平均ピーク効率は、380VAC および 400VAC 時、3kW~4.5kW で 98% です。

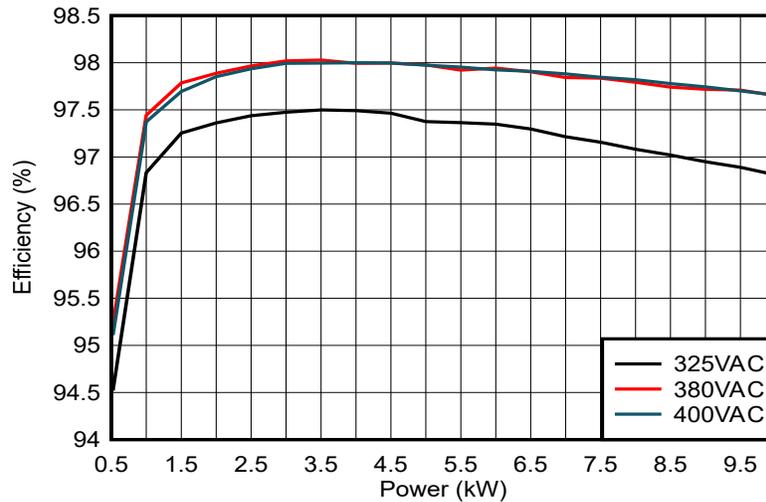


図 3-49. 効率のテスト結果

図 3-50 に、325VAC、380VAC、400VAC における各電力レベルでの平均 iTHD テスト結果を示します。平均 iTHD は、380VAC 時、> 2.5kW で 5%未満です。iTHD は、380VAC 時、10kW で 1.5% 未満です。

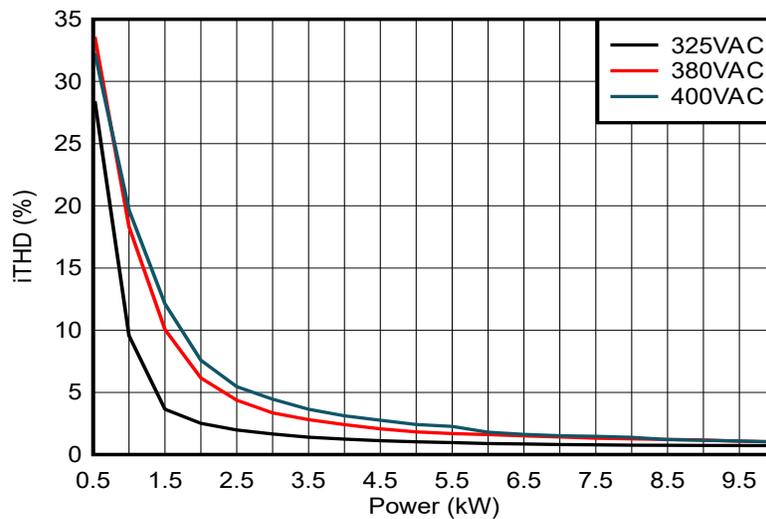


図 3-50. iTHD のテスト結果

図 3-51 に、325VAC、380VAC、400VAC における各電力レベルでの平均力率テスト結果を示します。

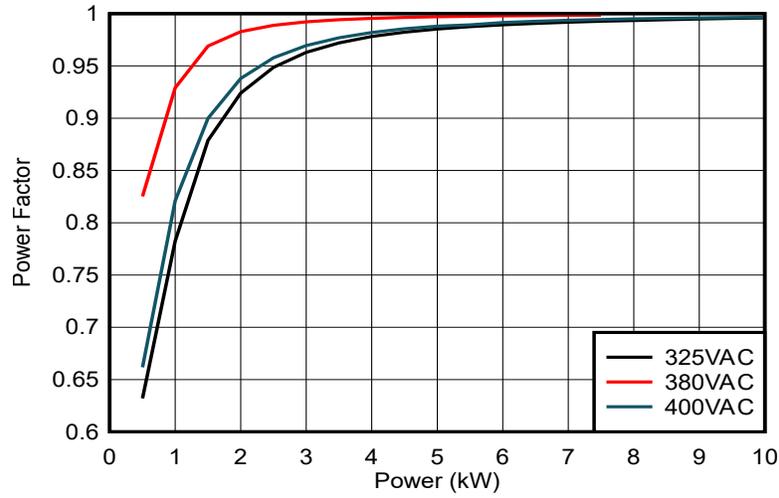


図 3-51. 力率のテスト結果

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-010257](#) のデザイン ファイルを参照してください。

4.1.2 部品表 (BOM)

BOM をダウンロードするには、[TIDA-010257](#) のデザイン 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

[LAUNCHXL-F2800137](#) TMS320F2800137 C2000™ リアルタイム マイコン向け LaunchPad™ 開発キット

ソフトウェア

[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、『[TMS320F280013x](#) 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、『[TMS320F280013x](#) リアルタイム マイクロコントローラ テクニカル リファレンス マニュアル』
3. テキサス・インスツルメンツ、『[C2000™](#) ソフトウェア周波数応答アナライザ (SFRA) ライブラリおよび補償デザイナー』ユーザー ガイド

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™, 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

HELY ZHANG はテキサス・インスツルメンツのシステム アプリケーション エンジニアで、家電製品に関連する電力供給とモーター インバータの開発を担当しています。Hely は 2002 年に安徽理工大学からパワー エレクトロニクスの修士号を取得し、テキサス・インスツルメンツに入社する前は SolarEdge と General Electric で働いていました。

JIM CHEN はテキサス・インスツルメンツのシステム エンジニアで、家電製品のシステム設計の仕様策定と開発を担当しています。Jim は、2022 年に南京航空航天大学で電気工学の修士号を取得しています。

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 © 2025, Texas Instruments Incorporated