参考資料

Design Guide: TIDA-010948

位置フィードバックと産業用通信プロトコル機能を搭載した 6 軸モーター制御のリファレンス デザイン

概要

このリファレンス デザインは、TI Sitara™ MCU-AM243x デバイスを 2 個使用して、6 軸制御機能を搭載した、シンプルでオープンなリアルタイム イーサネット ギガビット (SORTE_G) を接続したモーター ドライブを取り扱う能力を提示します。このデザインは、1 個の AM243x デバイスに 1 個の 800MHz R5F コアを搭載しており、インクリメンタル エンコーダを使用する 6 個の独立したモーターに対して、62.5µs のサイクル時間で閉ループのフィールド指向性制御を実行できます。この AM243x の 1 つのプログラマブル リアルタイム・ユニット (PRU) コアが SORTE_G コントローラとして機能し、リクエストを送信して 6 軸のモーター角度を受信します。SORTE_G デバイスとして動作する別の AM243x デバイスの PRU コアは、6 軸のモーター角度を送信します。この角度が、この AM243x の他の PRU コアによってデコードされ、各サイクルでコントローラ側に送信されます。また、AM243x は、マルチ プロトコル アブソリュート エンコーダとマルチ プロトコル産業用イーサネットの両方をサポートすることもできます。

リソース

TIDA-010948、TIDEP-01032	デザイン フォルダ
AM243x モーター制御 SDK, LP-AM243	ツール フォルダ
BLDC ブースタパック™ プラグイン モジュール	ツール フォルダ
AM2434、TQ-3P-SOM-TQMA243XL	プロダクト フォルダ
DRV8316、AMC1035	プロダクト フォルダ
DP83869、THVD1450	プロダクト フォルダ

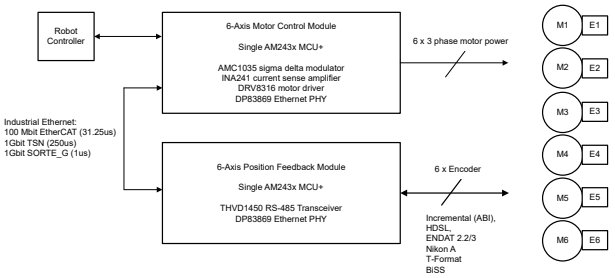
テキサス・インスツルメンツの E2E™ サポート エキスパートにお問い合わせください。

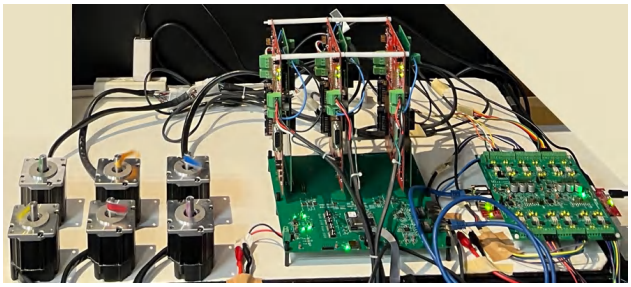
特長

- 6 つの FOC ループの出力に基づき PWM 波形を生成するための 18 チャネルの EPWM および 18 チャネルの ICSSG_PRU PWM
- 直接に接続された 6 個のモーターから位相電流フィードバックを行うために、PRU コア間で連続サンプリングと負荷共有を行う 12 チャネル シグマ デルタ フィルタリング ファームウェア
- ICSSG で実施された SORTE_G ファームウェア。EtherCAT®、PROFINET®、Gigbit TSN に対応する拡張オプションを搭載
- 6 つの独立した閉ループ FOC で、62.5µs サイクル時間で電流と速度を設定するのは可能
- 6 チャネルのインクリメンタル エンコーダ デコーディング、EQEP モジュールと PRU コア付き
- ICSSG を使用したマルチエンコーダ プロトコル用の拡張オプション。たとえば、EnDAT、HDSL、Biss C、Tamagawa や Nikon A など

アプリケーション

- ロボット向けサーボ ドライブ
- サーボ ドライブ位置フィードバック
- ロボット位置フィードバック アグリゲータ
- サーボ ドライブ制御モジュール





1 システムの説明

このリファレンス デザインでは、包括的な 6 軸リアルタイム サーボ モーター制御と産業用通信プロトコルを取り扱う AM243x デバイスの能力を示します。この実装は、2 つの TI AM243x SoC、1 つの AM243x ALV パッケージ、1 つの AM243x ALX パッケージで行われます。

AM243x ALV パッケージは、TQ-SOM 付きで制御ボード上に配置されており、EPWM および ICSSG_PRU PWM ペリフェラルの両方を使用して 36 の相補型 PWM 信号を生成し、12 チャネル デルタ シグマ フィルタ モジュールからの位相電流を測定します。この操作は、ICSSG ファームウェアを使用して実現されるほか、SORTE_G 経由で位置フィードバックボードと通信し、62.5µs サイクル時間ごとに 6 軸モーターの機械的角度に関するリクエストを送受信します。制御ボード上では、1 つの 800MHz R5F コアを使用してインクリメンタル エンコーダを備えた 6 つの独立したモーターで FOC ループを実現するか、複数のプロトコルを使用したアブソリュート エンコーダに拡張することも可能でうす。別の R5F コアを使用し、産業用イーサネット スタックをオプションとして実装できます。1 つの ICSSG1 がデルタ シグマ フィルタ モジュールとして使用され、もう 1 つの ICSSG0 は SORTE_G コントローラまたは追加オプション用の EtherCAT 2 次コントローラとして使用されます。

ALX パッケージを備えた LP_AM243 は、位置フィードバック ボード コントローラに使用されます。位置フィードバック ボード上の ICSSG0 は、産業用イーサネット ペリフェラル (IEP) を通じて ABI 信号の立ち上がり/立ち下がりエッジをキャプチャすることで、4 チャネル インクリメンタル エンコーダをデコードします。他の 2 チャネル エンコーダは、EQEP モジュールでデコードされます。ICSSG1 は SORTE_G デバイスとして使用され、62.5µs サイクル時間ごとに 6 軸モーターの機械的角度データを送信します。

1.1 用語

SoC	システム オン チップ
FOC	フィールド オリエンテッド コントロール
MCU	マイクロ プログラム制御ユニット
ALV、ALX	AM243x デバイスのパッケージ図
ABI	2 つの直交エンコードされた出力 A と B とインデックス I を持つインクリメンタル エンコーダ
RPM	回転数 / 分
LUT	ルック アップ テーブル
EnDAT、HDSL、BissC、Tamagawa、Nikon A	アブソリュート エンコーダ向けのマルチ デジタル型双方向インターフェイス プロトコル
EtherCAT	Ethernet for Control Automation Technology の略
Profinet	プロセスフィールドネットワークのためのかばん語
SORTE	シンプルかつオープンなリアルタイム イーサネット
ICSSG	産業用通信サブシステム ギガビット
PRU	プログラマブル リアルタイム ユニット
RTU	補助プログラマブル リアルタイム ユニット
SDFM	シグマ-デルタ フィルタリング モジュール
SDDF	シグマ デルタ デシメーション フィルタリング
SDM	シグマ-デルタ モジュレータ
IEP	産業用イーサネット ペリフェラル
CMP	イベント コンパレータ
CAP	イベントのキャプチャ
ISR	割り込みサービス ルーチン

EPWM	拡張パルス幅変調
EQEP	拡張型直交エンコーダ パルス
GPIO	汎用入出力
FIFO	先入れ先出し
TSR	時間同期ルータ (AM243x の汎用割り込みルータモジュールのインスタンス化)
TCM	密結合メモリ
DRAM	ダイナミック ランダム アクセス メモリ
RGMII	RGMII (Reduced Gigabit Media Independent Interface)
MII_G_RT	リアルタイムでメディア非依存のインターフェイス ギガビット
MDIO	管理データ入出力
TQ-SOM	社内製造サービスと設計サービスを提供する、TI の Arm® ベース プロセッサ向けシステム オン モジュールのベンダ

1.2 主なシステム仕様

表 1-1. 主なシステム仕様

サブシステム	仕様	備考
SoC EPWM	以下の機能を備えた 18 チャネルの相補型 PWM: • 構成可能なデッドバンド • シングルまたはダブル更新機能 • 可変スイッチング周波数: • 同期モードまたは位相シフト モード機能 • PRU-ICSS PWM および SDFM との同期	現在のデモでは、ターゲットは同期モードのみです。
ICSSG_PRU PWM	以下の機能を備えた 18 チャネルの相補型 PWM: • 構成可能なデッドバンド • シングルまたはダブル更新機能 • 可変スイッチング周波数: • 同期モードまたは位相シフト モード機能 • EPWM および SDFM と同期	現在のデモでは、ターゲットは同期モードのみです。
電流フィードバック – ICSS_SDFM	以下のものを含む位相電流フィードバック用の 12 チャネル SDFM（軸ごとに 2 チャネル）: • 通常電流 OSR:64 • RTU と PRU コアの間での負荷共有 • 連続サンプリング モード • シングルまたはダブル更新機能	
位置フィードバック – PRU EQEP	PRU コアと IEP_CAP を搭載したインクリメンタル ABI エンコーダから 4 チャネル モーターの角度をデコードします	
位置フィードバック – SoC EQEP	SoC EQEP モジュールを搭載したインクリメンタル ABI エンコーダからの 2 チャネル モーター角をデコードします	
制御アルゴリズム – FOC ループ	62.5µs サイクル時間内（PWM 周波数 16kHz）に電流と速度に対して 6 つの独立した FOC ループを達成	現在のデモでは、16kHz のダブルアップデートが目標となっています
制御アルゴリズム – 時間同期	制御ループ間（位置、速度、電流、PWM、産業用イーサネット）のタイミング同期	現在のデモでは、コントローラとデバイスの間の同期に SORTE_G と TSR が使用されています。
産業用通信 – SORTE_G	コントローラとデバイスの間のポイント ツー ポイントの通信。コントローラは、6 軸モーターの機械的角度に関するリクエストを送信し、ギガビット産業用イーサネット プロトコルを 62.5µs サイクル時間ごとにデータを受信します	SORTE_G を使用すると、EtherCAT や Profinet と比較して、レイテンシを短縮できます

2 システム概要

図 2-1 は、TIDA-010948 システムのセットアップを示します。本システムには、以下のような事項が含まれます。

- AM2434 ALV SoC をメインサーボ制御 MCU として使用した 6 軸制御ボードの TIDA-010948 リファレンス デザイン
- サーボ制御の電力段として 3 つの BP-AM2BLDCSERVO ボードと、各 BP-AM2BLDCSERVO ボードはデュアル軸をサポートできます
- 電力段から制御ボードに信号をルーティングするための 3 つのアダプターボード
- 6 チャネルエンコーダ信号を受信する 6 軸位置ボード
- モーターの角度と速度をデコードするための位置フィードバック MCU としての 1 つの AM2434 ALX SoC つき LP-AM243x

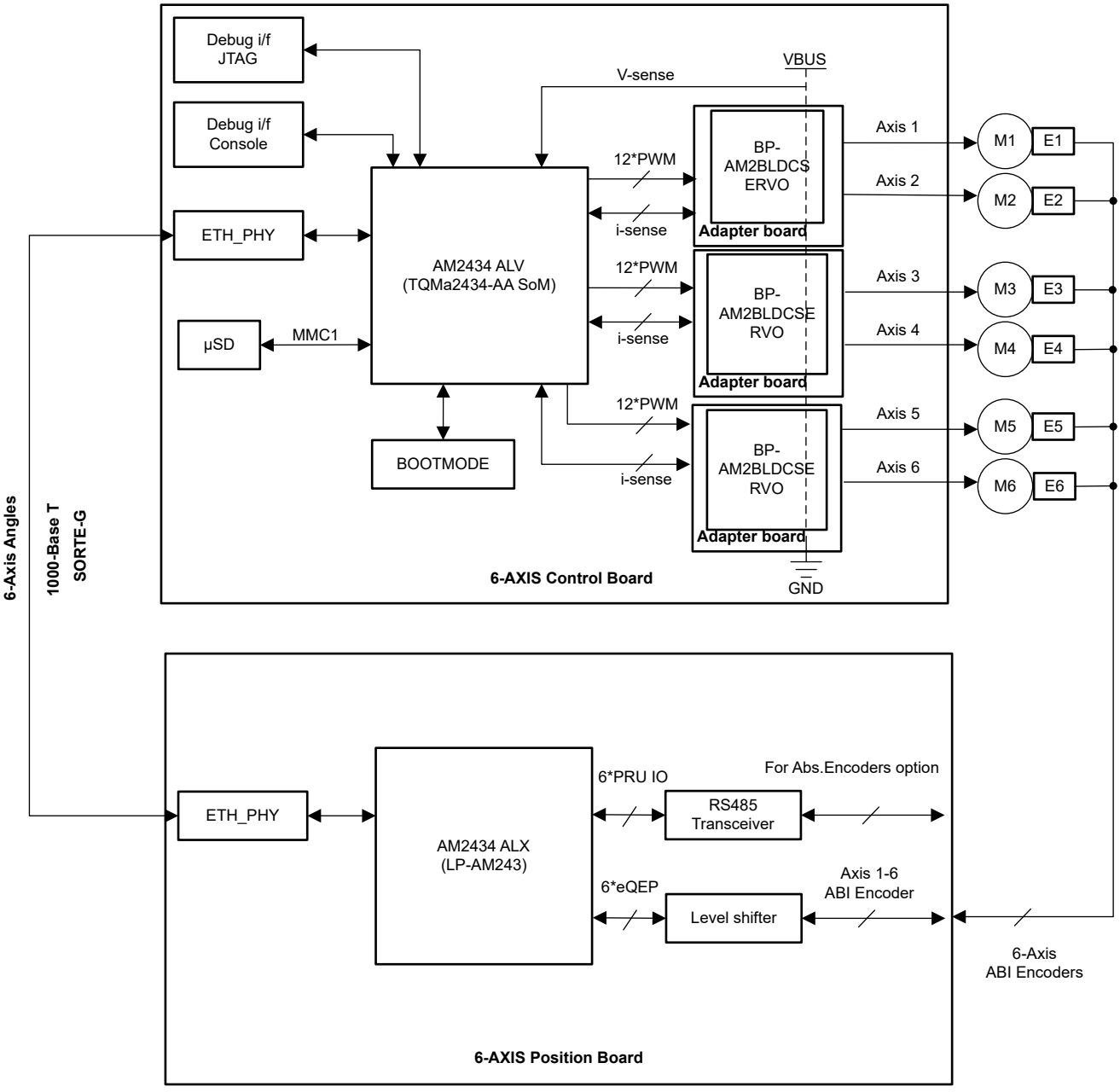


図 2-1. TIDA-010948 でのシステムセットアップ

2.1 ブロック図

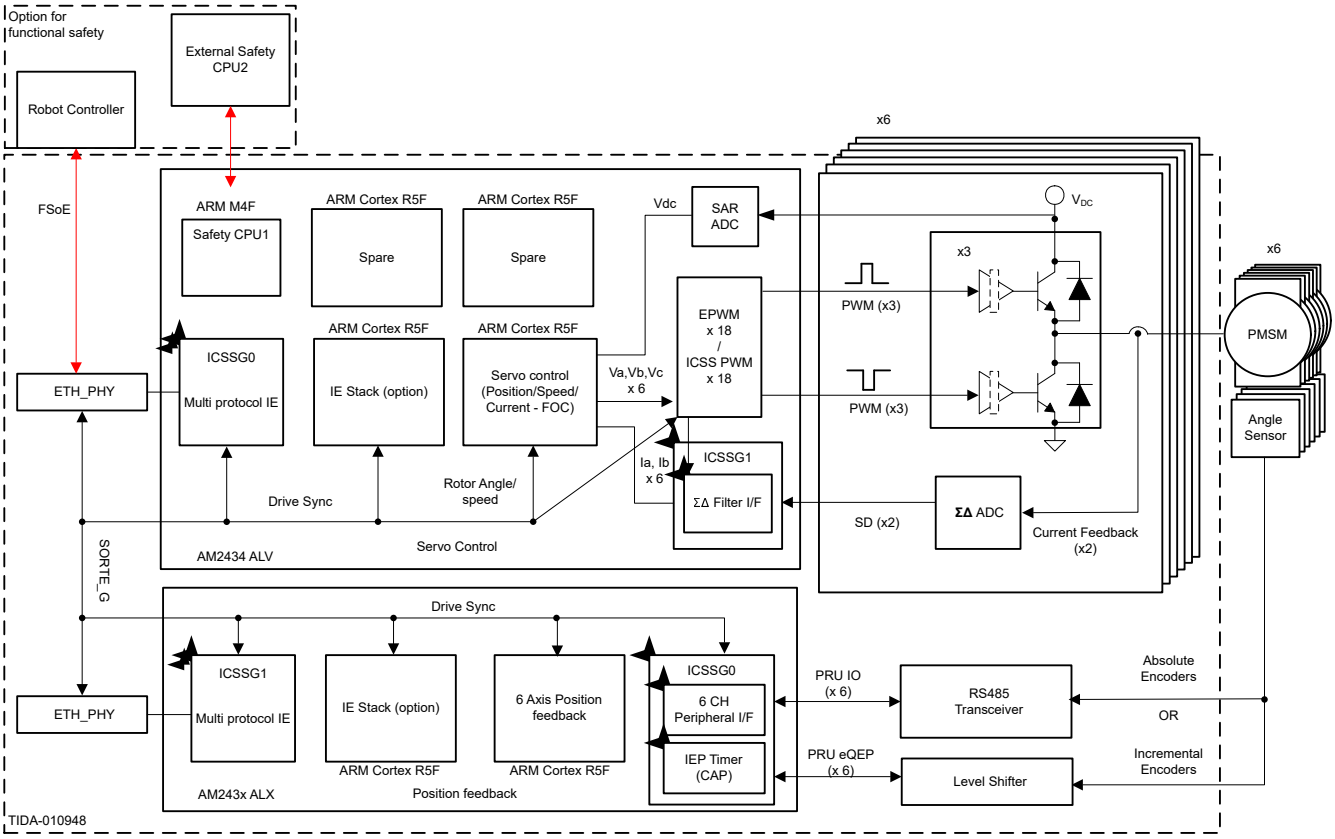


図 2-2. TIDA-010948 のシステムブロック図

2.2 設計上の考慮事項

シングルチップ 6 軸サーボ モーター制御の実装は、2 つのサブシステムで構成される集中型のリアルタイム パスを中心に設計されています。

- 以下を含む 6 軸制御ボード:
  - ICSSG0:拡張オプションに対応する SORTE_G コントローラ ファームウェアまたは EtherCAT セカンダリ コントローラのファームウェア。
  - R5FSS0_0:電流、速度、または位置のいずれかに対応できる 6 個の独立した閉ループ。エンコーダ付きの直接接続された 6 個のモーターに対応する、FOC 付き閉じループ。
  - R5FSS1_0:拡張オプションとして実装された EtherCAT セカンダリ スタック。
  - EPWM:3 軸 FOC ループの出力に基づいて波形を生成するための 18 チャネルの拡張 PWM ペリフェラル。
  - ICSSG1:直接接続された 6 個のモーターからの位相電流フィードバック向けに、slice0 と slice1 の両方にある RTU コアと PRU コアの間で連続サンプリングと負荷共有機能を持つデルタ シグマ フィルタリング ファームウェア。
  - ICSSG_PRU PWM (ICSSG1):他の 3 軸 FOC ループの出力に基づいて生成される、デッド バンド アサート付きの 18 チャネルの相補 ICSSG PWM 信号。
- 以下のものを含む 6 軸位置ボード:
  - ICSSG1:拡張オプションとしての SORTE_G デバイスのファームウェアまたは EtherCAT セカンダリ コントローラのファームウェア。
  - R5FSS0_0:システム初期化と LUT 生成。
  - R5FSS1_0:拡張オプションとして実装された EtherCAT セカンダリ スタック。
  - ICSSG0:4 チャネルのエンコーダ データをデコードするための PRU_eQEP ファームウェア。
  - EQEP:2 チャネル エンコーダのデータ デコーディング。

電力段は BP-AM2BLDCSERVO ボードを再利用します。詳細は、TIDEP-01032 を参照してください。計 3 つの BoosterPack™ プラグイン モジュール ボードと 3 つのアダプター ボードを備え、すべての AMC1035 デバイスから位相電流データを受信し、制御ボードから PWM 信号を送信し、すべての DRV8316 デバイスを駆動します。

図 2-3 は、IEP タイマと TSR モジュールを使用した SORTE_G のイーサネット タイムスタンプ機能を備えた位置ボードと制御ボードとの間の時間同期を示します。SORTE_G *IN* のパケット タイミングは確定的であり、サンプル時間と位置データとイーサネット遅延の計算結果に一致するように事前構成されています。パケット内の 1Gbps 64 バイトでは 1µs を超えませんが、イーサネット物理層での遅延とライン遅延のために追加で 1µs が必要となります。これにより、より多くの計算時間、またはダブル更新で PWM のサイクル時間を短縮できます。

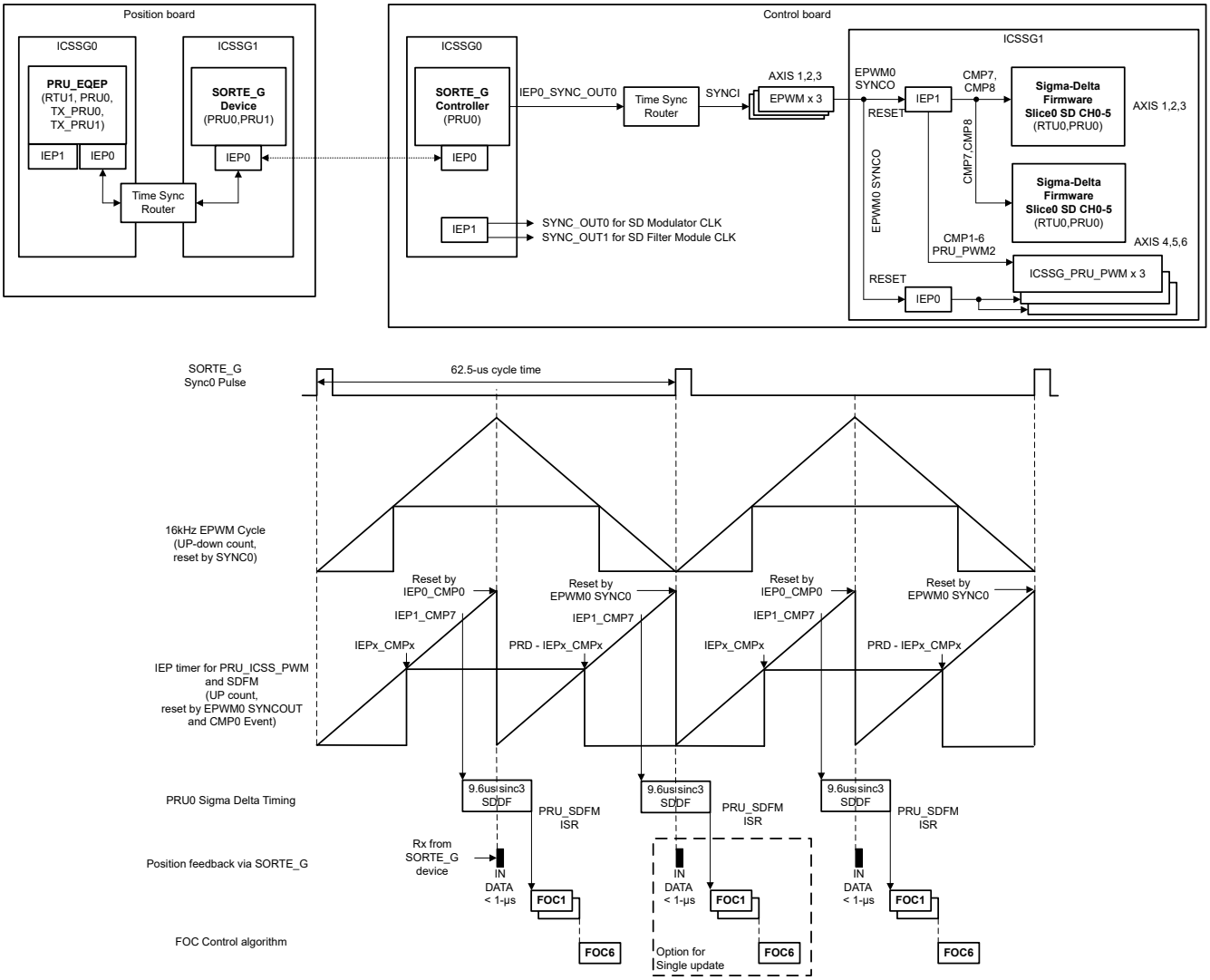


図 2-3. 時間同期およびトリガ FOC タイミング – 16kHz

2.3 主な製品 - AM243x サブシステム

2.3.1 制御ボード - SORTE_G コントローラ インターフェイス

SORTE_G コントローラ ファームウェア:

- PRU プロジェクト SORTE_g_master で生成された SORTE_g_master_PRU0.h。
- SORTE_G_MASTER プロジェクトのソースコード main_PRU0.asm には、次の関数が含まれています。
  - タスクマネージャにさまざまな状態をスケジュールします
  - ICSSG0 PRU0 レジスタを初期化します

  - RGMII インターフェイスと IEP クロックを構成します
  - イーサネット リンクのステータスを確認します
  - 検出パケットとパラメータ化パケットを生成し、同期の遅延を計算・保存し、データパケットを TX_L2 FIFO でデバイスに送信します

- sorte_g_master プロジェクトのソースコード rx.asm は、PRU0 ブロードサイド インターフェイスとレジスタ転送命令で receive 関数を実行します。

SORTE_G の構成と初期化:

- ソースコード sorte_g_app_tq_control_board.c を使用して、以下のための *IN* パケット格納アドレスを定義します。
  - 軸 1 は、TCM アドレス *0x78000808* でモーター角度データを受信しました
  - 軸 2 は、TCM アドレス *0x78000810* でモーター角度データを受信しました
  - 軸 3 は、TCM アドレス *0x78000818* でモーター角度データを受信しました
  - 軸 4 は、TCM アドレス *0x78000820* でモーター角度データを受信しました
  - 軸 5 は、TCM アドレス *0x78000828* でモーター角度データを受信しました
  - 軸 6 は、TCM アドレス *0x78000830* でモーター角度データを受信しました
- サイクル時間を 62.5µs、*IN* パケットオフセットを 31.25µs として構成します。ICSSG0、MDIO インターフェイス、IEP タイマを初期化し、その後、ファームウェアを読み込み、generic_pruss_init()関数を使用して PRU コアを有効にします。
- SORTE_G クライアントからの ICSSG0_IEP0_SYNCOUT0 は、EPWM ブロックを同期するために使用されます。

2.3.2 制御ボード – SDFM インターフェイス

図 2-4 は、シグマ デルタ フィルタ モジュールと変調器の両方に対するクロック ソースの分配を示します。表 2-1 は SDFM 信号を示します。

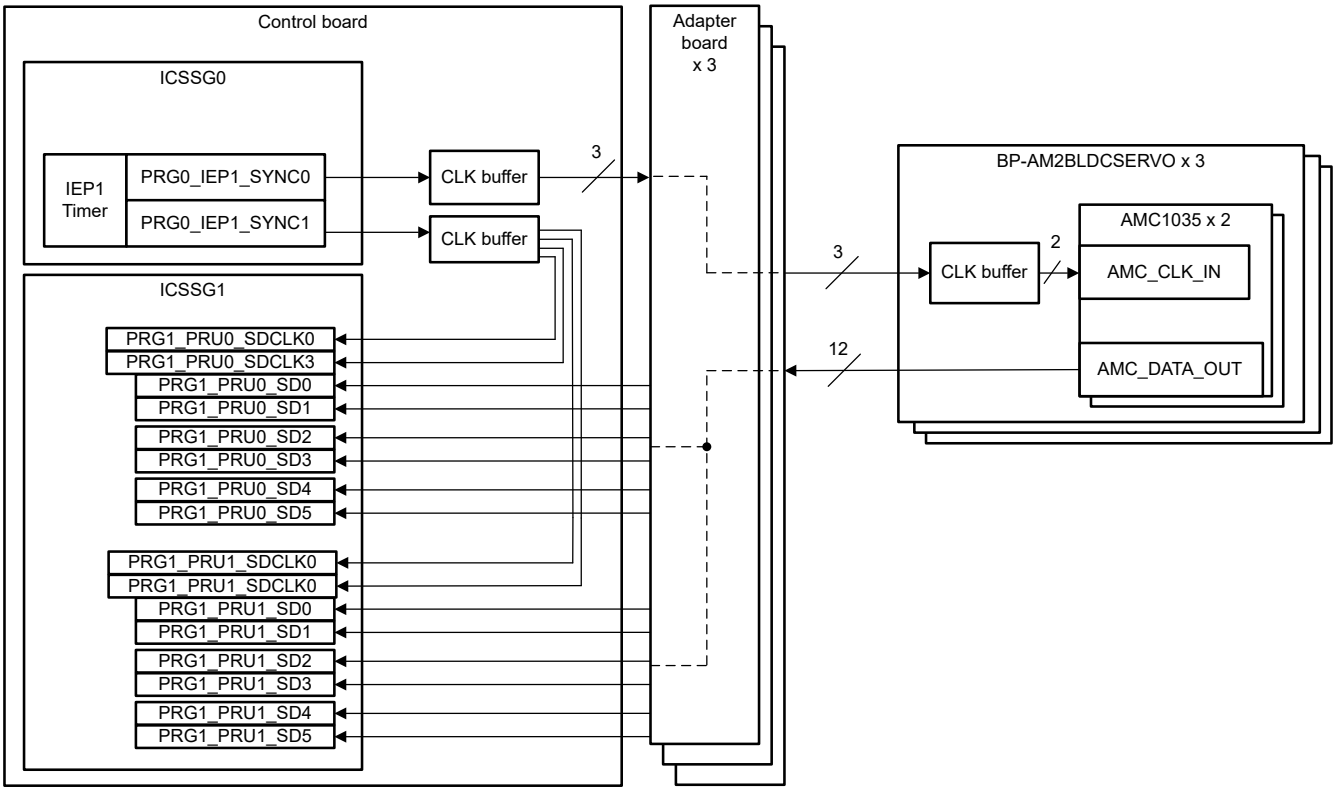


図 2-4. シグマ デルタ クロックとデータ分配

表 2-1. SDFM 信号

サブシステム	信号名	ピン名	AM243x ボールピン	TQ_SoM ピン
Axis1 - 相 A	DOUT_A1	PRG1_PRU0_SD0	U8	F5
Axis1 - 位相 B	DOUT_B1	PRG1_PRU0_SD1	V8	G3
Axis2 - 相 A	DOUT_A2	PRG1_PRU0_SD2	V13	H1
Axis2 - 位相 B	DOUT_B2	PRG1_PRU0_SD3	U13	H4
Axis3 - 相 A	DOUT_A3	PRG1_PRU0_SD4	U15	J2
Axis3 - 位相 B	DOUT_B3	PRG1_PRU0_SD5	AA8	J5
Slice0-SD0_CLK	Slice0_SD0_CLK	PRG1_PRU0_SD0_CLK	Y7	F4
Slice0-SD3_CLK	Slice0_SD3_CLK	PRG1_PRU0_SD3_CLK	AA7	H3
Axis4 - 相 A	DOUT_A4	PRG1_PRU1_SD0	V11	M4
Axis4 - 位相 B	DOUT_B4	PRG1_PRU1_SD1	Y12	N2
Axis5 - 相 A	DOUT_A5	PRG1_PRU1_SD2	AA13	N5
Axis5 - 位相 B	DOUT_B5	PRG1_PRU1_SD3	V15	P2
Axis6 - 相 A	DOUT_A6	PRG1_PRU1_SD4	V14	P5
Axis6 - 位相 B	DOUT_B6	PRG1_PRU1_SD5	AA10	R3
Slice1-SD0_CLK	Slice1_SD0_CLK	PRG1_PRU1_SD0_CLK	W11	M3
Slice1-SD3_CLK	Slice1_SD3_CLK	PRG1_PRU1_SD3_CLK	U11	P1
SDM_CLK_SOURCE	SDM_CLK	PRG0_IEP1_SYNC0	R2	A2
SDFM_CLK_SHIFT	SDFM_CLK	PRG0_IEP1_SYNC1	V5	B3

以下のリストは、SDFM について説明します。

- BoosterPack ボードにあるシグマ デルタ モジュレータのクロック ソースは、20MHz 付き ICSSG0 IEP1 の SYNC0 サイクリック出力によって供給されます。
- ICSSG1 のシグマ デルタ フィルタリング モジュール用のクロックソースは、ICSSG0 IEP1 の 20MHz 付き SYNC1 の周期的な出力によって供給されます。クロック選択値がオプション 2 に設定されます。これは、PRG1_PRUx_SD0_CLK が SD チャネル 0、1、2（SD0、SD1、SD2）用であることを意味します。PRG1_PRUx_SD3_CLK は SD チャネル 3、4、5（SD3、SD4、SD5）用で、ここでは slice0 および slice1 に関して x = 0 または 1 です。
- SDFM と SDM の間のクロック位相のシフトは、IEP SYNC0 と SYNC1 の間に遅延を設定して除去することができます。IEP の設定は、mclk_iep_sync.c の init_iep1_sync()関数で行います。

SDFM 割り込みは、以下のリストに定義されます。

- hwiPrms.intNum = ICSSG_PRU_SDDF_INT_NUM0、
- hwiPrms.callback = pruSddfIrqHandler0、

以下のリストは、SDFM 出力データバッファを示します。

- 軸 1、2、および 3 向け R5F_0_0 の TCMB の gSddfChSamps0[0-5]（ gSddfChSampsRaw0）
- 軸 4、5、および 6 向け R5F_0_0 の TCMB の gSddfChSamps[0-5]（ gSddfChSampsRaw）

以下のファイルは、SDFM ファームウェア用です。

- sdfm_rtu_bin.h は、スライス 0 の SD0、SD1、SD2 に対する PRU ファームウェアです
- sdfm_pru_bin.h は、スライス 0 の SD3、SD4、SD5 に対する PRU ファームウェアです
- sdfm_rtu1_bin.h は、スライス 1 の SD0、SD1、SD2 に対応する PRU ファームウェアです
- sdfm_pru1_bin.h は、スライス 1 の SD3、SD4、SD5 に対応する PRU ファームウェアです
- SDDF の初期化とファームウェアの読み込みは、、sdDF.c で行われます

SDFM パラメータ構成は、IEP クロック周波数、SD クロック周波数、最初のサンプルトリガ時間、SD クロックソースオプション、通常の電流オーバーサンプリングレート（OSR）を含め、軸 1、2、3 の構造 gTestSdfmPrms0 および 4、5、6 の構造 gTestSdfmPrms で設定されます。このデモでは、デフォルト設定では以下のものを使用しています。

- 250MHz IEP クロック
- 20.833333MHz SD クロック
- 26.45µs 最初のサンプリング トリガ時間
- トリガイベントは、RTU コアの場合は ICSSG1_IEP1_CMP7、PRU コアの場合は ICSSG1_IEP1_CMP8 です
- RTU と PRU コアの間の負荷共有
- SD クロックソースのオプション 2
- OSR 64 での通常電流サンプリング

詳細は、「AM243x モーター 制御 SDK：電流センス」を参照。

2.3.3 制御ボード - EPWM インターフェイス

表 2-2 は、軸 1、2、3 の EPWM*0*～*8* 信号を示します。

表 2-2. EPWM*0*～*8* 信号

サブシステム	信号名	ペリフェラル	AM243x ボールピン（ALV）	TQ_SoM ピン
Axis1_PWM	PWM_A1_H	EPWM6	B14	W21
	PWM_A1_L	EPWM6	A15	V19
	PWM_B1_H	EPWM8	V1	B9
	PWM_B1_L	EPWM8	W1	D9
	PWM_C1_H	EPWM7	W20	AA6
	PWM_C1_L	EPWM7	W21	AB6
Axis2_PWM	PWM_A2_H	EPWM5	T19	V9
	PWM_A2_L	EPWM5	W19	U5
	PWM_B2_H	EPWM4	R18	V7
	PWM_B2_L	EPWM4	T21	V6
	PWM_C2_H	EPWM3	V18	W5
	PWM_C2_L	EPWM3	Y21	Y5
Axis3_PWM	PWM_A3_H	EPWM2	V19	Y7
	PWM_A3_L	EPWM2	T17	AA7
	PWM_B3_H	EPWM1	U19	W8
	PWM_B3_L	EPWM1	V20	Y8
	PWM_C3_H	EPWM0	U20	AA9
	PWM_C3_L	EPWM0	U18	AB9

は以下のものを含む init_pwms()機能による EPWM 構成：

- 9 つの EPWM グループのすべてをデイジーチェーン接続として接続するために、SYNCI と SYNCO のマッピングを構成します。CTRLMMR_EPWM0_CTRL レジスタのビット[10–8]を 2h に設定すると、EPWM0_SYNCI は TSR モジュール出力の TIMESYNC_INTRTR0_OUT_38 によってトリガされます。時間同期ルータ入力 29 は出力 28 にルーティングされます。これは、SORTE_G コントローラの ICSSG0_IEP0_SYNC0 によって EPWM0 がトリガされることを意味します。また、EPWM0_event out は、*appEpwmCfg.cfgEt = TRUE* によって有効にされ、ICSSG_PRU_PWM および SDFM モジュールで使用されている IEP タイマをリセットします。CTRLMMR_EPWM3_CTRL」と CTRLMMR_EPWM6_CTRL レジスタのビット[10–8]をデフォルトから 1h に設定すると、デイジーチェーン接続として、EPWM3_SYNCI と EPWM6_SYNCI が EPWM2_SYNCO と EPWM5_SYNCO によってそれぞれトリガされます。

- *appEpwmCfg.epwmOutFreq = gEpwmOutFreq* による 16kHz に設定された EPWM 周波数。
- *appEpwmCfg.epwmTbCounterDir = EPWM_TB_COUNTER_DIR_UP_DOWN* によりアップダウンモードに設定された EPWM カウンタモード。
- EPWM デッドバンドは appEpwmCfg.cfgDb および appEpwmCfg.dbcfg.x のパラメータによって構成されます。
- App_epwmConfig()機能と出力データで計算された EPWM 周期および比較値は、すべての軸に対して gEpwmPrdVal です。
- *hwiPrms.intNum = PWM_C3_INTR*(EPWM0)によって構成された EPWM 割り込みとコールバック機能は *hwiPrms.callback =&App_epwmIntrISR* です。

EPWM 比較イベントは、FOC 計算結果に応じて writeCmpA()機能によって更新されます。*AM64x/AM243x テクニカルリファレンスマニュアル(TRM)*の *EPWM* モジュールセクションも参照してください。

2.3.4 制御ボード - ICSSG_PRU PWM インターフェイス

表 2-3 は、軸 4、5、6 の ICSSG_PRU PWM0-2 信号を示します。

表 2-3. ICSSG_PRU PWM0-2 信号

サブシステム	信号名	ペリフェラル	IEP_CMP	AM243x ボールピン (ALV)	TQ_SoM ピン
Axis4_PWM	PWM_A4_H	ICSSG1_PRU_PWM2	IEP1_CMP1	N16	U7
	PWM_A4_L		IEP1_CMP2	N17	U8
	PWM_B4_H		IEP1_CMP3	P17	V10
	PWM_B4_L		IEP1_CMP4	Y18	V4
	PWM_C4_H		IEP1_CMP5	V21	AB8
	PWM_C4_L		IEP1_CMP6	R16	W6
Axis5_PWM	PWM_A5_H	ICSSG1_PRU_PWM1	IEP0_CMP7	V10	R4
	PWM_A5_L		IEP0_CMP8	U10	T2
	PWM_B5_H		IEP0_CMP9	AA11	T3
	PWM_B5_L		IEP0_CMP10	Y11	T5
	PWM_C5_H		IEP0_CMP11	Y10	U1
	PWM_C5_L		IEP0_CMP12	AA14	U2
Axis6_PWM	PWM_A6_H	ICSSG1_PRU_PWM0	IEP0_CMP1	U9	K2
	PWM_A6_L		IEP0_CMP2	W9	K3
	PWM_B6_H		IEP0_CMP3	AA9	K5
	PWM_B6_L		IEP0_CMP4	Y9	L1
	PWM_C6_H		IEP0_CMP5	V9	L3
	PWM_C6_L		IEP0_CMP6	U7	L4

ICSSG_PRU PWM 構成 以下のものを含む app_pruicss_pwm.c の init_pruIcssPwm() 関数。

- API 関数 PRUICSS_PWM_stateInit()によって定義された、初期、アクティブ、トリップ状態が含む、PWM 信号出力状態。
- API 関数 PRUICSS_PWM_signalEnable() による PWM 信号の有効化。
- API 関数 PRUICSS_PWM_config() によって構成された、PWM 初期期間、デューティサイクル、デッドバンド。
- API 関数 PRUICSS_PWM_pruIcssPwmFrequencyInit() による PWM 周波数設定。

- PRUICSS_PWM_IEP_Config() 関数で構成された IEP タイマには、以下のことを行います。
  - IEP シャドウ モードを有効にします。
  - IEP クロックと PWM 周波数に基づき、CMP0 の値を PWM 周期として計算します。
  - EPWM0_SYNCO と IEP CMP0 イベントの両方で IEP リセットを有効にします。(IEP CMP0 の値は、ICSSG_PRU PWM 周期の半分よりも 1 クロックサイクル遅延があるため、IEP CMP0 イベントは、PWM タイマのゼロポイント *ではなく*、周期でのみ IEP をリセットします。)

ICSSG_PRU_PWM IEP_CMP0 割り込み App_pruicssIep1Compare0IrqSet() は、PWM 信号の次の立ち上がりエッジの比較イベントの値を FOC ループ計算の結果で更新するソフトウェアフラグを設定するために使用されます。

- hwiPrms.intNum = CSLR_R5FSS0_CORE0_INTR_CMP_EVENT_INTROUTER0_OUTP_16
- hwiPrms.callback = &App_pruIcssPwmHalfDoneIrq
- ICSSG1_IEP0 比較 0 イベント番号 TISCI_PRU_ICSSG1_IEP1_CMP0_SRC_INDEX のソース インデックスを 12U と定義します

PWM 信号の次の立ち下がりエッジの比較イベントは、EPWM0 ISR 内で更新されます。

詳細については、「*AM243x モーター制御 SDK: PRU-ICSS PWM デッドバンド EPWM 同期*」を参照。

2.3.5 制御ボード- ICSSG_PRU IEP タイマ

IEP の設定には、以下のパラメータが適用されます。

- ICSSG0:
  - IEP0 は SORTE_G コントローラ向けに 250MHz クロックに設定されます。
  - IEP0 SYNC_OUT0 で EPWM 時間ベースカウンタを同期させます。
  - IEP1 SYNC_OUT0 20MHz を SDM のクロックソースとして使用します。
  - IEP1 SYNC_OUT1 20MHz を SDFM のクロックソースとして使用します。
- ICSSG1:
  - ICSSG_PRU PWM の IEP クロックが 250MHz に設定されます(EPWM 時間ベースカウンタと同じ)。
  - IEP0 および IEP1 CMP0 は PWM 周期(250000000/16000/2)として 7812 に設定されます。
  - IEP0 CMP1 ～ CMP12 を軸 5 と 6 PWM の比較イベントとして使用します。
  - IEP1 Cmp1 ～ CMP6 を軸 4 の比較イベントとして使用します。
  - IEP1 CMP7 ～ CMP8 を SDFM の比較イベント(最初のサンプルトリガ)に使用します。

2.3.6 制御ボード– FOC ループ制御

6 つの独立した FOC ループは、PWM サイクルごとに SDFM ISR 内で AxisXFocLoopHandlerX() 機能 (ここでは、6 軸の場合の X = *1*～*6*) によって呼び出されます。図 2-3 は、PWM の更新と電流フィードバックの両方のタイミングを説明します。SDFM ISR は、各 PWM サイクル中に 2 回生成されます。単一の更新の場合、EPWM0 ISR で設定されたソフトウェアフラグ gEpwmSyncFlag と、IEP_CMP0 イベント ISR で設定された gUpdateNextRisingEdgeCmpValue の両方を判断し、FOC ループが呼び出されます。ダブル更新では、FOC ループを各 PWM サイクルで 2 回呼び出すことができます。

AxisXFocLoopHandlerX() 関数には、以下のものが含まれます。

- まず、産業用イーサネット SORTE_G 経由で位置フィードバック ボードを使用してデコードされた最新の機械的角度を受信します。未処理の角度データ形式は Q23 で、可変の mechThetaX として浮動小数点形式に変換する必要があります (ここでは、6 軸で X = *1*～*6*)
- FOC の計算に使用する機械的角度と電気角の間のオフセットは、以下の方法で較正します。
  - *I* パルスを使用するとモーター角度が元の *0* 度にトリガされるため、まず、純粋な開ループを使用してモーターを数サイクル回転させ、位置ボードで正しい角度を取得します。
  - 次に、電気角 (変数 elecTheta) を *0* 度とし、q 軸の電流 (変数 parkIqOut) を *0* とします。d 軸の電流 (変数 parkIdOut) に一定の値を設定すると、モーター用の逆起電力 (EMF) なしのトルク電流のみが生成されます。この時点では、モーターは回転しませんが、電気角が *0* となり、fmod() 関数を使用して mechThetaX と *90.0* 度の間の剰余を計算することにより、機械的オフセットを知ることができます。次に、機械的オフセットを変数 mechAngleOffsetX として保存します (ここでは、6 軸の X = *1*～*6*)。
- 純粋な開ループ、閉電流ループ、閉速度ループの各オプションによる FOC 計算:
  - FOC 計算に使用する elecTheta は、モーターが 4 極ペアであるため、mechAngleOffsetX で補償し、4 倍にする必要があります。ti_r5fmath_sincos() 関数の入力として、elecTheta が 0～360 度の範囲にあることを確認してください。
  - Clarke と Park は、CLARKE_run_twoInput() 関数と PARK_run() 関数によって変換されます。
  - 逆数 Park は、IPARK_run() 関数で変換され、空間ベクトル生成は、SVGEN_runCom() 関数で構成されます。
  - 純粋な開ループの場合、インクリメント モーター角は myMechDeltaX 固定値で、q 軸電流は glq で与えられます。
  - 閉ループの場合、DCL_runPIParallel() 関数を使用して電流と速度の両方の PI コントローラを実現します。電流および速度のターゲットは、glqArray および gSpdArray 配列で定義されます。PI 定数は init_pids() 関数で調整できます。

2.3.7 位置ボード– SORTE_G デバイス インターフェイス

SORTE_G デバイス ソフトウェアには以下のものが含まれています。

- ファームウェア SORTE_g_device_PRU0.h および SORTE_g_device_PRU1.h は、PRU プロジェクト SORTE_G_device で生成されます。LP-AM243 でジャンパを使用して J6 ピン 59 と J6 ピン 60 を短絡し、ボードをデバイスにします。

- ソースコード main.asm には、以下のような機能が含まれています。
  - PRU レジスタを初期化します
  - タスク マネージャをさまざまな状態に対し構成します
  - MII_G_RT モジュールをリセット・構成します
  - MDIO リンク イベントを生成し、現在のイーサネットリンクのステータスを読み取ります
  - 割り込みイベント マネージャと SORTE_G 状態を呼び出す制御ループを実行します
- ソース コード sorte_g_app.c は、以下のアドレスから読み込まれるデバイスが送信する IO 交換データを定義します。
  - チャネル 0 EQEP は、ICSSG1 DRAM0 アドレス *0x30081504* にモーター角を送信します
  - チャネル 1 EQEP は、ICSSG1 DRAM0 アドレス *0x3008150C* にモーター角を送信します
  - チャネル 2 EQEP は、ICSSG1 DRAM0 アドレス *0x30081514* にモーター角を送信します
  - チャネル 3 EQEP は、ICSSG1 DRAM0 アドレス *0x3008151C* にモーター角を送信します
  - チャネル 4 EQEP は、ICSSG1 DRAM0 アドレス 0x30081524 にモーター角を送信します
  - チャネル 5 EQEP は、ICSSG1 DRAM0 アドレス *0x3008152C* にモーター角を送信します

2.3.8 位置ボード:PRU_EQEP インターフェイス

図 2-5 は、インクリメンタル エンコーダからの 4 チャネル位置フィードバック用 PRU EQEP ファームウェアのアーキテクチャを示します。PRU_EQEP デザインは、A、B、I 信号を含む 4 つのインクリメンタル エンコーダ用インターフェイスで構成されます。これらの信号は TXB0106RGYR レベルシフタを通過し、エンコーダの信号レベルをマイコンのロジック レベルに一致させます。GPIO は、信号 A と B の立ち上がりおよび立ち下がりエッジで割り込みをトリガし、信号 I の立ち上がりエッジで割り込みをトリガするよう構成されています。これらの割り込みは ICSSG の IEP にルーティングされます。IEP には 64 ビット カウンタがあり、カウンタ値をキャプチャしてモーターの回転数と角度を測定します。PRU コアにはタスク マネージャがあり、PRU で実行される複数のタスクを構成します。これらのタスクは、カウンタの値をキャプチャし、モーターの回転数と角度を計算します。さらに、このタスクは、*XFR2VBUS* ウィジェットをトリガし、計算の基本的な入力であるエンコーダ信号のステータスを示す GPIO のステータスを読み取ります。

表 2-4 は、LP-AM243 側の 4 チャネル ABI 信号を示します。これらは、位置ボードのレベルシフタの出力です。

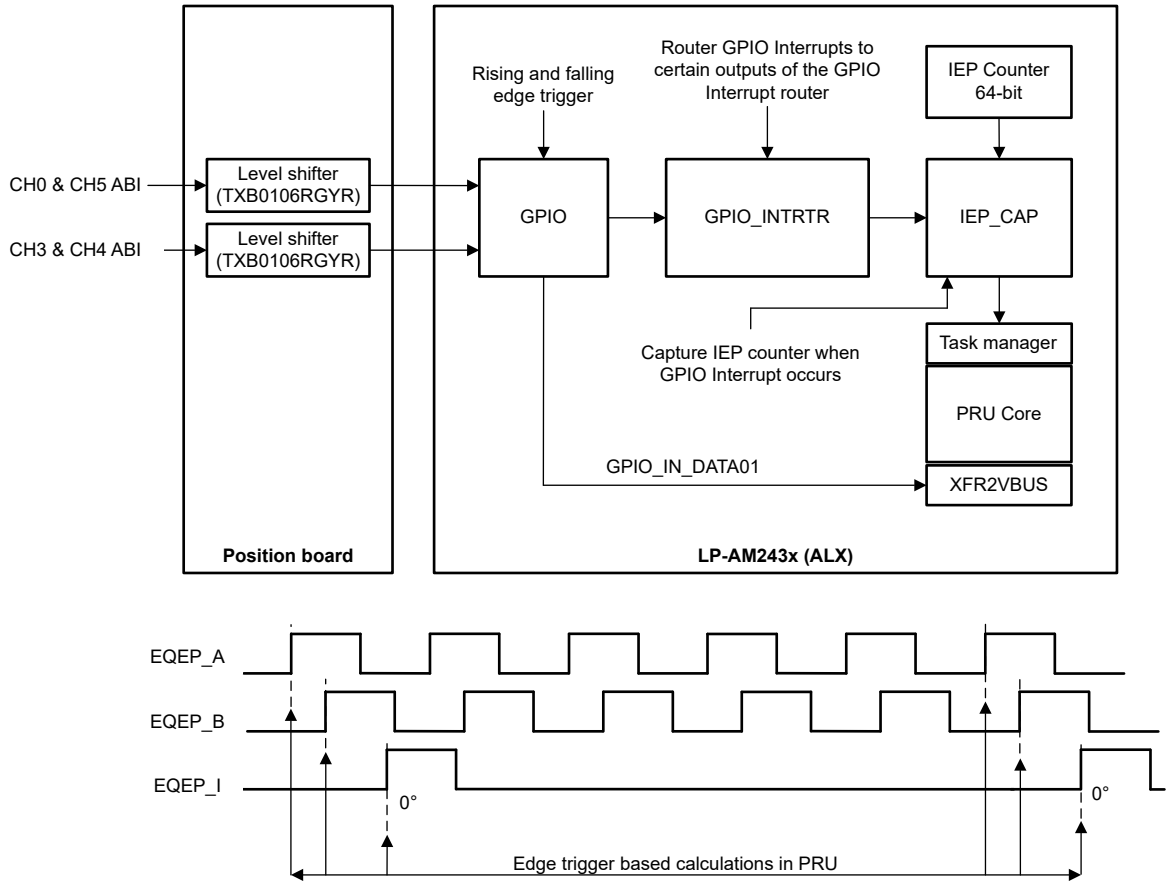


図 2-5. PRU_EQEP サブシステムのアーキテクチャとインターフェイス

表 2-4. PRU_EQEP 信号

サブシステム	信号名	ペリフェラル	PRU コア	LP-AM243 HEADER	AM243x ボールピン (ALX)
PRU_EQEP CH0	EQEP_A_CH0	GPIO1_20	ICSSG0_RTU1	BP.11	L5
	EQEP_B_CH0	GPIO1_21		BP.67	J2
	EQEP_I_CH0	GPIO1_33		BP.71	T4
PRU_EQEP CH3	EQEP_A_CH3	GPIO1_0	ICSSG0_PRU0	BP.33	J3
	EQEP_B_CH3	GPIO1_1		BP.32	J4
	EQEP_I_CH3	GPIO1_6		BP.48	H2
PRU_EQEP CH4	EQEP_A_CH3	GPIO1_2	ICSSG0_TXPRU0	BP.31	G1
	EQEP_B_CH3	GPIO1_3		BP.19	H1
	EQEP_I_CH3	GPIO1_7		BP.44	E2
PRU_EQEP CH5	EQEP_A_CH3	GPIO1_4	ICSSG0_TXPRU1	BP.17	K2
	EQEP_B_CH3	GPIO1_5		BP.13	F2
	EQEP_I_CH3	GPIO1_8		BP.51	H5

PRU_EQEP ソフトウェアには、以下のものが含まれています。

- ファームウェア ファイル:
  - チャネル 0 エンコーダ用ファームウェア EQEP_position_feedback_CH0_ICSS_G0_RTU_PRU1.h
  - チャネル 3 エンコーダ用ファームウェア EQEP_position_feedback_CH3_ICSS_G0_PRU0.h
  - チャネル 4 エンコーダ用ファームウェア EQEP_position_feedback_CH4_ICSS_G0_TX_PRU0.h
  - チャネル 5 エンコーダ用ファームウェア EQEP_position_feedback_CH5_ICSS_G0_TX_PRU1.h
- GPIO の構成:

  - GPIO_SET_RIS_TRIG レジスタによる立ち上がりエッジにトリガされた割り込みと、EQEP_A および EQEP_B 向けの *GPIO_SET_FAL_TRIG* による立ち下がりエッジ検出を有効にします。*I* 信号は、基準に使用される 1 回転あたり 1 つのパルスを表しており、立ち上がりエッジ割り込みのみが必要となります。
  - GPIO 割り込みルータ モジュールは、入力から出力先への GPIO 割り込み信号をマルチプレクシングするための重要な仲介役として機能します。GPIO エッジトリガ割り込みが構成されると、これらの割り込みは GPIO 割り込みルータの入力にルーティングされます。その後、ルータはこれらの入力を特定の出力にマッピングします。GPIO 割り込みルータの出力は、その後 ICSSG 内の IEP モジュールに送られます。

注

GPIO 割り込みルータ内での割り込みの構成とルーティングは、SCI クライアントのみが実行できます。

- ICSSG_PRU の構成:
  - ICSSG0 IEP タイマを 333Mhz に設定し、各カウンタは 1ns とします。
  - CMP0 をラップアラウンド モードで最大値に、CMP5 を 62.5µs に設定し、速度計算のためにタスクマネージャをトリガーします。
  - EXT_CAP_EN[5:0] は、GPIO 割り込みルータ出力の IEP キャプチャを有効にするために設定されます。
  - タスク sub_task_TS2_S0 は速度計算を処理します。sub_task_TS2_S1 は、XFR2VBUS ウィジェットを使用してモーター角の計算を行います。sub_task_TS2_S2 は、エンコーダ A 遷移の正確なタイム スタンプを使用して、EQEP_A 信号検出を行います。sub_task_TS2_S3 は、EQEP_B 信号検出を、エンコーダ B 遷移の正確なタイム スタンプとともに行います。sub_task_TS2_S4 は、EQEP_I 信号検出を処理して、エンコーダの回転サイクルでの基準位置をマークします。これにより、角度値を 0 度にリセットするためのポイントが提供されます。

2.3.9 位置ボード – SoC EQEP モジュール インターフェイス

インクリメンタル エンコーダからの 2 チャネルの位置フィードバックは、SoC EQEP モジュールを使用します。表 2-5 は、LP-AM243 側の 2 チャネル ABI 信号を示します。

表 2-5. SoC EQEP モジュール信号

サブシステム	信号名	ペリフェラル	LP-AM243 HEADER	AM243x ボールピン (ALX)
SoC EQEP CH1	EQEP_A_CH1	EQEP1	J12.1	L2
	EQEP_B_CH1		J12.2	L3
	EQEP_I_CH1		J12.3	R5
SoC EQEP CH2	EQEP_A_CH2	EQEP2	J21.1	B14
	EQEP_B_CH2		J21.2	A15
	EQEP_I_CH2		J21.3	B13

SoC EQEP は、次のものを含む generic_pruss_init() 機能で構成されます。

- QEP クロック 125MHz で QEP 期間を 16kHz に構成します。
- 位置カウンタをゼロで初期化し、モータ仕様に応じて最大カウンタ値を 4000 に設定します。
- QEP 位置カウンタ ソースを構成し、状態をラッチし、インデックス イベントでリセットします。
- ユニット タイムアウトによる割り込みの構成と有効化:
  - 割り込み番号は 144 で、チャネル 1 の割り込みコールバック機能 EQEP1_ISR
  - 割り込み番号は 145 で、チャネル 2 の割り込みコールバック機能 EQEP2_ISR

「*AM64x/AM243x テクニカル リファレンス マニュアル (TRM)*」の *EQEP* モジュール セクションも参照してください。

3 システム設計理論

3.1 位置ボード – システム初期化

以下の手順で、位置ボードを SORTE_G デバイスとして初期化し、generic_pruss_init() 機能を使用して R5F_0_0 コアの 6 つのエンコーダ チャネルすべての ABI 信号をデコードします。制御ボードを設定する*前*に、位置ボードコードを読み込み、実行します。

1. イメージ sbl_null_sciclient.release.hs_fs.tiimage を使用してセカンダリ ブート ローダ (SBL) 経由で構成を事前に読み込み、システム コマンド インタープリタ (SCI クライアント) を有効にします。
2. SORTE_G ファームウェアを使用するため、ワークスペース フォルダの下にある include.zip ファイルと pru_fw_common.zip ファイルをコピーします。
3. データ RAM をクリアし、エントリポイントを設定することで、ICSSG PRU を初期化します。
4. モーターの方向と速度 LUT を PRU データ RAM に書き込みます。
5. GPIO ピンモードを PRU_EQEP 用入力、そして SoC QEP 用 EQEP として設定し、GPIO 割り込みモードを立ち上がりエッジ検出として設定します。
6. SORTE_G デバイス向けに RGMII インターフェイスと MII_G_RT モジュールを設定します。
7. PRU_EQEP の ICSSG0 IEP タイマと SORTE_G の ICSSG1 IEP タイマを設定します。
8. SoC QEP モジュールのパラメータと割り込みを設定します。
9. SORTE_G デバイスと PRU_EQEP ファームウェアを読み込み実行します。その後、6 チャネルでデコードされたモーター角データが、事前定義されたアドレスを使って PRU データメモリにコピーされ、事前定義された PWM サイクルごとに送信する準備が整います。

3.2 位置ボード – 割り込み

- チャネル 1 EQEP1 割り込み番号は 144 で、割り込みコールバック機能 EQEP1_ISR は、チャネル 1 QEP ティックの数を読み出して、モーター角度を計算し、事前定義された PRU データメモリに角度データを書き込みます。
- チャネル 2 EQEP2 割り込み番号は 145 で、割り込みコールバック機能 EQEP2_ISR は、チャネル 2 QEP ティックの数を読み出して、モーター角度を計算し、事前定義された PRU データメモリに角度データを書き込みます。
- チャネル 0 およびチャネル 3 からチャネル 5 は、GPIO 割り込みを利用し相対 IO ピンの AB 信号の立ち上がり/立ち下がりエッジを検出します。

3.3 制御ボード – システム初期化

以下の手順で、制御ボードを SORTE_G コントローラとして初期化し、single_chip_servo_remote_core_start() 関数で R5F_0_0 コアの 6 軸モーターすべてを制御するための 6 チャネルモーター角データを受信します。位置ボードを設定した*後*、制御ボード コードを読み込み、実行します。

1. 制御ボードは TQ-SoM を使用しており、SOM 上のフラッシュを手動で構成する必要があります。フラッシュは、(mcu_plus_sdk\tools\ フォルダの下の) Python® uart_uniflash.py ツールとともに、tq_sbl_uart_uniflash.hs_fs.tiimage イメージを使用して、SBL をフラッシュする必要があります。tq_sbl_uart_unflash.hs_fs.tiimage と default_sbl_null_tq.cfg を SDK フォルダ mcu_plus_sdk\tools\boot\sbl_prebuilt\am243x-evm にコピーします。
2. GPIO ピンの方向と初期値を init_gpio_state() 関数で設定します。
3. enable_pwm_buffers(FALSE) 関数で PWM を無効にします。
4. ICSSG_PRU_PWM を軸 4、5、6 それぞれに 3 相相補用に構成し、init_pruIcssPwm() 関数で初期デューティサイクルを 50% で構成します。
5. 軸 1、2、3 それぞれに EPWM を 3 相相補用に構成し、init_pwms() 関数で初期デューティサイクルを 50% に設定します。
6. init_sddf() 関数を使用して、6 軸すべての SDFM の ICSSG1 RTU0、RTU1、PRU0、PRU1 コアを構成し、負荷共有モードで 4 つの SDFM ファームウェアを読み込みます。init_IEP1_SYNC() 関数で SD クロックの ICSSG0 IEP1 タイマ SYNC0 および SYNC1 を初期化します。start_ICSSG1_IEPx() 関数で ICSSG1 IEP タイマを起動します。
7. ICSSG0 PRU0 コアを SORTE_G コントローラ用に構成し、generic_pruss_init() 関数で SORTE_G コントローラ ファームウェアを読み込みます。
8. init_pids() 関数で FOC 制御用パラメータを初期化します。

9. 6 つすべての軸の EPWM 出力バッファを有効化します。

3.4 制御ボード – 割り込み

- EPWM 割り込み（16kHz）- 割り込み番号は 108（EPWM0）で、割り込みコールバック機能 App_epwmIntrISR は、IEP タイマをリセットし、軸 4、軸 5、軸 6 の次の ICSSG_PRU_PWM 立ち下がりエッジを、PWM 周期の半分から立ち上がりエッジを引いた値で更新するために使用されます。この割り込みは、単一の更新スキーム、または FOC の計算結果によって、ダブル更新スキームとして値を比較します。
- ICSSG_PRU_PWM 割り込み（16kHz）– IEP_CMP0 割り込み App_pruicssIep1Compare0IrqSet()は、PWM 信号の次の立ち上がりエッジの比較イベント値を FOC ループ計算の結果で更新するためにソフトウェアフラグ gUpdateNextRisingEdgeCmpValue を設定するために使用されます。
- SDFM 割り込み（32kHz）- 割り込み番号は 251（PRU_ICSSG1_PR1_HOST_INTR_PEND_3）で、割り込みコールバック機能 pruSddfIrqHandler0 は、6 軸の FOC ループをトリガするために使用されます。サンプル 8192 から 16384 までの範囲で、SDFM チャネルのオフセット gSddfChOffset[x]を計算し、FOC ループの電流フィードバックを補償します。

4 ハードウェア、ソフトウェア、テスト要件、テスト結果

4.1 ハードウェア要件

このリファレンス デザインをテストするには、以下の装置が必要です。

- 制御ボード - TIDA-010948_CB (図 4-1 と 図 4-2 の図を参照)
- 3 つのアダプター ボード - TIDA-010948_DB (図 4-3 と 図 4-4 を参照)
- 3 つの BP-AM2BLDCSERVO — AM2x ブラシレス DC (BLDC) サーボ モーター ブースタパック ™
- 位置ボード - TIDA-010948_PB (図 4-5 と 図 4-6 を)
- LP-AM243 評価ボード – Arm® ベース MCU 向け AM243x 汎用 LaunchPad™ 開発キット
- 6 つの LVSERVOMTR モーター - 低電圧サーボ モーター - 低電圧サーボ (エンコーダ) モーターおよびワイヤ ハーネス

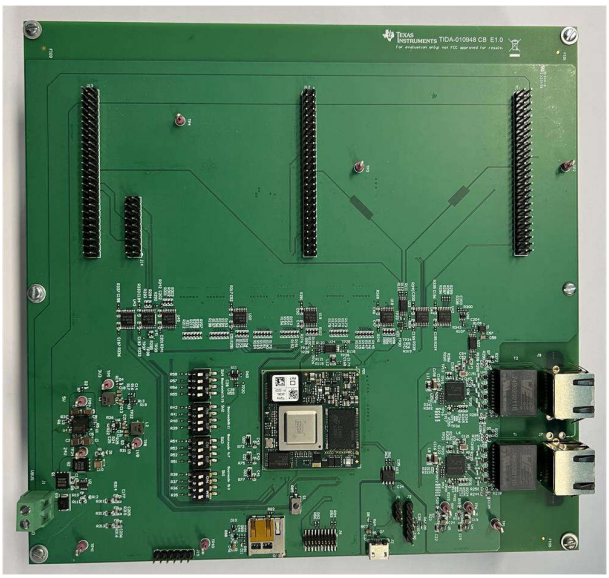


図 4-1. TIDA-010948_CB PCB 上面図

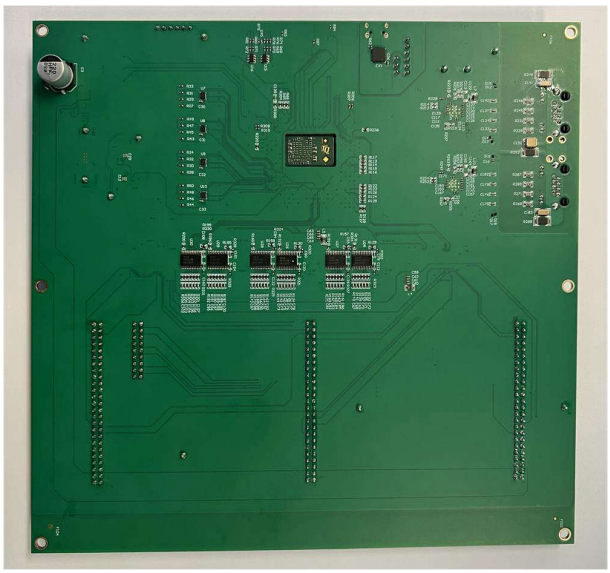


図 4-2. TIDA-010948_CB PCB 底面図

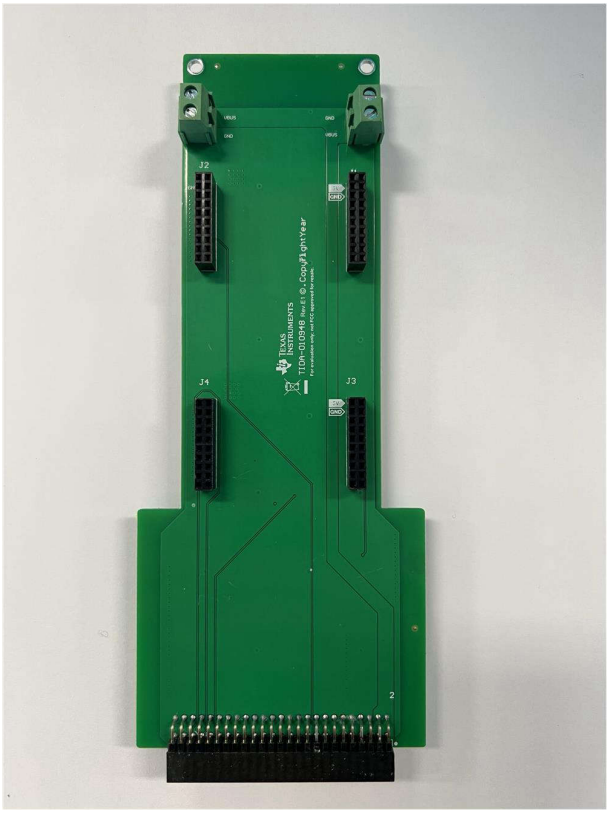


図 4-3. TIDA-010948_DB PCB 上面図

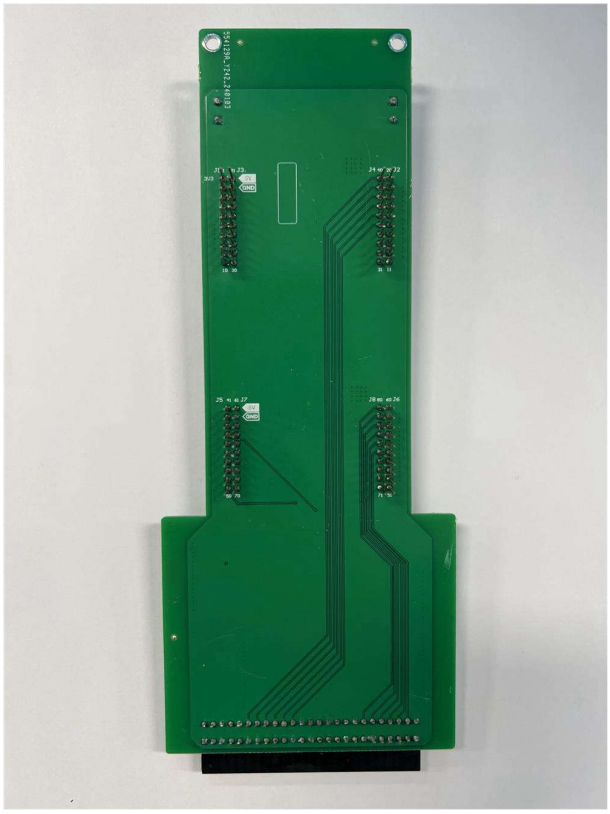


図 4-4. TIDA-010948_DB PCB 底面図

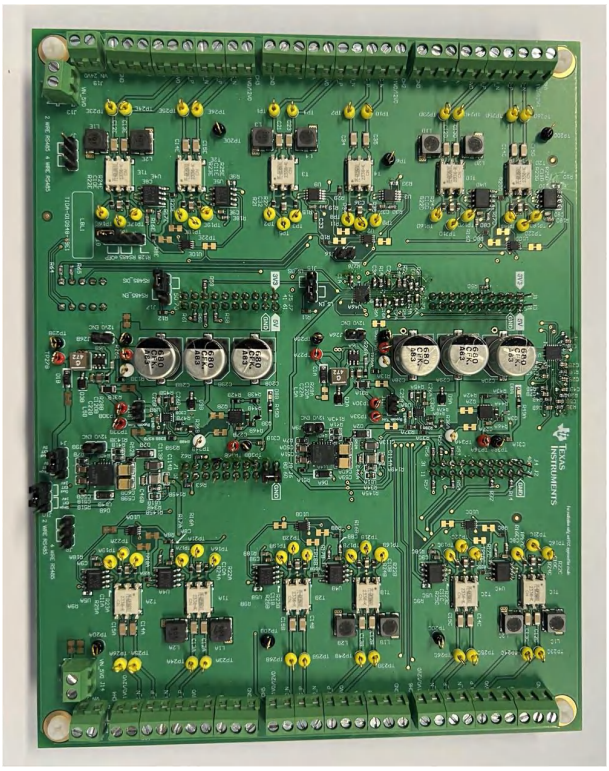


図 4-5. TIDA-010948_PB PCB 上面図

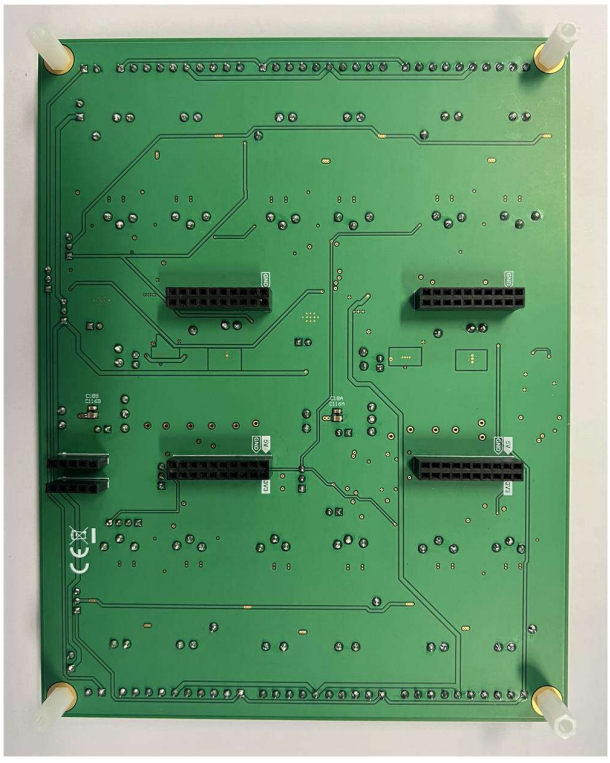


図 4-6. TIDA-010948_PB PCB 底面図

図 4-7 は、システムデモの概要を示します。

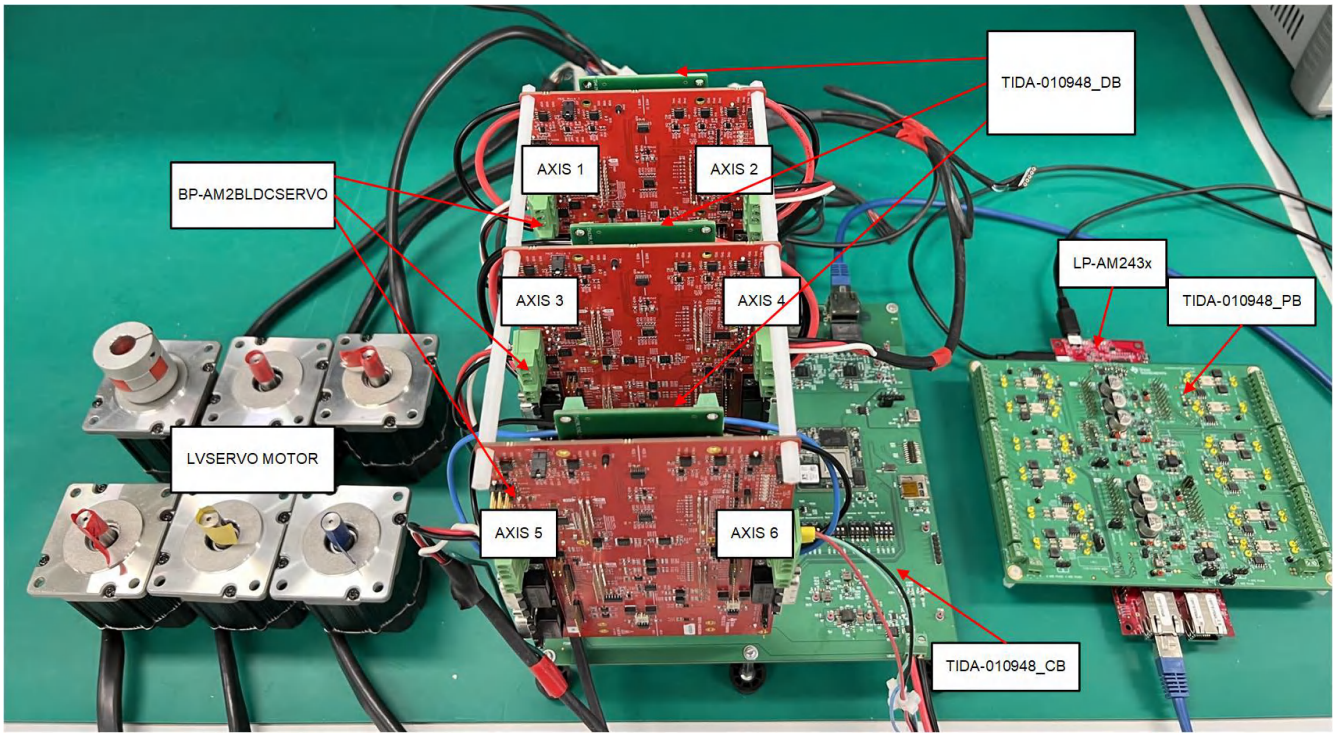


図 4-7. TIDA-010948 システムデモの概要

4.1.1 システム デモのセットアップ

以下の手順では、システム デモのセットアップについて説明します。

1. 3 つのアダプター ボード (TIDA-010948_DB) を 3 つの BP-AM2BLDCSERVO ボード (BoosterPack ボード) に接続してから、このセットアップをベース制御ボード (TIDA-010948_CB) に接続します。1 つの BoosterPack ボードと 1 つのアダプタボードを 2 つの軸の電力段として使用します (アダプターボードの J1～J4 は、それぞれ BoosterPack ボードの J1～J4 に接続します)。
2. 1 つのアダプターボードの J5 を、軸 1 と軸 2 の電力段として制御ボードの J11 に接続します。2 つ目のアダプターボードの J5 は、軸 3 と軸 4 の電力段として制御ボード上の J12 に接続します。3 つ目のアダプターボードの J5 は、軸 5 と軸 6 の電力段として制御ボード上の J13 に接続します。
3. 制御ボードの J1 接続部は、システム入力 $24V_{DC}$ です。
4. アダプターボードの J6 接続は、制御ボードから DC リンク電力が供給されます。この J6 をケーブルで BoosterPack ボードの J5 に接続し、電源を BoosterPack ボードに供給します。アダプターボードの J7 接続は、制御ボードから DC リンク電力が供給されます。この J7 をケーブルで BoosterPack ボードの J6 に接続し、電源を BoosterPack ボードに供給します。3 つのアダプターボードすべてを同じ接続で接続し、6 つの軸すべてに電源を供給します。2 つの BoosterPack ボード間の距離は約 83mm で、BoosterPack ボードとアダプターボードと制御ボードの間の固定にはスタンドオフが必要です。図 4-8 と 図 4-9 は、DC リンクとモーター電源ケーブルの接続を示します。
5. 制御ボードのブート:
   - 制御ボードの J4 のピン 1 とピン 2 を短絡し、UART 電源に電力を供給します
   - USB ケーブルを J2 に UART 端子のために接続します
   - JTAG ケーブルを J6 に接続します (デフォルトのオスヘッダは 0.05 インチの CM20 ピンで、ピン 6 をブロックする必要があります)
   - ブートモードを UART BOOT に設定し、SW4「*0000*」、SW2「*1011*」、SW3「*1100*」、SW1「*1101*」に設定します
   - UART 端子を開くと、文字 *C* が 2～3 秒で印刷されます。このプロセスが完了したら、UART 端末を閉じます。
   - Python® uart_uniflash.py を使用して、SBL_null をフラッシュします。unflash の TQ イメージを使用し、tq_sbl_uart_uniflash.hs_fs.tiimage と default_sbl_null_tq.cfg を SDK フォルダ mcu_plus_sdk\tools\boot\sbl_prebuilt\am243x-evm にコピーします

- ボードの電源をオフにし、SW4「*0000*」、SW2「*0100*」、SW3「*1110*」、SW1「*1100*」に設定して OSPI BOOT MODE を設定し、その後、ボードの電源をオンにします。SBL_NULL 情報が、UART 端末に表示されます。図 4-10 は、JTAG および UART のブートモードスイッチと接続を示します。
- 図 4-11 に示すように、ターゲットの ccxml ファイル内で JTAG 電源電圧を 1.8V に設定します。

6. 位置ボード (TIDA-010948_PB) には、J13 の $5V_{DC}$ で電力を供給します。J4 のデュアルチャネルに外部の 5V を、J10 のデュアルチャネルには外部の 24V を選択します。J15 のレバーシフタを有効にし、J17 の RS-485 インターフェイスを無効にします。J6 ピン 59 を J6 ピン 60 (GND) に短絡し、位置ボードを SORTE_G デバイスとして設定します。J3A～J3F は、それぞれ 6 チャネルのエンコーダ信号を接続するために使用されます。J1、J2、J5、J6、J12、J21 は LaunchPad LP-AM243x に接続されています。図 4-12 は、位置ボードの設定を示します。
7. LP-AM243x は、6 軸エンコーダ信号と SORTE_G デバイスをデコードするための MCU プラットフォームとして使用されます。セットアップとブート初期化情報については、AM243x MCU+ SDK：EVM のセットアップを参照してください。セクション 2.3.8 が示すように、LP-AM243x は sbl_null_sciclient.release.hs_fs.tiimage イメージを使用して SBL 経由で構成をプリロードして、SCI クライアントを有効にする必要があります。イメージファイルを *mcu plus sdk* フォルダ：
mcu sdk folder\tools\boot\sbl_prebuilt\am243x-lp\ にコピーします。

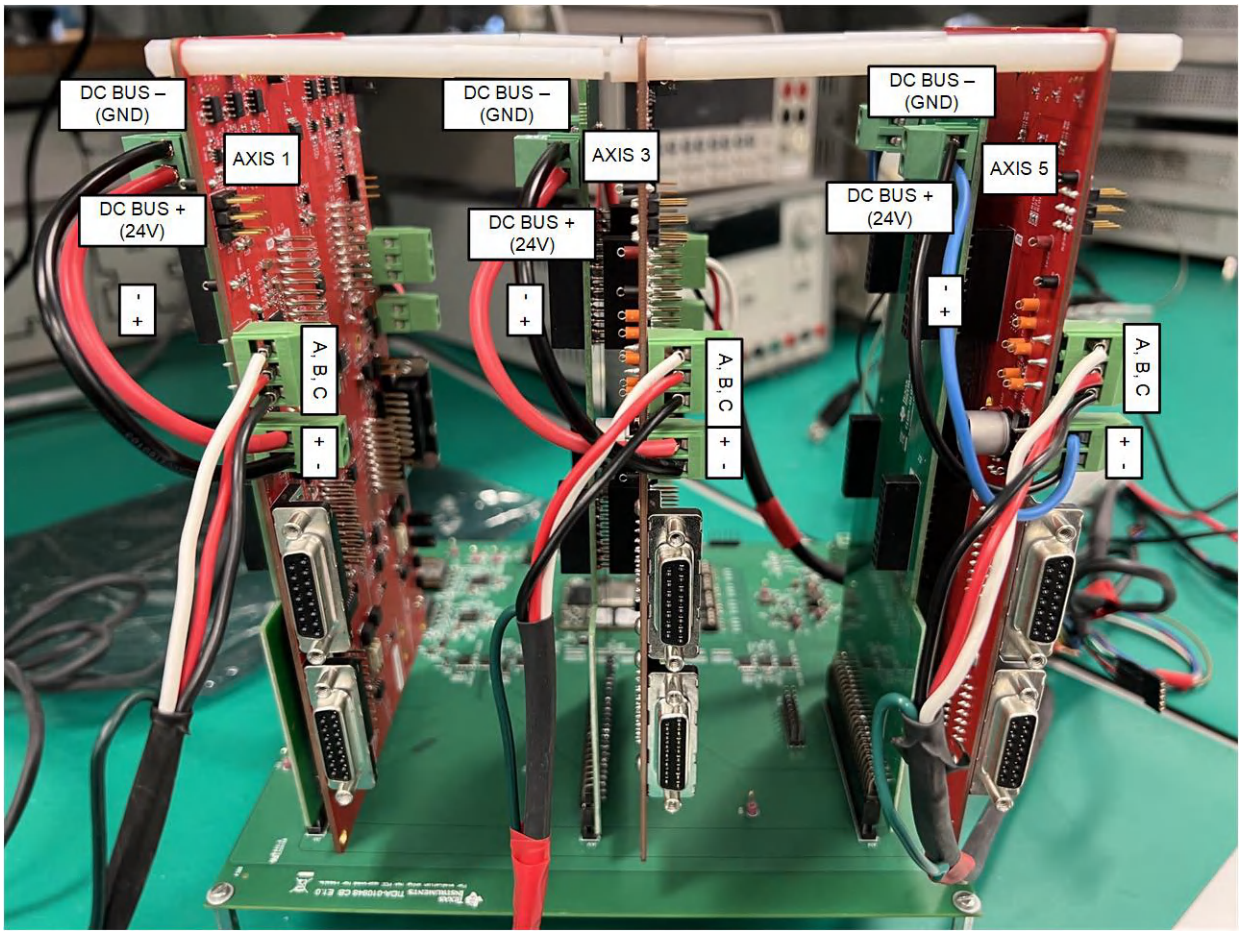


図 4-8. DC リンク接続とモータ位相電力 – 軸 1、3、5

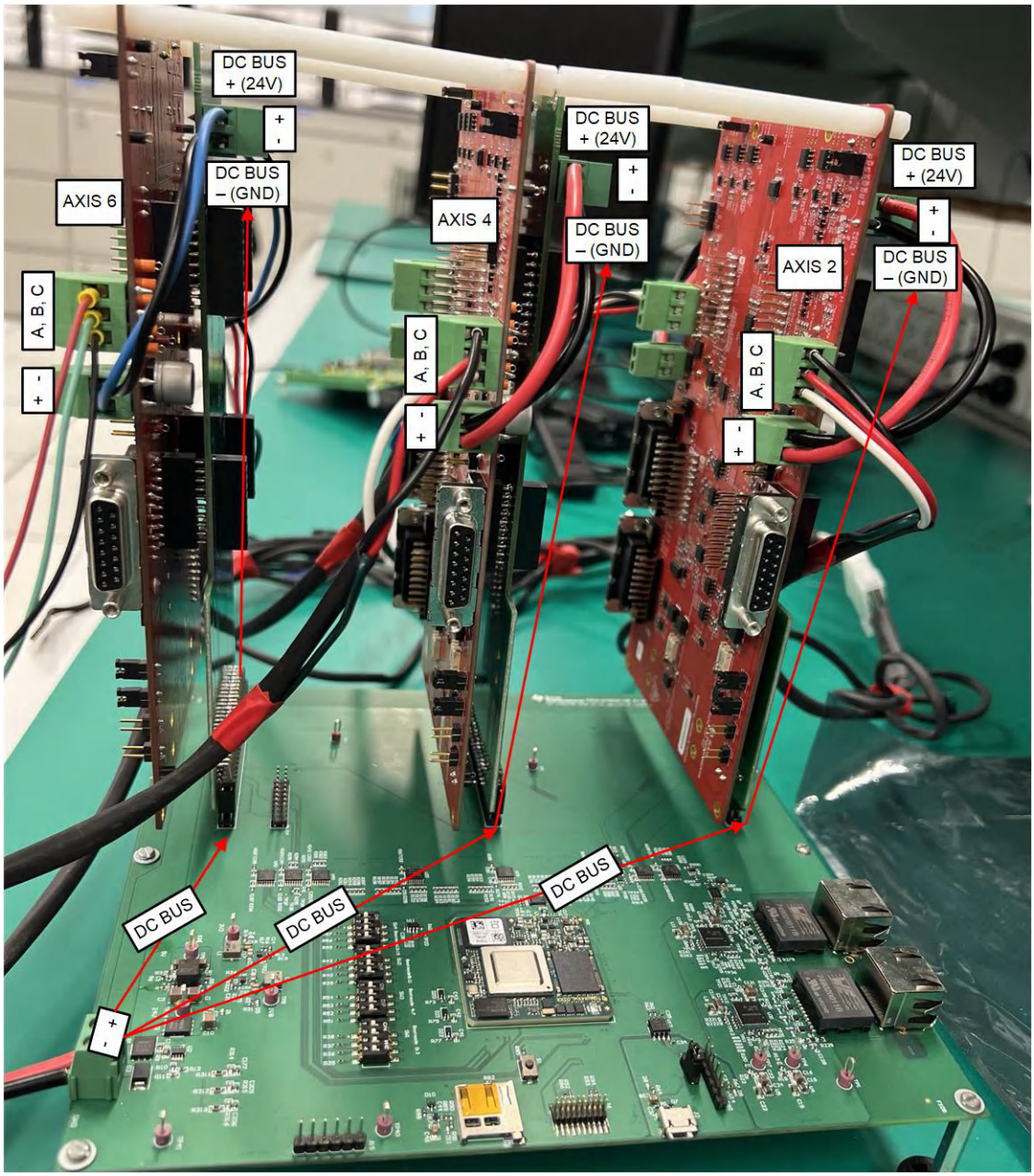


図 4-9. DC リンク接続とモータ位相電力 – 軸 2、4、6

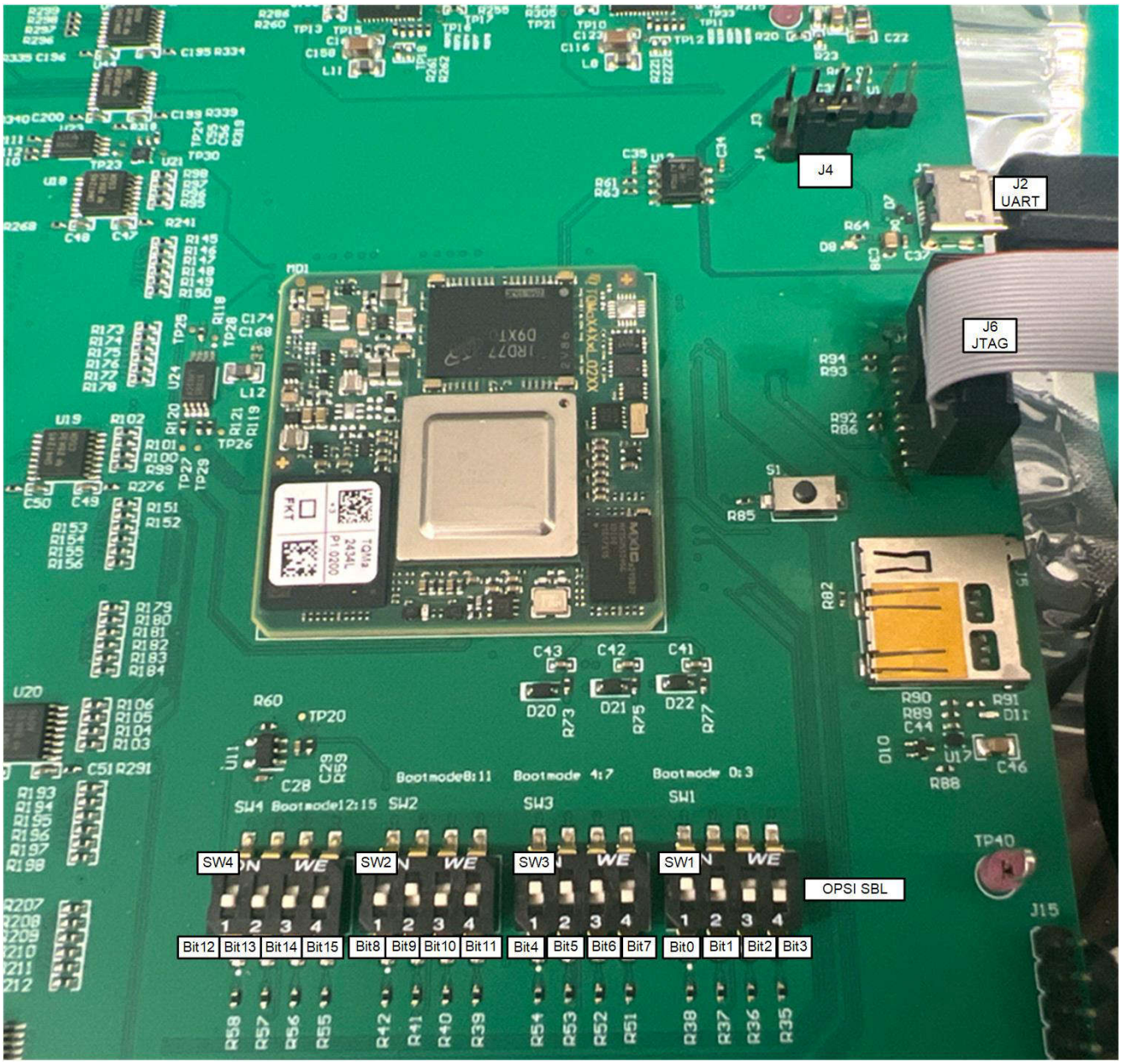


図 4-10. 制御ボード上の JTAG と UART のブートモードスイッチと接続

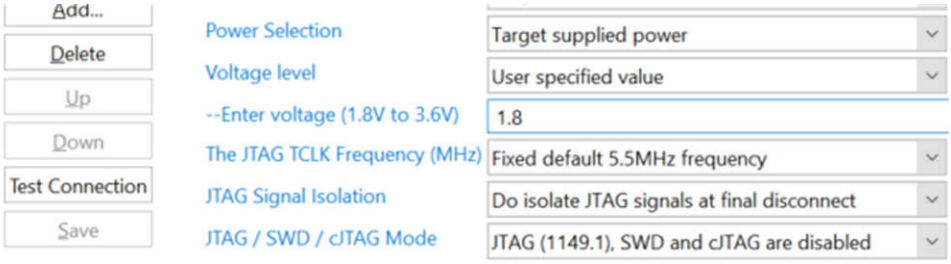


図 4-11. JTAG 電源のターゲットファイルの設定

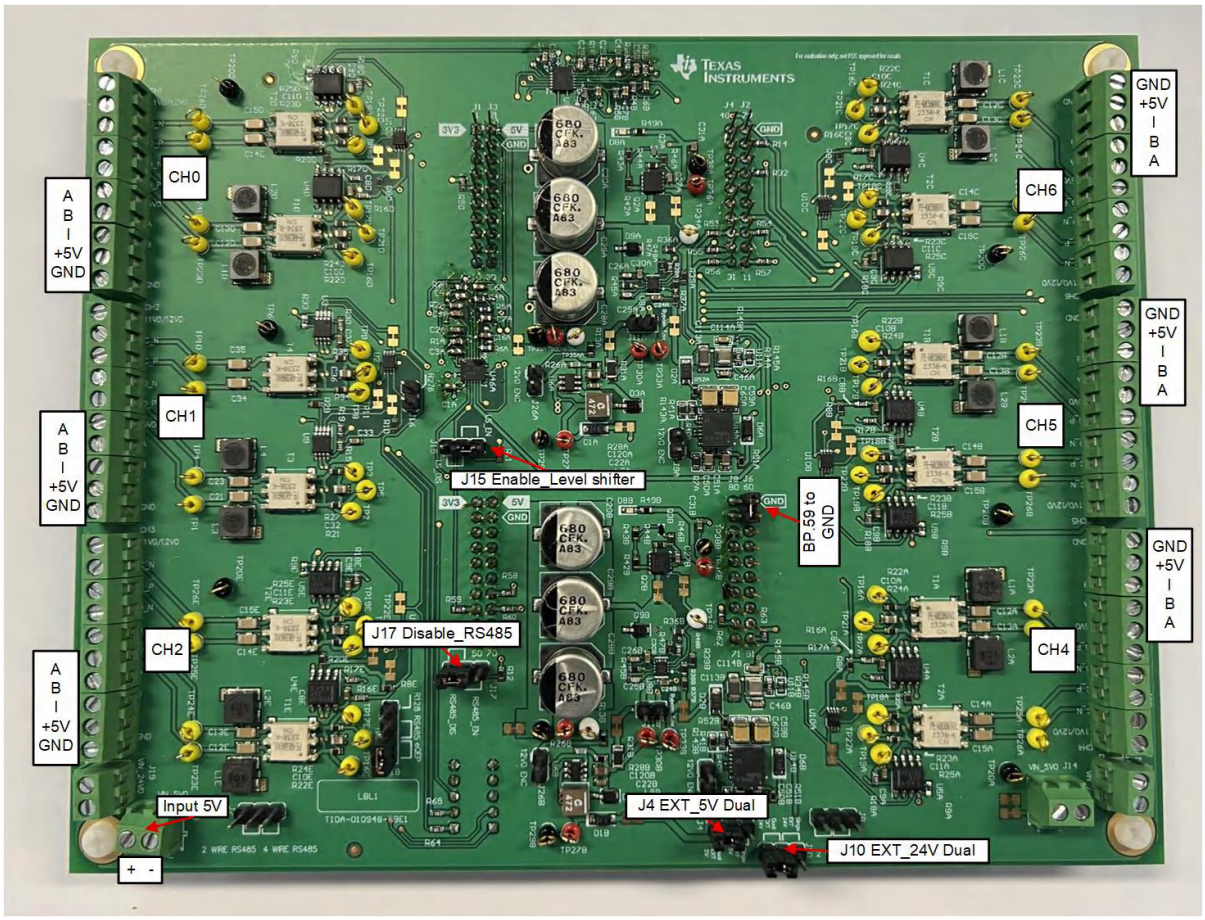


図 4-12. 位置ボードのセットアップ

4.2 ソフトウェア要件

このリファレンス デザインを検証するため、AM243x MCU 向けソフトウェア開発キット motor_control_sdk_am243x_09_02_00_09 を使用し、でテキサス・インスツルメンツの社内テスト ソフトウェアが開発されました。このソフトウェアは、一般向けには提供していません。AM243x ソフトウェアのサポートについては、MCU-PLUS-SDK-AM243X ソフトウェア開発キット（SDK）および Arm ベースマイクロコントローラフォーラム（Arm ベースマイクロコントローラ - TI E2E サポートフォーラム）も参照してください。

表 4-1. 主なソフトウェア構成

サブシステム	仕様	値
SOC ePWM および ICSSG_PRU PWM	周波数	16 kHz
	同期または位相シフトモード	同期モード
	カウント モード	アップダウン カウント
	デッドバンド	200ns
電流フィードバック – ICSS SDFM	通常電流 OSR	OSR 64
	サンプリング モード	連続サンプリング
	シングルまたはダブル更新機能	あり
位置フィードバック – PRU EQEP および SoC QEP	QEP 向け最大チャネル数	6
制御アルゴリズム – FOC ループ	サイクル時間	ダブル更新のオプションとしての 62.5µs または 31.25µs
産業用通信 – SORTE_G	サイクル時間	62.5µs
	リアルタイム制御	確定的なネットワーク時間

4.3 テスト設定と結果

4.3.1 電流フィードバック – SDF

図 4-13 は、変調器 AMC1035 側の SDFM クロックとデータを示します。青の曲線のチャネル 1 は、クロックバッファ LMK1C1104 の後に制御ボード ICSSG0_IEP1_SYNC0 によって生成された 20MHz クロックです。赤の曲線のチャネル 2 は、AMC1035 によって生成された SD データです。

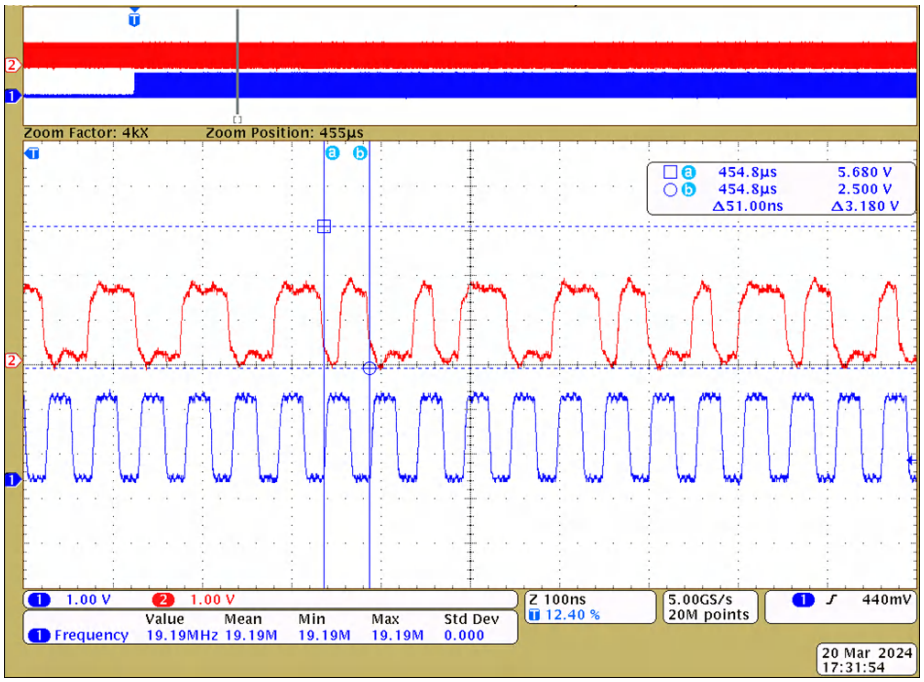


図 4-13. SDM クロックおよびデータ ライン信号

図 4-14 および図 4-15 は、開ループ制御を備えた電流プローブを用いてテストされたテスト構成と 2 相電流を示します。

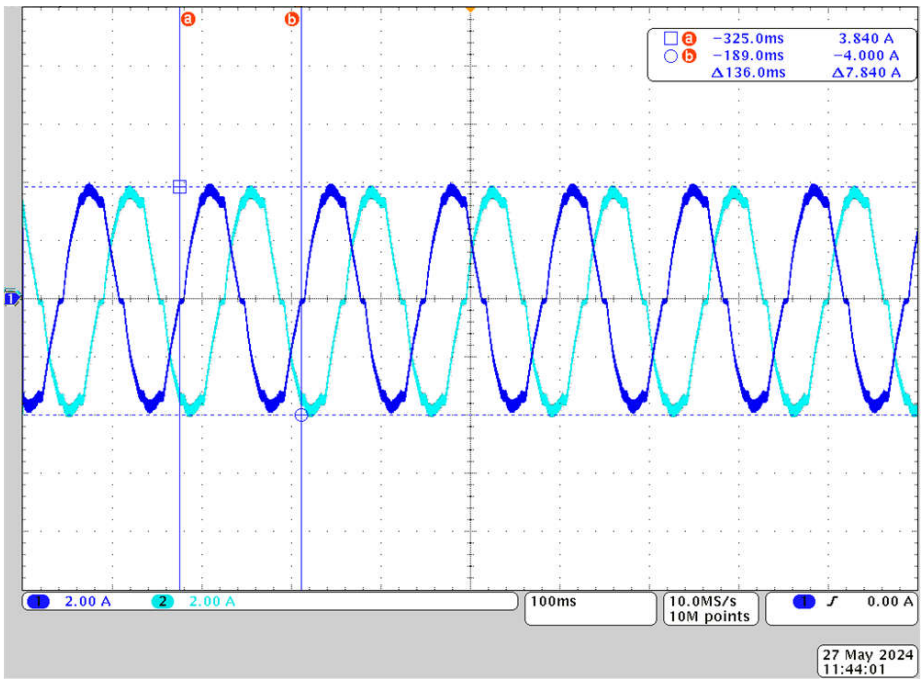


図 4-14. 位相 A と位相 B の電流

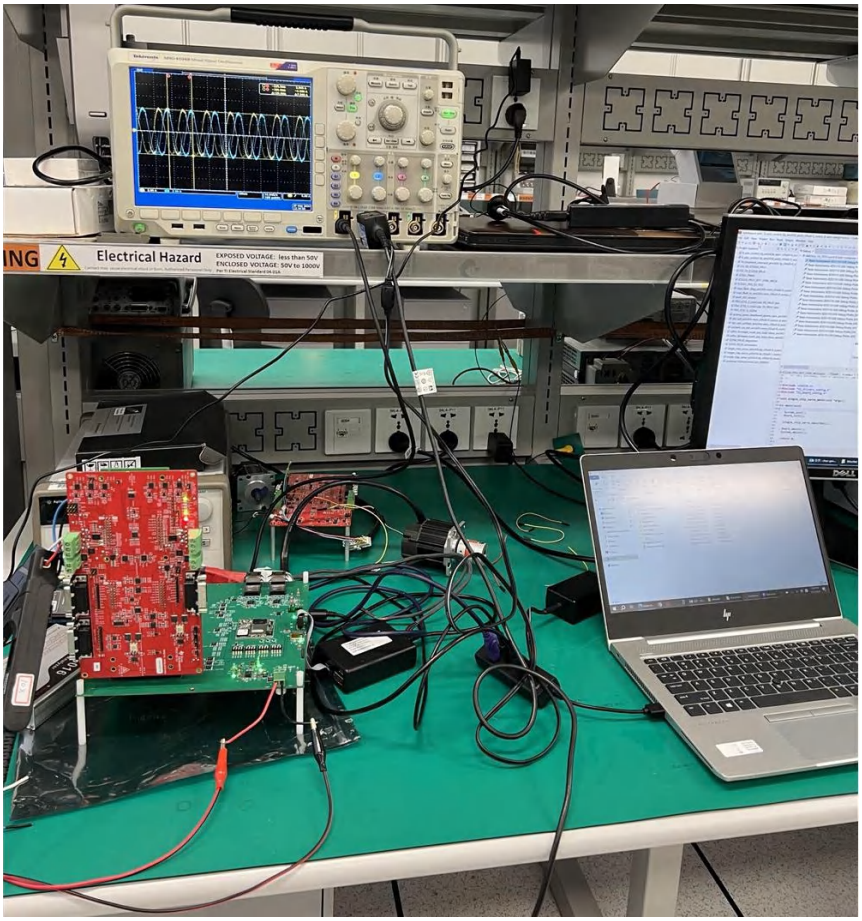


図 4-15. SDFM テスト設定

4.3.2 産業用イーサネット（SORTE_G）と PWM インターフェイス間の時間同期

図 4-16 は、位置ボード（SORTE_G デバイス）の ICSSG1_IEP0_SYNCOUT0 によって生成される同期パルス、SoC EPWM 信号、ICSSG_PRU_PWM 信号を示します。チャネル 0 は位置ボードの同期パルス、チャネル 8～13 は制御ボードの EPWM0、1、2 信号で、チャネル 14～15 は制御ボードの ICSSG_PRU_PWM 信号です。図 2-3 は、時間同期フローを示します。SORTE_G は、TSR モジュールを用いて EPWM0 と同期するため、事前定義されたサイクル時間で周期的なパルスを生成しました。ICSSG_PRU_PWM に使用される IEP タイマは、EPWM0 SYNC0 によってリセットされます。

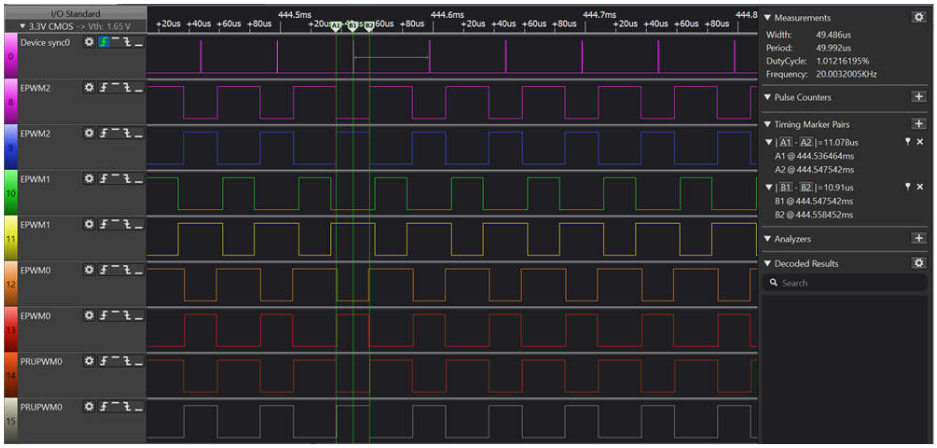


図 4-16. 産業用イーサネット（SORTE_G）と PWM インターフェイス間の時間同期

4.3.3 FOC ループの検証

4.3.3.1 FOC ループタイミング

図 4-17 は、PWM および SDFM インターフェイスを使用した FOC ループ（20kHz、ダブル更新）のタイミングに関するテストデータを示します。

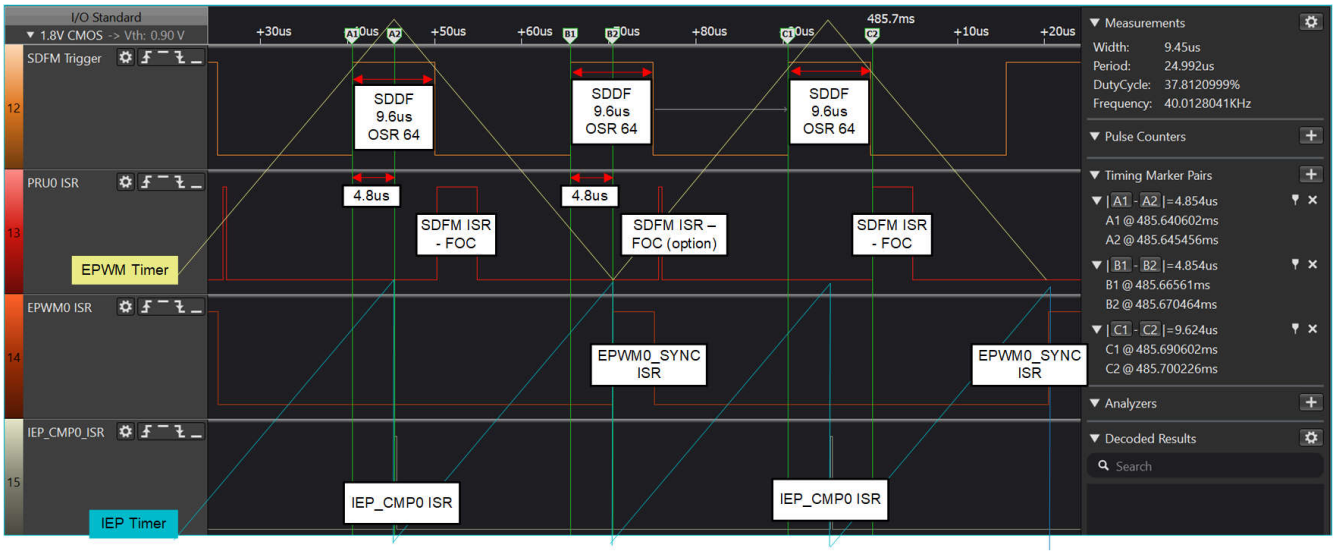


図 4-17. FOC タイミング検証

4.3.3.2 FOC ループ処理時間の検証

図 4-18 は、閉速度ループを使用した FOC ループの処理時間のテストデータを示します。FOC ループの計算時間は 1.056µs の前後で、6 軸の閉速度ループは、62.5µs のサイクル時間によって 7.584µs かかります。

図 4-18. 6 軸 FOC ループの処理時間

4.3.4 PI コントローラによる閉ループ制御の検証

PI コントローラは、電流ループと速度ループの両方に対し DCL_runPIParallel()機能によって制御を行います。PI 定数は init_pids()機能で調整できます。

図 4-19 は、閉ループ FOC ブロック図の検証を示します。閉速度ループ中には、I_Q リファレンスは速度 PI コントローラの出力となります。ただし、PI コントローラを検証するには、glqRef は一定の目標値として提供され、ParkIqMeasured は、PI 調整後のフィードバック値として Park および Clarke 変換後のモーター位相電流値となります。次に、parkIqMeasured を記録し、変数がターゲット値 glqRef で続くことができるかどうかを確認します。

図 4-20 は、以下のパラメータを使用した PI コントローラを使ったモーター位相 A フィードバック波形と電流ループのステップ応答を示します。

- gPilq.KP = 0.245
- gPilq.Ki = 0.09
- gPilq.Umax = 0.2

- gPiIq.Umin = 0.2

ここでは、DRV8316 の能力に応じて、電流ループ PI 制御出力は 0.2 に制限されます。検証のため、glqRef = 0.3 を閉電流ループの電流ターゲット値とします。機能付きの PI コントローラは、次の行で実行されます。

- *parkIqOut = DCL_runPIParallel(&gPiIq, glqRef, parkIqMeasured);*

これらの結果は、位相電流が PI によって適切に制御されていること(ピーク値は約 0.3A で、ターゲットである I_Q リファレンスに等しい)、電流ループのステップ応答時間は 32kHz の制御周波数で約 90µs であることを示しています。したがって、電流ループの帯域幅は、式 1 を用いて約 3.54kHz となります。

$$\text{Bandwidth} = \frac{1}{\pi \times t_r} \quad (1)$$

ここで、

- t_r = 応答時間

モーターは、以下のパラメータとともに LVSERVO を使用しています。

- 位相間の抵抗 = 0.72Ω
- 位相間インダクタンス = 0.4mH
- 電気時間定数 = 0.56
- 逆起電力(V_{peak} /K_{rpm})= 4.64

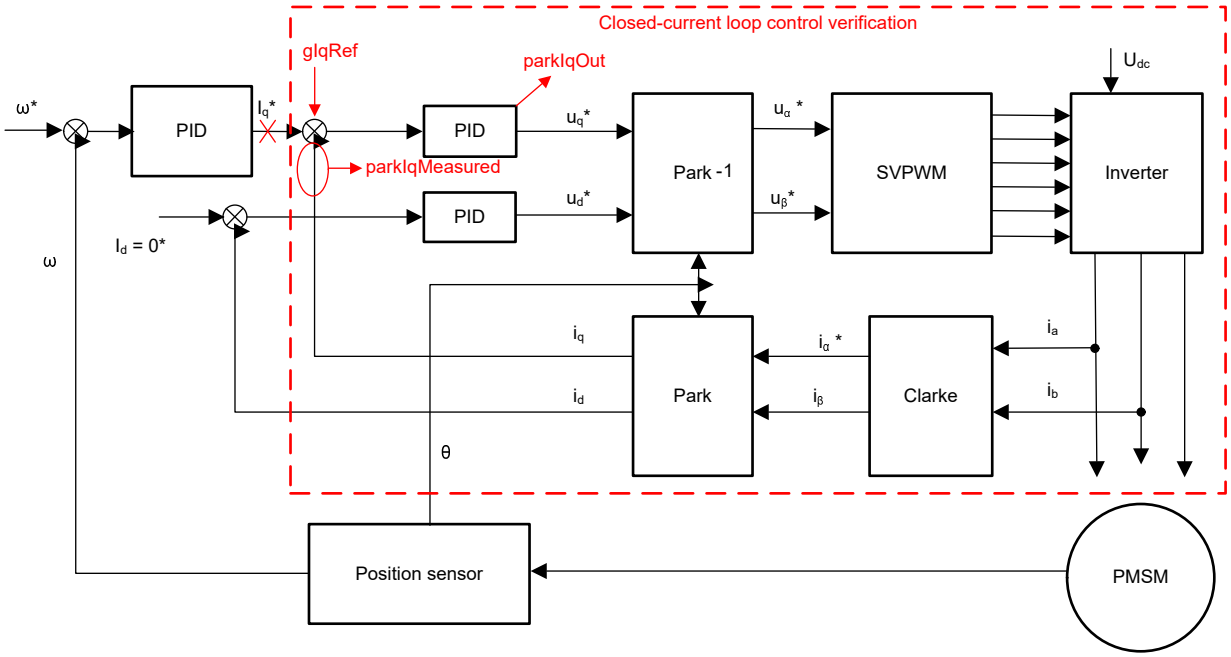


図 4-19. PI コントローラによる閉電流ループの検証図

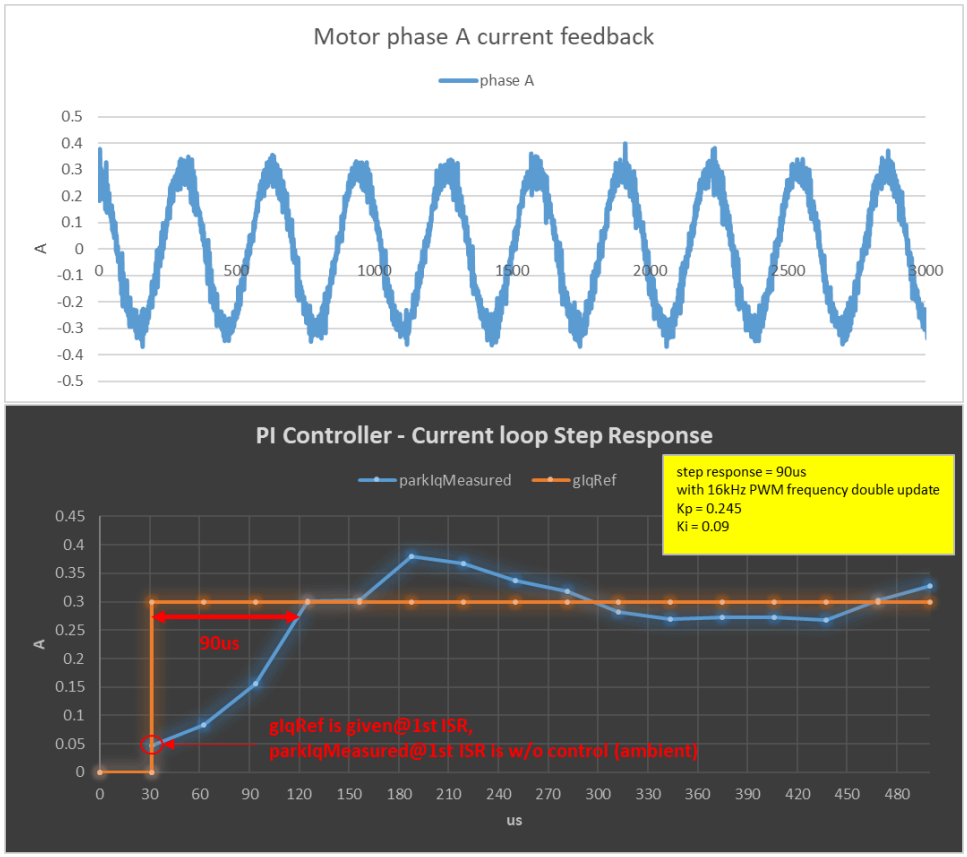


図 4-20. モーター位相電流フィードバックと PI コントローラのステップ時間応答

5 設計とドキュメントのサポート

5.1 デザイン ファイル

5.1.1 回路図

回路図をダウンロードするには、TIDA-010948 のデザイン ファイルを参照してください。

BLDC BoosterPack の回路図をダウンロードするには、BP-AM2BLDCSERVO デザイン パッケージのデザイン ファイルを参照してください。

AM243x LaunchPad の回路図をダウンロードするには、LP-AM243 デザイン パッケージのデザイン ファイルを参照してください。

5.1.2 BOM

部品表 (BOM) をダウンロードするには、TIDA-010948 のデザイン ファイルを参照してください。

BLDC BoosterPack の BOM をダウンロードするには、BP-AM2BLDCSERVO デザイン パッケージのデザイン ファイルを参照してください。

AM243x LaunchPad の BOM をダウンロードするには、LP-AM243 デザイン パッケージのデザイン ファイルを参照してください。

5.1.3 レイヤ プロット

レイヤ プロットをダウンロードするには、TIDA-010948 のデザイン ファイルを参照してください。

BLDC BoosterPack レイヤ プロットをダウンロードするには、BP-AM2BLDCSERVO デザイン パッケージのデザイン ファイルを参照してください。

AM243x LaunchPad レイヤ プロットをダウンロードするには、LP-AM243 デザイン パッケージのデザイン ファイルを参照してください。

5.1.4 Altium プロジェクト

Altium Designer® のプロジェクト ファイルをダウンロードするには、TIDA-010948 のデザイン ファイルを参照してください。

BLDC BoosterPack の Altium プロジェクトファイルをダウンロードするには、BP-AM2BLDCSERVO デザイン パッケージのデザイン ファイルを参照してください。

AM243x LaunchPad の Altium プロジェクトファイルをダウンロードするには、LP-AM243 デザイン パッケージのデザイン ファイルを参照してください。

5.1.5 ガーバー ファイル

ガーバー ファイルをダウンロードするには、TIDA-010948 のデザイン ファイルを参照してください。

BLDC BoosterPack のガーバーファイルをダウンロードするには、BP-AM2BLDCSERVO デザイン パッケージのデザイン ファイルを参照してください。

AM243x LaunchPad のガーバーファイルをダウンロードするには、LP-AM243 デザイン パッケージのデザイン ファイルを参照してください。

5.1.6 アセンブリの図面

アセンブリの図面をダウンロードするには、TIDA-010948 のデザイン ファイルを参照してください。

BLDC BoosterPack アセンブリの図面をダウンロードするには、BP-AM2BLDCSERVO デザイン パッケージのデザイン ファイルを参照してください。

AM243x LaunchPad アセンブリの図面をダウンロードするには、LP-AM243 デザイン パッケージのデザイン ファイルを参照してください。

5.2 ツールとソフトウェア

ツール

CCSTTUDIO	Code Composer Studio™ 統合開発環境 (IDE)：Microsoft® Windows® または Linux® 向けの CCS 20.0.0 バージョンをダウンロード
ARM-CGT-CLANG	Arm® コード生成ツール - コンパイラ：Microsoft Windows または Linux 向け TI ARM CLANG 3.2.0.LTS をダウンロード
SYSCONFIG	SysConfig のスタンドアロン デスクトップ バージョン：Microsoft Windows または Linux 向けの SysConfig 1.22.0 をダウンロード

ソフトウェア

AM243x モーター制御 SDK	モーター制御 SDK Microsoft Windows インストーラ
AM243x 産業用通信 SDK	産業用通信 SDK Microsoft Windows インストーラ
AM243x MCU+ SDK	MCU PLUS SDK Microsoft Windows インストーラ

5.3 ドキュメントのサポート

1. テキサス・インスツルメンツ、*「AM64x/AM243x プロセッサ シリコン テクニカル リファレンス・マニュアル」*
2. テキサス・インスツルメンツ、*「AM2x BLDC サーボ・モーター ブースタパック (BPAM2BLDCSERVO) EVM ユーザー ガイド」*
3. テキサス・インスツルメント、*TIDEP-0103 EtheCat® 接続型、シングルチップ、デュアル サーボ モーター ドライブのリファレンス デザイン*
4. テキサス・インスツルメンツ、*「AM243x モーター制御 SDK 09.02.00 の下にある TIDEP-01032 の例」*

5.4 サポート・リソース

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

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

5.5 商標

Sitara™, E2E™, BoosterPack™, LaunchPad™, Code Composer Studio™, and テキサス・インスツルメンツ E2E™ are trademarks of Texas Instruments.
EtherCAT® is a registered trademark of Beckhoff Automation GmbH.
PROFINET® is a registered trademark of PROFIBUS Nutzerorganisation e.V..
Arm® is a registered trademark of Arm Limited.
Python® is a registered trademark of Python Software Foundation.
Altium Designer® is a registered trademark of Altium LLC.
Microsoft® and Windows® are registered trademarks of Microsoft Corporation.
Linux® is a registered trademark of Linus Torvalds.
すべての商標は、それぞれの所有者に帰属します。

6 著者について

CHEN GAO は、テキサス・インスツルメンツの産業システム モーター ドライブ チームのシステム エンジニアであり、産業用ドライブおよびロボット向けリファレンス デザインの仕様策定と開発を担当しています。

SABARI KANNAN MUTHALAGU は、テキサス・インスツルメンツの産業システム ロボット チームのシステム エンジニアであり、ロボット向けリファレンス デザインの仕様策定と開発を担当しています。

THOMAS LEYRER は、テキサス・インスツルメンツの産業システム ファクトリー オートメーションとコントローラチームのシステム アーキテクト兼テクニカル スタッフの主要メンバーであり、産業用ドライブ、ロボットおよびファクトリー オートメーション向けリファレンス デザインの仕様策定と開発を担当しています。

謝辞:

TIDA-010948 リファレンスデザインのソフトウェア開発を支援するために、**Pratheesh Gangadhar TK**、**Dhaval Khandla**、**Achala Ram**、**Manoj Koppolu** からの多大なご協力に感謝いたします

重要なお知らせと免責事項

テキサス・インスツルメンツは、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、 テキサス・インスツルメンツ製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した テキサス・インスツルメンツ製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとします。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている テキサス・インスツルメンツ製品を使用するアプリケーションの開発の目的でのみ、 テキサス・インスツルメンツはその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。 テキサス・インスツルメンツや第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、 テキサス・インスツルメンツおよびその代理人を完全に補償するものとし、 テキサス・インスツルメンツは一切の責任を拒否します。

テキサス・インスツルメンツの製品は、 テキサス・インスツルメンツの販売条件、または ti.com 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