iMOTION™ Motion Control Engine

Software Reference Manual

本書について

本書は、英文版が正本であり、日本語版は参考として作成されています。これら両言語版の間に齟齬がある場合、英文版が優先されます。

適用範囲および目的

iMOTION™デバイスはハードウエアとソフトウェアを統合することにより、永久磁石モータの制御を行います。

これらのデバイスは、深い弱め界磁での安定した制御を含む、モータの全回転速度範囲でセンサレスあるいはセンサベースのベクトル制御（FOC, field oriented control）を行うことができます。これ以降、iMOTION™モータ制御ソフトウェアはモーション制御エンジン（MCE, Motion Control Engine）と記載します。MCE は、モータの制御以外にも力率改善（PFC）制御オプションを提供します。さらに、MCE はスクリプト機能をサポートしているため、ユーザはモータ制御や PFC を超えたシステムレベルの機能を実行したり、MCE の機能性を拡張したりできます。

本ソフトウェアリファレンスマニュアルは、以下のようなトピックを始めとする、MCEがサポートするさまざまな機能について説明しています。

- モータ、PFC、および電源ボードのパラメータ設定に使用するアプリケーション固有のレジスタ
- アプリケーション固有のハードウエアの設計、テスト、および最適化についてのガイド
- 磁束推定器、速度および電流制御ループの調整、ならびにモータ起動パラメータの最適化
- モータ駆動性能の検証とトラブルシューティングの方法

本リファレンスマニュアルには MCE のすべての機能、保護、および設定オプションが説明されていますが、製品によってはその一部しか実行できない場合があります。例えば、力率改善は専用のデバイスでしか実行できません。詳しくは、個々のデバイスのデータシートをご参照ください。

iMOTION™製品の電気的、機械的、タイミング、および品質に関するパラメータは、個々のデータシートに記載されています。データシートには、本マニュアルに記載された各機能の具体的な IO ピンについても説明があります。

本マニュアルは MCE ソフトウェアバージョン MCE FW_V1.03 を対象としています。

対象読者

本書は、モーション制御エンジン（MCE）と組み合わせて iMOTION™デバイスを使用するユーザを対象にしています。

目次

1 はじめに

本書は、iMOTION™ ソフトウェアのモータ制御、力率改善、および追加機能について説明しています。本ソフトウェアはモーション制御エンジン（MCE, Motion Control Engine）という名前で提供されています。本ソフトウェアの主な特徴は以下の通りです。

- センサレスベクトル制御：iMOTION™ デバイスの高速 ADC、内蔵オペアンプ、コンパレータ、およびモーション周辺機器を活用した、永久磁石同期モータ（表面型磁石と内蔵型磁石モータ）による高性能センサレスベクトル制御（FOC, field oriented control）。
- ホールセンサベースのベクトル制御：2/3 デジタルホールセンサおよび 2 アナログホールセンサ構成をサポート。補完的な Atan 角オプション付き。
- 初期ロータ角度検出のための角度センシング：直接クローズドループ起動と共に、初期角度センシングはモータ始動特性を改善します。
- 1 シャントまたは 3 シャントのモータ電流センシング：他に類を見ない 1 シャントおよび 3 シャントの電流再構築を行います。設定可能なゲインを持つ内蔵オペアンプおよび A/D コンバータにより、追加のアナログ/デジタル回路なしに、iMOTION™ デバイスへの直接シャント抵抗インターフェイスが可能です。1 シャントオプションでは、最小パルス方式または位相シフト方式のいずれかを使用できます。位相シフト PWM 方式では、1 シャント構成でよりよい起動性と低速性能が得られます。
- 3 相および 2 相 PWM 変調をサポート：3 相 SVPWM（ゼロベクトルの対称的配置）に比べてスイッチング損失を抑制できる 2 相 SVPWM（タイプ 3）。
- 高速かつスムーズな始動を実現する改良型磁束ベースの制御アルゴリズム：比例積分コントローラおよび過変調機能を持つ空間ベクトル変調を使用して、トルクと固定子磁束（弱め界磁）の両方の直接クローズドループ制御を実現。
- ブーストモードとトーテムポール力率改善（PFC）をサポート。
- ユーザモード UART によるネットワーキング能力：マスタモードとスレーブモードが使用可能。最大 15 ノードで、各ノードがそれぞれのアドレスを保有。同報通信機能により、すべてのスレーブを同時に更新可能。
- 15 個の再プログラム可能なパラメータブロック：15 個の設定ブロックをプログラムして制御パラメータを保存することが可能。各パラメータブロックのサイズは 256 バイト。各ブロックは個別にプログラムすることも、MCEDesigner を使って 15 個すべてを同時にプログラムすることも可能。
- 複数のモータ変数パラメータをサポート：各パラメータブロックは異なるモータまたはハードウェアプラットフォームに割り当てることが可能。
- スクリプト機能のサポートにより、ユーザはモータ制御や PFC を超えたシステムレベルの機能を実現することが可能。

表 1　　モータ制御および共通保護のリスト

保護のタイプ	説明	UL60730-1 認証
過電流（ゲートキル）（Over Current (Gate kill)）	このフォールトは過電流があったときにセットされ、PWM を遮断します。このフォールトはマスクできません。	はい
限界過電圧（Critical Over Voltage）	このフォールトは電圧がしきい値を超えたときにセットされます。ローサイドスイッチはすべてクランプされ（ゼロベクトル制動）、モータの回転を保護して制動します。ゼロベクトルはフォールトがクリアされるまで保持されます。このフォールトはマスクできません。	いいえ
DC 過電圧（DC Over Voltage）	このフォールトは DC バス電圧がしきい値を超えたときにセットされます。	いいえ
DC 低電圧（DC Under Voltage）	このフォールトは DC バス電圧がしきい値未満のときにセットされます。	いいえ
磁束 PLL 制御不能（Flux PLL Out Of Control）	このフォールトはモータの磁束 PLL がロックされていないときにセットされます。これはパラメータ設定の誤りによる可能性があります。	はい
過温度（Over Temperature）	このフォールトは温度がしきい値を超えたときにセットされます。	いいえ
ロータロック（Rotor Lock）	このフォールトはロータがロックされたときにセットされます。	はい
実行（Execution）	このフォールトは CPU 負荷が 95%を超えたときに発生します。	はい
欠相（Phase Loss）	このフォールトはモータの 1 つまたは複数の相が接続されていないときにセットされます。	はい
パラメータ読込エラー（Parameter Load）	このフォールトはフラッシュのパラメータブロックが誤っている場合に発生します。	はい
リンク切れ保護（Link Break Protection）	このフォールトは一定時間 UART 通信がない場合にセットされます。	はい
ホール信号異常（Hall Invalid Protection）	このフォールトはホールインターフェイスが無効なホールパターンを受信した場合にセットされます。	いいえ
ホールタイムアウト保護（Hall Timeout Protection）	このフォールトは一定時間ホール入力遷移が検出されなかった場合にセットされます。このフォールトは、ホールセンサ/ハイブリッドモードでロータのロック状態を検出するためのものです。	いいえ

表 2　　PFC 保護のリスト

保護のタイプ	説明	UL60730-1 認証
過電流（ゲートキル）（Over Current (Gate kill)）	このフォールトは過電流があったときにセットされ、PWM を遮断します。マスクできません。	はい
DC 過電圧（DC Over Voltage）	このフォールトは DC バス電圧がしきい値を超えたときにセットされます。	いいえ
DC 低電圧（DC Under Voltage）	このフォールトは DC バス電圧がしきい値未満のときにセットされます。	いいえ
AC 過電圧（AC over voltage）	このフォールトは PFC への AC 入力電圧がしきい値を超えたときにセットされます。	はい
AC 低電圧（AC under voltage）	このフォールトは PFC への AC 入力電圧がしきい値未満のときにセットされます。	はい
周波数異常（Frequency fault）	このフォールトは PFC への AC 入力周波数値が設定値と異なる場合にセットされます。	はい
パラメータ読込エラー（Parameter Load）	このフォールトはフラッシュのパラメータブロックに誤った値があるときに発生します。	はい

2 ソフトウェアの説明

このセクションでは、MCE のモータ制御と力率改善の特性および機能について説明します。

2.1 Motor Control: Sensor-less / Hall Sensor Based FOC

MCE は先進的なセンサレスあるいはホールセンサを用いたベクトル制御(FOC, field oriented control)アルゴリズムを用いて永久磁石同期モータ(PMSM, permanent magnet synchronous motor)を駆動します。エアギャップ一定の表面型磁石モータ(SPM, surface mounted permanent magnet motor)およびリラクタンスが変化する内蔵型磁石モータ(IPM, interior permanent magenet motor)の双方に対応しています。ハイレベルなセンサレス/ホールセンサ付きベクトル制御アルゴリズム構造を図 1 に示します。制御アルゴリズムは外側の速度制御ループと内側の電流制御ループにより構成されており、モータが目標速度で回転するようにモータの巻線電圧を制御します。弱め界磁ブロックにより、モータの回転速度範囲を上げることが可能です。

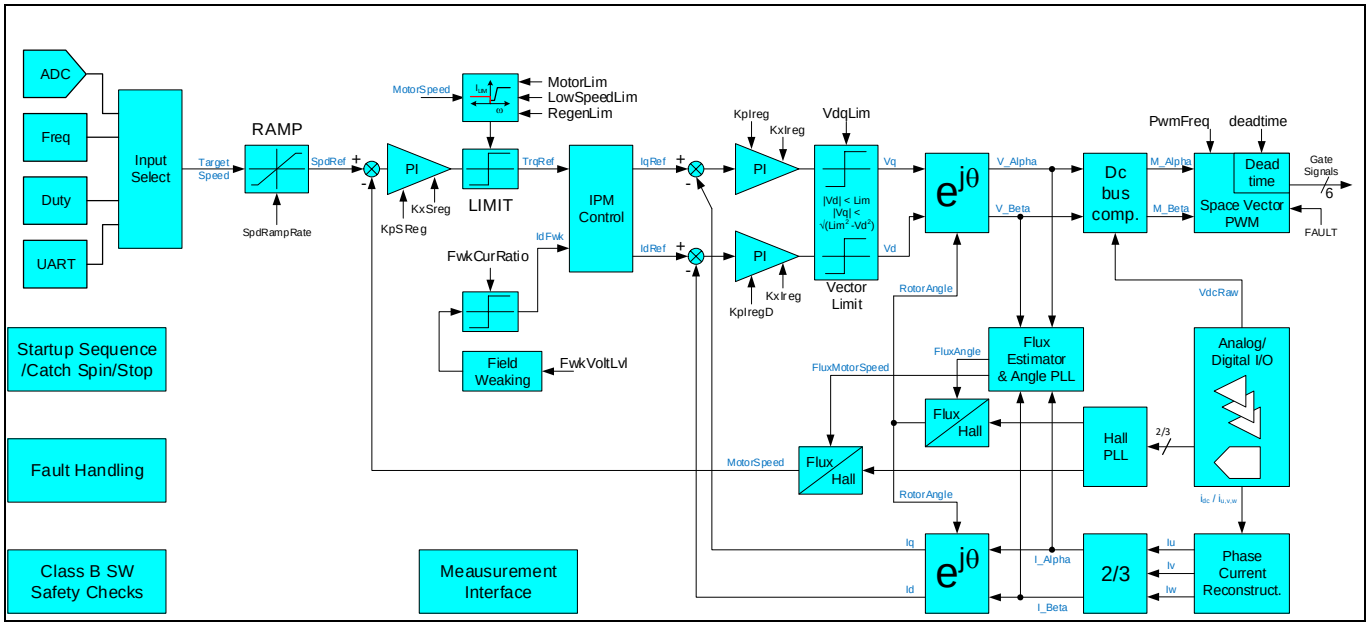


図 1 速度制御ループおよびセンサレス FOC の最上位レベル図

速度コントローラは目標速度に従って必要となるモータトルクを計算します。電流制御系はこのトルクを発生させるのに必要となるモータ電流を制御します。比例積分(PI)速度制御補償器は目標速度と実際の(推定された)速度との差分から制御を行います。積分項は定常偏差をゼロに、比例項は高速応答を改善するように働きます。PI 補償器のゲインはモータと負荷特性に依存し、目標となる動的特性に合うように調整されます。PI 補償器出力上にあるリミッタ機能は、積分器の誤差蓄積によるオーバーシュートを防止し、モータ電流がモータおよびドライブの定格内に収まるように働きます。

電流制御系は目標トルクを生成するために必要となるモータ電流を発生させるインバータ電圧を計算します。ベクトル制御はクラーク変換とベクトル回転を用いて、モータ巻線電流を 2 つの疑似的な DC 成分に変換します。一つは I_d 成分でロータの磁界の強めおよび弱めに用いられ、もう一つは I_q 成分でモータのトルクを生成するために用いられます。

2 つの独立したコントローラが I_d と I_q 電流情報から V_d および V_q 出力電圧を生成し、順方向ベクトル回転により V_d と V_q が 2 相 AC 成分(V_α、V_β)に変換されます。DC バス電圧のリップル影響を取り除くために、DC バス補償機能により出力デューティ比が DC バス電圧の関数として調整されます。これは電流制御系の安定性も改善させます。空間ベクトル PWM(SVPWM, space vector pulse width modulation)により、V_α および V_β 電圧を入力として 3 相インバータのスイッチング信号が生成されます。

一般的には、I_q制御入力は速度コントローラからの参照トルクとなり、I_dはゼロに設定されます。しかしながら、特定の速度(いわゆる base speed)以上になると、インバータ出力電圧は DC バス電圧により制約されます。このような状態では、弱め界磁制御によりマイナスのI_d電流が生成されて、巻き線の逆起電力を減少させるロータ磁界が発生します。この制御により、高速での動作が可能となりますが、トルク出力は減少します。制御系の補償器は、I_d電流がバス電圧の制約範囲内になるようにモータ電圧を維持するよう働きます。

ロータ磁石位置推定器は磁束推定器と PLL で構成されます。磁束は、フィードバック電流、推定電圧（DC バスのフィードバック電圧およびデューティ比に基づく）、およびモータのパラメータ（インダクタンスと抵抗）に基づいて計算されます。磁束推定器の出力はロータ磁石磁束を α-β（静止直交座標、U 相と α は一致）の 2 相の成分で表します。角度と周波数の位相ロックループ（PLL）は、ロータ磁束ベクトルの α-β 成分から磁束角度と速度を推定します。ベクトル回転はロータ磁束角度と推定角度の間の差分を計算します。クローズドループパス内の PI 補償器および積分器は、角度および周波数の推定値がロータ磁束の角度および周波数に追随するように動作します。モータ速度はロータの極数に応じてロータ周波数から求めます。

埋込磁石同期（IPM）モータを駆動する場合、ロータの突極性によってリラクタンストルク成分を生成して、ロータの磁石によって作られたトルクを増強することができます。表面磁石モータを駆動する場合、突極性はゼロ($L_d = L_q$)になっており、効率を最大化するためにI_dはゼロにセットされています。突極性が($L_d < L_q$)のIPMモータの場合、負のI_dによって正のリラクタンストルクが生じます。最も効率のよい動作点は、与えられた電流の大きさに対して合計トルクが最大になるときです。

2.1.1 可変スケーリング

MCE は、物理的な電圧と電流の信号が固定小数点整数によって表される固定点 CPU コア上に制御アルゴリズムを実装します。MCE アルゴリズムは、制御パラメータおよび変数に適切なスケーリングを使用して、モータおよび PFC の制御計算の精度を最適化します。制御パラメータおよび変数はすべて整数として保存される一方、MCE エコシステムツール(MCE Designer)は制御変数およびパラメータ設定を物理量に合わせてスケーリングした実数として表示する機能をサポートしています。

以下の図 2 に異なる領域で使用されるスケーリングを示します。ハードウエアの参照フレームでは、電流および電圧の測定値は入力回路スケーリングと ADC の分解能に合わせてスケーリングされます。α-β および d-q の疑似的な DC 電圧は、PWM 変調器の分解能およびインバータ DC バスの電圧能力によって定義されます。AC および制御参照フレームには異なるモータ電流スケーリングがあります。α-β 電流スケーリングは測定値のスケーリングによって定義されるのに対し、d-q スケーリングはモータの定格電流によって定義されます。モータの速度スケーリングはアプリケーション要件によって定義されます。時間スケールには 3 つの異なるスケールがあります。ハードウェアタイミングは IC 周辺機器クロックによって定義されます。サンプリングおよび制御タイミングは PWM 周波数によって設定される一方、アプリケーションの参照フレームタイミングは固定です。変数スケーリングについては後節にて詳述します。すべての制御パラメータのスケーリングは、関連する参照フレームに対する制御変数および時間スケーリングから求められます。

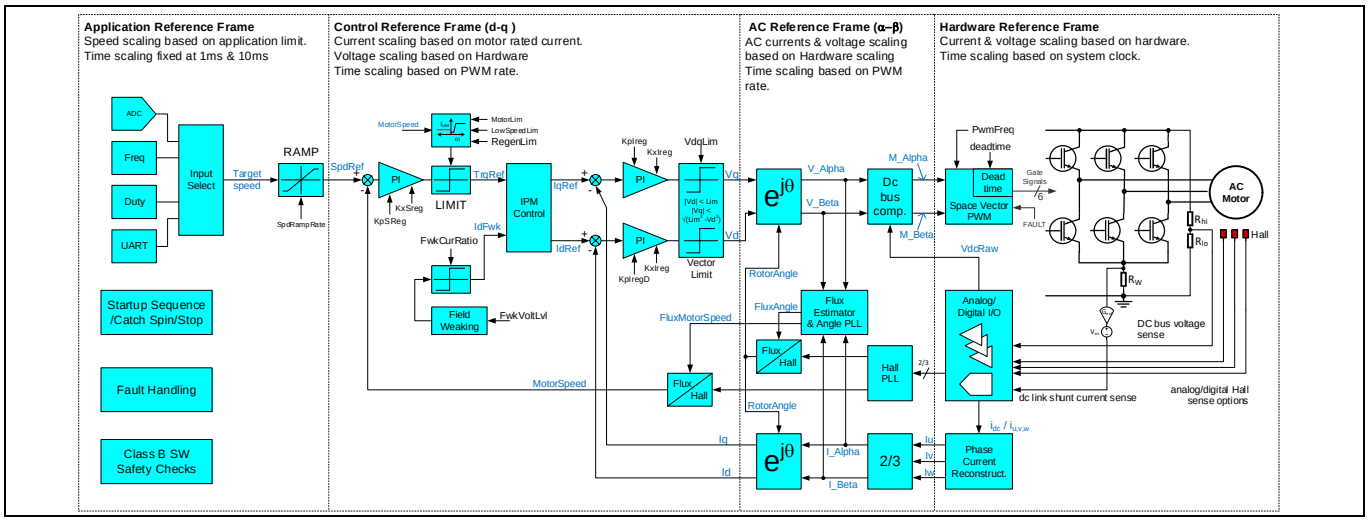


図 2　　制御変数およびパラメータのスケーリング領域

2.1.2　　ステート処理

制御ソフトウェアには、駆動電源投入からクローズドループセンサレス制御下でのモータの定常回転まで、さまざまな過渡動作条件をサポートするための異なる動作ステートがあります。例えば、始動準備ステート、磁束推定器が安定した作動状態に達するまでモータを駆動するステート、すでに回転しているモータを始動するステート、およびフォールト状態を処理するステートなどがあります。モーション制御エンジンは、始動、停止、および起動のためのすべてのステートを処理するステートマシンを内蔵しています。ステートマシン機能は定期的に実行されます（デフォルトでは 1ms 毎）。合計 10 のステートがあります。各ステートには 0 から 9 の値が割り当てられており、シーケンサの現在のステートが“SequencerState“変数に保存されます。

表 3　　ステートの説明と遷移

ステート No.	シーケンスステート	ステートの機能	遷移イベント	次のシーケンスステート
0	IDLE	コントローラの電源投入後、制御はこの状態に入ります。有効なパラメータブロックがない場合、シーケンサはこのステートに留まります。	パラメータが正常に読み込まれたとき。	STOP
1	STOP	スタートコマンド待ち。保護のための電流および電圧測定は実行されます。	電流増幅器のオフセット計算が未実行のとき。	OFFSETCAL
			スタートコマンド。	BTSCHARGE
2	OFFSETCAL	モータ電流センシング入力のためのオフセット計算。	電流オフセット計算完了。	STOP
3	BTSCHARGE	ブートストラップコンデンサのプリチャージ。保護のための電流および電圧測定は実行されます。	ブートストラップコンデンサの充電完了。	CATCHSPIN
4	MOTORRUN	正常なモータ駆動モード。	ストップコマンド	STOP

ステート No.	シーケンスステート	ステートの機能	遷移イベント	次のシーケンスステート
5	FAULT	何らかのフォールトが検出されると、モータは停止し（それまで駆動していた場合）、それ以外のステートから FAULT ステートに入ります。	UART 制御モードでは、“FaultClear“変数に 1 を書き込むことによりフォールトクリアコマンドとします。	STOP
			周波数/デューティ/VSP 入力制御モードでは、10 秒後。	STOP
6	CATCHSPIN	ロータ位置を検出し、駆動するモータの速度を測定するために、磁束推定器および磁束 PLL は動作しています。速度調整器は無効化され、電流コマンド Id と Iq はゼロに設定されます。	測定されたモータ速度の絶対値がしきい値を超えたとき（“DirectStartThr“パラメータ）。	MOTORRUN
			測定されたモータ速度の絶対値がしきい値未満（“DirectStartThr“パラメータ）。	ANGLESENSING
			“TCatchSpin“変数の値がゼロに設定されている場合、次のステートに切り替わります。	ANGLESENSING
7	PARKING	PARKING(直流励磁)ステートでは、線形的に増加する電流を注入することによりロータを既知の位置に保持します。最終電流振幅は低速電流制限値によって決まります。このステートの合計継続時間は“ParkTime“変数によって調整されます。	直流励磁完了。	OPEN_LOOP
			“ParkTime“パラメータがゼロに設定されている場合、直ちに次のステートに切り替わります。	OPEN_LOOP
8	OPENLOOP	オープンループ角度を使って、ロータを起動し、速度ゼロから MinSpd まで加速させます(強制転流)。MOTOR_RUN ステートへのスムーズな移行を実現するため、磁束推定器と磁束 PLL はこのステートで稼働します。オープンループ角度の加速レートは“OpenLoopRamp“変数によって設定されます。	速度が“MinSpd“パラメータ値に到達したとき。	MOTOR_RUN
			“OpenLoopRamp“パラメータがゼロに設定されている場合、直ちに次のステートに切り替わります。	MOTOR_RUN
9	ANGLESENSING	初期ロータ角度を測定。各センシングパルスの長さは“IS_Pulses“変数（単位: PWM サイクル）によって設定されます。	角度センシング完了。	MOTORRUN
			“IS_Pulses“パラメータがゼロに設定されている場合、直ちに次のステートに切り替わります。	PARKING

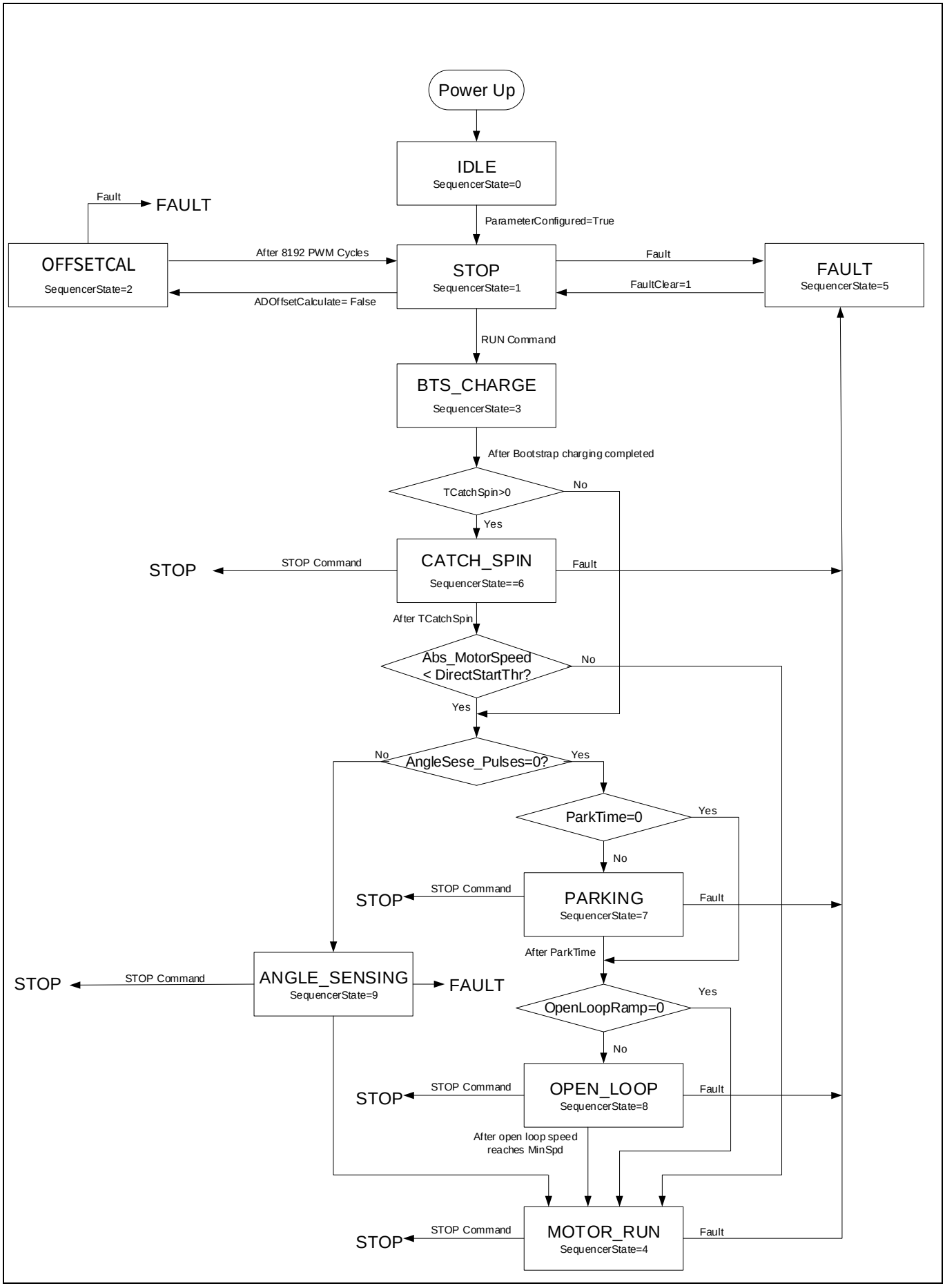


図 3　　ステート処理および始動制御フローチャート

2.1.3 電流センシングオフセット測定

電流センシングオフセットは、インバータが停止していて、モータ相電流がシャント抵抗を流れていないときに、OFFSETCAL ステート中に MCE によって測定されます。OFFSETCAL ステートの間、MCE は電流センシングオフセット値を測定します。3 シャント構成では U 相の値を IU ピンで、V 相の値を IV ピンで測定します。1 シャント構成では ISS ピンで測定します。測定はモータ PWM サイクル毎に行われ、測定サイクル数は調整可能です（$N = 2^{13-CurrentOffsetCal\_Psc}$、このとき$CurrentOffsetCal_Psc$は‘HwConfig’パラメータのビットフィールド[15:13]の値）。OFFSETCAL ステートの終わりに、各相の N 個の電流オフセット測定値が平均されて、それぞれ‘CurrentAmpOffset0’変数および‘CurrentAmpOffset1’変数に保存されます。

OFFSETCAL ステートの継続時間T_{Offset_Cal}は以下の式によって推定できます。

$$T_{Offset\_Cal} = \frac{Fast\_Control\_Rate \times 2^{13-CurrentOffsetCal\_Psc}}{F_{PWM}}$$

デフォルトでは、$CurrentOffsetCal_{Psc} = 0$で、そのため OFFSETCAL ステートの継続時間は 8192 PWM サイクルとなります。

2.1.4 ブートストラップコンデンサの充電

ブートストラップコンデンサの充電は 3 つのすべてのローサイドスイッチをオンにすることで行います。充電電流は内蔵プリチャージ制御機能によって制限されます。

すべてのローサイドデバイスを同時に充電するのではなく、ゲートプリチャージ制御が交互の（U、V、W 相）充電シーケンスのスケジュールを設定します。各相はブートストラップコンデンサを PWM サイクルの 1/3 の間充電するため、各コンデンサの充電時間は合計プリチャージ時間の 1/3 になります。

図 4 にブートストラップコンデンサのプリチャージステート中の PWM 信号を図示します。

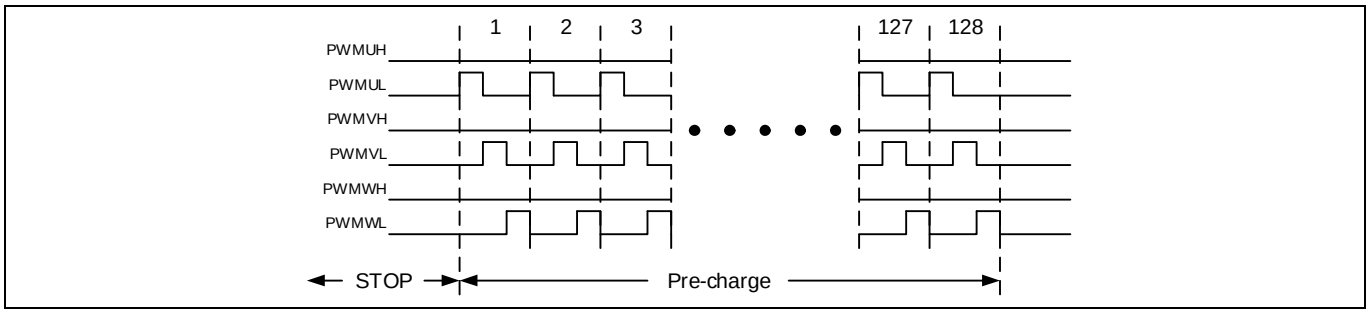


図 4　ブートストラップコンデンサのプリチャージ

各相の合計プリチャージ時間は次の式で計算できます。

$$T_{Charge} = \frac{BtsChargeTime}{3 * F_{PWM}}$$

このとき、‘BtsChargeTime’パラメータはプリチャージ PWM サイクル数です。

たとえば、PWM 周波数が 10kHz で BtsChargeTime が 100 のとき、各相のプリチャージ時間は次のようになります。

$$\frac{100}{3 * 10000} = 3.333\ (\text{ms})$$

2.1.5 電圧測定

インバータ基板の DC バス電圧の測定は、電圧保護および DC バス電圧補償を行うために必要です。電圧は PWM サイクル毎に測定されます。インバータの DC バス電圧は 12 ビット ADC を使って分圧回路経由で測定されます。測定された DC バス電圧は内部的には 12 ビットフォーマットで表されます。

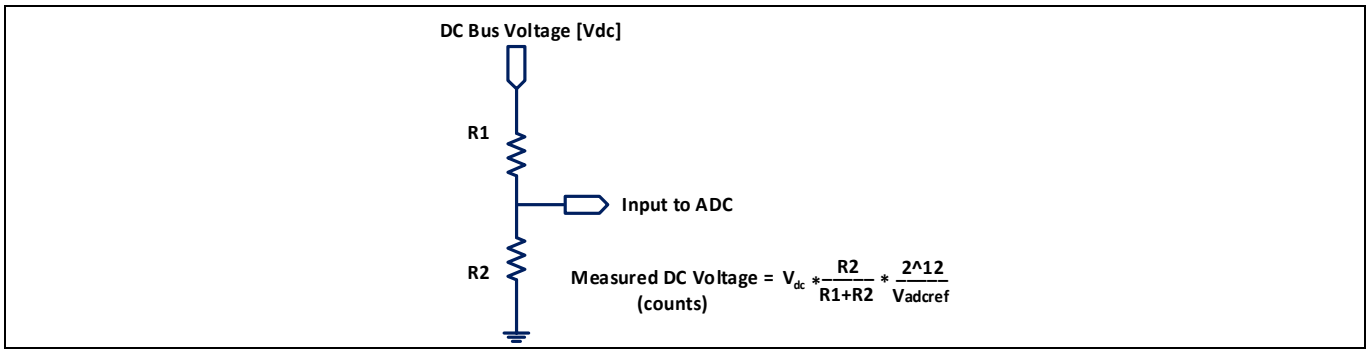


図 5　DC バス電圧フィードバック信号パス

例：　R1 = 2MΩ、R2 = 13.3kΩ、V_{adcref}= 3.3V、および V_{dc} = 320V；測定された DC バス電圧 = 2623 カウント

重要：　MCEWizard では R1 と R2 の値は実際のハードウエアに即して調整して下さい。調整を誤ると、誤った低電圧/過電圧/限界過電圧フォールトが発生したり、低電圧/過電圧/限界過電圧フォールトが正しく検出できなくなったりする可能性があります。

2.1.6 DC バス補償

一般的に、DC バス電圧には高周波リップルと AC 入力ライン周波数（Fline）の 2 倍の周波数のリップルがあります。DC バスコンデンサのサイズには制限があるため、低周波リップルが優勢です。瞬時 DC バス電圧はモータ電流制御ループゲインの一部であるため、電流制御帯域幅が狭く 2 x F_{line} 周波数で十分なループゲインが得られない場合、電流ループはそれに応じてデューティ比（MI）を調整し、安定したインバータ出力電圧を確保することができなくなります。その結果、モータ相電流は不可避的に 2 x F_{line} 周波数 DC バス電圧リップルによって変調されることになります。

MCE は DC バス電圧にフィードフォワード機能を提供して電流制御ループゲインへの DC バス電圧変動の影響を補償し、実際の MI が DC バス電圧リップルによって影響されないようにします。

以下の図 6 に示すように、DC バス補償を有効にすると、必要なインバータ出力電圧の 2 つの直交成分である V_Alpha と V_Beta は、DC バスの最大範囲電圧の 50%と瞬時 DC バス電圧の比率によって調整されます。調整された結果である M_Alpha と M_Beta は、必要な出力電圧ベクトルの 2 つの直交成分で、これに基づいて SVPWM ブロックは 3 相の PWM スイッチング信号を生成します。（M_Alpha と M_Beta は内部変数で、ユーザはアクセスできません。）DC バス補償を無効にすると、V_Alpha と V_Beta は追加の調整なしに直接 M_Alpha と M_Beta に結合されます。

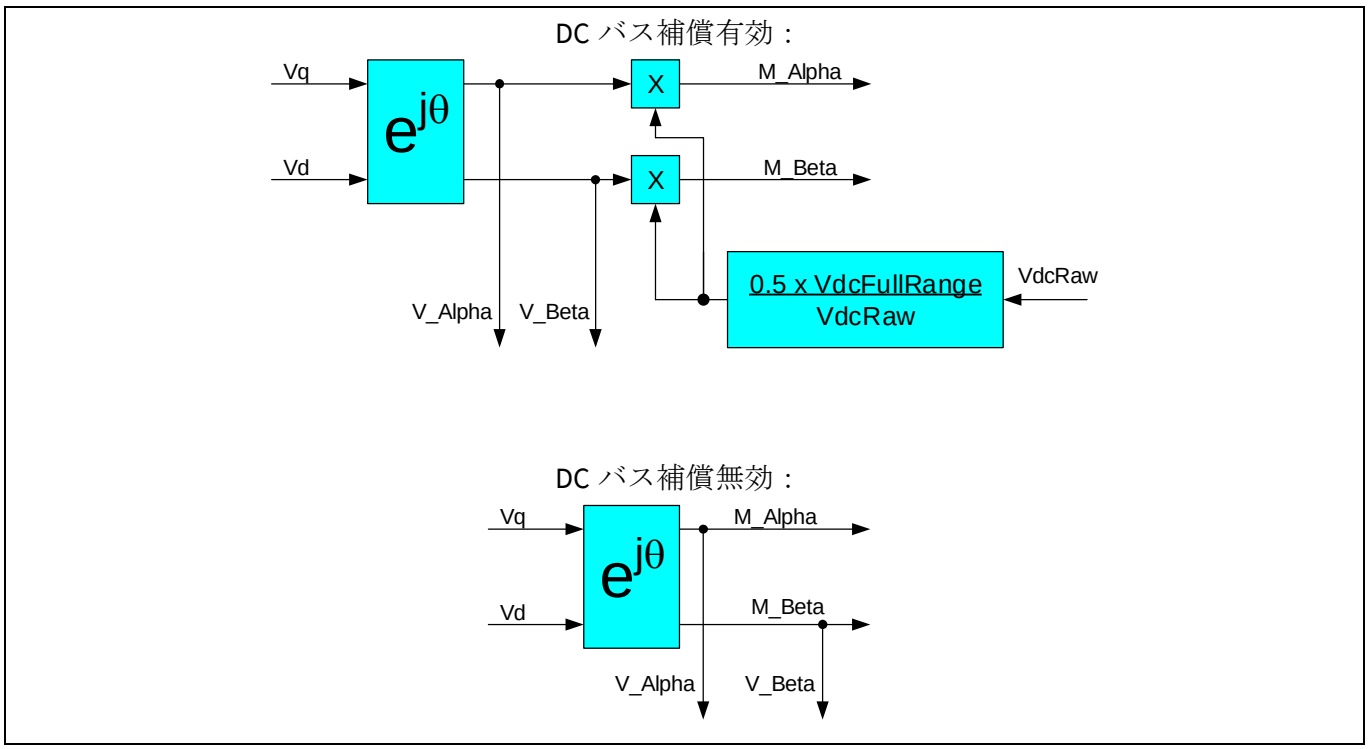


図 6　　DC バス補償機能の図

DC バスの全電圧範囲は、抵抗分圧された範囲にて ADC が測定できる最大 DC バス電圧です。図 5 を参照して、以下の数式を使って計算することができます。

$$V_{DCFullRange} = V_{ADC\_REF} \times \frac{R1 + R2}{R2}$$

DC バス補償機能は‘SysConfig’パラメータの[0]ビットを設定することで有効にすることができます。

DC バス補償機能を無効にすると、実際の MI は次の数式により‘MotorVoltage’変数を使って推定することができます。

$$MI = \frac{MotorVoltage}{4974}$$

DC バス補償機能を有効にすると、実際の MI は次の数式により‘MotorVoltage’変数と‘VdcRaw’変数を使って推定することができます。

$$MI = \frac{MotorVoltage \times \frac{2048}{VdcRaw}}{4974} \times 100\%$$

2.1.7　　電流測定

センサレスベクトル制御を実行するに当たっては、モータ巻線電流を正確に測定することが非常に重要です。モータ巻線電流の測定値は電流制御と磁束推定に用いられます。電流は PWM サイクル毎に測定されます。表 4 に、MCE での電流測定条件を示します。それぞれの詳細条件およびそれに対応した PWM 方式については、後節にて詳述します。

表 4　電流測定条件と PWM 波形

電流測定条件	シャント抵抗数	PWM 波形
3 シャント	3	センターアライン対称
1 シャント	1	センターアライン対称 　最小パルス幅 PWM センターアライン非対称 PWM 　位相シフト PWM 　ローノイズ位相シフト PWM

電流測定には内蔵アンプを用いることができ、外付けオペアンプは必要ありません。内蔵アンプのゲインは、MCEWizard で調整可能です。

図 7 に、モータ相電流のフィードバック信号処理の流れを示します。

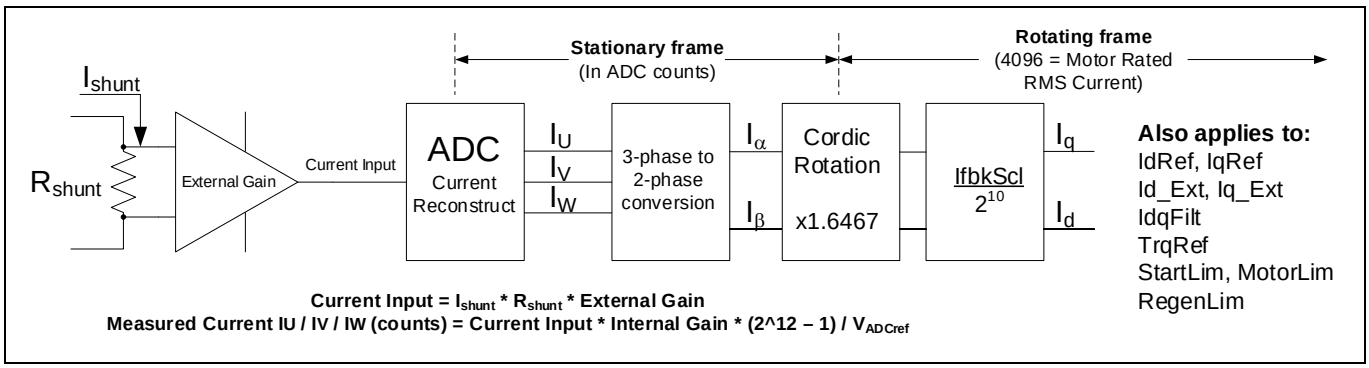


図 7　電流フィードバック信号パス（TminPhaseShift ≠ 0）

重要：　MCEWizard では電流入力値は実際のハードウエアに即して調整する必要があります。調整を誤ると、意図しない過電流保護が働いたり、過電流状態が正確に検知できなくなる可能性があります。

2.1.7.1　3 シャント電流測定

3 個のシャント抵抗を用いてインバータの 3 相電流を計測する、3 シャントでの回路構成例を図 8 に示します。MCE は U 相と V 相の電流を測定し、W 相電流は 3 相の電流の総和がゼロであるという条件に基づき計算により求められます。モータ相電流はローサイドのスイッチが ON のときのみにシャント抵抗を流れます。したがって、MCE は PWM サイクルの開始近傍のゼロベクトル[000]期間内に U 相と V 相の電流を測定します。図 9 に示されるゼロベクトル[000]の最短継続時間(T_{GB_min})は、モータ相電流測定のためのサンプリングをするのに十分なだけ維持される必要があります。この最短継続時間は 'PwmGuardBand'パラメータで規定され、TGB_min は

$$T_{GB_{min}} = PwmGuardBand \times 10.417ns$$

により計算されます。このゼロベクトル[000]の最小継続時間により、各 PWM 信号の ON 時間は、TPWM を PWM 周期としたとき、$T_{PWM} - T_{BGmin}$ を超えることはありません。

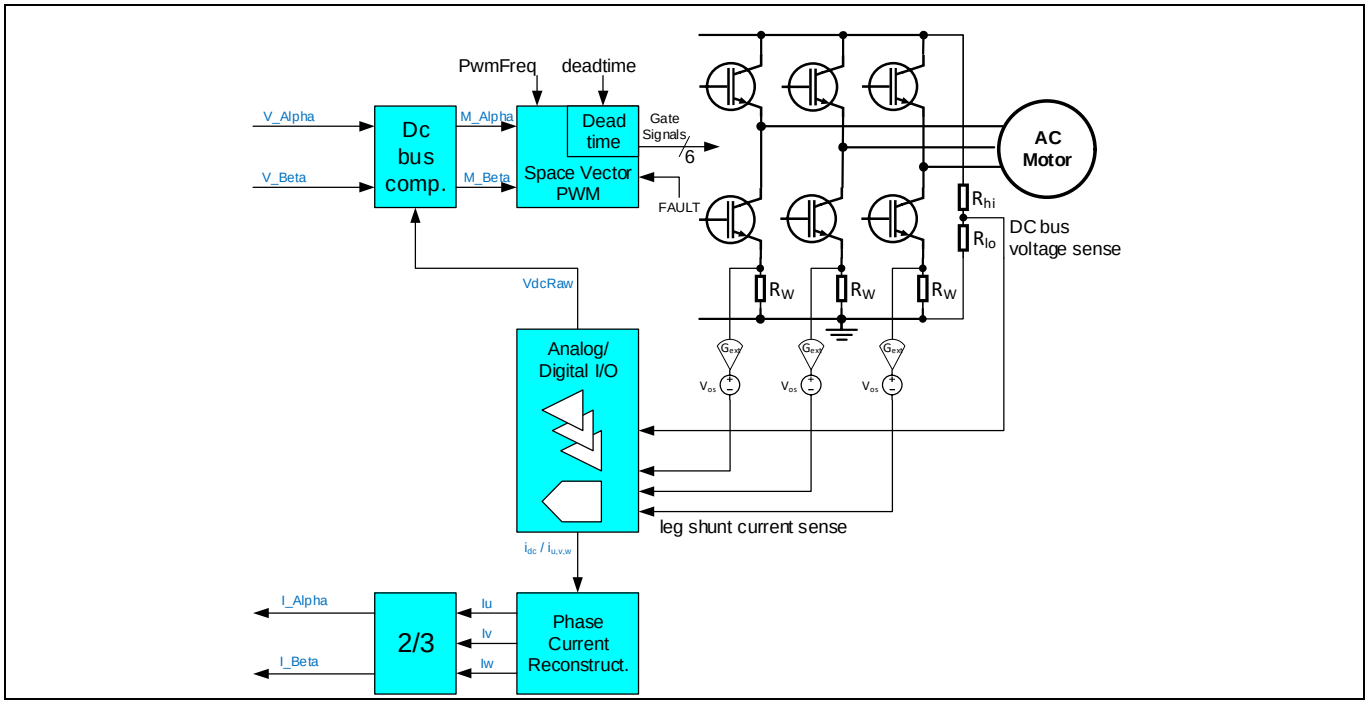


図 8　　3 シャント電流測定の回路構成例

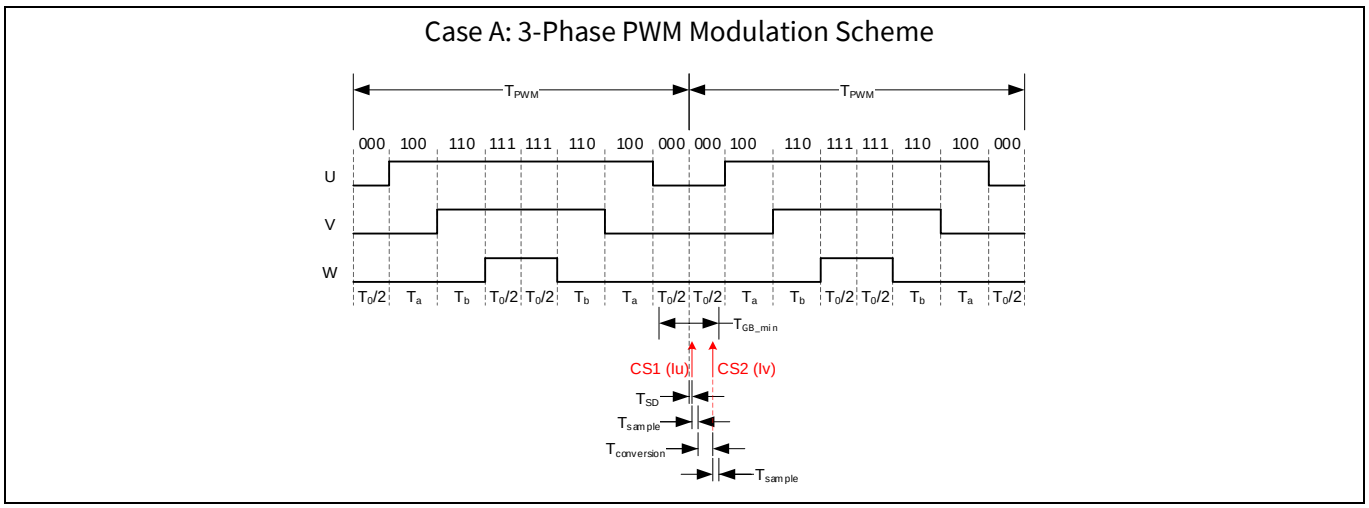
3 シャント構成での電流計測タイミングを図 9 に示します。各 PWM サイクルにおいて、T_aおよびT_bは、それぞれ 2 つのアクティブベクトル([100], [110])の継続時間を示します。また、2 x T_0は、ゼロベクトル([000]、[111])の継続時間を示します。ゼロベクトル[111]の継続時間は[000]と等しくなります。最初の電流計測点(CS1)は U 相電流の測定点で、PWM サイクルが開始されてから下記の時間経過後に発生します。

$$T_{SD} + \frac{4}{f_{PCLK}} = T_{SD} + 41.668ns$$

ここでf_{PCLK}はクロック周波数(96 MHz)、T_{SD}は ADC のサンプリング遅延時間で、下記の数式により'SHDelay'パラメータを用いて調整します。

$$T_{SD} = SHDelay \times 10.417ns$$

ADC のサンプリング時間 T_{sample} は約 $0.333\mu s$で、ADC の変換時間$T_{conversion}$は約$0.854\mu s$です。2 回目の電流計測点(CS2)では、V 相電流を計測します。CS2 は、CS1 のサンプリングと変換操作が完了後ただちに発生します。



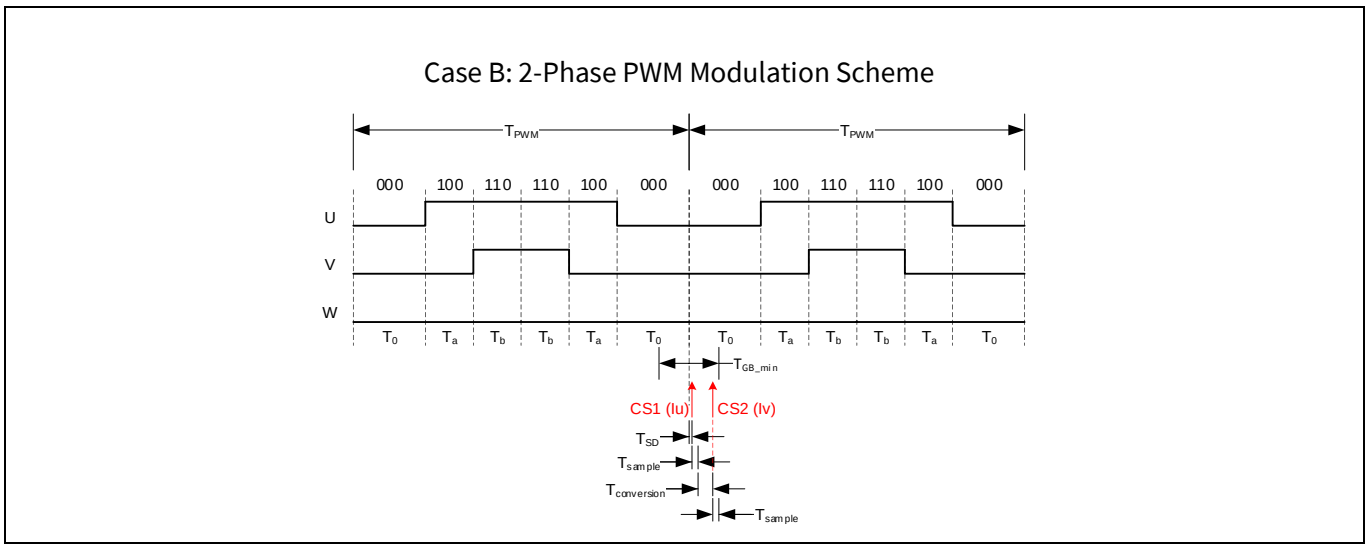


図 9　　3 シャント時の電流計測タイミング

2.1.7.2　1 シャント電流測定

1 個のシャント抵抗で DC リンクの電流を測定する、1 シャントでの回路構成例を図 10 に示します。1 シャント方式は、システムコスト削減に有効です。この構成においては、DC リンク電流のみが MCE にてサンプリング計測され、モータの相電流情報は、各 PWM サイクルにてアクティブベクトル(ゼロベクトルでない)が与えられている期間に測定された DC リンク電流から計算されます。各 PWM サイクルで異なる 2 つのアクティブベクトルが与えられ、そのアクティブベクトル期間の DC リンク電流は、セクタ情報に依存する特定のモータ電流により決定されます。3 つめのモータ相電流は、相電流の総和がゼロであるという条件により計算されます。

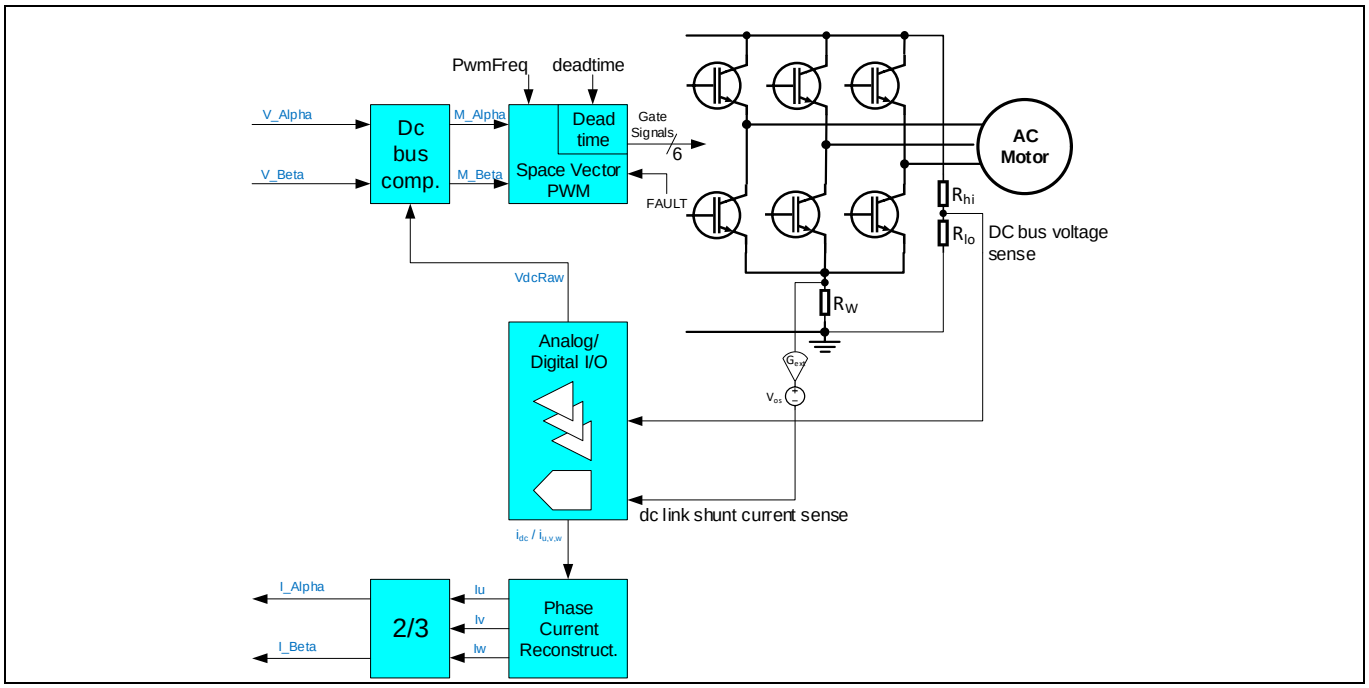


図 10　　1 シャント電流測定の回路構成例

1 シャント構成における電流計測タイミングを図 11 に示します。最初の電流計測点(CS1)にて、アクティブベクトル期間での電流値(図 11 の例ではアクティブベクトル[110]の期間でマイナスの W 相電流値)を計測します。CS1 は、このアクティブベクトル(図 11 の例では[110])の開始から下記の時間経過後に発生します。

$$\frac{T_b}{2} + T_{SD}$$

2 番目の電流計測点(CS2)にて、第二のアクティブベクトル期間での電流値(図 11 の例ではアクティブベクトル[100]の期間で U 相電流値)を計測します。CS2 は、このアクティブベクトル(図 11 の例では[100])の開始から下記の時間経過後に発生します。

$$\frac{T_a}{2} + T_{SD}$$

T_{SD}は ADC のサンプリング遅延時間で、下記の数式により'SHDelay'パラメータを用いて調整します。

$$T_{SD} = SHDelay \times 10.417ns$$

上述の数式により計算された CS1 と CS2 の計測点が PWM サイクルの終点より後になった場合は、実際の CS1 と CS2 は PWM サイクルの終了直前に発生するように調整され、次の PWM サイクルの開始時点で最新の電流値がベクトル制御の計算に利用可能となります。これについては、この後で詳細に説明します。

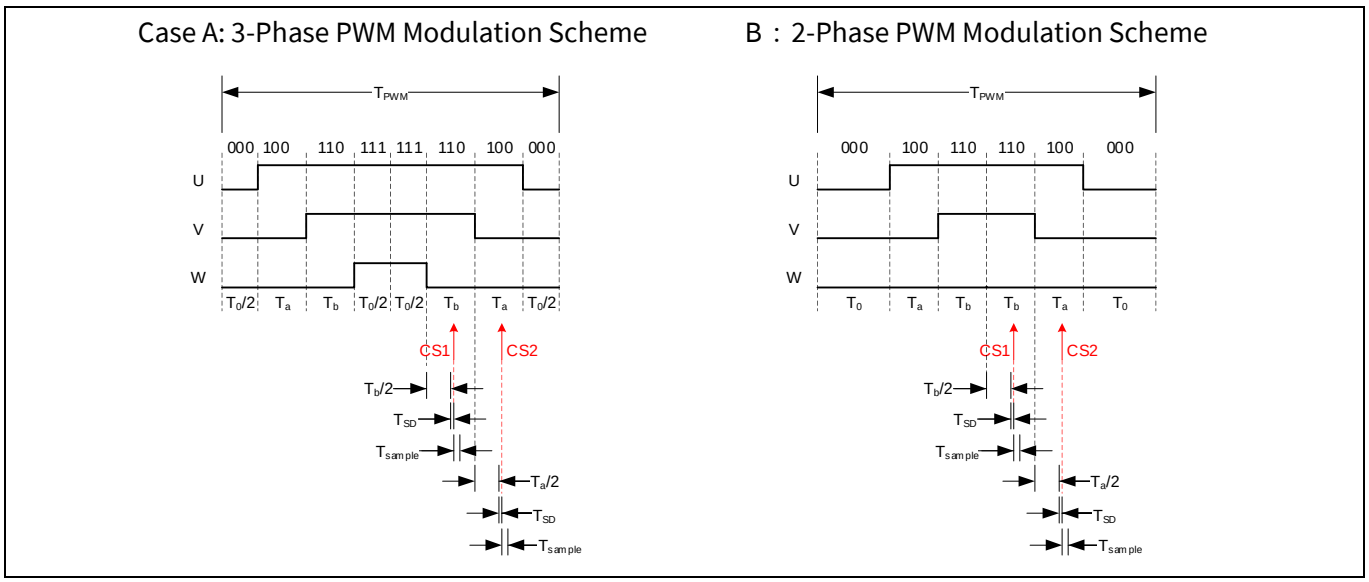


図 11　1 シャント構成の電流計測タイミング

2.1.7.2.1　最小パルス幅 PWM

1 シャントでの電流再構築においては、単一相の電流値は各アクティブベクトル期間にシャント抵抗を流れる電流にて計測されます。しかし、目標となる電圧ベクトルがセクタ境界近傍あるいは電圧ベクトルの長さが短い(デューティ比が小さい)等のある特定の動作条件においては、電流計測を行うアクティブベクトルの 1 つあるいは 2 つとも期間が短く、正しい巻線電流の計測を行うのが難しいことがあります。そのような動作条件は図 12 の空間ベクトル図の斜線で示した領域になります。図 12 の例では、アクティブベクトル[110]の継続期間 T_bが短く正確な電流計測を行うのには不十分となります。

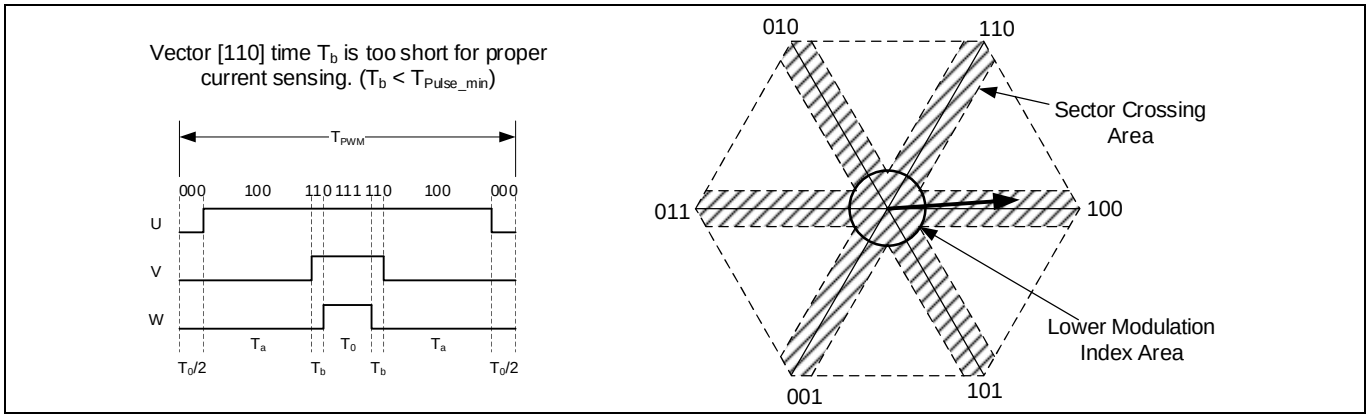


図 12　1 シャント電流計測におけるパルス幅が短くなる例

巻線電流の正確な計測を行うため、最小パルス幅制限(T_{Pulse_min})が PWM サイクル内の各アクティブベクトルに対して挿入されます。この最小パルス幅は、'TcntMin'パラメータにより設定されます。TPulse_min と TcntMin の関係は下記の式のようになります。

$$T_{Pulse\_min} = TcntMin \times 10.417ns$$

このモードにおいては、制御性能を最適化するために、'SHDelay'パラメータを実際のハードウエアに即して調整する必要があります。この最小パルス幅制限は、デューティ比が小さい場合あるいは目標となる電圧ベクトルがセクタ間を遷移する場合に出力電圧の乱れを引き起こします。これは出力電圧の目標値と実際の値に違いが生じるためです。この電圧乱れは、特に低速運転時において、音響ノイズの原因となり、制御性能を劣化させます。図 12 の空間ベクトル図の斜線部分は、出力電圧の乱れが発生する領域を示しています。図 13 に、電圧ベクトルの目標値がセクタ間を遷移する際に生じる電圧ベクトルの乱れを示します。図 13 に示すように、セクタ間遷移期間中の電流計測を行うため、アクティブベクトル[101]と[110]の継続時間 Tb が$T_b' = T_{Pulse_min}$に延長されています。この際、目標出力(黒線)と実際の出力(赤線)との間に違いが発生します。

最小パルス幅 PWM 方式を用いた 1 シャント構成での電流計測のタイミング(CS1、CS2)は、図 11 に示されている通りです。

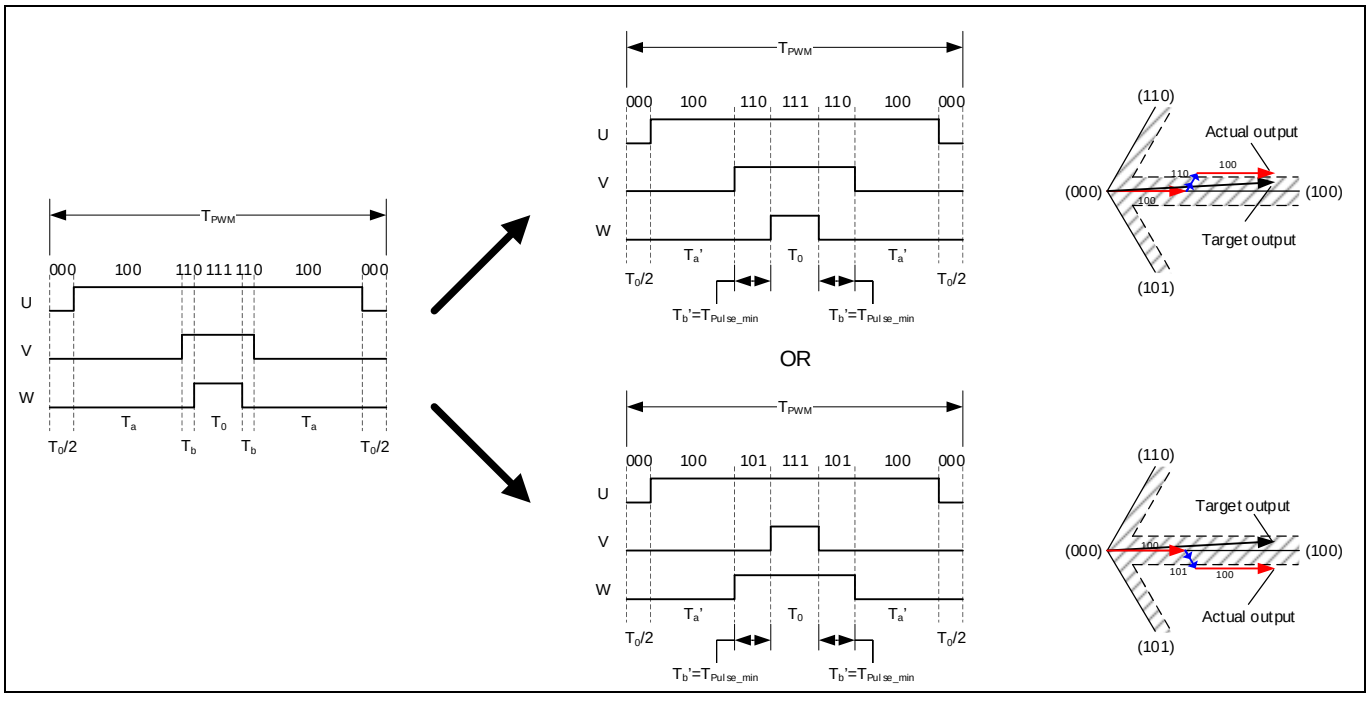


図 13　最小パルス幅制限

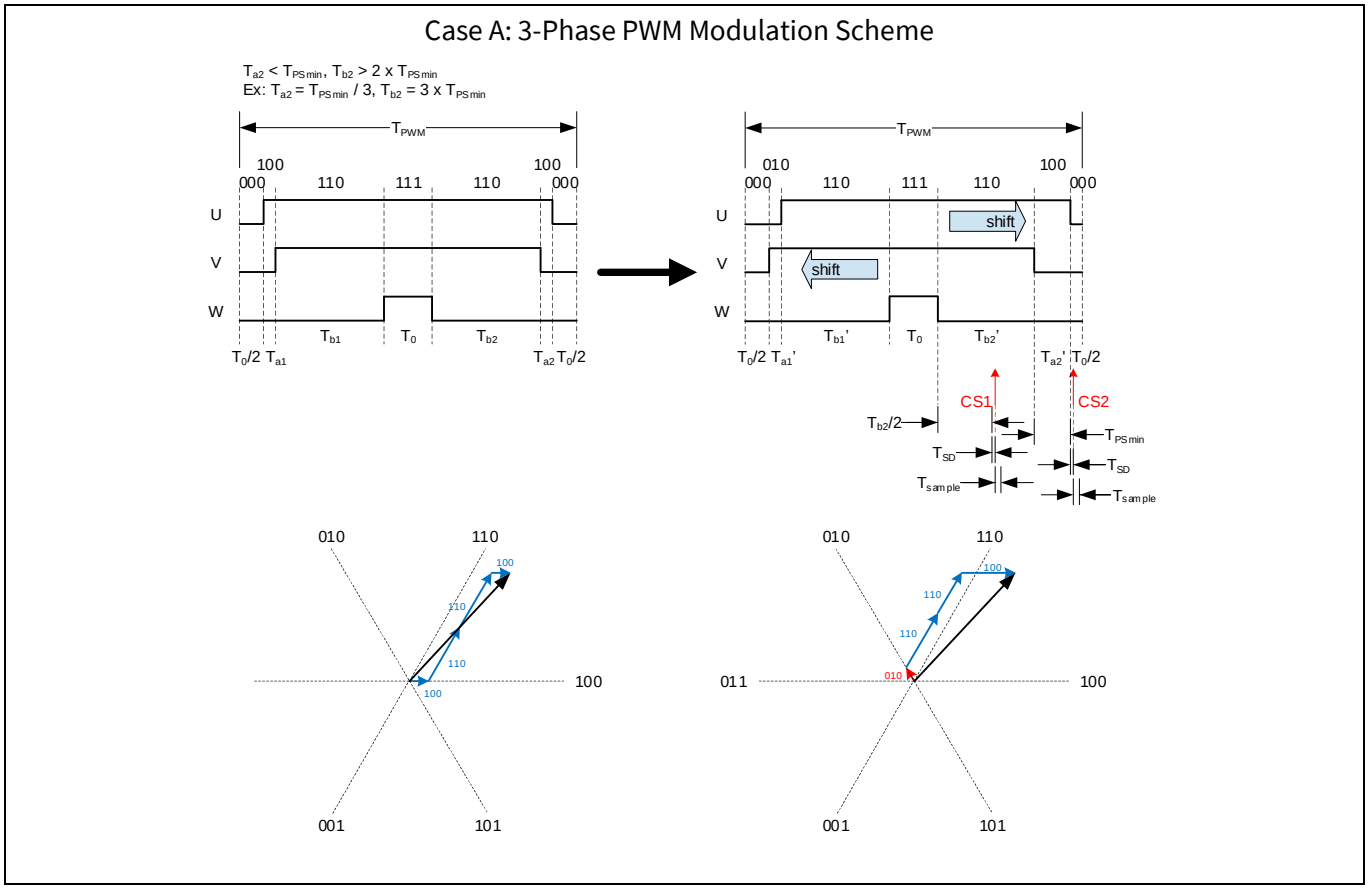
2.1.7.2.2 位相シフト PWM

最小パルス制限での弊害を防止するために、MCE では位相シフト PWM 方式がオプションとして用意されています。位相シフト PWM 方式では、各 PWM 出力はセンターアラインではありません。相電流の適切なサンプリングを保証するために、最小のアクティブベクトル継続時間(T_{PSmin})が設定されます。T_{PSmin} は'TminPhaseShift'パラメータを用いて設定されます。T_{PSmin} と TMinPhaseShift の関係は下記になります。

$$T_{PSmin} = TminPhaseShift \times 10.417ns$$

アクティブベクトル継続時間の目標値(T_a あるいは T_b)が T_{PSmin} よりも長い場合、PWM パターンは変化しません。アクティブベクトル継続時間の目標値(T_a あるいは T_b)が T_{PSmin} よりも短い場合、フォーリングエッジ側(電流計測を行う時間領域)におけるアクティブベクトル継続時間が T_{PSmin} 以上となるように、3 相 PWM パターンがシフトします。

図 14 に示すように、フォーリングエッジ側におけるアクティブベクトル[110]および[100]の継続時間がそれぞれ T_{b2}、T_{a2} となっています。要求される最小のアクティブベクトル継続時間 $T_{PSmin} = 3 \times T_{a2}$ の場合、T_{a2} は電流計測には不十分である一方、$T_{b2}T_{b2}$ は十分です。その結果、アクティブベクトル継続[100]に対して十分な継続時間($T_{a2}' = T_{PSmin}$)を確保するために、U 相 PWM を右側に、V 相 PWM を左側にシフトさせる必要が生じます。図 14 にあるように、PWM 位相シフト動作では、元々は存在していない赤線で示されるアクティブベクトル[010]を追加しています。この追加されたベクトルの影響は、フォーリングエッジ側のベクトル[100]の延長とライジングエッジ側の[110]の縮小により相殺されます。



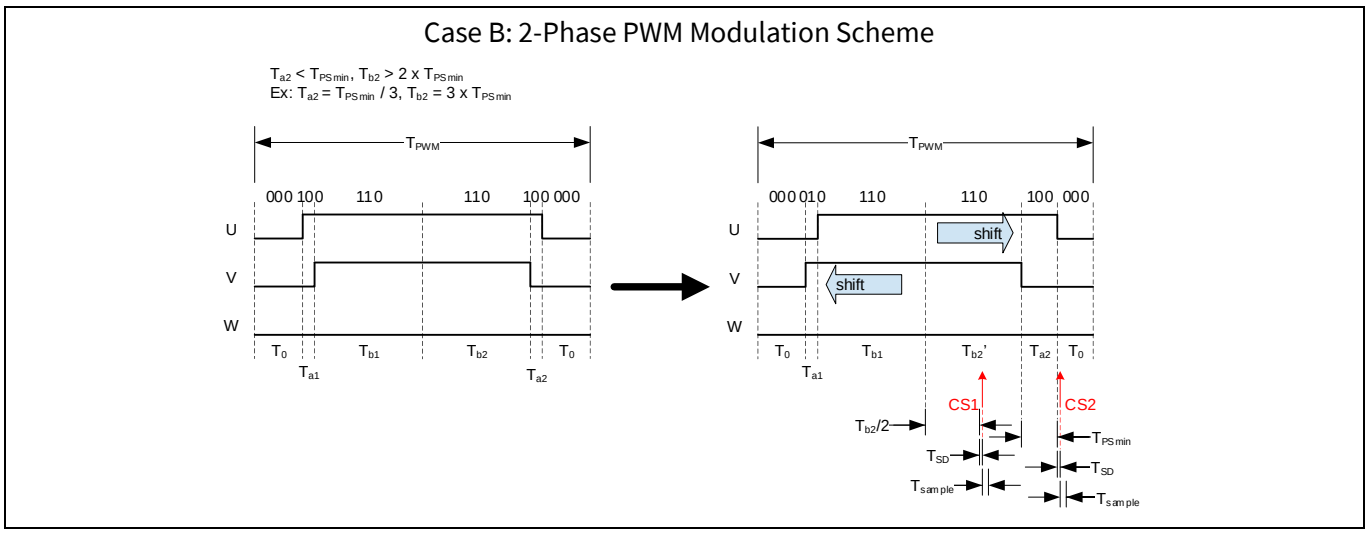


図 14　位相シフト PWM を使用した 1 シャント構成での電流計測タイミング (Case 1: $T_{a2} < T_{PSmin}$、 $T_{b2} > 2 \times T_{PSmin}$)

図 15 では、$T_{PSmin} = 3 \times T_{b2}$ の場合を示しています。この場合、T_{b2} は電流計測には十分でない一方、T_{a2} は十分です。その結果、アクティブベクトル[110]の継続期間を十分確保するため($T_{b2}' = T_{PSmin}$)、W 相 PWM を左側にシフトさせる必要が生じます。図 15 に示されているように、位相シフト動作では元々は存在していない赤線で示されるアクティブベクトル[101]を追加しています。この追加されたベクトルの影響は、フォーリングエッジ側のベクトル[110]の延長とライジングエッジ側の[100]の縮小により相殺されます。

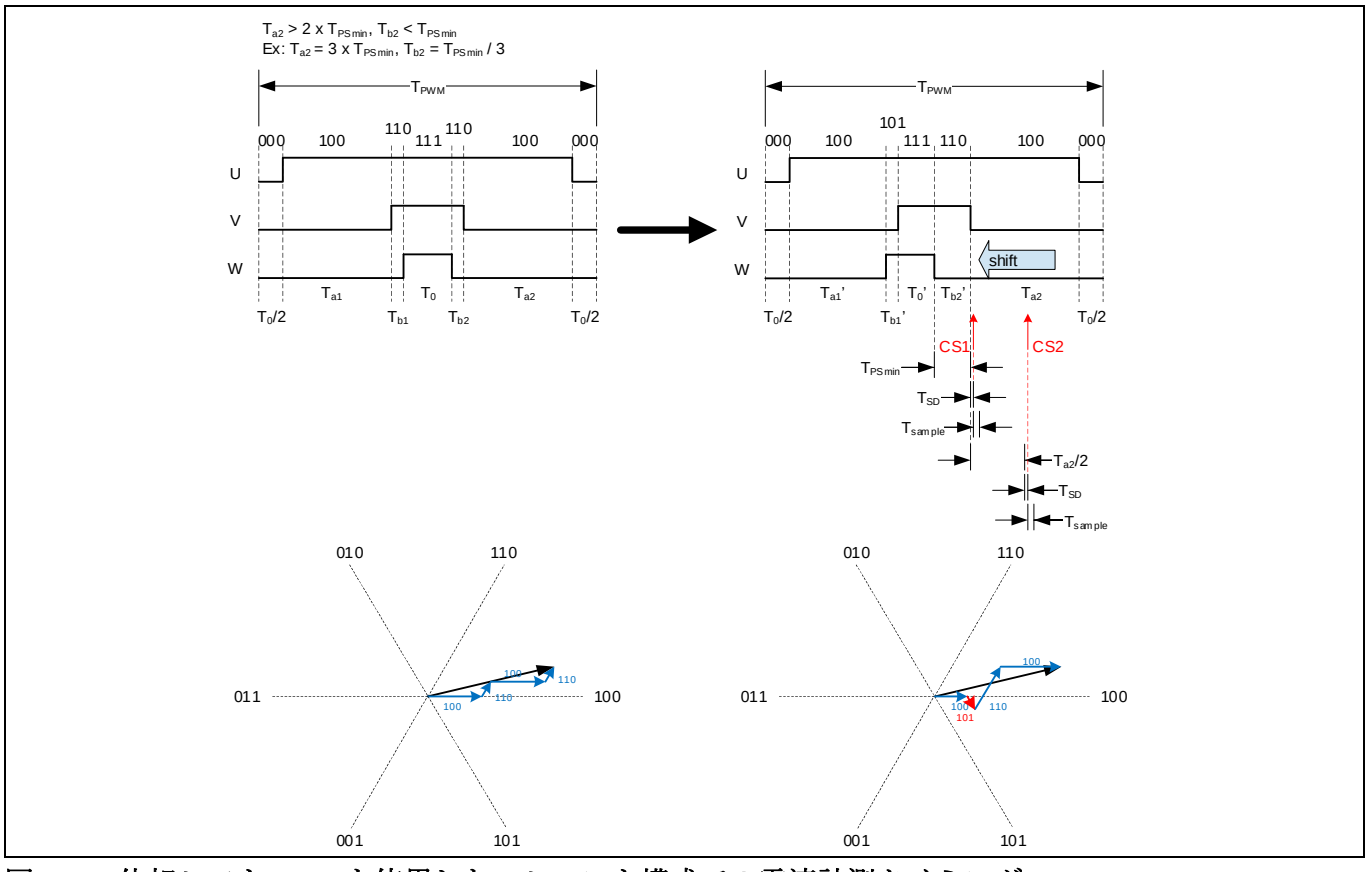


図 15　位相シフト PWM を使用した 1 シャント構成での電流計測タイミング (Case 2: $T_{a2} > 2 \times T_{PSmin}$, $T_{b2} < T_{PSmin}$)

位相シフト PWM 方式を用いた 1 シャント構成の電流計測タイミングは、アクティブベクトル継続時間(T_a、T_b)と最短アクティブベクトル継続時間(T_{PSmin})の関係に依存します。

Ta あるいは T_b が T_{PSmin} の 2 倍以上である場合、対応する電流計測点はそのアクティブベクトル時間の中心にサンプリング遅延 T_{SD} を加えた点になります。T_b が T_{PSmin} の 2 倍以上である場合が図 14 に、T_a が T_{PSmin} の 2 倍以上である場合が図 15 に示されています。これは図 11 に示される電流計測タイミングと一致しています。

T_a あるいは T_b が T_{PSmin} 以上 2×T_{PSmin} 以下の範囲にある場合、対応する電流計測点(CS1 あるいは CS2)はアクティブベクトルが開始してから$T_{PSmin} + T_{SD}$の時点となります。図 16 の事例では、T_a と T_b はともに T_{PSmin} と 2xT_{SD} の間にあります。そのため、CS1 はアクティブベクトル[110]の始点から T_{PSmin}+T_{SD} 経過後に発生し、CS2 はアクティブベクトル[100]の始点から T_{PSmin}+T_{SD} 経過後に発生します。

T_a あるいは T_b が T_{PSmin} より小さい場合、最小のアクティブベクトル継続時間 T_{PSmin} を確保するために、必要とされる位相シフトが適用されます。したがって、対応する電流計測点は T_{PSmin} に T_{SD} を加えた時点となります。図 14 では、T_{a2} は T_{PSmin} より小さいため、T_{a2}' = T_{PSmin}.となるように位相シフトが適用されます。対応する CS2 は T_{a2}'の終点からT_{SD} 経過後となります。図 15 では、T_{b2} は T_{PSmin} より小さいため、T_{b2}' = T_{PSmin}.となるように位相シフトが適用されます。対応する CS1 は T_{b2}'の終点からT_{SD} 経過後となります。

上記で算出される CS1 および CS2 が PWM サイクルの終点以降となる場合は、実際の CS1 および CS2 は、次のベクトル制御演算が実行される PWM サイクルにて電流値を利用できるよう、該当となる PWM サイクルの終点直前に実行されます。

位相シフト方式を利用することで、各 PW サイクル中の出力電圧値が目標値と完全に一致し、低速での制御性が最小パルス幅 PWM 方式に対して改善します。このモードで最適な制御性能を実現するために、2 つのパラメータ'TMinPhaseShift'(=TPSmin)と'SHDelay'(=TSD)を適切な値に調整する必要があります。

a

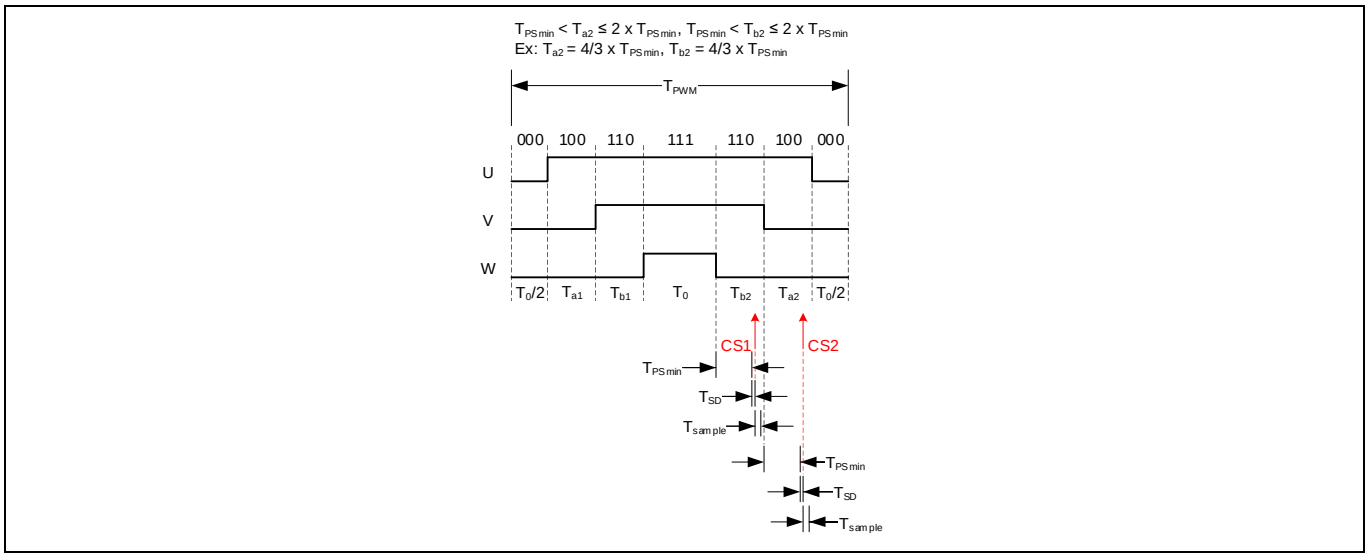


図 16　1 シャント構成/位相シフト PWM での電流計測タイミング (Case 3: $T_{PSmin} \le T_{a2} \le 2 \times T_{PSmin}$、$T_{PSmin} \le T_{b2} \le 2 \times T_{PSmin}$)

2.1.7.2.3 ローノイズ位相シフト PWM

上述した位相シフトによる欠点の一つとして、シフトパターンがセクタ毎に異なり、セクタ遷移期間中のシフトパターンの変化が、特に低速運転時において、音響ノイズを発生させる恐れがあります。

MCE は低速運転時の音響ノイズを更に低減させるため、「ローノイズ位相シフト PWM 方式」の機能を有しています。通常の位相シフト PWM 方式と比較して、ローノイズ位相シフト PWM 方式は全ての 6 種類の PWM セクタ対して、シフトパターンを固定させます。これにより、シフトパターンの変化による音響ノイズを低減させることが可能です。

図 17 に示すように、W→V→U 相の順で固定したシフトパターンが選択され、1 シャント電流計測に利用できるベクトルは[110]と[100]となります。これら 2 つのアクティブベクトルを用いて、W 相と U 相モータ電流を連続して計測します。これら 2 つのベクトルの継続時間(T_{PSmin})は'TMinPhaseShift'パラメータを用いて調整可能です。T_{PSmin} と TMinPhaseShift の関係式は下記のようになります。

$$T_{PSmin} = TminPhaseShift \times 10.417ns$$

図 17 に、ローノイズ位相シフト PWM 方式を用いて、セクタ境界領域(斜線の領域)内にある 5 種類の出力電圧ベクトル例(A~E)を示します。

例 A においては、ベクトル[110]と[100]が利用可能ですが、ベクトル[100]は U 相電流を正確に測定するには短すぎます。ローノイズ位相シフト PWM 方式を用いて、V 相および W 相 PWM が非対称にシフトし、ベクトル[100]の期間を延長して U 相電流計測に必要なウインドウ幅を確保します。

例 B においては、ベクトル[110]と[100]が利用可能ですが、ベクトル[110]は W 相電流を正確に測定するには短すぎます。ローノイズ位相シフト PWM 方式を用いて、V 相および W 相 PWM が非対称にシフトし、ベクトル[110]の期間を延長して W 相電流計測に必要なウインドウ幅を確保します。

例 C においては、ベクトル[100]は利用可能ですが[110]が利用できません。ローノイズ位相シフト PWM 方式を用いて、V 相および W 相 PWM を非対称にシフトさせることでベクトル[110]が追加され、W 相電流を計測するのに十分なウインドウ幅を確保します。追加ベクトル[110]を導入したことによる影響は、ベクトル[101]の延長と、[100]の短縮により相殺されます。

例 D においては、ベクトル[100]は利用可能ですが[110]が利用できません。ローノイズ位相シフト PWM 方式を用いて、V 相および W 相 PWM を非対称にシフトさせることでベクトル[110]が追加され、W 相電流を計測するのに十分なウインドウ幅を確保します。追加ベクトル[110]を導入したことによる影響は、ベクトル[001]を追加することにより相殺されます。

例 E においては、ベクトル[100]は利用可能ですが[110]が利用できません。ローノイズ位相シフト PWM 方式を用いて，V 相および W 相 PWM を非対称にシフトさせることでベクトル[110]が追加され，W 相電流を計測するのに十分なウインドウ幅を確保します。追加ベクトル[110]を導入したことによる影響は，ベクトル[001]を追加することにより相殺されます。

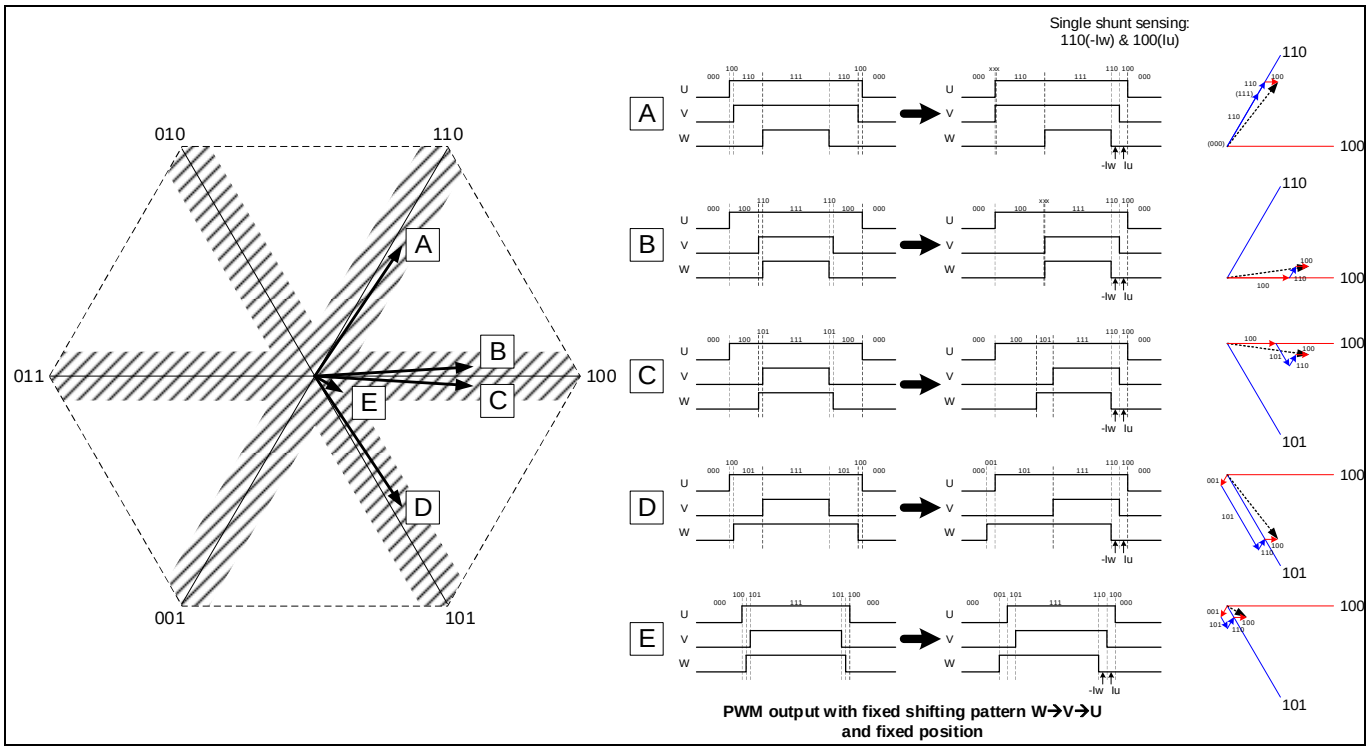


図 17　ローノイズ位相シフト PWM 方式

ローノイズ位相シフト PWM 方式を用いた 1 シャント構成での電流計測タイミングを図 18 に示します。ローノイズ位相シフト PWM 方式では、アクティブベクトル継続時間 T_{a2} あるいは T_{b2} が、要求される最小アクティブベクトル継続時間 T_{PSmin} より大きいか小さいかに関わらず、電流計測条件を満たすために、常にアクティブベクトル[110]と[100]の継続時間が T_{PSmin} ($T_{a2}' = T_{PSmin}$、$T_{b2}' = T_{PSmin}$) となるように、PWM 波形がシフトされます。結果として、最初の電流計測点(CS1)はアクティブベクトル[110]の終点 T_{b2}' からT_{SD} 遅れた時点となり、2 番目の電流計測点(CS2)はアクティブベクトル[100]の終点 T_{a2}'からT_{SD}遅れた時点となります。

電流計測点 CS1 および CS2 が PWM サイクルの終点以降となる場合は、実際の計測点 CS1 および CS2 は、次のベクトル制御演算が実行される PWM サイクルにて電流値を利用できるよう、該当となる PWM サイクルの終点直前に移動されます。

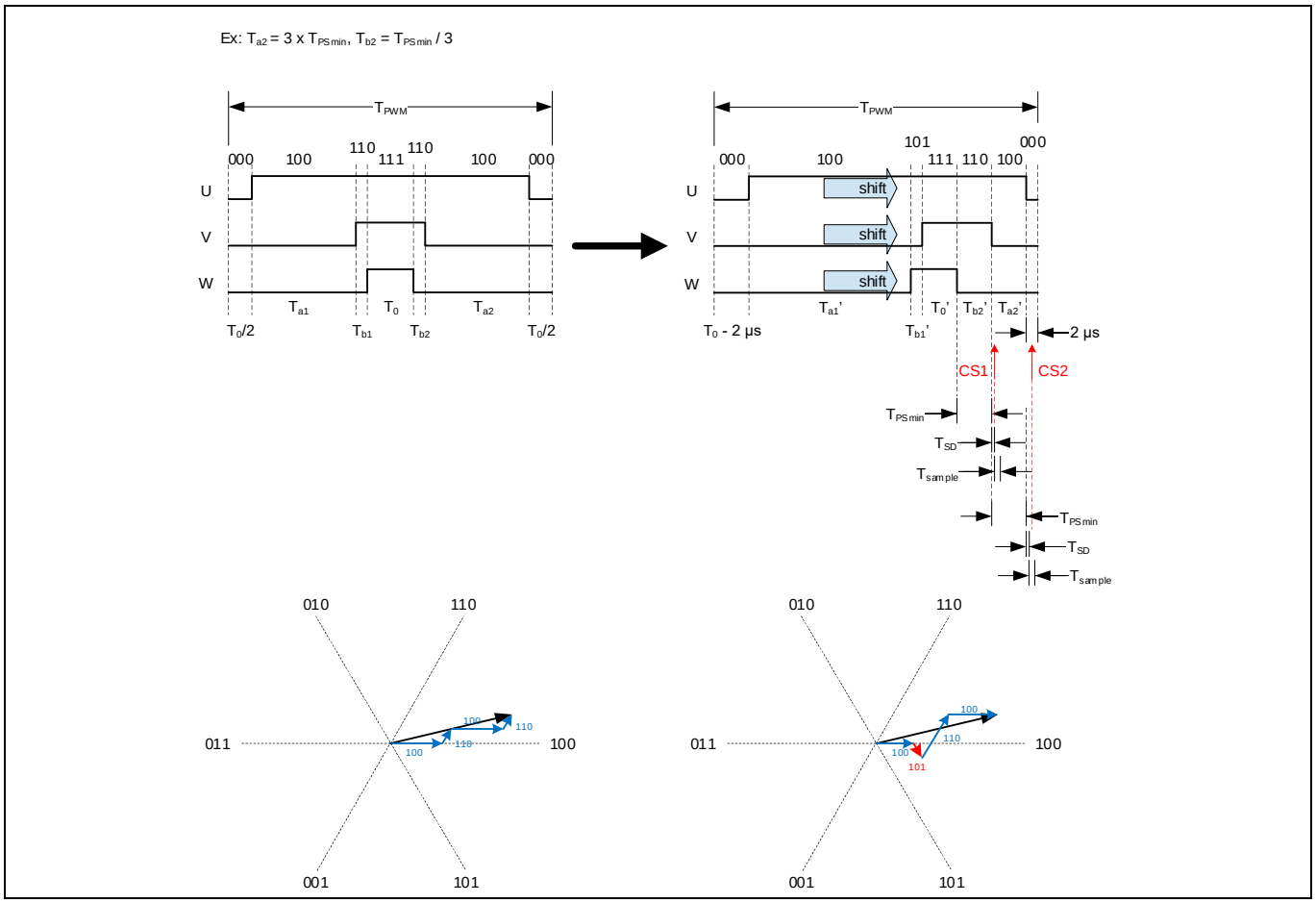


図 18　1 シャント構成/ローノイズ位相シフト PWM 方式での電流計測タイミング

シフトパターンが固定されているため、ローノイズ位相シフト PWM は 3 相 PWM でのみ利用可能であり、また適用可能な PWM のデューティ比には上限があります。ローノイズ位相シフト PWM が有効化された場合、デューティ比が 50%を超えると通常の位相シフト PWM に自動的に移行し、デューティ比が 35%を下回ると自動的にローノイズ位相シフト PWM に復帰します。

ローノイズ位相シフト方式を利用することで、各 PWM サイクル中の出力電圧値が目標値と完全に一致し、低速での音響ノイズと起動性が通常の位相シフト PWM 方式に対して大幅に改善します。このモードで最適な制御性能を実現するために、2 つのパラメータ'TMinPhaseShift'(=T_{PSmin})と'SHDelay'(=T_{SD})を適切な値に調整する必要があります。

2.1.7.2.4　位相シフトウインドウ幅が無い場合のピーク電流トラッキング

AC ファン制御では、特に低速運転時の音響ノイズに対する要求が厳しい場合があります。この場合、最小パルス幅制限 PWM を用いないデューティ制御を行うことで、電圧のサイン波からの乱れを抑制し、また音響ノイズを抑えることが可能となります。1 シャント構成においては、最小のサンプリングウインドウを確保できないために、インバータの電流サンプリングは最小パルス幅よりも大きいアクティブベクトルをもつ PWM サイクルに制限されます。この非連続な電流サンプリングでは巻線電流の再構築が行えず、オープンループのデューティ/電圧制御しか行えません。これは低速での駆動特性に大きな影響を与えるわけではありませんが、過負荷状態での電流制限は必要です。利用可能な電流サンプリング情報に基づき過負荷に対する保護を行うことは可能ですが、サンプリングレートは PWM 周期よりも長くなります。MCE では、サンプリングウインドウが完全に閉じている場合でもピーク電流制限を実現するために、ピーク電流トラッキング機能を備えています。

1 シャント構成で TMinPhaseShift=0 の場合、MCE は自動的にピーク電流トラッキングモードに切り替わります。このモードでは、各 PWM サイクルで 2 つの連続した電流サンプリングが実行され、その大きい方が'Ipeak'変数に代入されます。各セクタの切り替え直後には、'Ipeak'変数はゼロにリセットされ、新しいセクタでのピーク電流トラッキングに備えます。'Ipeak'変数は PWM サイクル毎に直接'Iq'に代入され、q 軸コントローラが電流制限を行います。その一方、'Id'変数はピーク電流トラッキングモードでは常にゼロに設定されます。

TMinPhaseShift=0 の場合のモータ相電流フィードバック信号の流れを図 19 に示します。'Ipeak'のスケーリング係数は、'Ipeak'の値が'Iq'と等しくなるように設定されています。このピーク電流トラッキングモードを用いることで、1 シャント構成にて TMinPhaseShift=0 の場合に、Iq 電流制御系をピーク電流の検知および制限に使用することが出来ます。

TMinPhaseShift がゼロでない場合、MCE は通常の 1 シャント構成の相電流再構築モードに戻ります。

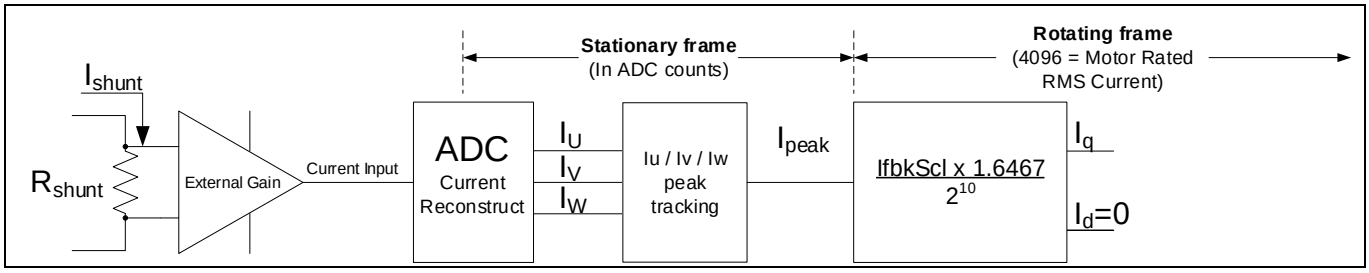


図 19　モータ電流フィードバック信号パス (TminPhaseShift = 0、single shunt)

ピーク電流トラッキングモードでは、モータ電流の計測タイミングを必要に応じて調整します。

図 20 に、通常の位相シフト PWM 方式に対するピーク電流トラッキングモードでの電流計測タイミングを示します。2 つのアクティブベクトル継続時間(T_a と T_b)が 1 us より長い場合(Case 1)、最初の電流計測点(CS1)はアクティブベクトル継続時間 T_b の中心から$T_{DT} + T_{SD}$ 後に発生します。ここで T_{DT} はインバータのデッドタイム、T_{SD} は ADC サンプリング時間('SHDelay'変数により規定)です。2 番目の電流計測点(CS2)はアクティブベクトル継続時間 T_a の中心から$T_{DT} + T_{SD}$ 後に発生します。

アクティブベクトル継続時間 T_a が 1 us より短い場合(Case 2、Case 4)、無効な電流計測値を得るのを避けるため、CS2 は現行 PWM サイクルの終点から 0.5 us 前の時点に再設定されます。

アクティブベクトル継続時間 Tb が 1 us より短い場合(Case 3、Case 4)、無効な電流計測値を得るのを避けるため、CS1 は現行 PWM サイクルの始点から 0.5 us 後の時点に再設定されます。

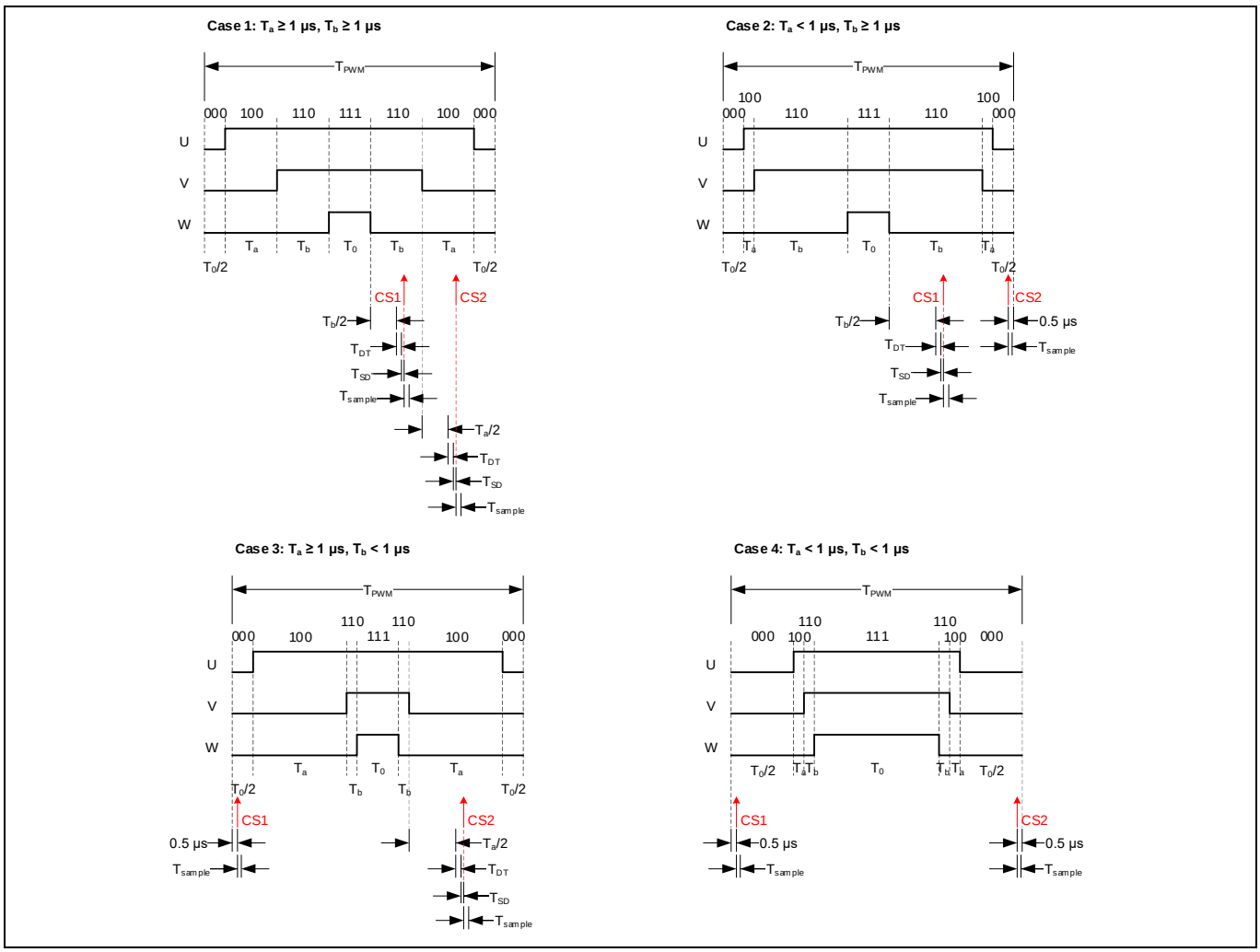


図 20　1 シャント構成/位相シフト PWM 方式/ピーク電流トラッキングモード時の電流計測タイミングチャート

図 21 に、ローノイズ位相シフト PWM 方式対するピーク電流トラッキングモードでの電流計測タイミングを示します。アクティブベクトル継続時間の合計が 2×T_a と 2×T_b として、T_a と T_b がともに 1 us より長い場合(Case 1)、最初の電流計測点 CS1 はアクティブベクトル継続時間 2×T_a の始点から$T_a + T_{DT} + T_{SD} + 1\mu s$後に発生します。2 番目の電流計測点(CS2)はアクティブベクトル継続時間 2×T_b の始点から$T_b + T_{DT} + T_{SD} + 1\mu s$後に発生します。

T_a が 1 µs より短い場合(Case 2、Case 4)、無効な電流計測値を得るのを避けるため、CS1 はアクティブベクトル継続時間 2×T_a の始点から 1 µs 前の時点に再設定されます。その CS1 発生点が PWM サイクルの開始より前に来る場合は、無効な電流計測値を得るのを避けるため、CS1 は PWM サイクル開始の直後に発生するように調整されます。

T_b が 1 µs より短い場合(Case 3、Case 4)、無効な電流計測値を得るのを避けるため、CS2 はゼロベクトル[111]の始点から 4 us 後に再設定されます。

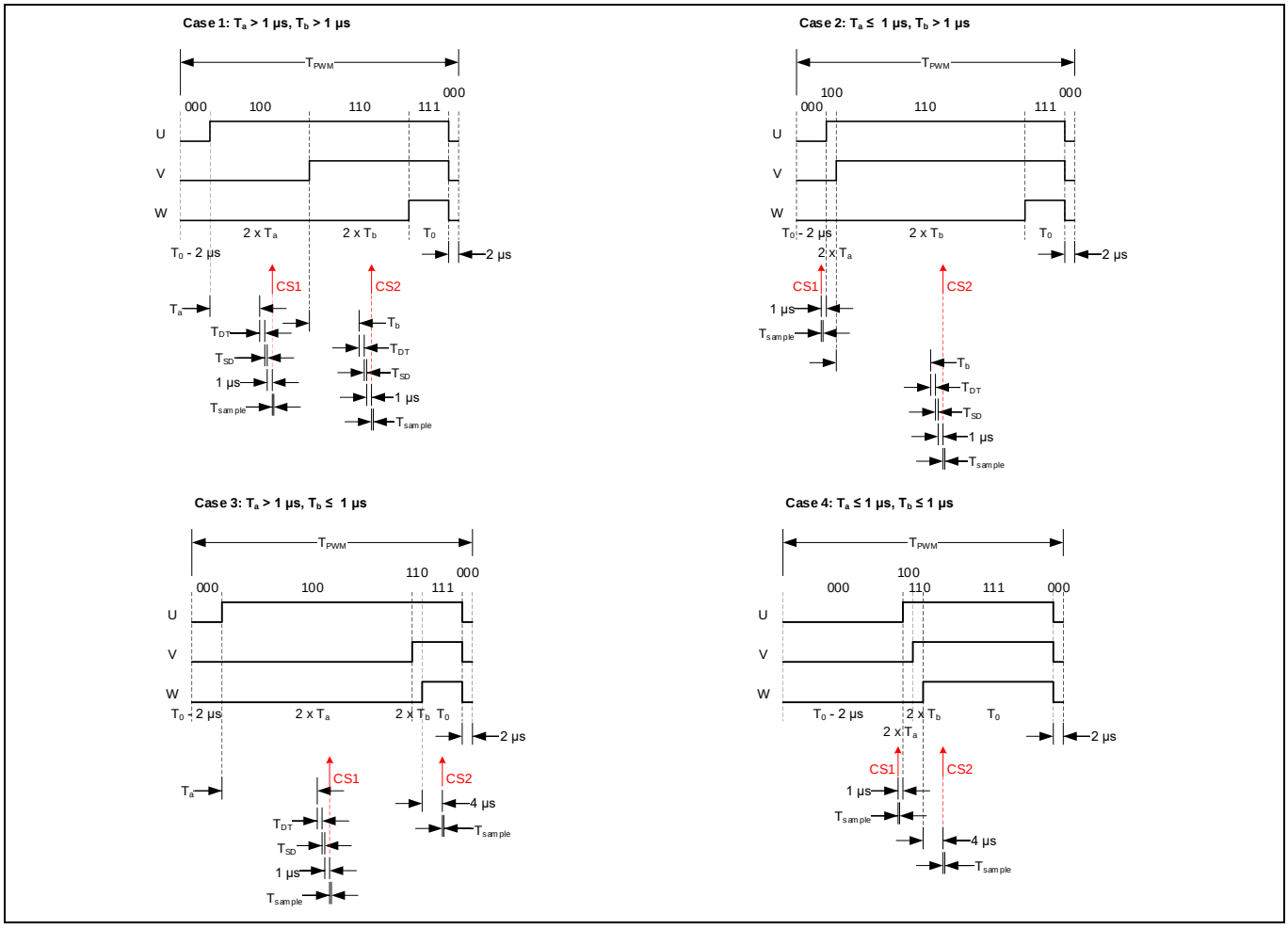


図 21　1 シャント構成/ローノイズ位相シフト PWM 方式/ピーク電流トラッキングモード時の電流計測タイミングチャート

2.1.8　モータ電流制限プロファイル

ファンのようなアプリケーションでは低速で大電流を必要としません。言い換えれば、大トルクはある特定速度以上で必要となります。MCE はスムーズな起動のために、低速領域では電流制限値を低くする、動的な電流制限調整機能を有しています。この機能は、スムーズで静かな起動を可能とし、ロータロック電流を低減することが出来ます。

図 22 に、モータ電流制限がモータ速度の関数として動的に変化する様子を示します。MCE はモータ負荷がモータモード(図 22 の第 1 および第 3 象限)と回生モード(図 22 の第 2 および第 4 象限)の両方で動作することを可能にします。

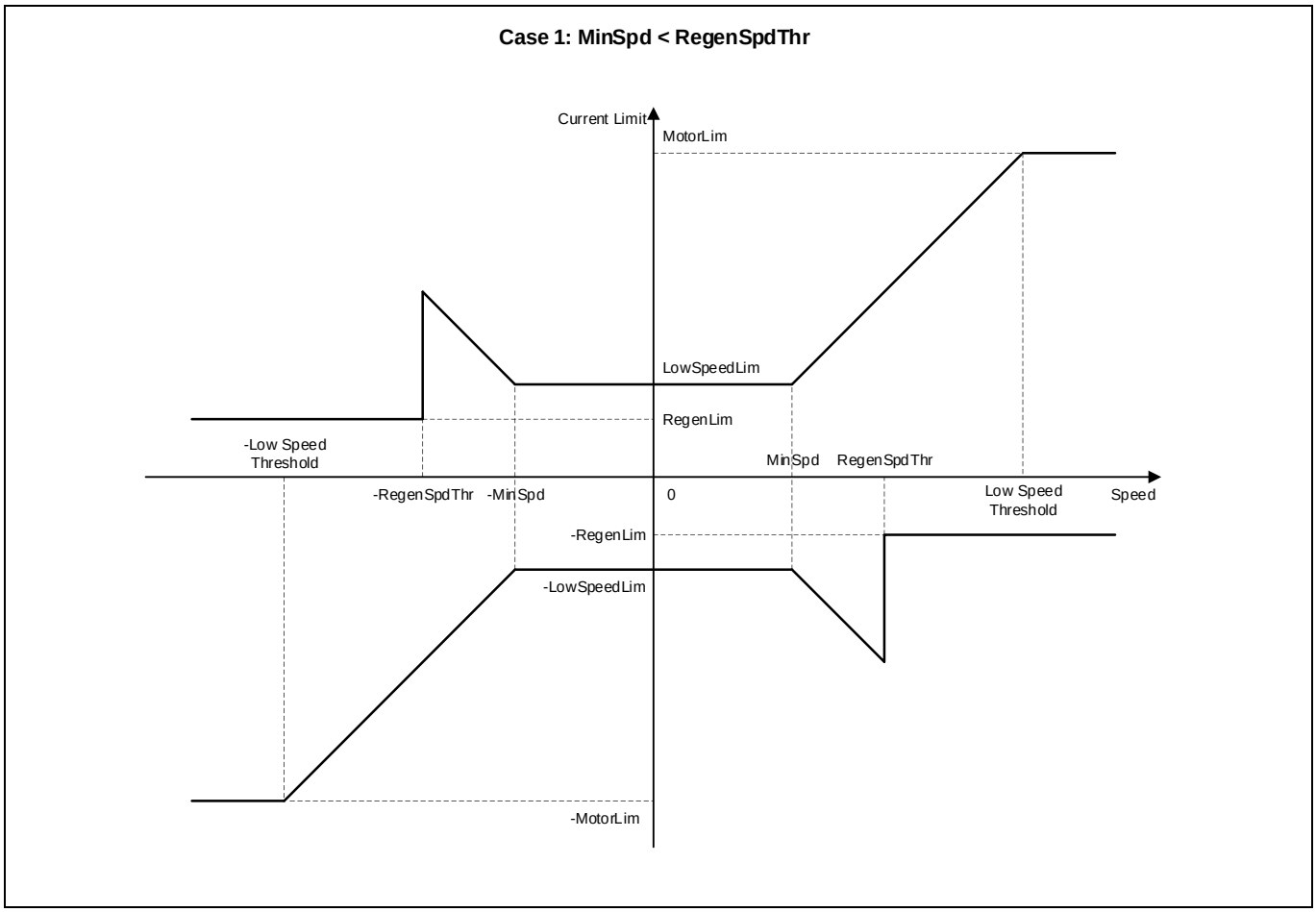
モータモードにおいては、モータ速度の絶対値が'MinSpd'パラメータで規定される最低速度以下の場合($|MotorSpeed| \leq MinSpd$)、モータの最大電流は'LowSpeedLim'パラメータで規定される閾値に制限されます。モータ速度の絶対値が最低速度と低速閾値の間にある場合、下記の式に従って、モータ電流制限値はモータ速度増加と共に線形に増加します。

$$Motor\ Current\ Limit = LowSpeedLim + (|MotorSpeed| - MinSpd) \times LowSpeedGain$$

モータ速度が低速閾値を上回った場合、モータの最大電流は'MotorLim'パラメータで規定される上限値に制限されます。

回生モードでは、モータ速度の絶対値が'RegenSpdThr'パラメータで規定される閾値より小さい場合、モータ電流制限値は上記の式に示される線形関係に従います。モータ速度の絶対値が'RegenSpdThr'パラメータで規定される閾値を上回った場合、モータの最大電流は'RegenLim'パラメータで規定される閾値に制限されます。

モータモードと回生モードで異なるモータ電流制限値の設定が可能なため、最適な動作特性を得るために加速トルクと回生ブレーキトルクを個別に調整することが出来ます。更なるモータ電流制限のカスタマイズが必要な場合は、スクリプトを使用することで、モータ電流制限('MotorLim'パラメータ)を自由なプロファイルに変更することが出来ます。



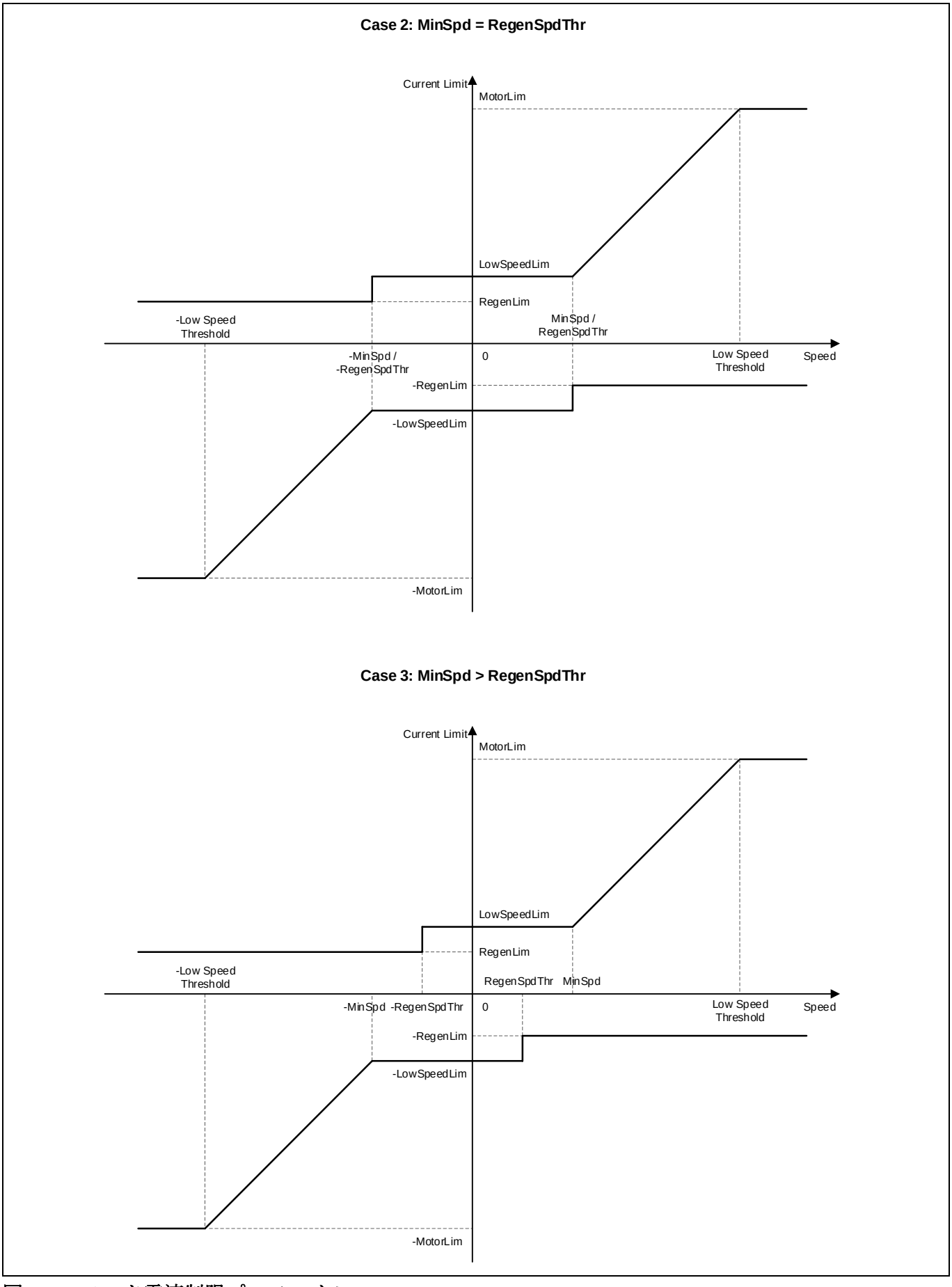


図 22　モータ電流制限プロファイル

2.1.9 初期角度センシング

ファンのアプリケーションによっては、モータ始動時に逆方向への回転動作をすることなく正しい方向へ回転することが求められます。通常の直流励磁＋オープンループの方式では、時々逆方向の回転動作が発生します。ダイレクトスタート方式を用いた場合でも、モータ低速域での逆起電力が不十分なため、始動に失敗することがあります。

MCE は、始動前に数 ms 幅の、異なる角度の 6 つの電流パルスをモータに流すことで、ロータ位置を推定する初期角度センシング機能を有しています(特許取得済み)。初期角度はこれらのパルスの電流強度より計算されます。ANGLE_SENSING ステートが完了すると、モータのステートマシンは MOTOR_RUN ステートに遷移し、直接クローズドループのベクトル制御を実行します。

初期角度センシング機能を使用することで、常にモータが正しい方向に起動し、センサレスベクトル制御を用いた際の直流励磁時に時々発生する逆方向の回転動作を防止することが出来ます。初期角度推定はロータ磁石の突極性に依存し、モータの L_d と L_q の比が 95%より小さく、L_d と L_q の平均値が 0.1 mH より大きいときに有効に働きます。

関連する制御パラメータ(IS_Pulses、IS_Duty、IS_IqInit)は、入力された L_d と L_q の値を元に、MCEWizard により自動計算されます。

この手法は初期角度推定のみに用いられ、慣性の大きい負荷を駆動する場合には磁束推定器の調整が必要となります。モータ速度が始動時にゼロで無い場合、推測された角度は正しくないことがあります。その場合は、キャッチスピン機能の使用を推奨します。

2.1.10 過変調

図 23 に示すように、線形変調の範囲はアクティブ電圧ベクトル(a、b)の作る六角形に内接する円の内部で定義されます。変調が線形の範囲内にある場合、デューティ比は$\frac{\sqrt{3}}{2} = 0.866$が上限となります。出力最大化が優先され非線形の変調が許容される場合は、デューティ比を 1 まで上げることが出来、アクティブ電圧ベクトルは DC バス電圧をフルに活用するために内接円の外側の領域に達します。

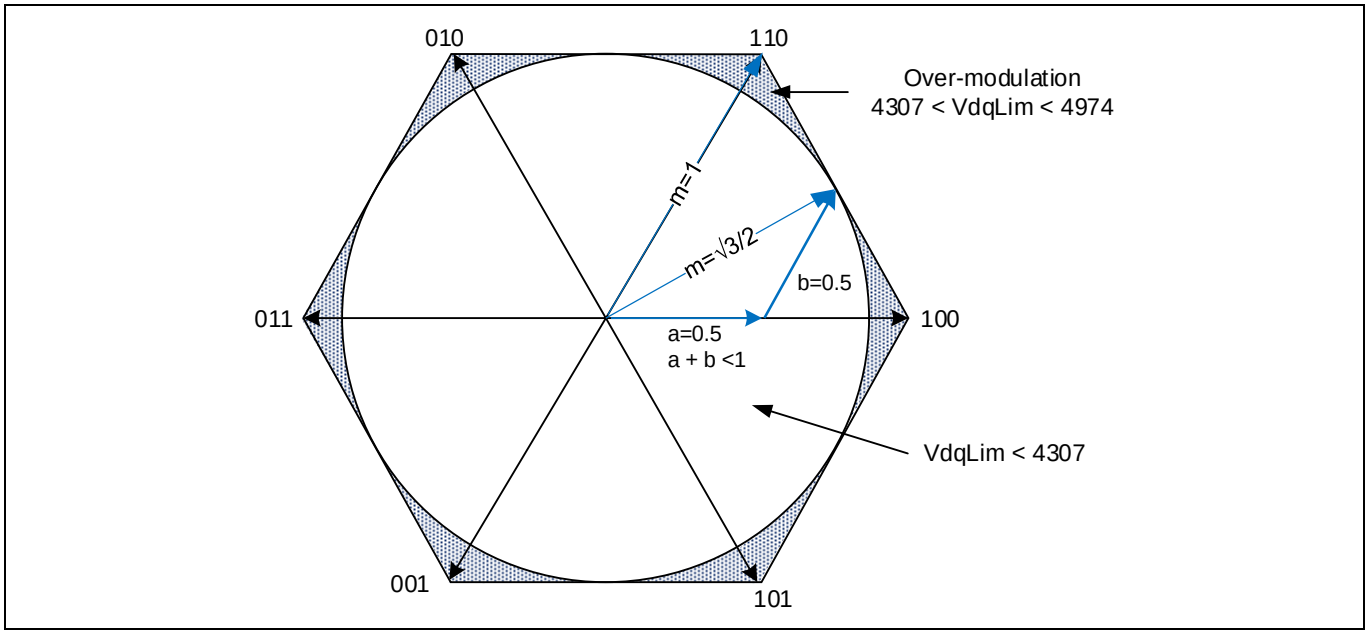


図 23 SVPWM ベクトルタイミング制限

MCE はデューティ比の制約を調整するために、'VdqLim'パラメータを用意しています。100%デューティは'VdqLim'パラメータが 4974 となることに相当します。変調を線形領域に制限する必要がある場合は、'VdqLim'パラメータを 4307(=4974 x 0.866)に設定する必要があります。過変調による出力向上が必要となる場合は、'VdqLim'パラメータを 4974 に設定してください。

過変調は DC バスの利用効率を最大化する一方、高調波成分による音響ノイズを引き起こし、また磁束 PLL 動作を損なうために、電力あるいはトルク計算に基づく RMS 電流および電圧の誤差発生を招きます。

2.1.11 2 相変調

MCE は、切り替え速度閾値を調整可能な 2 相タイプ 3(ローサイドクランプ)の空間ベクトル PWM 機能を有しています。図 24 に示すように、2 相タイプ 3PWM は巻線の 1 相をインバータのマイナス側のレールにクランプします。そのため、各セクタにて 3 つのインバータレグのうちの一つはスイッチングせず、出力電圧を 3 相 PWM の場合と比較して等しく保ちつつ、スイッチング損失を削減することが出来ます。これはゼロベクトル[111]を利用せず、すべてのゼロベクトルの時間を[000]にアサインすることで実現します。

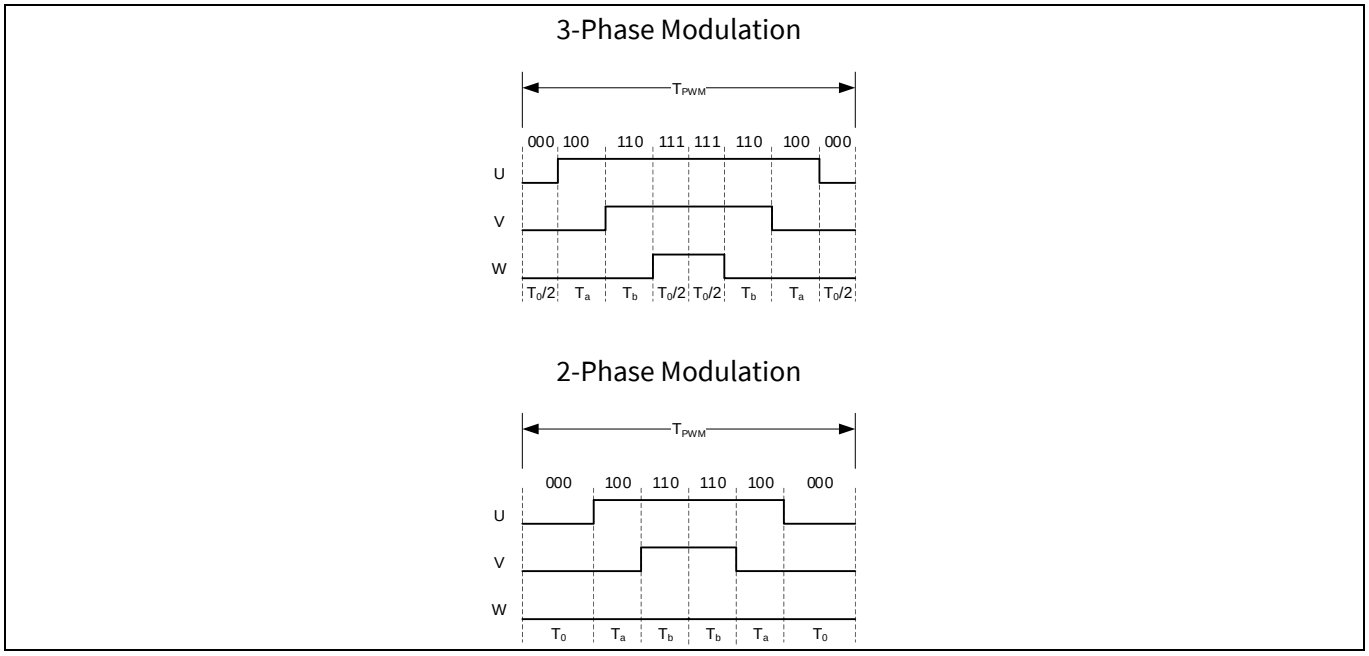


図 24 3 相/2 相タイプ 3 空間ベクトル PWM ダイアグラム

ハイサイドのゲートドライバがブートストラップコンデンサを充電するためにブートストラップダイオードを使用している場合、2 相タイプ 3PWM は低速では利用できません。ブートストラップコンデンサは、空間ベクトル PWM セクタの全期間でハイサイドのゲートを駆動できるよう、十分な容量を持たなければなりません。

'HwConfig“変数の Bit[4:3]が、2 相タイプ 3PWM の有効化に使われます。図 25、に示すように、2 相タイプ 3PW が有効化されると、起動時には 3 相 PWM が使われます。モータ速度の絶対値('Abs_MotorSpeed'変数)が'Pwm2PhThr'変数で規定される閾値を超えると、MCE は 2 相タイプ 3 に切り替えます。モータ速度の絶対値が'Pwm2PhThr'変数で規定される閾値から 256 カウント(モータ最大速度の 1.6%)のヒステリシス分だけ小さい値を下回ると、MCE は 3 相 PWM に戻ります。

'Pwm2PhThr'変数の値が 256 以下(モータ最大速度の 1.6%以下)で、2 相 PWM が有効化されている場合、MCE が 2 相 PWM に移行した後は自動的に 3 相 PWM には戻らず、モータが停止して再起動してから 3 相 PWM に戻ります。

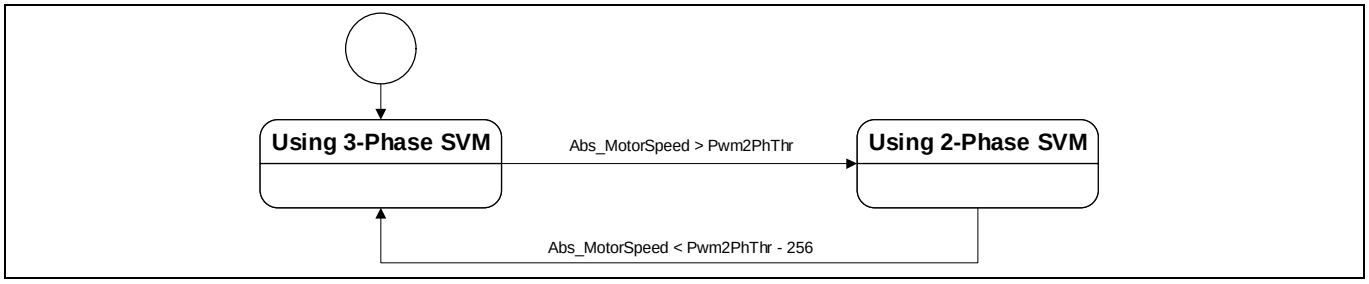


図 25　3 相/2 相 SVM ステート遷移図

2.1.12　キャッチスピン

例えばファンのアプリケーションでは、風等による外力により、インバータが起動する前にモータが回転している場合があります。MCE は「キャッチスピン」機能を有しており、モータ駆動トルクを供給する前にモータの実際の速度に合わせて磁束推定器と磁束 PLL を同期するようにします。キャッチスピンはモータの逆起電力が DC バス電圧より高いときには利用できません。これはモータが定格速度よりも速く回転している場合に発生します。従って、キャッチスピンは一般的にはモータの定格速度以下の範囲で有効です。キャッチスピンの始動プロセスはモータのステートマシンの一部であり、キャッチスピンが有効化されていればモータ始動時に実行されます。

キャッチスピンにおいては、モータが回転しているか、もし回転している場合はどちらの方向に回転しているかを検出するために、コントローラは逆起電力を監視しています。キャッチスピン動作はブートストラップ容量の充電が完了してから開始します。キャッチスピン実行中は、IqRef と IdRef の両方が 0 に設定され(速度コントローラは無効化)、その間磁束 PLL は実際のモータ速度('MotorSpeed'変数)とロータ角度('RotorAngle'変数)にロックするよう試みます。'TCatchSpin'パラメータで定義されるキャッチスピン時間が経過したら、モータ速度が'DirectStartThr'パラメータの値と比較されます。モータ速度が'DirectStartThr'以上であれば、通常の速度制御が開始し、その時点のモータ速度が初期参照速度となり速度上昇の開始点となります。設定された目標速度によって、モータは減速(回生ブレーキによる)か加速し、目標速度に達します。モータ速度が'DirectStartThr'以下であれば、モータのステートは'ANGLESENSING'ステートに変化します。

回転方向により、下記の 3 種類のキャッチスピン方法が用いられます。

- ゼロスピードキャッチスピン
- 正回転キャッチスピン
- 逆回転キャッチスピン

2.1.12.1　ゼロスピードキャッチスピン

モータが停止している場合、キャッチスピン動作は「ゼロスピードキャッチスピン」となります。図 26(A)に、ゼロスピードキャッチスピンの一例を示します。この例では、スタートコマンド実行時モータは停止しています。スタートコマンド実行後、「ゼロスピードキャッチスピン」動作が開始します。キャッチスピン動作中、モータ電流は出力されません。キャッチスピン時間経過後、その状態のモータ速度(この場合は 0 rpm)初期の参照速度および速度加速の始点となります。モータは基準となる加減速レートに従って加速を続け、設定された目標速度に達します。

キャッチスピンが無効化されている場合、PLL ロックを待たずに、スタートコマンド実行後ただちに通常の速度制御が開始します。図 26 (B)に示すように、スタートコマンド実行後、キャッチスピン動作を行うことなく直ちにモータ電流が出力されます。モータは基準となる加減速レートに従って加速を開始し、設定された目標速度に達します。

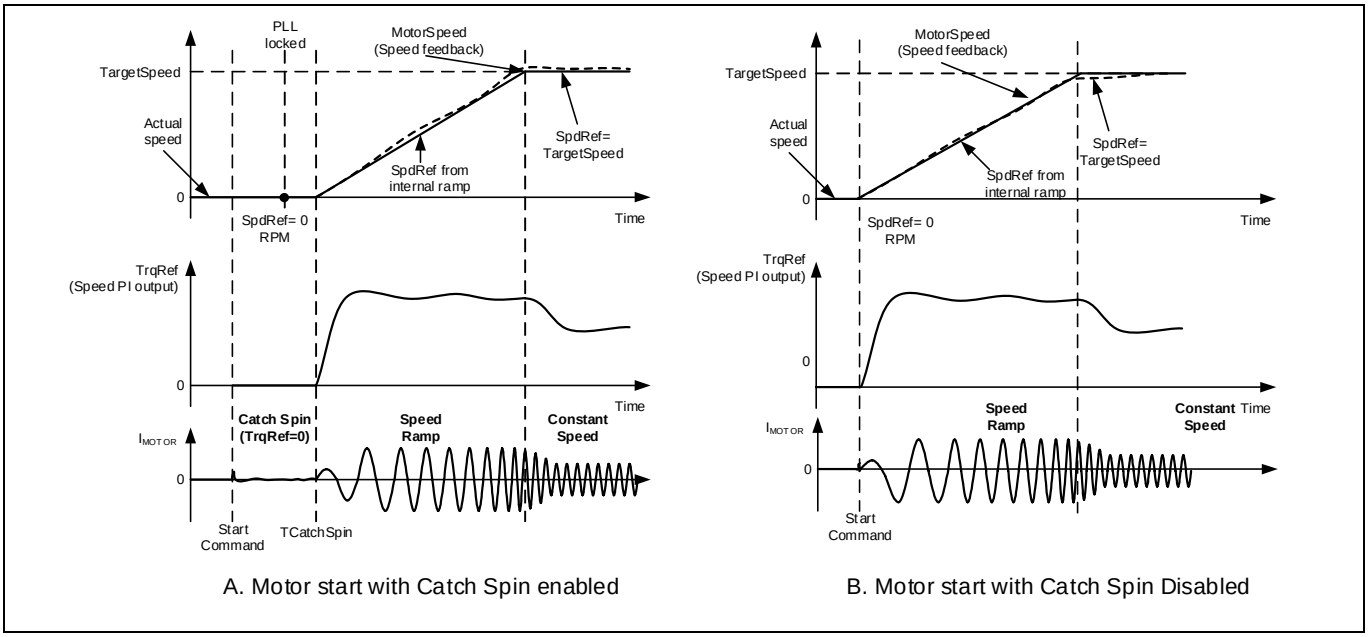


図 26　ゼロスピードキャッチスピン - キャッチスピン有効/無効の場合のモータ始動シークエンス

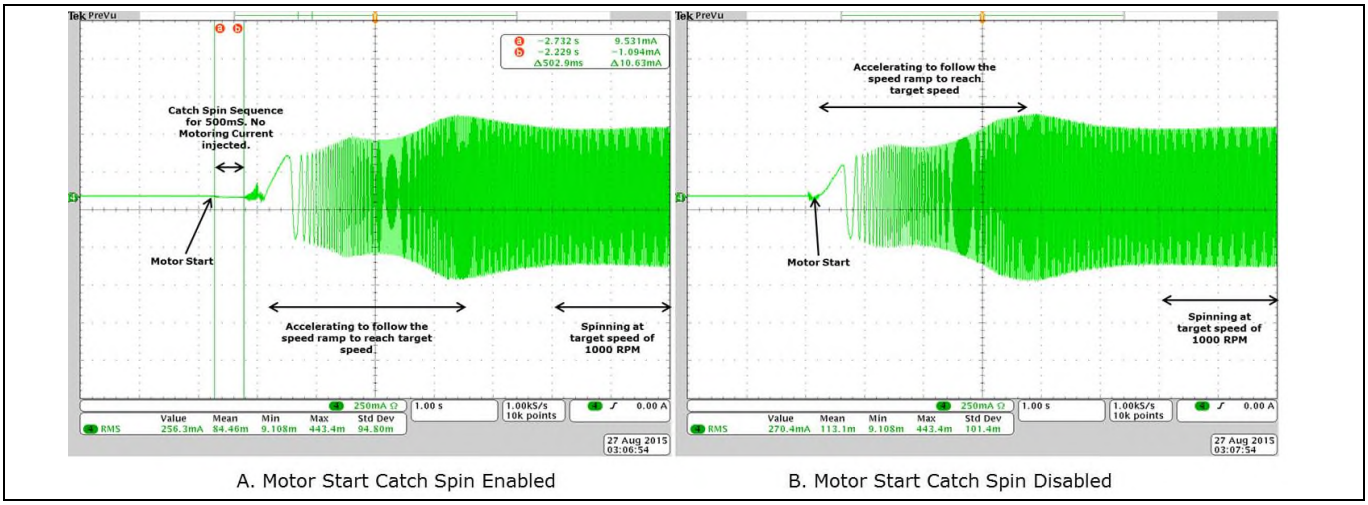


図 27　モータ相電流(ゼロスピードキャッチスピン) - キャッチスピン有効/無効の場合

2.1.12.2　正回転キャッチスピン

モータが正方向に回転している場合、キャッチスピン動作は「正回転キャッチスピン」となります。図 28 (A)に「正回転キャッチスピン」の一例を示します。この例では、スタートコマンド実行時にモータは正方向に回転しています。キャッチスピン動作中、モータへ電流は印加されません。キャッチスピン時間経過後、磁束 PLL が実際のモータ速度にロックしたと想定して、その時点のモータ速度が初期の参照速度および速度加速の始点となります。モータは基準となる加減速レートに従って加速を続け、設定された目標速度に達します。

キャッチスピンが無効化されている場合、PLL ロックを待たずに、スタートコマンド実行後ただちに通常の速度制御が開始します。通常は制御システムは回転中のモータを起動することが出来ますが、モー

タ速度はスムーズに加減速しない場合があります。図 28 (B)に示すように、スタートコマンド実行後、実際のモータ速度は参照速度('SpdRef'変数)より大きくなっています。従って、モータは速度参照('SpdRef'変数)に従うように減速します(回生ブレーキを使用)。モータ速度が回生速度閾値('RegenSpdThr'変数)よりも大きい場合、モータが減速するために注入される負のトルクは、'RegenLim'パラメータの値に制限されます。モータが速度参照と一致したら、モータは基準となる加減速レートに従って加速を開始し、設定された目標速度に達します。

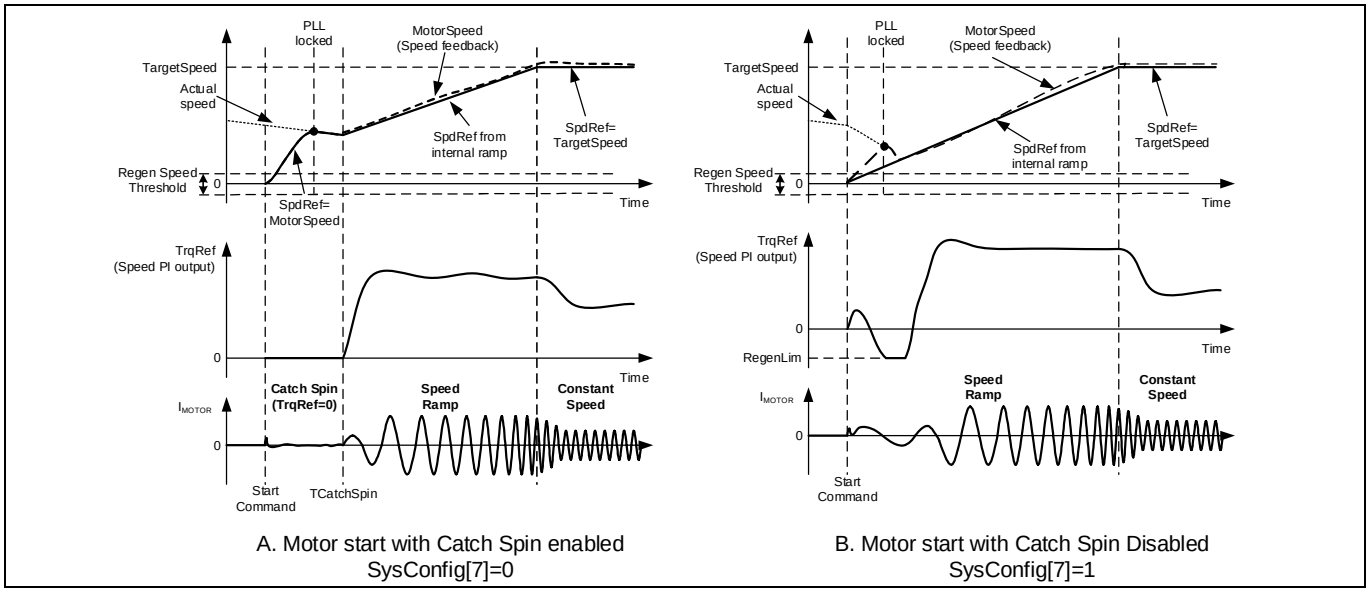


図 28　正回転キャッチスピン - キャッチスピン有効/無効の場合のモータ始動シークエンス

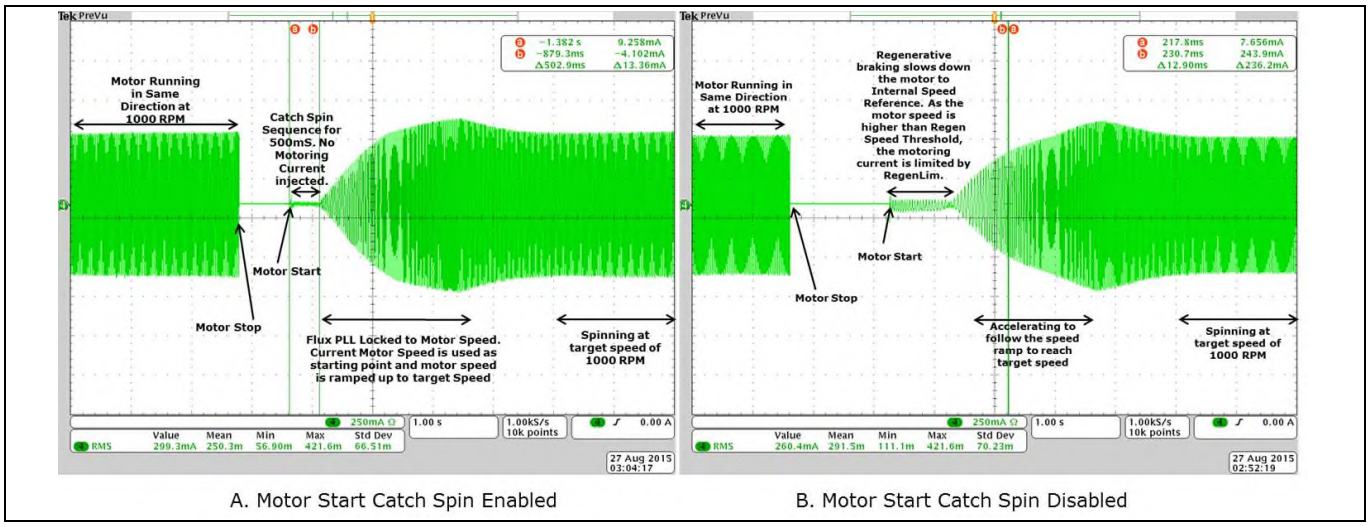


図 29　モータ相電流(正回転キャッチスピン) - キャッチスピン有効/無効の場合

2.1.12.3　逆回転キャッチスピン

モータが逆方向に回転している場合、キャッチスピン動作は「逆回転キャッチスピン」となります。図 30 (A)に「逆回転キャッチスピン」の一例を示します。この例では、スタートコマンド実行時にモータは逆方向に回転しています。キャッチスピン動作中、モータへ電流は印加されません。キャッチスピン時間(TCatchSpin)経過後、モータが逆方向に回生速度閾値(RegenSpdThr)よりも速く回転している場合、RegenLim パラメータで定義された値に制限されるトルクが出力され、モータを回生ブレーキにより減速させます。逆回転速度が回生速度閾値(RegenSpdThr)を下回ると、出力トルクは MotorLim(RegenLim<=MotorLim)に制限されます。出力トルクはモータを停止させ、モータは基準となる加減速レートに従って正方向への加速を開始し、設定された目標速度に達します。

キャッチスピンが無効化されている場合、PLL ロックを待たずに、スタートコマンド実行後ただちに通常の速度制御が開始します。通常は制御システムは回転中のモータを起動することが出来ますが、モータ速度はスムーズに加減速しない場合があります。図 30 (B)に示すように、スタートコマンド実行後、モータは回生速度閾値(RegenSpdThr)よりも速く回転しているため、RegenLim パラメータで定義される値で制限されるトルクが逆回転しているモータを回生ブレーキで減速させます。逆回転速度が回生速度閾値(RegenSpdThr)を下回ると、出力トルクは MotorLim(RegenLim<=MotorLim)に制限されます。出力トルクはモータを停止させ、モータは基準となる加減速レートに従って正方向への加速を開始し、設定された目標速度に達します。

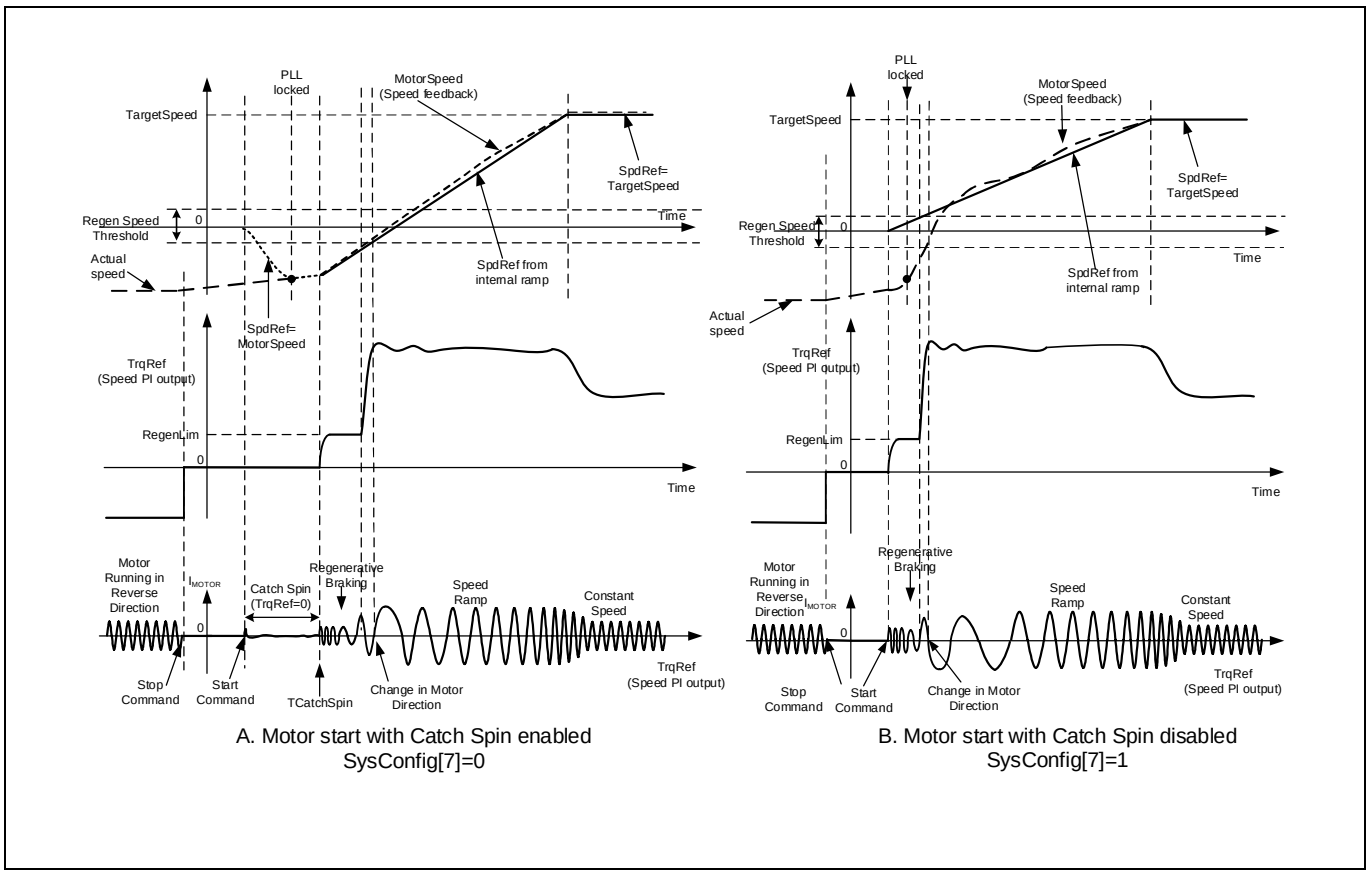


図 30　正回転キャッチスピン - キャッチスピン有効/無効の場合のモータ始動シークエンス

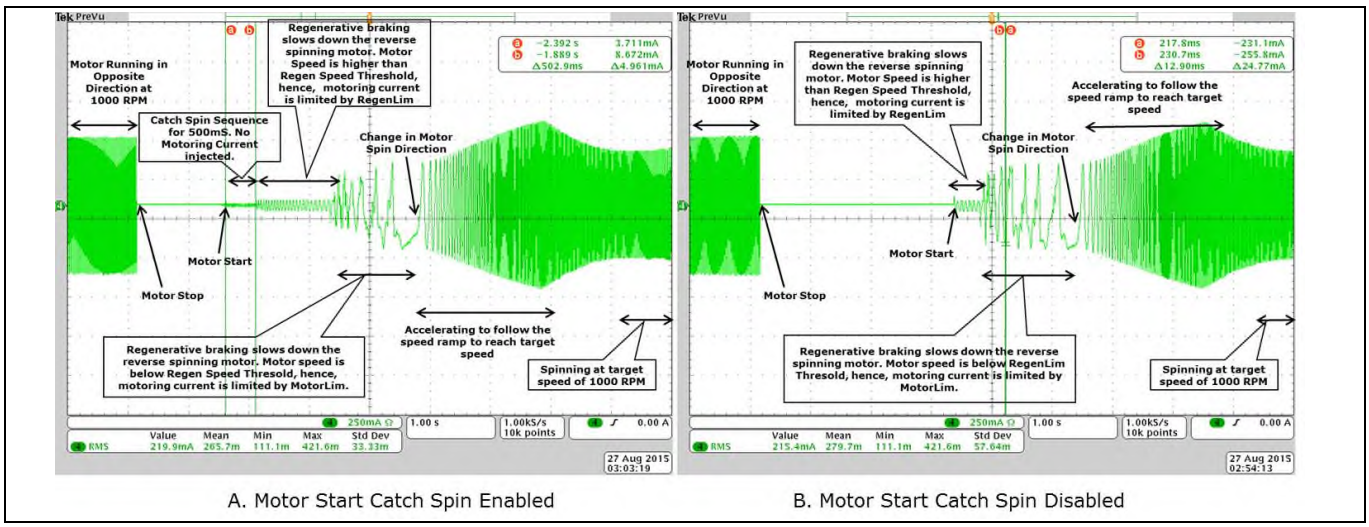


図 31　モータ相電流(逆回転キャッチスピン) - キャッチスピン有効/無効の場合

2.1.13 制御入力

MCE は 4 種類の入力からモータを制御することができます。制御入力の種類は MCEWizard を使って設定することができます。

- UART 制御
- Vsp アナログ入力
- 周波数入力
- デューティサイクル入力

2.1.13.1 UART 制御

UART 制御モードでは、モータの始動、停止、および速度変更が UART コマンドによって制御されます。目標速度は正または負とすることが可能で、目標速度が負の場合はモータが逆方向に回転します。何らかのフォールト条件が発生した場合、モータは停止して、フォールト状態に留まります。いつフォールトをクリアしてモータを再始動するかはマスタコントローラが決定します。

2.1.13.2 Vsp アナログ入力

Vsp アナログ入力制御モードでは、モータの始動、停止、速度変更といったモータの運転はアナログ電圧信号の印加によって制御されます。モータの回転方向は個別のピンによって制御されています。方向ピンが LOW の場合、目標速度は正の値として設定され、方向ピンが HIGH の場合、目標速度は負の値として設定されます。目標速度が負の値の場合、モータは逆方向に回転します。MCE は“VSP“ピンを Vsp アナログ入力として使用し、“DIR“ピンをモータの回転方向入力として使用します。Vsp 電圧とモータの目標速度の関係を図 32 に示します。

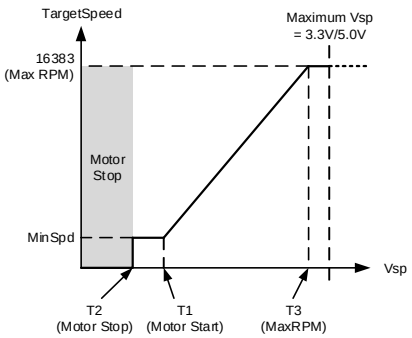


図 32 Vsp アナログ入力

入力電圧と目標速度の関係を定義するのに使用する 3 つの入力しきい値があります。

- T1（モータ始動の入力しきい値）：Vsp アナログ電圧がこのしきい値を超えると、モータは始動します。
- T2（モータ停止の入力しきい値）：Vsp アナログ電圧がこのしきい値未満になると、モータは停止します。
- T3（最大 RPM の入力しきい値）：Vsp アナログ電圧がこのしきい値以上になると、“TargetSpeed“変数が最大速度である 16383 になります。

MCEWizard はこれら 3 つの入力しきい値を使って 3 つのパラメータの値(“CmdStart“、“CmdStop“、および“CmdGain“)を計算します。

$$CmdStop = Integer\left\{\left(\frac{T2 * 2}{Vadcref} * 2048\right) + 0.5\right\}$$

ここで、T2 = アナログ Vsp モータ停止電圧（単位は V）。

$$CmdStart = Integer\left\{\left(\frac{T1 * 2}{Vadcref} * 2048\right) + 0.5\right\}$$

ここで、T1 = アナログ Vsp モータ始動電圧（単位は V）。

$$CmdGain = Integer\left\{\left(\frac{Speed_{Max} - Speed_{Min}}{Speed_{Max}} * 2^{12}\right) * \left(\frac{2^{14}}{\left(\left(4096 * 32 * \frac{T3}{Vadcref}\right) - (CmdStart * 32)\right)}\right) + 0.5\right\}$$

ここで、T3 = アナログ Vsp モータ最大 RPM 電圧（単位は V）

$Speed_{Max}$ = モータ最大速度（単位は RPM）

$Speed_{Min}$ = モータ最小速度（単位は RPM）

表 5　アナログ入力電圧の仕様

推奨入力範囲	Vsp アナログ入力（0.1V～V_{adcref}）
T1	<50% of V_{adcref}
T2*	<50% of V_{adcref}
T3**	< V_{adcref}

注： *T2< T1、**T3>T2 である必要があります。*

特定のデバイスの入力範囲およびピンの詳細については、IMC データシートを参照してください。この機能は UART 制御モードでは使用できません。

2.1.13.3 周波数入力

周波数入力制御モードでは、モータの始動、停止、速度変更といったモータの運転は、デジタル IO ピンに矩形波信号を送ることによって制御されます。モータの回転方向は別の方向ピンによって制御されています。方向ピンが LOW の場合、目標速度は正の値として設定され、方向ピンが HIGH の場合、目標速度は負の値として設定されます。目標速度が負の値の場合、モータは逆方向に回転します。MCE は“DUTYFREQ“ピンを周波数入力として使用し、“DIR“ピンをモータの回転方向入力として使用します。周波数とモータの目標速度の関係を図 33 に示します。

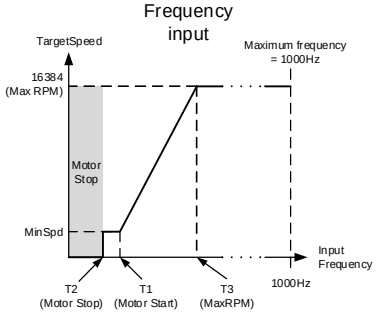


図 33　周波数入力

周波数入力と目標速度の関係を定義するのに使用する 3 つの入力しきい値があります。

- T1（モータ始動の入力しきい値）：周波数入力がこのしきい値を超えると、モータは始動します。
- T2（モータ停止の入力しきい値）：周波数入力がこのしきい値未満になると、モータは停止します。
- T3（最大 RPM の入力しきい値）：周波数入力がこのしきい値以上になると、目標速度が最大速度である 16383 になります。

MCEWizard はこれら 3 つの入力しきい値を使って 3 つのパラメータの値(“CmdStart“、“CmdStop“、および“CmdGain“)を計算します。

$$CmdStop = Integer\{T2 * 10 + 0.5\}$$

ここで、T2 = モータの停止周波数（単位は Hz）。

$$CmdStart = Integer\{T1 * 10 + 0.5\}$$

ここで、T1 = モータの始動周波数（単位は Hz）。

$$CmdGain = Integer\left\{\left(2^{12} * \frac{\left(16384 - \left(\frac{Speed_{Min}}{Speed_{Max}} * 16384\right)\right)}{(T3 - T1) * 32 * 10}\right) + 0.5\right\}$$

ここで、T1 = モータの始動周波数（単位は Hz）、

T3 = モータの最大速度周波数（単位は Hz）、

$Speed_{Max}$ = モータ最大速度（単位は RPM）、

$Speed_{Min}$ = モータ最小速度（単位は RPM）。

表 6　周波数入力の仕様

推奨入力範囲	周波数入力（(5～1000Hz、10～90%デューティサイクル)
T1	≤ 255Hz
T2*	≤ 255Hz
T3**	≤ 1000Hz

注： ** T2< T1、**T3>T2 である必要があります。*

特定のデバイスの入力範囲およびピンの詳細については、IMC データシートを参照してください。この機能は UART 制御モードでは使用できません。

2.1.13.4 デューティサイクル入力制御

デューティサイクル入力制御モードでは、モータの始動、停止、速度変更といったモータの運転は、デジタル IO ピンへの矩形波信号のデューティサイクルを変化させることによって制御されます。モータの方向は別の方向ピンによって制御されています。方向ピンが LOW の場合、目標速度は正の値として設定され、方向ピンが HIGH の場合、目標速度は負の値として設定されます。目標速度が負の値の場合、モータは逆方向に回転します。MCE は“DUTYFREQ“ピンをデューティ入力として使用し、“DIR“ピンをモータの方向入力として使用します。デューティサイクルとモータの目標速度の関係を図 34 に示します。

デューティサイクル制御モードでは、キャプチャタイマのプリスケーラの範囲が周波数制御モードよりも大きくなっています。このため、デューティサイクル制御モードではより高い入力周波数が可能です。推奨入力周波数範囲は 5Hz～20kHz です。入力ピンに対するどのような外部 R/C ローパスフィルタ

もデューティサイクル測定値に影響する可能性がある点に注意してください。入力周波数が 1kHz 以上の場合は特に顕著です。

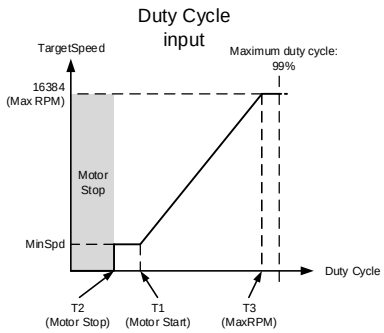


図 34　デューティサイクル入力

デューティサイクル入力と目標速度の関係を定義するのに使用する 3 つの入力しきい値があります。

- T1（モータ始動の入力しきい値）：デューティサイクル入力がこのしきい値を超えると、モータは始動します。
- T2（モータ停止の入力しきい値）：デューティサイクル入力がこのしきい値未満になると、モータは停止します。
- T3（最大 RPM の入力しきい値）：入力がこのしきい値以上になると、“TargetSpeed“変数が最大速度である 16383 になります。

MCEWizard はこれら 3 つの入力しきい値を使って 3 つのパラメータの値(“CmdStart“、“CmdStop“、および“CmdGain“)を計算します。

$$CmdStop = Integer\{T2 * 10 + 0.5\}$$

ここで、T2 = モータの停止デューティサイクル（単位は%）。

$$CmdStart = Integer\{T1 * 10 + 0.5\}$$

ここで、T1 = モータの始動デューティサイクル（単位は%）。

$$CmdGain = Integer\left\{\left(\frac{Speed_{Max} - Speed_{Min}}{Speed_{Max}} * 2^{12}\right) * \left(\frac{2^{14}}{((T3 * 10) - (CmdStart)) * 32}\right) + 0.5\right\}$$

ここで　T1 = モータの始動速度デューティサイクル（単位は%）、

T3 = モータの最大速度デューティサイクル（単位は%）、

SpeedMax = モータ最大速度（単位は RPM）

SpeedMin = モータ最小速度（単位は RPM）

表 7　デューティサイクル入力の仕様

推奨入力範囲	デューティサイクル入力（(5Hz～20kHz、1～99%デューティサイクル）
T1	< 50%
T2*	< 50%
T3**	≤ 99%

注： *T2<T1、**T3>T2 である必要があります。*

特定のデバイスの入力範囲およびピンの詳細については、IMC データシートを参照してください。この機能は UART 制御モードでは使用できません。

2.1.13.5 自動再起動

Vsp、周波数、またはデューティサイクル制御入力モードでは、何らかのフォールトが発生してモータが停止した場合に再起動までのリトライ回数およびリトライ間隔をユーザが指定できるオプションがあります。リトライ回数およびリトライ間隔は‘FaultRetryPeriod’パラメータを使用して設定します。

この機能は UART 制御モードでは使用できません。

2.1.13.6 強制制御入力変更

何らかのデバッグ目的で必要な場合、マスタコントローラ（または PC）から UART コマンドを送って制御入力を変更し、直ちに新しいモードを有効にすることが可能です。制御入力が他の 3 つの入力から UART 制御に切り替えられた場合、新しいモータ制御コマンドを受け取るまでモータの状態（駆動/停止および“TargetSpeed“変数）は変わりません。

2.1.13.7 制御入力のカスタマイズ

デフォルトでは、制御入力（VSP アナログ入力/周波数入力/デューティサイクル入力）とモータ目標速度の関係は、図 32、図 33、および図 34 に示すように線形です。アプリケーションで制御入力とモータ目標速度の間に任意の関係を実装する必要がある場合は、デフォルトの線形制御入力方法を無効にして、スクリプト言語を使って制御入力をカスタマイズすることができます。

VSP アナログ入力制御方法では、スクリプトを使って、アナログ入力電圧を‘ADC_Result0’変数から読み取ることができます。

周波数またはデューティ入力制御をカスタマイズするには、‘AppConfig’変数の 6 番目のビットをセットして、‘ControlFreq’変数と‘ControlDuty’変数を 10ms 毎に関連する周波数およびデューティサイクルの測定結果によって更新する必要があります。サポートされている入力周波数範囲は 5～5000Hz、入力デューティサイクル範囲は 1～99%です。

周波数入力制御方法では、スクリプトを使って、測定された入力周波数を‘ControlFreq’変数から読み取ることができます。

デューティサイクル入力制御方法では、スクリプトを使って、測定された入力デューティサイクルを‘ControlDuty’変数から読み取ることができます。

2.1.14 ホールセンサインターフェイス

MCE ホール角度抽出アルゴリズムは、モータ PWM サイクル毎のロータ角度および速度信号を、電気サイクル毎に 4 回（2 ホールセンサ）または 6 回（3 ホールセンサ）発生するデジタルホール入力遷移イベントから推定します。オプションの Atan 角度アルゴリズムは、2 つのアナログホールセンサ信号からモータ PWM サイクル毎のロータ角度および速度信号を抽出します。

MCE ホールセンサインターフェイスは、表 8 に示すように、以下のホールセンサ構成をサポートします。

表 8 サポートされるホールセンサ構成

インターフェイスのタイプ	サポートされる構成	センサ変位（電気角）
デジタル	2/3 デジタルホールセンサ	120°
アナログ	2 アナログホールセンサ	120°

2.1.14.1 インターフェイス構成

以下の図 35 に示すように、アナログホールセンサの正および負の出力は、それぞれ設定可能なヒステリシス（‘SysConfig’パラメータのビットフィールド[15:14]）を持つ内部コンパレータの非反転および反転入力に連結しています。AHALLx+および AHALLx-（x = 1, 2）間のすべてのホールゼロクロスイベントの間、関連するコンパレータ出力はそれに応じてトグルします。内部コンパレータ出力はマルチプレクサ経由でホールイベントキャプチャブロックの H1 および H2 入力に接続しています。また、アナログホールセンサ出力は、アナログホールセンサ出力電圧値をサンプリングする目的で、1 の等価ゲインステージを通じて 4 つの内部 ADC チャンネルに接続しています。これらの値はホール Atan 角度計算方法を有効にしたときに、Atan 角度を計算するのに使われます。

デジタルホールセンサ出力はマルチプレクサ経由でホールイベントキャプチャブロックの対応する H1、H2、H3 入力に直接接続しています。これらの出力はホールイベントタイミング情報およびホールパターンで、ホール PLL ブロックによってホール角度およびホール速度を推定するのに使われます。

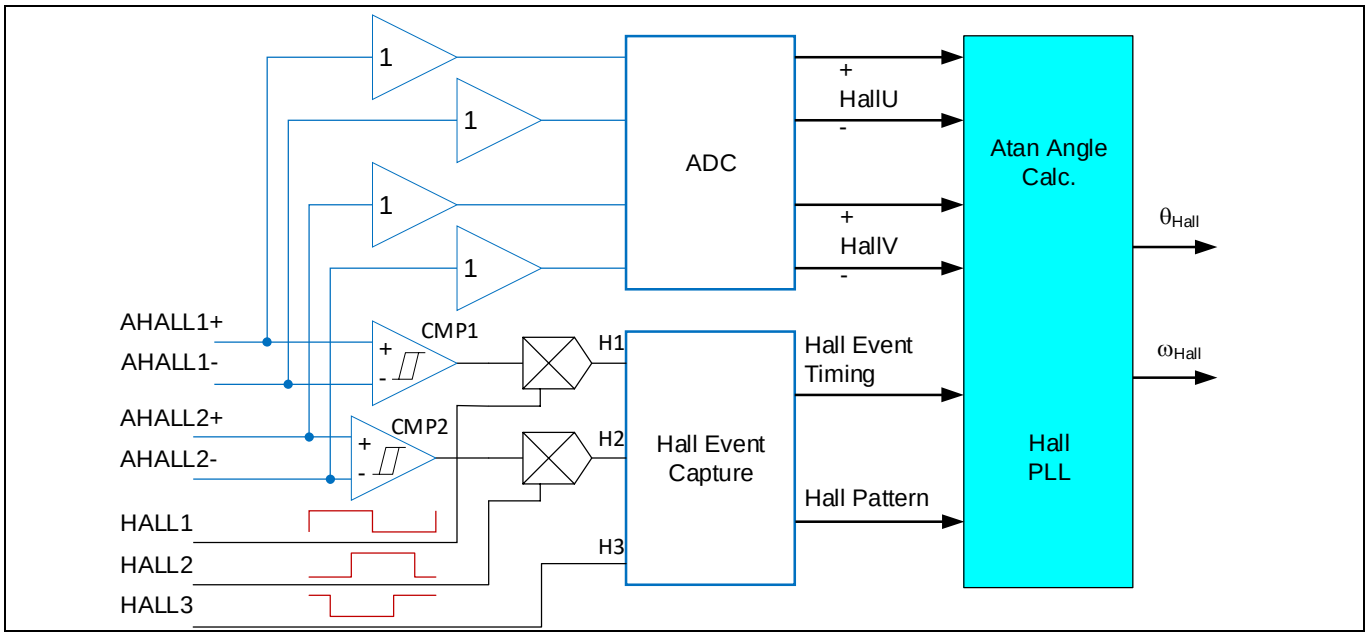


図 35　ホールセンサインターフェイスのハイレベルストラクチャの概要

2.1.14.2 ホールサンプルデバウンスフィルタ

ホールイベントキャプチャブロックには、ホール入力をサンプリングする前にデバウンスチェック機構を提供するハードウェアノイズフィルタが入っています。ノイズフィルタタイミング機構を以下の図 36 に示します。H1、H2、および H3 入力で遷移が検出された場合にはいつでも、設定可能なデバウンス時間（T_{DB}）が経過するまで、その状態はサンプルされません。T_{DB} が経過する前に別の遷移が発生した場合、予定されたその後のサンプリング操作はキャンセルされて、また最初からデバウンス時間の計測が行われます。このデバウンス時間は‘HallSampleFilter’パラメータを使用して設定でき、数式$T_{DB} = HallSampleFilter \times 10.417ns$で計算されます。

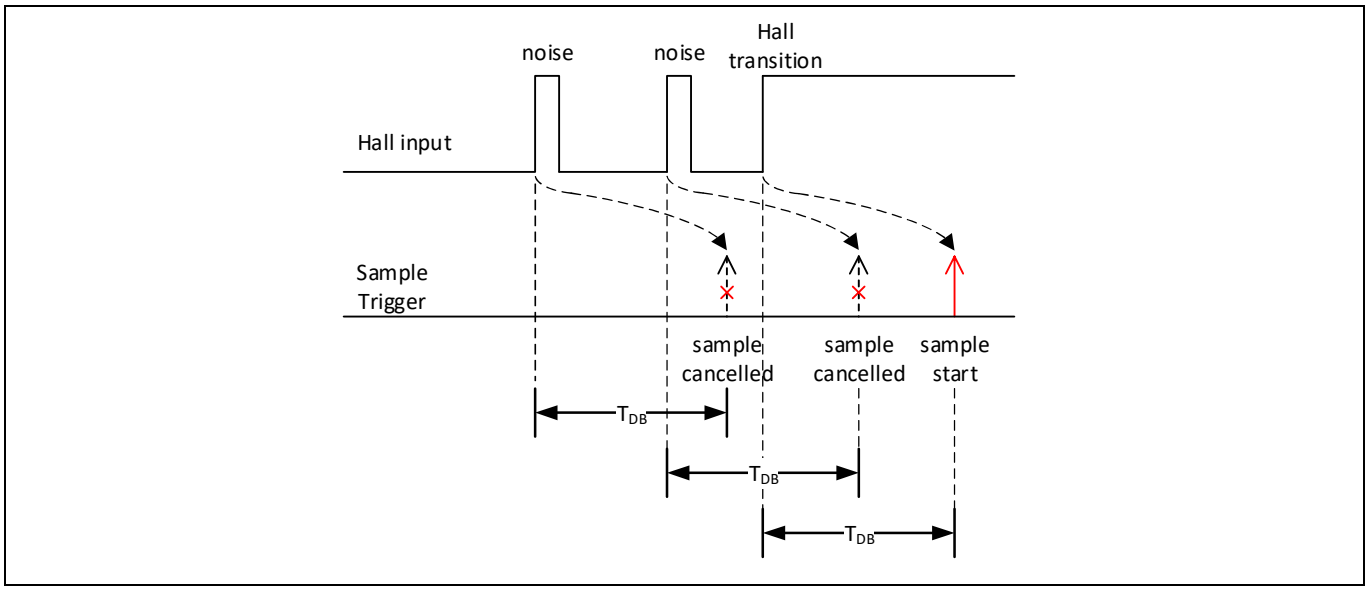


図 36　ホールセンサノイズフィルタのタイミング図

2.1.14.3 ホール角度推定

デジタルホールセンサまたはコンパレータベースのアナログホールセンサインターフェイスは、各ホール遷移イベントで離散角度入力を提供します。3 デジタルホールセンサ構成では、ホール遷移イベントは 60°電気角毎に発生します。2 デジタルホールセンサ構成では、ホール入力遷移イベントは 60°電気角毎（通常セクタ）に、または 120°電気角毎（ワイドセクタ）に交互に発生します。2 アナログホールセンサ構成では、2 つの内部コンパレータを使用してゼロクロスイベントを検出しており、2 デジタルホールセンサ構成の場合と同じように、対応するホール遷移イベントが発生します。

MCE のホール角度推定アルゴリズムは、ホール周波数推定を統合して、連続するホール遷移イベント間のホール角度を推定します。PLL ループを利用して実際のホール周波数を追跡し、ホール周波数積分器から補正項を引くことによって、次のホール遷移イベントサイクルの間に角度推定誤差を修正します。

ホールイベントキャプチャブロックのデジタル入力の状態は、ホール遷移イベントが発生したときに、MCE のハードウエア周辺機器によってサンプルされます。サンプルされたホール入力は、セクション 2.1.14.3.1 の説明のように、一定のホールパターンを形成します。

MCE のホール角度推定ルーチンは各モータ PWM サイクル中に実行されます。ホール角度推定プロセスの詳細は、以下の 2 つのサブセクションで説明します。

2.1.14.3.1 PLL によるホール角度推定

ホール PLL を有効化すると（KpHallPLL > 0）、ホール角度推定アルゴリズムは図 37 に示すようになります。

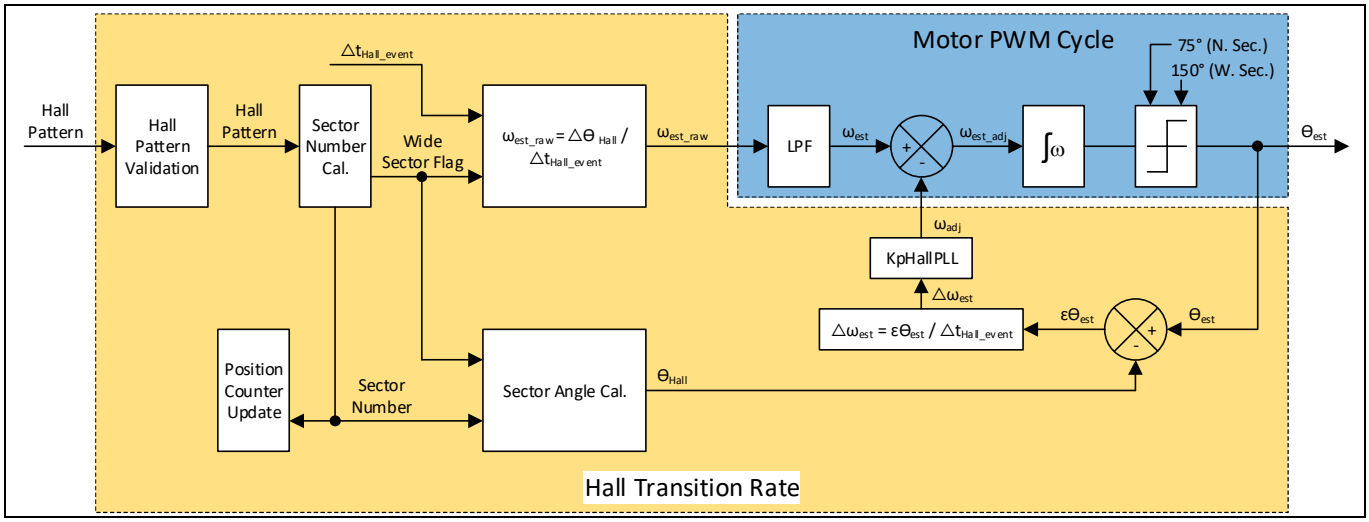


図 37　ホール角度推定アルゴリズムの図（ホール PLL 有効）

各モータ PWM サイクルの間、MCE のホール角度推定ルーチンはローパスフィルタ処理されたホール周波数推定値ω_{est}（‘Hall_FrequencyOut’変数）と補正項ω_{adj}（‘HallPLL_FrequencyAdjust’変数）の差ω_{est_adj}を積分して、ホール角度推定値θ_{est}を生成します。これは図 37 の青色のブロックに示されています。ホール角度推定値の増分が最後のホール遷移イベントから最大 75°まで（通常セクタ）、または 150°まで（ワイドセクタ）増加した場合、それ以上の積分は行われず、次のホール遷移イベントの発生までθ_{est}は一定値のままになります。

MCE はまた、各モータ PWM サイクル中に、最後のチェック以降に新しいホール遷移イベントが発生しなかったかどうかをチェックします。新しいホール遷移イベントがあった場合、図 37 のオレンジのブロックに示されるようなステップが実行されます。

新しくサンプルされたホールパターンは、まず、回転方向に基づいて予想されるパターンによって検証を行います。

検証が成功したら、次に図 39 に示すように、ホールパターンとセクタ番号の間のマッピング関係に基づいて、対応する新しいセクタ番号（‘HallStatus’変数のビットフィールド[6:4]）が計算されます。

2.1.14.6 に説明があるホール位置カウンタは、セクタ番号が更新されるときに、回転方向に基づいて増加または減少されます。

次に、PWM サイクル毎の角度変化量を表すホール周波数推定値（フィルタ前）ω_{est_raw}は、2 つの連続するホール遷移イベント間の角度差$\Delta\theta_{Hall}$と 2 つの連続するホール遷移イベント間の時間間隔の除算の結果として計算されます（$\omega_{est_raw} = \frac{\Delta\theta_{Hall}}{\Delta t_{Hall_event}}$）。更新された未処理のホール周波数推定値は設定可能な時定数T_{decay}でローパスフィルタ処理されたものとなります。このローパスフィルタに対して必要な帯域幅$\omega_c = \frac{1}{T_{decay}}$を求めるには、以下の数式を使って‘HallSpdFiltBW’変数の値を計算してください。

$$HallSpdFiltBW = 2^{16} \cdot (1 - e^{-\frac{\omega_c \cdot Fast\_Control\_Rate}{F_{PWM}}})$$

フィルタされたホール周波数推定値ω_{est}（‘Hall_FrequencyOut’変数）は、ホール角度推定に使用できます。

次に、更新されたセクタ番号と回転方向およびワイドセクタフラグに基づいて、実際のホール角度θ_{Hall}を計算します。ホール角度推定値θ_{est}は、ホール遷移イベント毎に直ちに調整されるわけではありません。次のホール遷移イベントサイクルの間にホール周波数積分器に補正項ω_{adj}を加算することにより、ホール角度推定誤差$\varepsilon\theta_{est}$を修正します。周波数補正項ω_{adj}は、比例係数（‘KpHallPLL’パラメータ）と、角度推定誤差$\varepsilon\theta_{est}$を 2 つの連続して起きるホール遷移イベント間の時間間隔で除算したものとを掛け合

わせた積として計算されます（$\omega_{adj} = KpHallPLL \times \frac{\varepsilon\theta_{est}}{\Delta t_{Hall\_event}}$）。角度推定誤差$\varepsilon\theta_{est}$が 15°（通常セクタ）または 30°（ワイドセクタ）を超える場合には、ホール角度推定値θ_{est}は実際のホール角度θ_{Hall}にリセットされ、補正項ω_{adj}は 0 にリセットされます。

最後に、‘HallAngle’変数は以下の数式を使って更新されます。

$$HallAngle = \theta_{est} + HallAngleOffset$$

‘HallAngleOffset’パラメータの設定についてはセクション 2.1.14.7 に説明があります。‘HallMotorSpeed’パラメータは、ローパスフィルタ処理されたホール周波数推定値ω_{est}と対応するスケーリング係数の積によって更新されます。

ホール PLL を有効にした場合、角度推定誤差$\varepsilon\theta_{est}$はその後に続く複数のホールイベントサイクルの間に修正されるため、ホール角度推定値θ_{est}はホール遷移イベント毎に急変動することはありません。このため、モータは加速時または減速時によりスムーズに動作することが期待できます。ホールセンサを使用するときは、よりよい性能を得るために、‘KpHallPLL’パラメータの値を 0 から 4096 の間で選択してホール PLL を活用することを推奨します。

‘KpHallPLL’パラメータの値を選択するときは、アプリケーションの要件に応じて、トルク/速度応答性とスムーズな作動のどちらを優先するかをよく検討してください。たとえば、ドア開閉アプリケーションの場合、トルクの応答性が高い方が好まれるかもしれませんし、ファンのアプリケーションの場合には、トルクの応答性よりもスムーズな作動の方が優先されるかもしれません。‘KpHallPLL’変数の値を低くすると素早い速度/トルク反応が得られますが、推定ホール速度と角度が急に変化するため、作動のスムーズさは低下します。‘KpHallPLL’変数の値を高くすると、トルク/速度変化はスムーズになりますが、応答性は低下します。

2.1.14.3.2 PLL なしのホール角度推定

ホール PLL を無効化すると（KpHallPLL = 0）、ホール角度推定アルゴリズムは図 38 に図示したようになります。

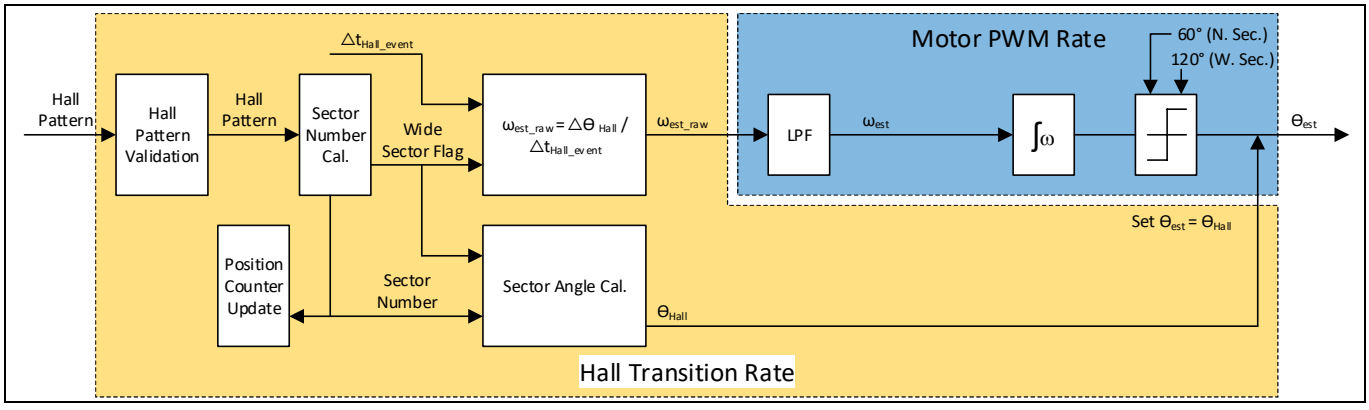


図 38　ホール角度推定アルゴリズムの図（ホール PLL 無効）

各モータ PWM サイクルの間、MCE のホール角度推定ルーチンはローパスフィルタ処理されたホール周波数推定ω_{est}を積分して、ホール角度推定値θ_{est} を生成します。これは図 38 の青色のブロックに示されています。ホール角度推定値の増分が最後のホール遷移イベントから最大 60°まで（通常セクタ）、または 120°まで（ワイドセクタ）増加した場合、それ以上の積分は行われず、次のホール遷移イベントの発生までθ_{est}は一定値のままになります。

MCE はまた、各モータ PWM サイクル中に、最後のチェック以降に新しいホール遷移イベントが発生しなかったかどうかをチェックします。新しいホール遷移イベントがあった場合、MCE は図 38 のオレンジのブロックに示されたような PLL 有効の場合に似た一連の手順を取ります。

ホールパターンの検証、セクタ番号の計算、位置カウンタの更新、ホール周波数推定計算、実際のホール角度計算、‘HallAngle’変数と‘HallMotorSpeed’変数の更新手順は、PLL 有効の場合と同じです。

手順が異なるのは、角度推定誤差の修正です。角度推定誤差 $\varepsilon\theta_{est}$は、ホール遷移イベント毎に、最新の角度推定誤差$\varepsilon\theta_{est}$をホール角度推定値θ_{est}に加算することによって修正します。つまり、ホール角度推定値θ_{est}はホール遷移イベント毎に実際のホール角度θ_{Hall}にリセットされることになります。

ホール PLL を無効にした場合、モータを加速または減速すると、ホール遷移イベント毎にホール角度推定値θ_{est}が急変動する場合があります。

2.1.14.4 ホールゼロ速度チェック

モータ制御ステートマシンが‘MOTORRUN’ステートの場合、2 つの連続するホール遷移イベントの間の時間間隔がしきい値T_{zf}よりも長くなると、ホールゼロ周波数フォールトと見なされて、‘HallStatus’変数のビット[2]がオンになります。しきい値T_{zf}は数式

$$T_{zf} = 4096 \times T_{PWM}$$

を使って計算します。2 つの連続するホール遷移イベントの間の時間間隔がしきい値T_{zf}よりも短くなると、このフォールトは自動的にクリアされます。

2 または 3 デジタルホールセンサ構成で、常にホールゼロ周波数フォールトを発生させるモータ速度は、次の数式を使って計算することができます。

$$\omega_{zf\_3Hall}(rpm) = \frac{1}{4096 \times T_{PWM}} \times \frac{1}{6} \times \frac{60}{pole\_pair}$$

このホールゼロ周波数フォールトが$T_{HallTimeOut}$の間続く場合は、「ホールタイムアウト」フォールトとなります。しきい値$T_{HallTimeOut}$は、以下の数式により‘HallTimeoutPeriod’パラメータを使って設定できます。

$$T_{HallTimeOut} = HallTimeoutPeriod \times 10ms$$

このフォールトは、ホールセンサの使用中にロータのロック状態を検出するためのものです。

2.1.14.5 ホールパターン検証

ホールパターンは、2 進数（$[H3, H2, H1]_b$)で表されます。これは、以下の図 39 に示すように、ホールイベントキャプチャブロックの 3 つのデジタル入力、H1、H2、および H3 を使って、H3 がビット 2、H2 がビット 1、H1 がビット 0 であると見なして行われます。たとえば、H3 がロジックハイ、H2 がロジックロー、H3 がロジックハイの場合、ホールパターンは$[101]_b = 5$ であると認識されます。

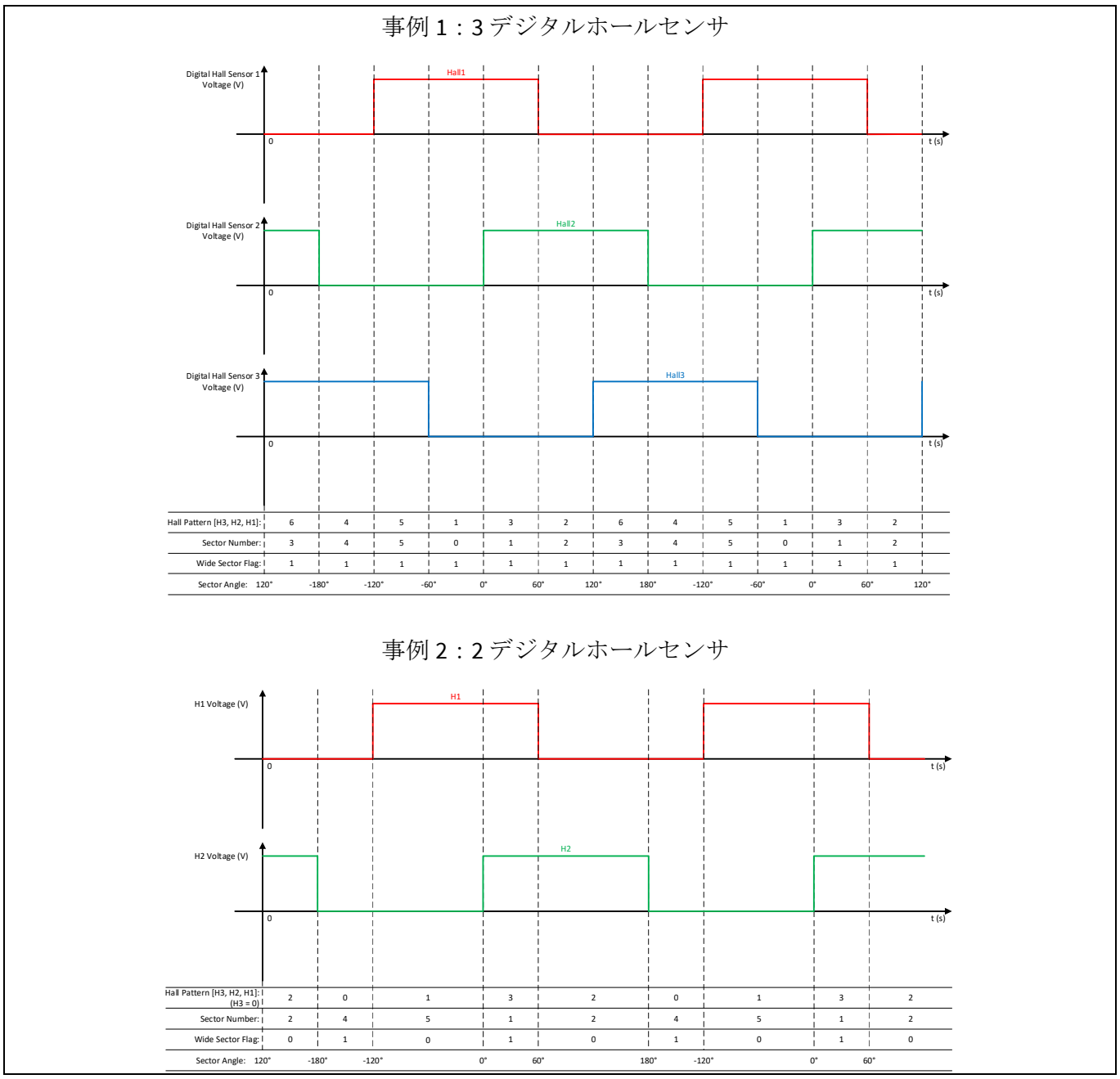


図 39　ホール入力からのホールパターン、セクタ番号、ワイドセクタフラグ、およびセクタ角度の計算

ホールパターン検証ではまず初めに、新しくサンプルされたホールパターンと、モータの回転方向に基づいてあらかじめ決められたホールパターンシーケンスから求めた推定ホールパターンとを比較します。

新しくサンプルされたホールパターンが[111]または[000]の場合、無効なパターンフォールトと見なされます。無効なパターンフォールトの発生が 2 回連続して検出されると、‘Hall Invalid’フォールトとなり、‘FaultFlags’変数の 15 番目のビットがオンになります。この無効パターンチェックは 3 デジタルホール構成にのみ適用される点に注意してください。

新しくサンプルされたホールパターンは有効であるが、CW 回転のホールパターンシーケンスから得られた推定ホールパターンとも、CCW 回転のホールパターンシーケンスから得られた推定ホールパターンとも一致しない場合は、予期しないパターンフォールトと見なされます。予期しないパターンフォール

トの発生が 3 回連続して検出されると‘Hall Invalid’フォールトとなり、‘FaultFlags’変数の 15 番目のビットがオンになります。

新しくサンプルされたホールパターンの検証が成功すると、図 39 に示されたホールパターンとセクタ番号のマッピング関係に基づいて新しいセクタ番号（0～5）が抽出されます。

2.1.14.6 ホール位置カウンタ

MCE には、ガレージドアやブラインドなどの一部のアプリケーションで、モータの負荷の位置を追跡するのに使用できる 32 ビットの位置カウンタが搭載されています。新しいホール遷移イベントが検証されて、セクタ番号が MCE によって更新されると、位置カウンタが CW 方向の場合は増加し、CCW 方向の場合は減少します。増加または減少のステップは、通常セクタ（60°変位）では 1 カウント、ワイドセクタ（120°変位）では 2 カウントです。つまり、位置カウンタはセクタカウンタということになります。

位置カウンタの値は‘PositionCounter’パラメータと‘PositionCounter_H’パラメータから読み取ることができます。

2.1.14.7 ホール角度オフセット

2 アナログホールセンサ構成の場合、UV ラインからラインへの逆起電力電圧波形のゼロクロスとアナログホール 1 差動波形のゼロクロスの角度の差は、以下の図 40 に示すようにθ_{offset}として定義されます。

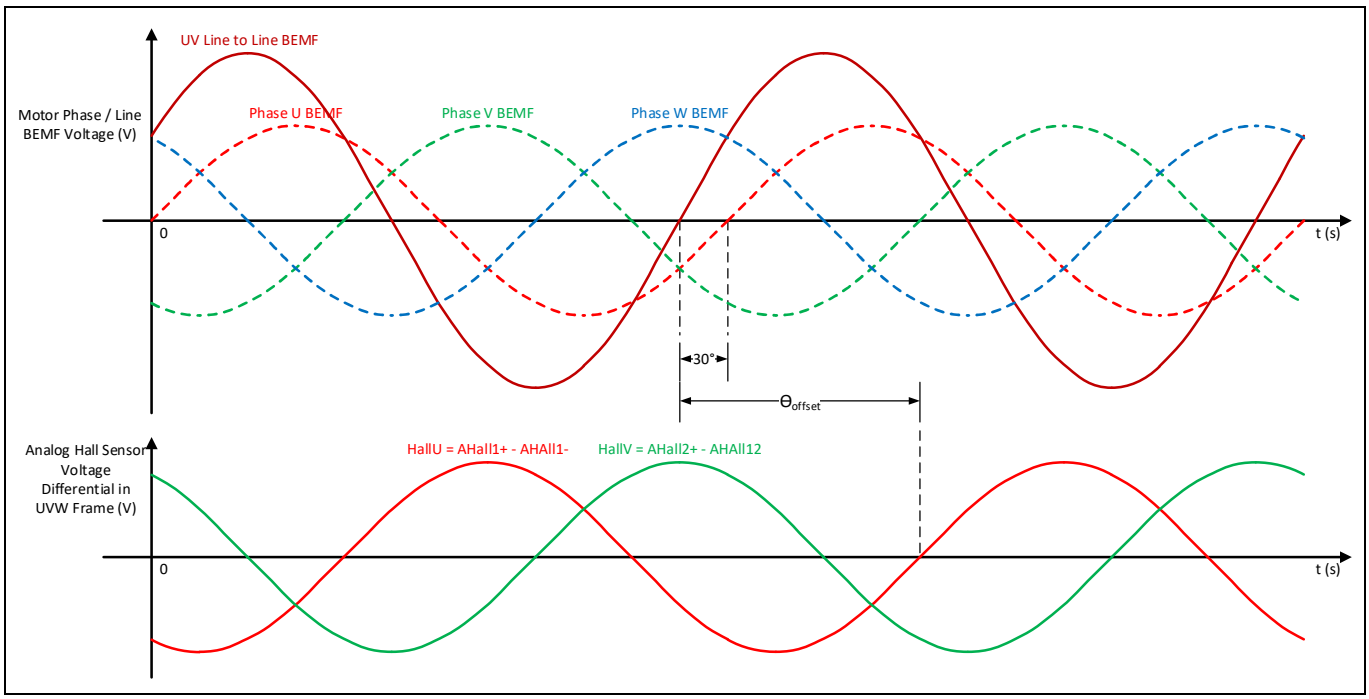


図 40　2 アナログホールセンサ構成の角度オフセット定義図

2 または 3 デジタルホールセンサ構成の場合、UV ラインからラインへの逆起電力電圧波形のゼロクロスロスとアナログホール 1 差動波形のゼロクロスの角度の差は、以下の図 41 に示すようにθ_{offset}として定義されます。

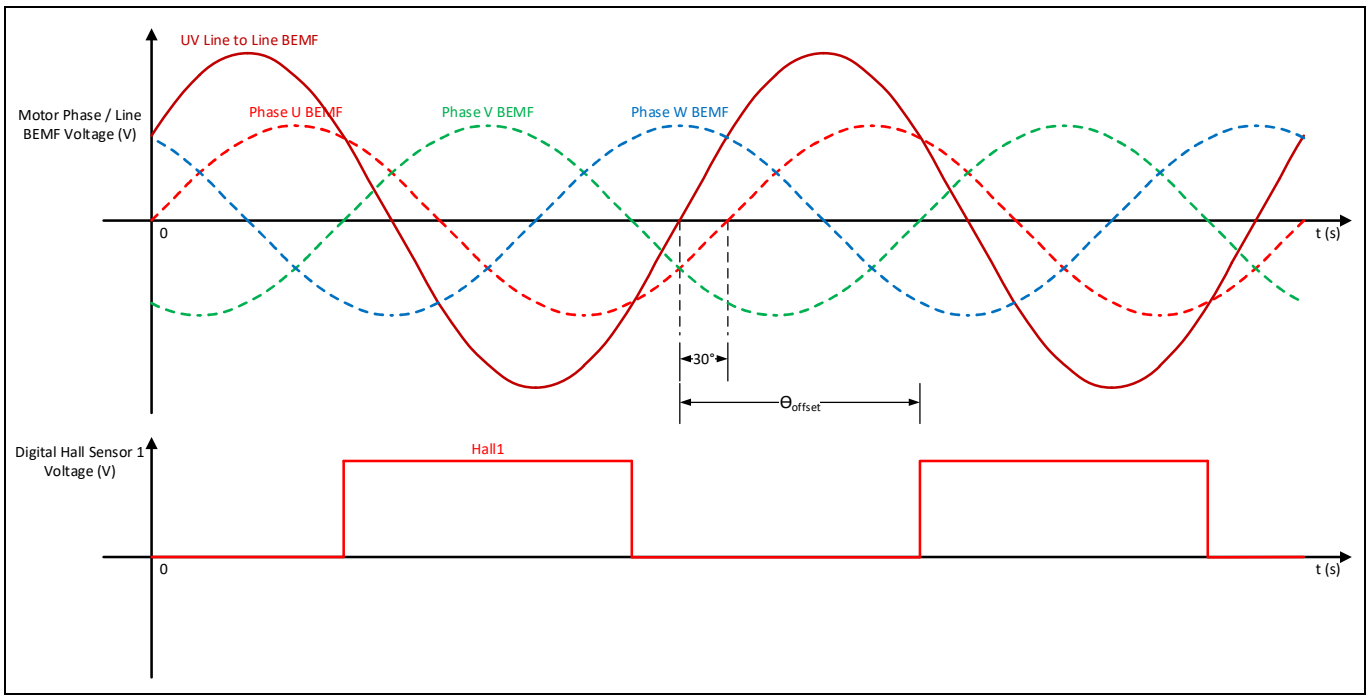


図 41　2 または 3 デジタルホールセンサ構成の角度オフセット定義図

'HallAngleOffset'パラメータはこの数式によって計算します。

$$HallAngleOffset = (\theta_{offset} - 90°) \times \frac{16384}{90°}$$

このパラメータは、各モータ PWM サイクル中に、ロータ位置とホールセンサ（DHALL1 または AHALL1）の取り付け位置の角度差を補償する'HallAngle'変数の最終計算において使用します。

2.1.14.8　Atan 角度計算

上記のような、コンパレータを使用してホールゼロクロスイベントに基づいて行うホール角度計算方法からは、変数推定角度誤差修正率が求められます。モータの速度が低いほど、ホール入力遷移イベントが起きるまでに時間がかかり、推定角度誤差修正率も低くなります。その結果、モータがこのホール角度計算方法を使って起動する場合には、ホール入力遷移イベントの発生率が低いために、最初の数セクタの間は不可避的にホール角度推定値がスムーズに蓄積されません。このことから、望ましくない音響ノイズが発生したり、モータがスムーズに起動しなかったりする場合があります。

MCE は、2 アナログホールセンサ構成に対して、起動中にホール角度推定値計算を補完する Atan 角度計算方法を提供します。Atan 角度計算方法は'SysConfig'パラメータの 13 番目のビットを使って有効または無効にできます。Atan 角度計算方法が有効で、'AngleSelect'パラメータによってホール角度またはハイブリッド角度が選択されているとき、ユーザは'APPConfig'パラメータの上位 8 ビットを使って、'Hall_Atan_Angle'パラメータによって表されたホール Atan 角度が起動中にロータ角度として使用されるセクタの数を指定することができます。

Atan 角度計算プロセスを以下の図 42 に示します。アナログホールセンサ入力（AHALL1+、AHALL1-、AHALL2+、および AHALL2-）の電圧レベルは各モータ PWM サイクル中に測定され、各アナログホールセンサの電圧差は HallU = AHALL1+ - AHALL1-、および HallV = AHALL2+ - AHALL2-として計算されます。MCE はクラーク変換を実行して、UVW 参照フレーム内の HallU と HallV 成分を静止 αβ 参照座標内の Hallα と Hallβ 成分に変換します。その後、$\mathrm{Atan}(\frac{Hall_\beta}{Hall_\alpha})$計算を行って Atan 角度を生成します。これは

'Hall_Atan_Angle'変数によって表され、'HallAngleOffset'パラメータによって指定されるホール角度オフセットを加算して得られる値です。

補完的な Atan 角度計算方法を使用することにより、Atan 角度を使用するロータ角度は、上記のホール角度推定方法を使った場合と比較して、モータの起動中の音響ノイズが非常に低くなり、よりスムーズな累積が得られることが期待されます。アナログホールセンサ信号では、高次高調波が高くなる場合があります。その結果、Atan 角度計算はロータの速度変動を反映したものとは異なり、意図しない変動が生じることになります。そのため、'APPConfig'パラメータの上位 8 ビットを設定することにより、ホール Atan 角度計算の使用を、起動時の短い期間のわずか数セクタのみに制限することが推奨されます。

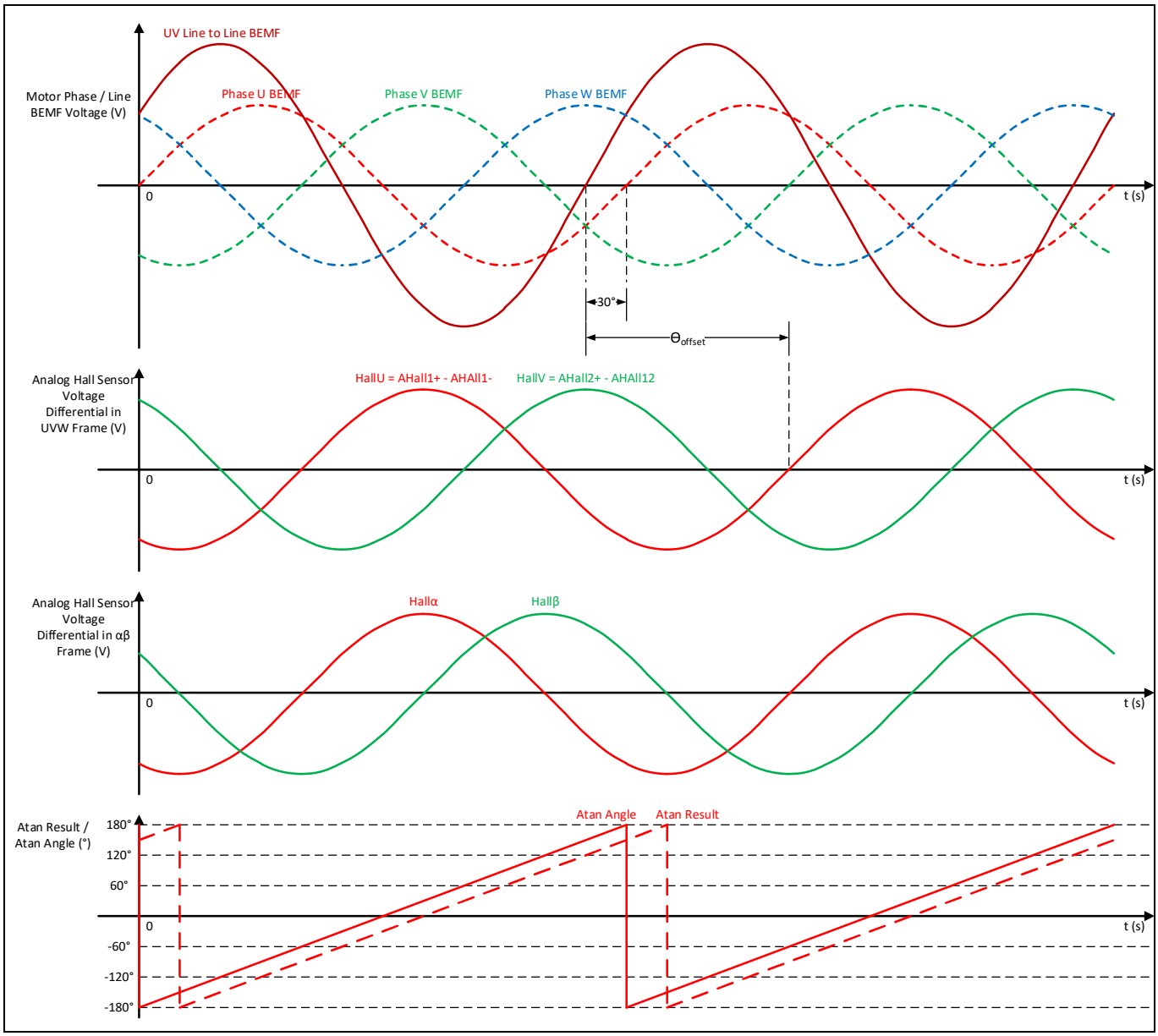


図 42　2 アナログホールセンサ入力に基づく Atan 角度計算（AHall1+、AHall1-、AHall2+、AHall2-）

2.1.14.9　ホール初期位置推定

デジタル 2 または 3 ホール構成の場合、および Atan 角度計算方法を使用しないアナログ 2 ホール構成の場合、起動時の初期ロータ位置は初期ホール入力に基づいて MCE が推定します。MCE は、初期ホー

ルパターンから解釈された角度範囲の中央からロータが始動すると見なします。以下の表 9 と表 10 に、3 ホールおよび 2 ホール構成の初期角度推定の詳細を示します。

表 9　　ホール初期位置推定（3 ホール、HallAngleOffset = 0）

ホールパターン [H3, H2, H1]	1 (001_b)	3 (011_b)	2 (010_b)	6 (110_b)	4 (100_b)	5 (101_b)
角度範囲	-60～0°	0～60°	60～120°	120～180°	180～240°	240～300°
初期角度	-30°	30°	90°	150°	210°	270°

表 10　　ホール初期位置推定（2 ホール、HallAngleOffset = 0）

ホールパターン [H3, H2, H1] (H3 = 0)	1 (001_b)	3 (011_b)	2 (010_b)	0 (000_b)
角度範囲	-120～0°	0～60°	60～180°	180～240°
初期角度	-60°	30°	120°	210°

2.1.14.10 ホールセンサ/センサレスハイブリッド動作

MCE は、ホールセンサインターフェイスドライバおよび磁束推定器と磁束 PLL の両方が有効になるハイブリッドモードをサポートしています。以下の図 43 に示すとおり、起動時にロータ角度はホールセンサインターフェイスドライバからのホール角度推定値を使用します。モータの速度が‘Hall2FluxThr’パラメータによって設定されたホール/磁束速度しきい値を超えて増加すると、ロータ角度は磁束推定器と磁束 PLL からの推定磁束角度を使ったものへと切り替わります。ロータ角度が磁束角度から供給されている間は、モータ速度が‘Flux2HallThr’パラメータによって設定された磁束ホール速度しきい値未満まで下がった場合、ロータ角度はホールセンサインターフェイスドライバからのホール角度推定値を使ったものに再び切り替わります。ハイブリッドモードでは、ホールセンサインターフェイスドライバおよび磁束推定器と磁束 PLL の両方が同時に作動していますが、2 つの出力のうち 1 つのみが角度ループを閉じるためのロータ角度として使用されます。

MCE は優れた性能を誇る高度なセンサレスアルゴリズムを提供しますが、一部のアプリケーションは起動時および/または低速運転時にさらに高い性能を必要とします。この場合、優れた起動性能と低速性能を提供するホールセンサを使用することで、センサレスオプションを補完することができます。このように、センサレスモードとホールセンサモードの両方の利点を活用することで、起動時を含む幅広い速度範囲において駆動システムの高い性能を一貫して使用できるハイブリッドモードを選択することをお勧めします。

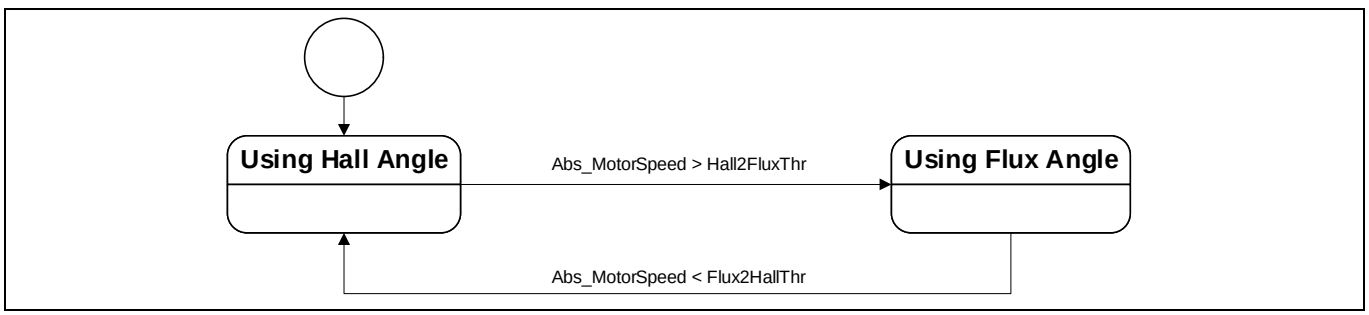


図 43　　ホールセンサ/センサレスハイブリッドモード図

2.1.15 トルク補償

シングルロータリコンプレッサベースのエアコンや冷蔵庫のアプリケーションでは、吸収ステージから圧縮ステージのメカニカルサイクル内でトルク要求に大きな変動があります。速度制御帯域幅には制限があるため、1 メカニカルサイクル内でトルク要求が変動することによりモータ速度が変化し、明らかな機械的振動や望ましくないノイズを引き起こします。この問題を解決するため、MCE にはトルク補償機能が搭載されています。この機能は、メカニカルサイクルを検出して同期し、フィードフォワードループを使ってメカニカルサイクル毎に正弦波補償曲線によってトルク参照値を変調することにより、速度の変動を最小限に抑え、振動を減少します。この機能は、トルク参照値と磁束角度（センサレスモードでのロータ電気角）を入力として使用します。設定可能な速度範囲の中に 2 つの主な作動モードがあります。一方のモードはメカニカルサイクル内のピーク負荷トルクに同期し、もう一方のモードはフィードフォワード補償トルクを計算します。MCE のトルク補償機能は 4 極または 6 極のコンプレッサモータをサポートします。

トルク補償機能は“AppConfig“’パラメータのビット[4]を使って有効または無効にできます。

図 44 に、MCE トルク補償機能を有効にしたときの状態遷移を図示します。まず、TC_Speed_Not_Valid 状態から開始します。この状態ではトルク補償機能は無効です。TC_Speed_Not_Valid 状態に入ると、初期化プロセス（TorqueComp_init()）が実行されます。このとき、“TrqCompBaseAngle“、“TrqCompStatus“、および‘TrqRef_Ext’などの関連する変数がリセットされます。モータ速度参照値（‘SpdRef’変数）が‘TrqCompOnSpeed’パラメータによって設定されるターンオンしきい値よりも低くなると、TC_Speed_Valid 状態に切り替わり、トルク補償機能が有効になります。TC_Speed_Valid 状態のとき、モータ速度参照値が‘TrqCompOffSpeed’パラメータによって設定されるターンオフしきい値よりも高くなると、TC_Speed_Not_Valid 状態に戻ります。

トルク補償機能の有効ステータスは‘TrqCompStatus’変数のビット[0]に反映されます。

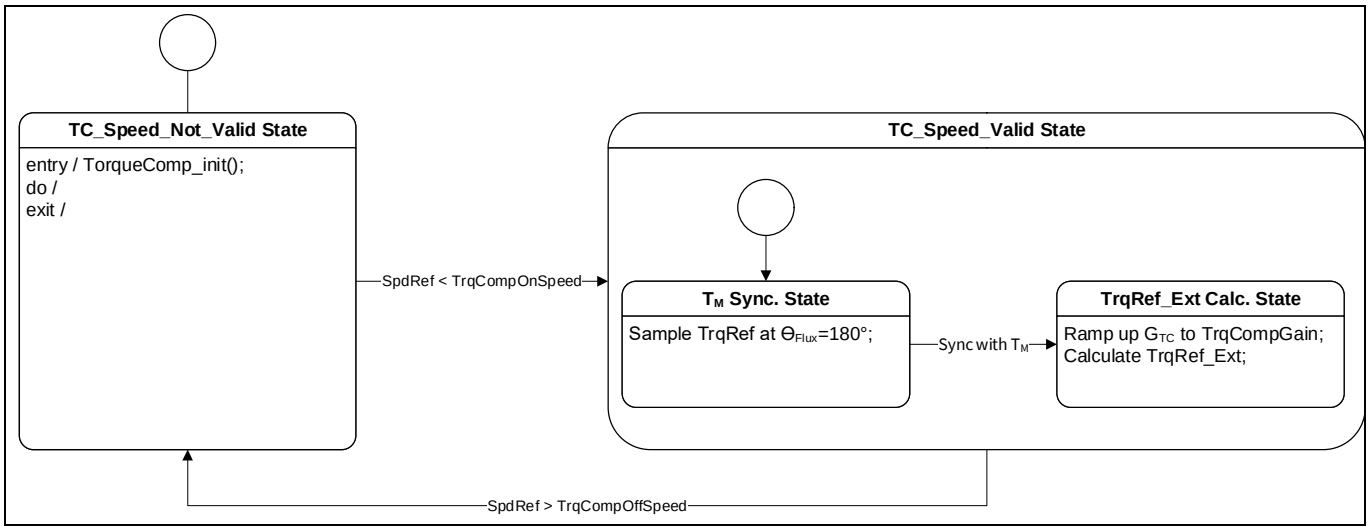


図 44　トルク補償状態遷移図

図 44 および図 45 に示すとおり、TC_Speed_Valid 状態内には 2 つのサブ状態があります。TC_Speed_Valid 状態に入るとき、磁束角度$\theta_{Flux} = 180°$のときに 1 電気サイクル毎に 1 回トルク参照値（‘TrqRef’変数）をサンプルする T_M Synchronization サブ状態から開始します。k 番目のサンプル時間のトルク参照サンプルが TrqRef[k] > TrqRef[k-1]、TrqRef[k] > TrqRef[k-2]、TrqRef[k-3] > TrqRef[k-1]、TrqRef[k-3] > TrqRef[k-2]の関係を連続する 10 回のメカニカルサイクル TM の間満足した場合、メカニカルサイクル T_M 内のピーク負荷トルクと同期したものとみなし、その時点をトルク補償ベース角度θ_{base}（‘TrqCompBaseAngle’変数）のゼロ点として設定します。その後、TrqRef_Ext Calculation サブ状態へと移行します。この状態で、希望する正弦波補償トルク参照値（‘TrqRef_Ext’変数）が以下の数式に従って合成されます。

$$TrqRef\_Ext = G_{TC} \times TrqRef_{Filt} \times \cos(\frac{\theta_{Flux} - 180°}{pole\_pair} + \theta_{base} + \theta_{os}) \times k_{CORDIC}$$

G_{TC}は、要求される補償トルク参照値'TrqRef_Ext'のゲイン係数を表します。

$TrqRef_{Filt}$は、速度 PI レギュレータ出力から求めた要求されるトルク参照値の平均値を表します。これは、上限値で'TrqRef'変数をローパスフィルタ処理した結果です。図 45 に示されたとおり、$TrqRef_{Filt}$が'TrqCompLim'パラメータの値よりも大きい場合、値は'TrqCompLim'の値に制限されます。

k_{CORDIC}は CORDIC ゲイン係数で$k_{CORDIC} = 1.647$です。

要求される正弦波補償トルク参照値の振幅は$G_{TC} \times TrqRef_{Filt} \times k_{CORDIC}$です。$G_{TC}$はゼロから開始して、'TrqCompGain'パラメータの値に達するまで 1 電気サイクル毎に 8 カウントの割合で増加します。

トルク補償ベース角度θ_{base}は、1 電気サイクル毎に 120°（6 極）または 180°（4 極）ずつ増加します。

トルク補償角度オフセットθ_{TCos}（'TrqCompAngOfst'パラメータ)は、メカニカルサイクル内でのピーク負荷トルクと合成された正弦波補償トルク参照値のピークとの角度差を規定します。

メカニカルサイクル T_M との同期ステータスは'TrqCompStatus'変数のビット[1]に反映されます。

図 45 に示すように、合成補償トルク参照値'TrqRef_Ext'は'TrqRef'と合算されて合計トルク参照値（'TrqRef_Total'変数）となります。この値は後続の IPM 制御ブロックで使用されて、d 軸および q 軸電流ループのための電流参照値を生成します。

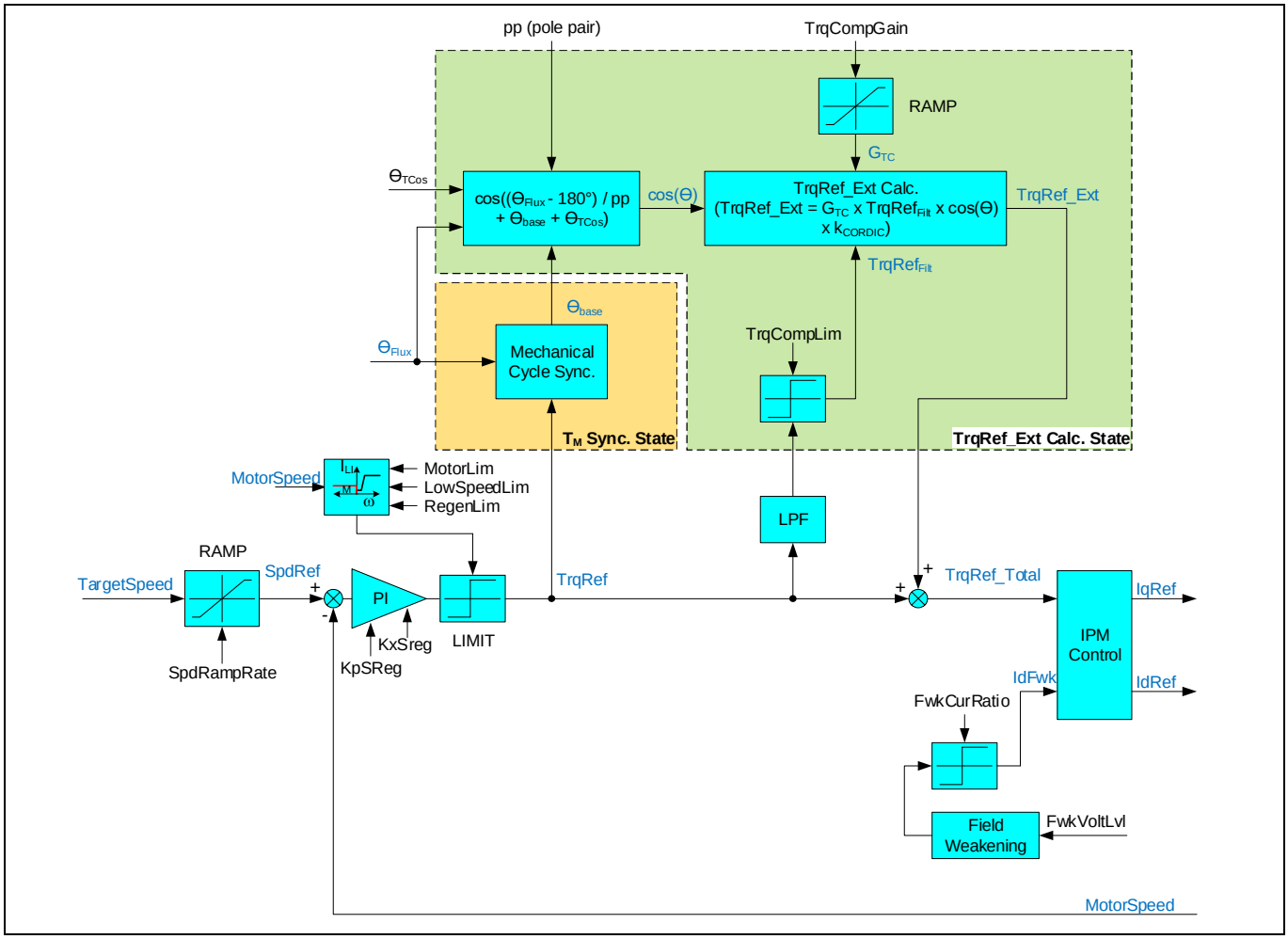


図 45　トルク補償のトップレベルアルゴリズム図

トルク補償関連のパラメータを調整する際は、以下の手順で行うことを推奨します。

a) 初期パラメータ値：$G_{TC} = 0.5$（TrqCompGain = 128）および 50%補償トルク制限値（TrqCompLim = 2048）を設定する。
b) トルク補償機能が無効になる速度上昇しきい値（TrqCompOffSpeed）を設定する。
トルク補償機能が有効になる速度減少しきい値（TrqCompOnSpeed）を設定する。これら 2 つのパラメータには振動回避のために約 2%のヒステリシスが設定されている。
c) MCEDesigner のトレース機能を使って‘SpeedError’変数をプロットする。
d) ‘TrqCompAngOfst’の値を‘SpeedError’の振幅とコンプレッサの振動が最小になる値に調整する。
e) 必要な場合は、‘TrqCompGain’の値を上げてコンプレッサの振動をさらに抑制する。

2.1.16 保護機能

2.1.16.1 磁束 PLL 制御不能保護

磁束 PLL が正しいロータ角度にロックされている場合、モータの永久磁石の磁束を表す Flx_M は 2048 カウントで正規化された DC 値になります。反対に、PLL が正しいロータ角度でロックされていない場合、Flx_M は不安定になるか、2048 カウントから大きく乖離した値になります。磁束 PLL 制御不能保護は、このようなフォールト状態を検出するために設計されたメカニズムです。

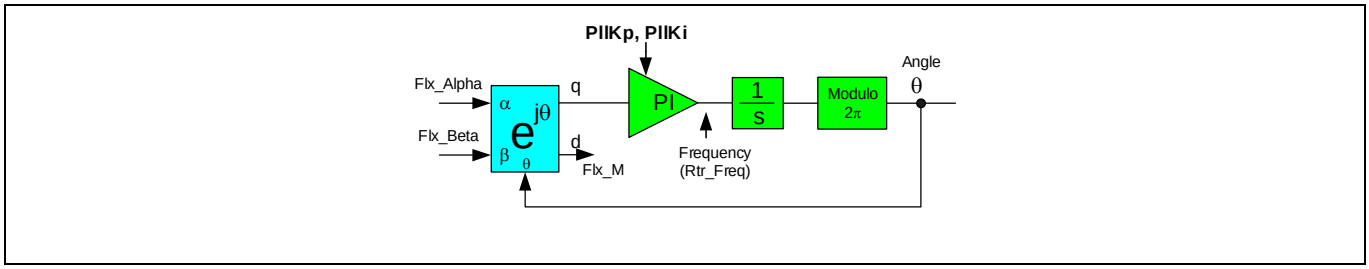


図 46 磁束 PLL の簡略化されたブロック図

Flx_M の値が 512 を下回ったり、8192 を超えたりすると、MCE は特定のタイムスロット（‘FluxFaultTime’パラメータで設定）内で Flx_M をモニタし続けます。連続 8 個のタイムスロットでこの状態が起きた場合（各スロットの時間は FluxFaultTime/8 と等しい）、磁束 PLL は「制御不能」とみなされます。詳細は、図 47 を参照してください。

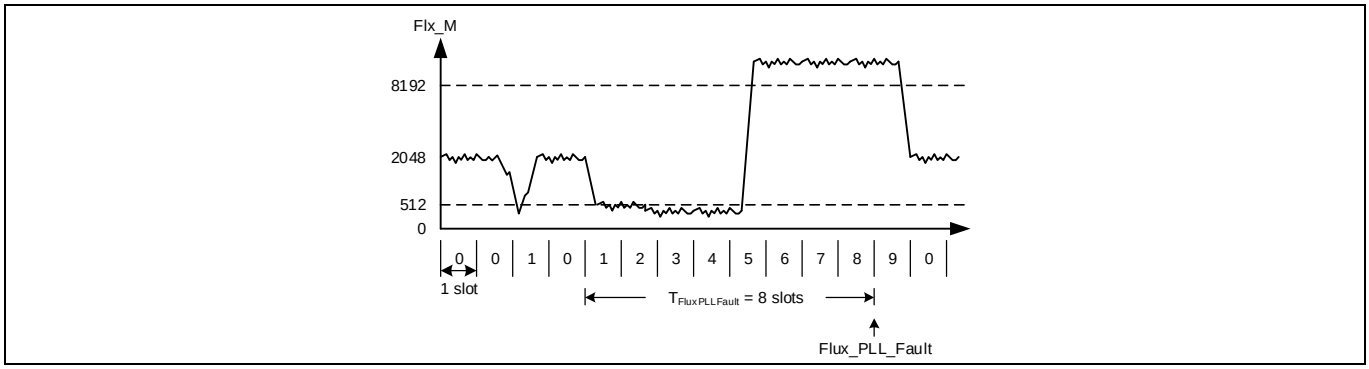


図 47 **磁束 PLL 制御不能保護**

磁束 PLL 制御不能フォールトが確認されると、’FaultFlags’モータ変数のビット 4 をセットして報告し、モータ速度制御ループはリセットされます。‘FaultEnable’モータダイナミックパラメータのビット 4 が

セットされると、このフォールトは‘SwFaults’モータ変数に反映され、モータステートマシンは FAULT ステートへ移行して、モータの駆動を停止します。このビットがセットされない場合、‘SwFaults’変数の対応するビットは‘FaultEnable’パラメータによってマスクされ、このフォールトは‘SwFaults’変数に反映されなくなります。この場合、モータステートマシンは FAULT 状態へ移行しません。この保護は欠相状態を検出することもできます。

PLL 制御不能フォールトの応答時間は、‘FluxFaultTime’モータパラメータをセットして設定できます。この値の有効範囲は 0～65535 です。値が 1 であれば 0.01 秒に相当します。デフォルト値は 800 にセットされています。これは 8 秒の応答時間に相当します。

2.1.16.2 ロータロック保護

以下の図 48 に示すとおり、ロータロックフォールトは、速度 PI レギュレータの出力（‘TrqRef’変数）が定義された時間T_{Rotor_Lock}飽和状態を継続した場合に検出されます。ロータロック検出時間T_{Rotor_Lock}は、‘RotorLocktime’パラメータを使って、数式

$$T_{Rotor\_Lock} = RotorLockTime \times 10ms$$

によって設定できます。ロータロック保護は、モータ速度がモータ最小速度から最大速度の 25%までの範囲で有効になります。ロータロック保護は、モータ速度が最大速度の 25%を超えると、誤ったフォールト報告が生じないようにするために無効になります。

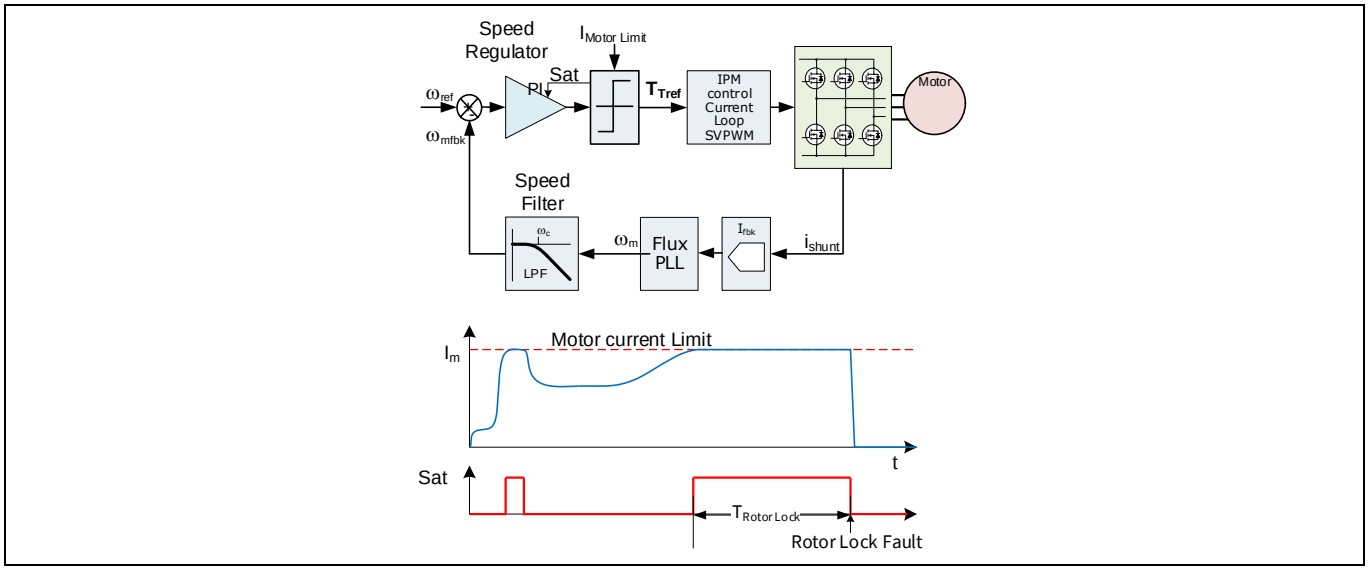


図 48　ロータロック保護のメカニズム図

ロータロックフォールトが確認されると、’FaultFlags’モータ変数のビット 7 をセットすることによって報告されます。‘FaultEnable’モータダイナミックパラメータのビット 7 がセットされると、このフォールトは’SwFaults’モータ変数に反映され、モータステートマシンは FAULT 状態へ移行して、モータの駆動を停止します。このビットがセットされない場合、’SwFaults’変数の対応するビットは’FaultEnable’パラメータによってマスクされ、このフォールトは’SwFaults’変数に反映されなくなります。この場合、モータステートマシンは FAULT ステートへ移行せず、モータは駆動し続けます。

ロータロック検出時間T_{Rotor_Lock} の設定が短かすぎる場合は、加速時や瞬間的な高負荷条件時にフォールトを発生させる場合があります。

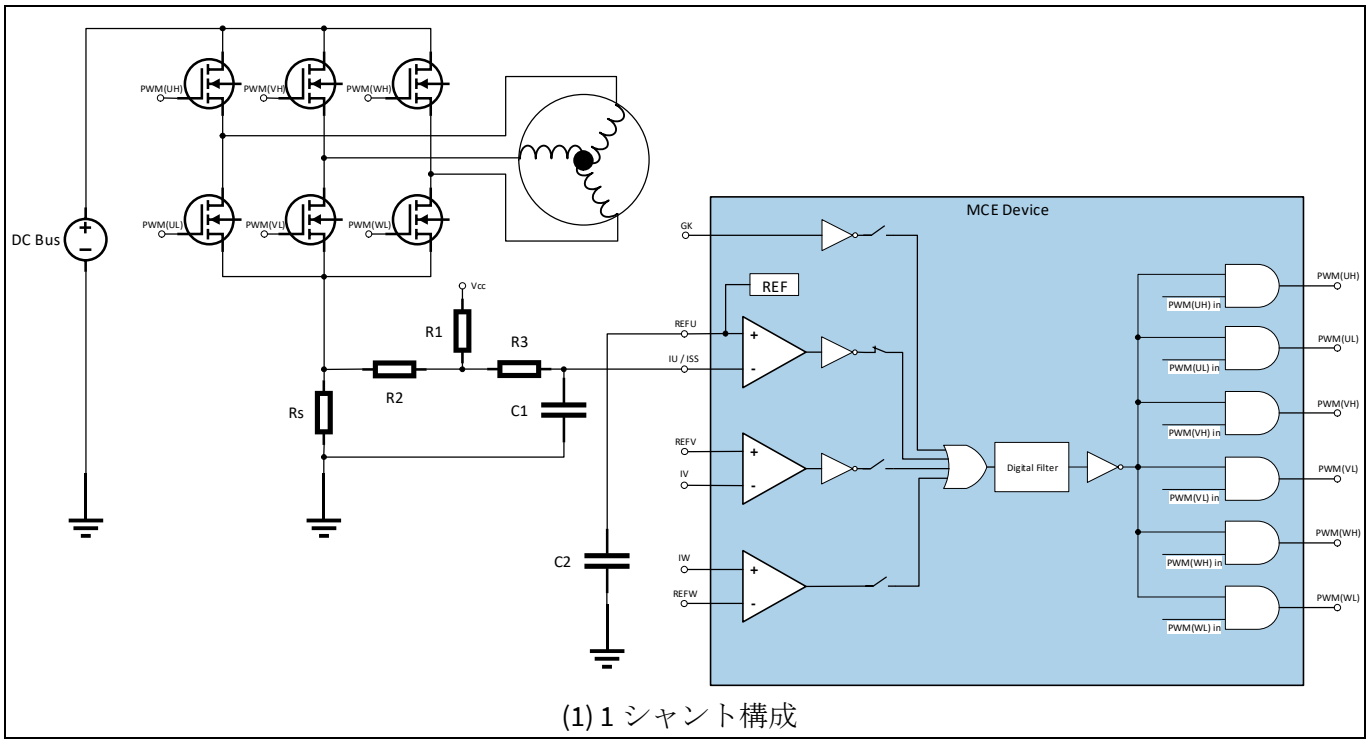
ロータロック検出は、特にモータが低速で駆動している場合には、フォールト報告が 100%保証されているわけではありません。その理由は、ロータロック状態では、速度 PI 出力の飽和が起きないような誤った速度で PLL がロックされてしまう場合があるためです。

2.1.16.3 モータ過電流保護（OCP）

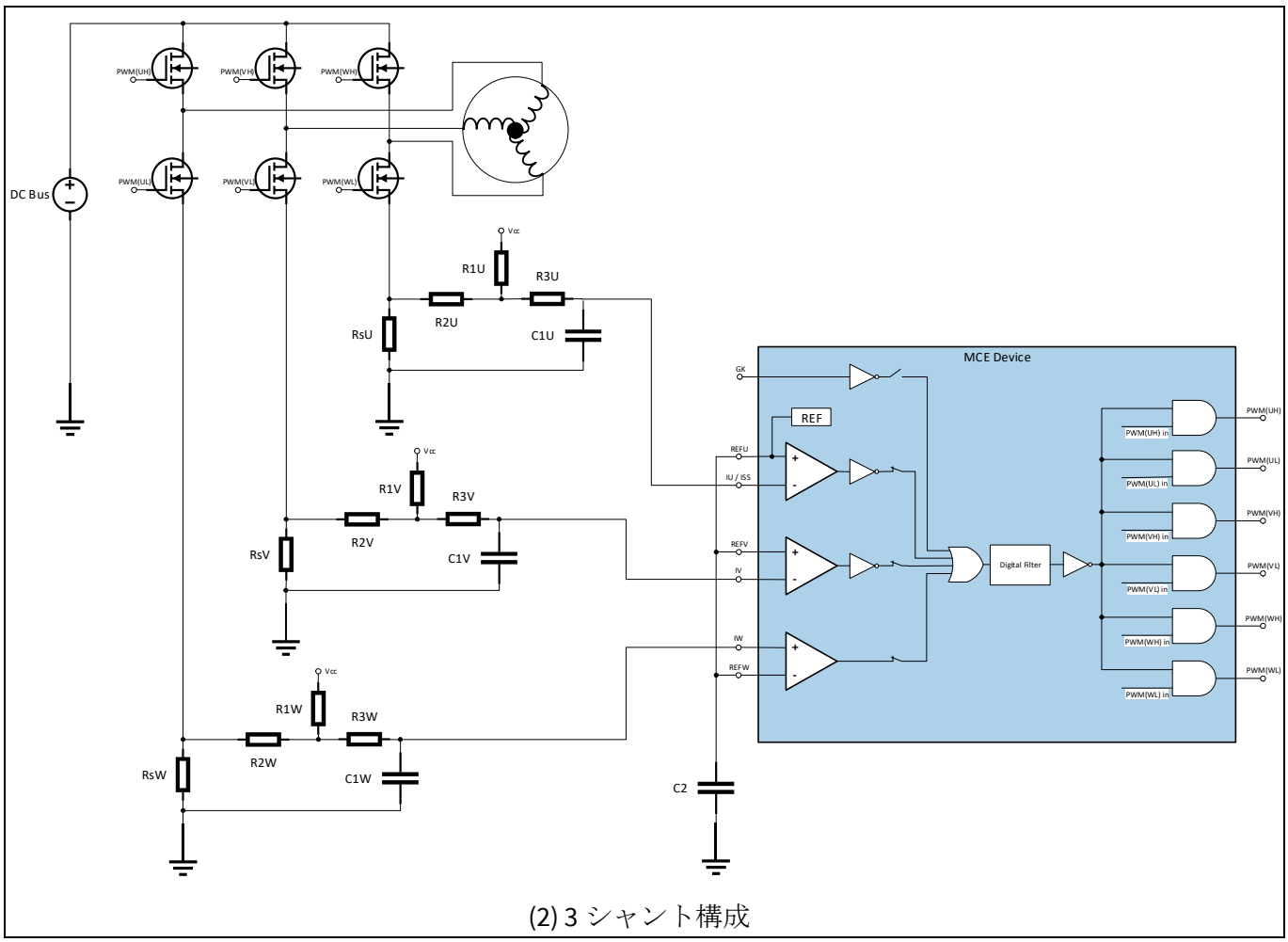
モータのゲートキルフォールトは過電流発生時にセットされます。この過電流状態は、以下の2つの入力源から検出される可能性があります。

1. 直接 GK ピン：入力が LOW のとき、ゲートキルフォールトがセットされる。
2. 内部コンパレータ

2つの検出源の両方またはいずれかを過電流検出ロジックに選択することが可能です。過電流検出源はMCEWizardによって選択できます。



(1) 1 シャント構成



(2) 3 シャント構成

図 49　内部コンパレータを使った一般的なモータ OCP 実装

ユーザは、専用 GK ピンまたは内部コンパレータのいずれかの使用を選択して、過電流保護機能を設定できます。GK ピンを使用する場合、LOW のときにアクティブになるように設定されます。内部コンパレータを使用する場合、正確なトリップ電圧レベルは‘CompRef’モータパラメータをセットすることによって設定できます。内部コンパレータに対する電流トリップレベルは MCEWizard を使って設定できます。‘CompRef’パラメータが電流トリップレベル値を保持します。図 49 (1)に示した通り、1 シャント電流測定構成では、1 つの内部コンパレータのみが使用されています。3 シャント電流測定構成では、図 49 (2)に示すように、3 つの内部コンパレータを使って過電流状態を検出します。

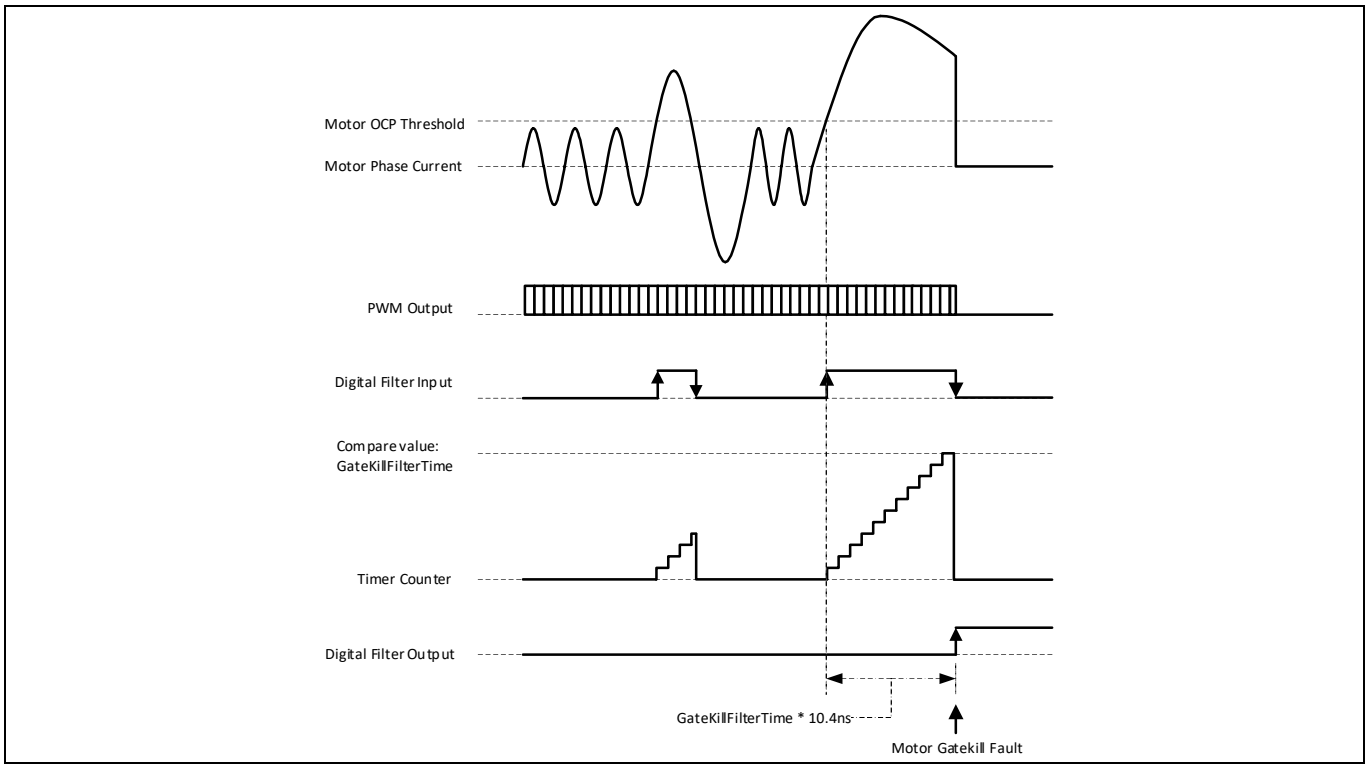


図 50　モータゲートキルフォールトのデジタルフィルタタイミング図

内部的に設定可能なデジタルフィルタを使って入力信号をデバウンスし、誤ってゲートキルフォールトが発生することによって生じる高周波ノイズを防止します。“GatekillfilterTime“パラメータはゲートキルフィルタ時間値をクロックサイクルで表します。フォールト状態を発生させるには、指定されたゲートキルフィルタ時間の間、入力信号が安定した状態を保つ必要があります。

ゲートキルフィルタタイマは、外部 GK ピンまたは内部コンパレータ出力によってトリガされるレベルに設定されます。図 50 に示すとおり、相電流が指定された OCP しきい値を超えた場合、デジタルフィルタのタイマがカウントを開始します。デジタルフィルタの入力値がロジック LOW になる（外部 GK ピンがロジック HIGH になり、内部コンパレータの出力電圧レベルがロジック LOW に変わる）と、タイマはリセットされます。タイマが‘GateKillFilterTime’値までカウントしても過電流状態が続く場合、デジタルフィルタの出力は直ちにロジック HIGH になり、トラップ状態に入ります。この状態では、すべての PWM 出力値がプログラムされたパッシブレベルに入ります。モータゲートキルフォールトは‘FaultClear’モータ変数に 1 を書き込むことによってのみクリアされます。このフォールトはマスクすることができません。このフォールトは’SwFaults’モータ変数に反映され、モータステートマシンは FAULT ステートへ移行して、モータの駆動を停止します。

GateKillFilterTime は静的モータパラメータの一種で、過電流フォールト検出に対するゲートキル応答時間を設定します。この値の有効範囲は 4～960 クロックサイクルです。値が 1 であれば 1/96MHz = 10.4167ns に相当します。デフォルト値は 96 で、これは 1µs に相当します。

2.1.16.4　過温度保護

以下の図 51 に示すとおり、MCE は外部 NTC サーミスタを使って過温度保護（OTP）機能を提供します。一般的には、NTC サーミスタとプルアップ抵抗を使って分圧回路を形成します。MCE の場合、分圧回路の出力をセンシングして、設定可能な OTP シャットダウンしきい値$V_{Shutdown}$と比較します。このしきい値はシステムがシャットダウンされる希望温度$T_{Shutdown}$に対応します。サーミスタの分圧回路の出力が$V_{Shutdown}$を下回ると、OTP フォールトが報告されます。OTP シャットダウンしきい値$V_{Shutdown}$は‘Tshutdown’パラメータを使って設定できます。

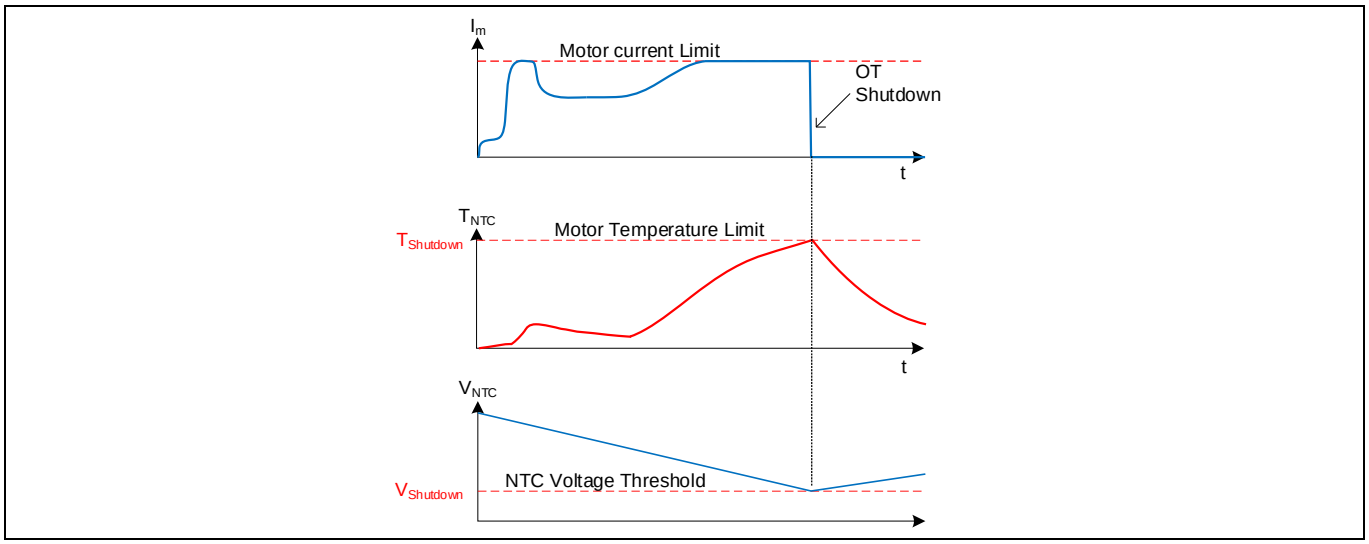


図 51　過温度保護のメカニズム図

過温度フォールトの発生に対応するアクションは、FaultEnable ダイナミックモータパラメータのビット 6 を使用して設定できます。このビットがセットされた場合、モータステートマシンが FAULT ステートになり、モータは駆動を停止します。このビットがセットされない場合、モータステートマシンは FAULT ステートにならず、モータは駆動を継続します。

2.1.16.5 DC バス過電圧/低電圧保護

過電圧/低電圧フォールトは、DC バス電圧が関連する保護電圧しきい値を上回ったり下回ったりしたときに検出されます。

DC バス電圧はモータ PWM サイクル毎に測定されます。測定された DC バス電圧はローパスフィルタを通過して高周波ノイズを減衰します。この値は‘VdcFilt’変数から読み取ることができます。LPF の時定数はモータ制御 PWM 周波数によって決まります。たとえば、モータ制御 PWM 周波数が 15kHz の場合、DC バス電圧のサンプリングレートは 15kHz になります。この場合、時定数 T_{decay} は約 2.1ms となり、カットオフ周波数は約 76Hz となります。

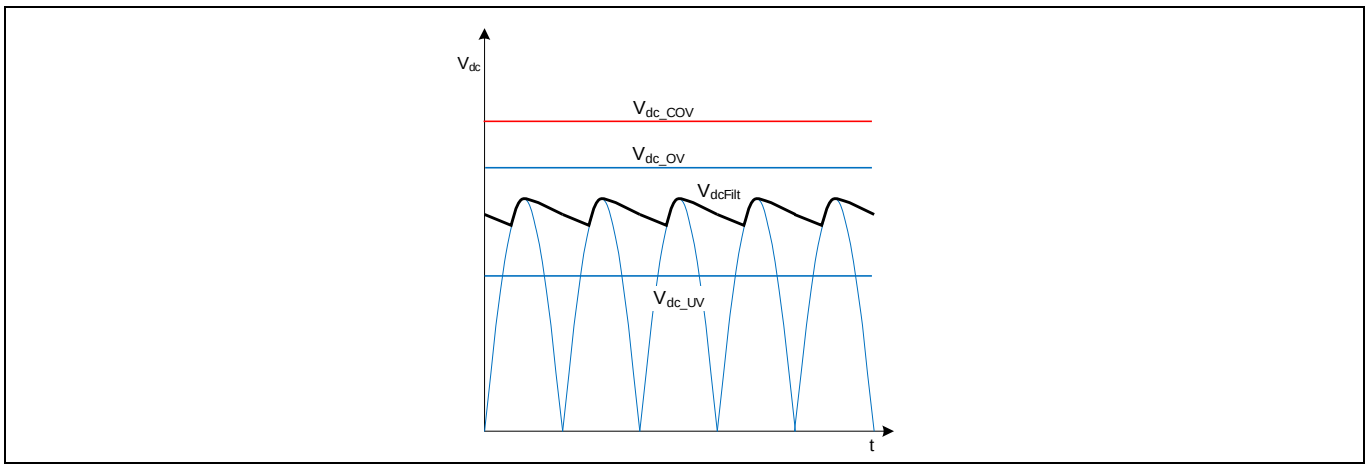


図 52　DC バス過電圧/低電圧保護のしきい値の図

図 52 に示すように、‘VdcFilt’の値が V_{dc_ov}（‘DcBusOvLevel’変数によって設定可能）を上回る場合、対応する’FaultFlags’モータ変数のビット 2 がセットされます。‘FaultEnable’パラメータのビット 2 がセットされると、このフォールトは’SwFaults’モータ変数に反映され、モータステートマシンは FAULT ステート

へ移行して、モータの駆動を停止します。このビットがセットされない場合、'SwFaults'変数の対応するビットは'FaultEnable'パラメータによってマスクされ、このフォールトは'SwFaults'変数に反映されなくなります。この場合、モータステートマシンは FAULT ステートへ移行せず、モータは駆動し続けます。

'VdcFilt'の値が V_{dc_uv}（'DcBusLvLevel'変数によって設定可能）を下回る場合、対応する'FaultFlags'モータ変数のビット 3 がセットされます。'FaultEnable'パラメータのビット 3 がセットされると、このフォールトは'SwFaults'モータ変数に反映され、モータステートマシンは FAULT ステートへ移行して、モータの駆動を停止します。このビットがセットされない場合、'SwFaults'変数の対応するビットは'FaultEnable'パラメータによってマスクされ、このフォールトは'SwFaults'変数に反映されなくなります。この場合、モータステートマシンは FAULT ステートへ移行せず、モータは駆動し続けます。

'VdcFilt'の値が V_{dc_cov}（'CriticalOvLevel'変数によって設定可能）を上回る場合、モータは直ちに停止され、フォールトがクリアされるまでゼロベクトル[000]が適用されます。この間、「限界過電圧（Critical Over Voltage）」フォールトが報告されます。この「限界過電圧」フォールトは無効にできません。

2.1.16.6 欠相保護

MCE はモータの欠相フォールトを検出できます。モータのいずれかの相が接続されていない場合、あるいは複数のモータ巻線間で短絡が発生している場合、直流励磁電流は正しい値になりません。PARKING ステートの最後にいずれかの相電流値がI_{phase_loss}を下回った場合、欠相フォールトが確認されます。

I_{phase_loss}は'PhaseLossLevel'パラメータを使って設定できます。'PhaseLossLevel'のデフォルト値は、MCEWizard によって以下の数式を使って自動的に計算されます。

$$PhaseLossLevel = 25\% \cdot \frac{LowSpeedLim}{4096} \cdot I_{rated\_rms} \cdot \sqrt{2} \cdot R_s \cdot G_{ext} \cdot G_{int} \cdot \frac{4096}{V_{ref\_ADC}}$$

欠相フォールトが確認されると'FaultEnable'パラメータのビット[8]がセットされます。このフォールトは'SwFaults'変数に反映されて、モータステートマシンが FAULT ステートへ移行し、モータの駆動を停止します。このビットがセットされない場合、'SwFaults'変数の対応するビットは'FaultEnable'パラメータによってマスクされ、このフォールトは'SwFaults'変数に反映されなくなります。この場合、モータステートマシンは FAULT ステートへ移行せず、モータは駆動し続けます。

2.2 力率改善

力率改善（PFC）は、特定のアプリケーションにおいて政府規制の要求に応じて、入力電流波形を入力電圧と一致させる手法です。力率は 0～1 の間で変化し、負荷に含まれる有効電力と皮相電力の比率を表します。力率が高いと送電損失を抑制し、電圧制御を改善できます。政府規制では、PFC の効率を実証する条件が指定されています。

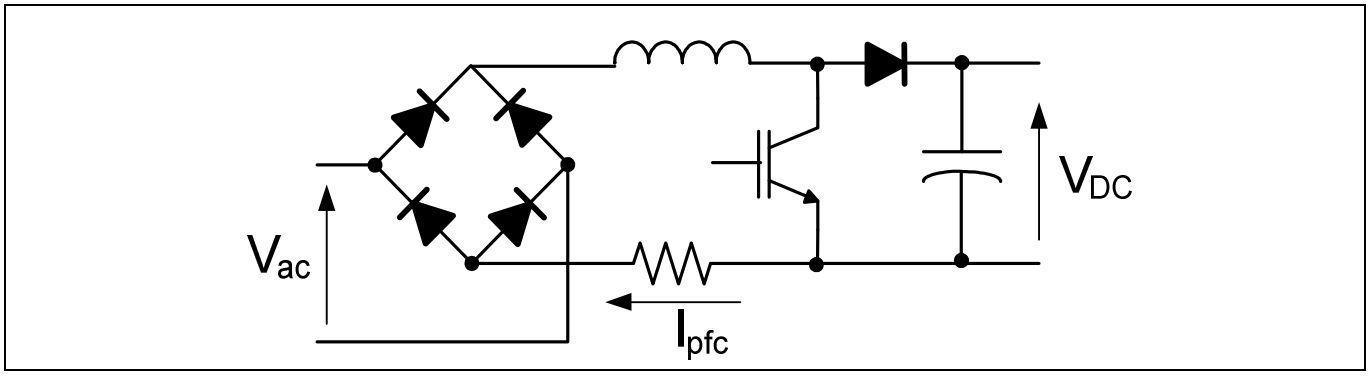


図 53 基本ブースト PFC 回路

図 53 はブースト PFC トポロジの簡略化した回路図を示します。

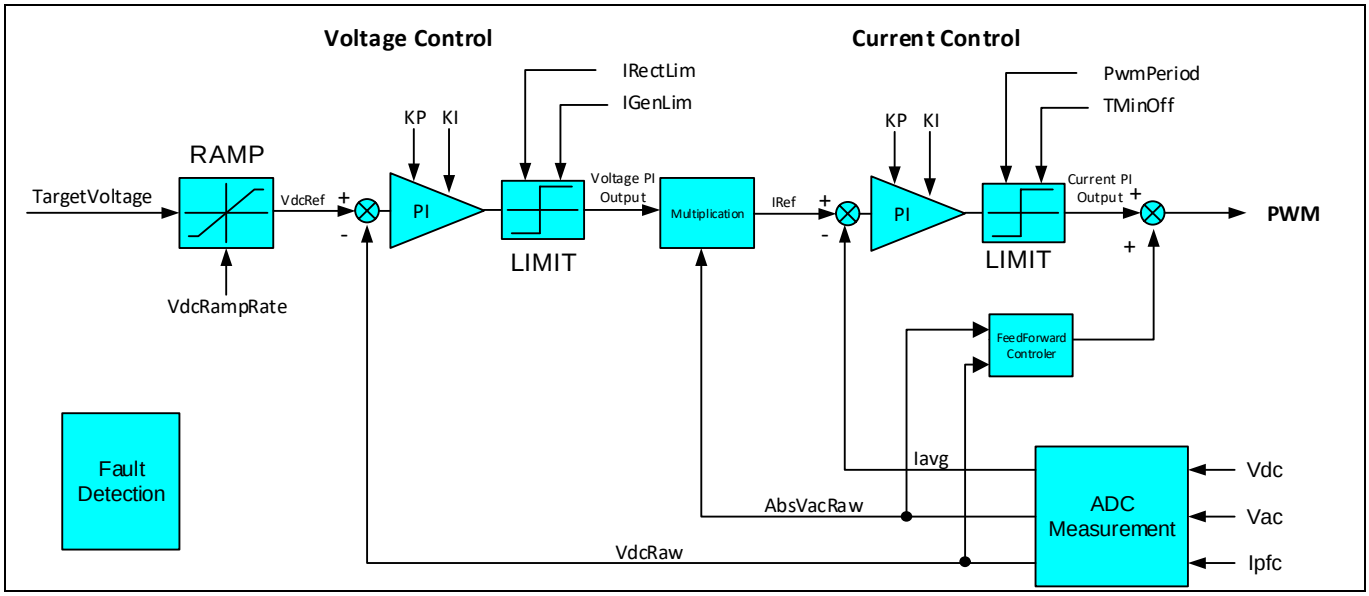


図 54　力率改善のトップレベル図

MCE の PFC は乗算器ベースの制御を行います。このため、PFC 内にはフィードフォワード成分の他に、内部電流ループと外部電圧ループの 2 つの制御ループがあります。電圧コントローラの出力に整流された AC 電圧を乗じて、電流参照値を求めます。電流コントローラの出力はフィードフォワード出力に加算され、変調コマンドを生成します。この PFC 制御方式には、インダクタ電流、AC ライン電圧、および DC バス電圧のセンシングが必要です。

MCE は 2 種類の PFC トポロジをサポートします。

1. ブーストモード PFC
2. トーテムポール PFC

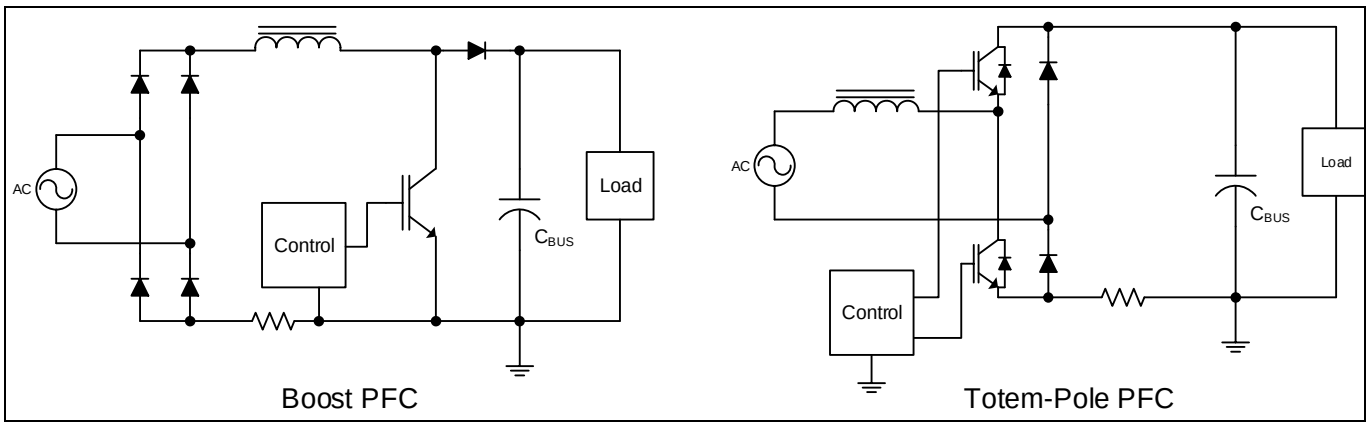


図 55　PFC トポロジ

ブースト PFC は制御しやすいため、最も一般的な PFC トポロジです。ブーストトポロジはブリッジダイオードでの損失が大きいため、あまり効率的ではありません。ブリッジダイオードによる損失を抑制することを目的としたブリッジレスの設計もありますが、ブリッジレス PFC ソリューションの大半は EMI の問題があるため、インバータエアコンなどの家電製品の用途には使用できません。トーテムポール PFC はブリッジレス PFC の一種ですが、EMI の問題はありません。高速 IGBT の開発と、SiC や GaN といった高バンドギャップ素子が商業的に入手可能になったことから、トーテムポール PFC は従来のブースト PFC に替わりうる存在として多くの注目を集めています。

AC 電圧やインダクタ電流センシングのために高価なセンサを使わずにトーテムポール PFC 制御回路を設計するのは非常に困難です。トーテムポール PFC トポロジは、その性質として、ブースト PFC に比べてより複雑な制御回路を必要とします。MCE トーテムポール設計の主な目的は制御回路ハードウエアの複雑性を最小限にすることです。AC 電圧には差動センシングを使用し、インダクタ電流センシングには DC リンクに 1 シャント抵抗を使用しています。AC 電圧の極性を検出する追加のハードウエアはありません。デジタル制御のため、DC リンクの 1 シャントからインダクタの電流情報を再構築することも可能です。

2.2.1 ステート処理

モーション制御エンジン（MCE）は、始動、停止、および起動のすべてのステートを処理するステートマシンを内蔵しています。ステートマシン機能は 1ms 毎に実行されます。合計 5 つのステートがあります。シーケンサの現在の状態は“PFC_SequencerState“変数に保存されます。

表 11　状態の説明と遷移

ステート No.	シーケンスステート	ステートの機能	遷移イベント	次のシーケンスステート
0	IDLE	コントローラの電源投入後、制御はこのステートに入ります。有効なパラメータブロックがない場合、制御はこのステートに留まります。	パラメータが正常に読み込まれたとき。	STOP
1	STOP	スタートコマンド待ち。保護のための電流および電圧測定が実行されます。	電流増幅器のオフセット計算が未実行。	OFFSETCAL
			スタートコマンド。	RUN
2	OFFSETCAL	PFC 電流センシング入力のためのオフセット計算。このステートは 256 PWM サイクル継続します。	電流オフセット計算完了。	STOP
4	RUN	正常な PFC 駆動モード。	ストップコマンド	STOP
5	FAULT	何らかのフォールトが検出されると、PFC は停止し（それまで駆動していた場合）、それ以外のステートから FAULT ステートに入ります。	“PFC_FaultClear“変数に 1 を書き込むことによりフォールトクリアコマンドとします。	STOP

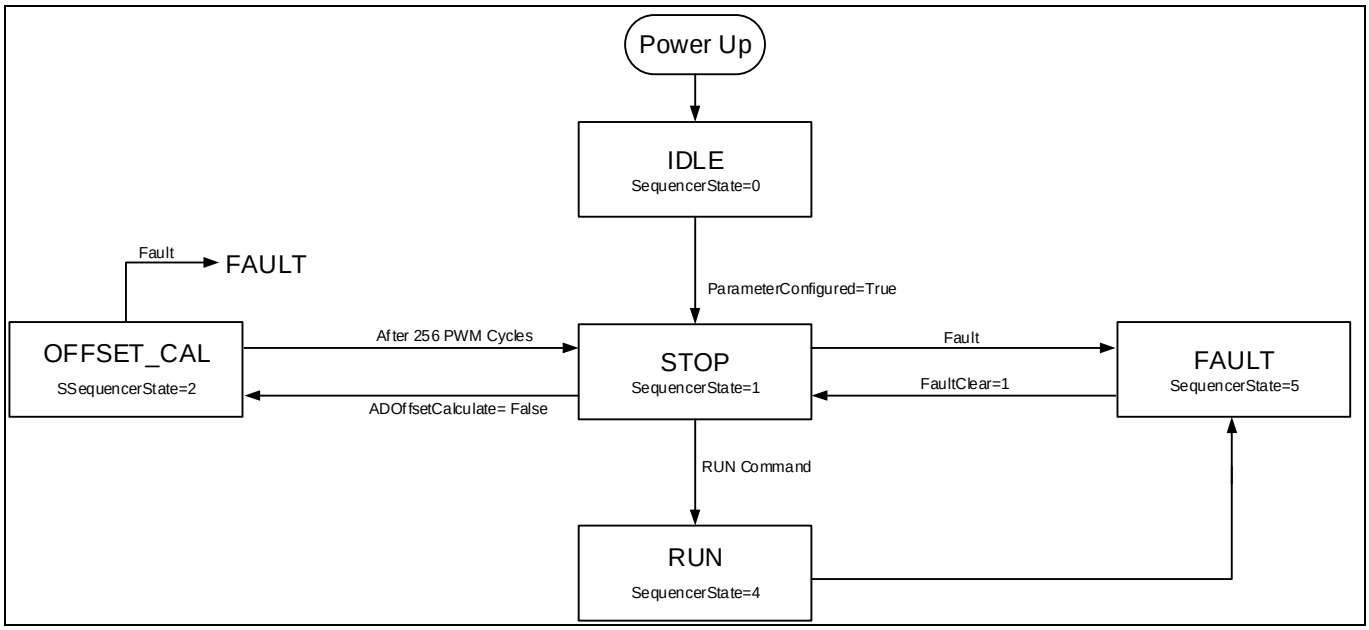
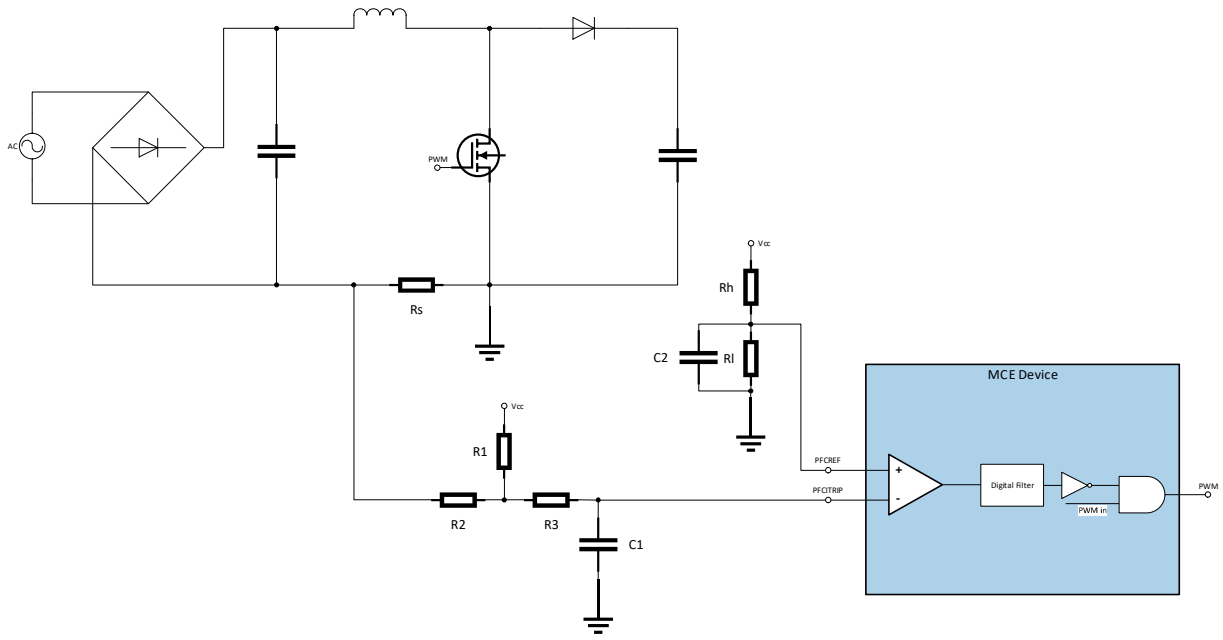


図 56　ステート処理フローチャート

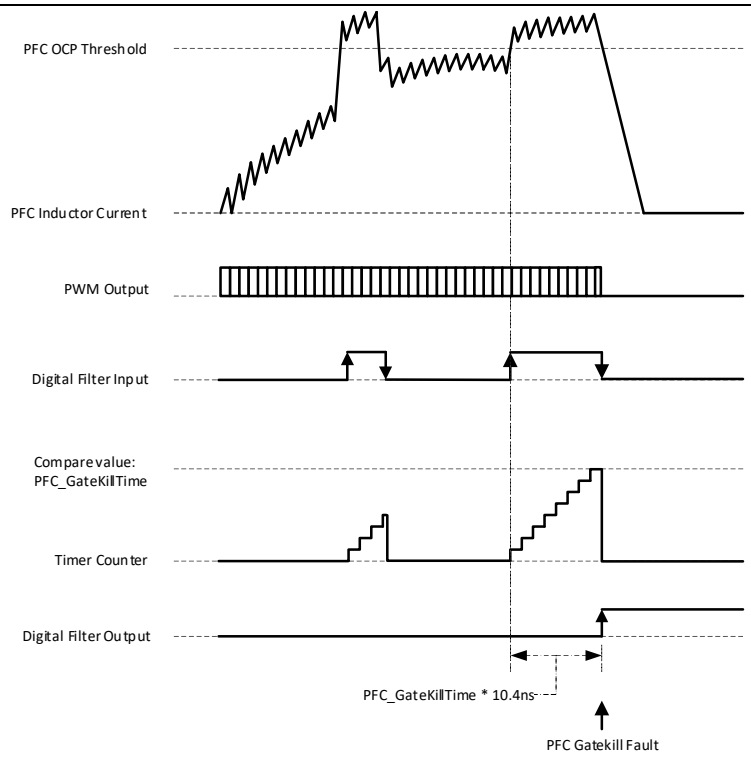
2.2.2　保護機能

2.2.2.1　PFC 過電流保護（OCP）

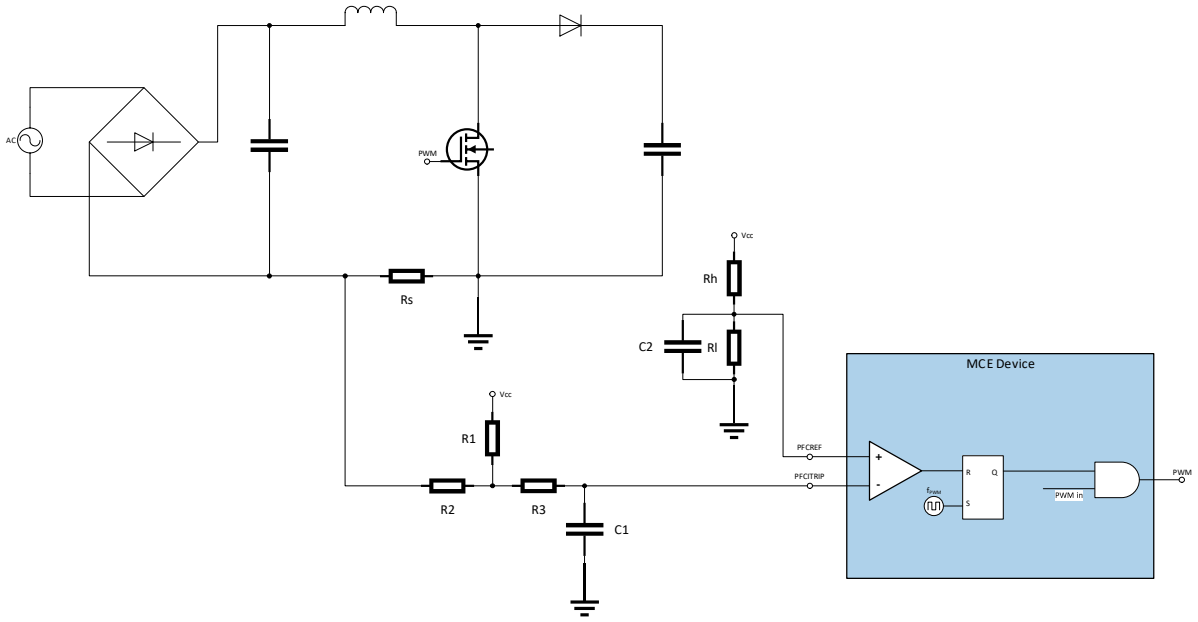
MCE には、検出された瞬時 PFC インダクタ電流をあらかじめ設定された OCP しきい値と比較する過電流保護機能があります。この機能は検出されたインダクタ電流が OCP しきい値を超えると PWM 出力を無効にします。



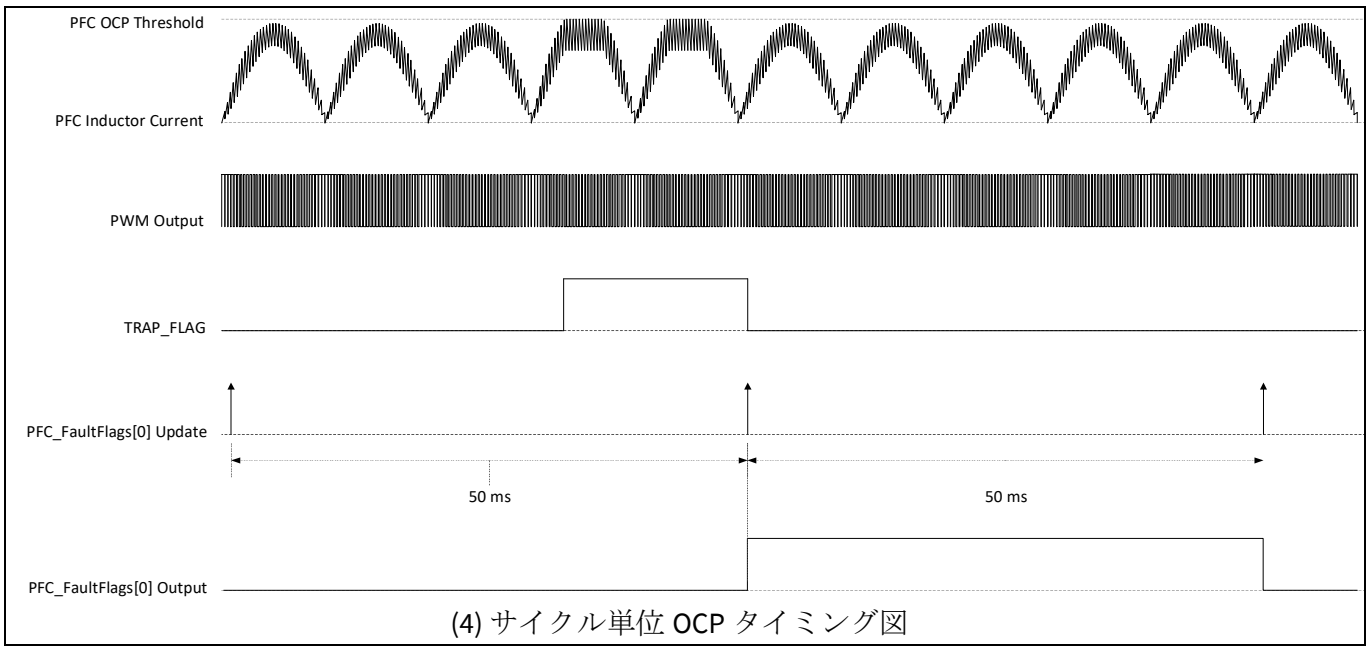
(1) ラッチオフ OCP アプリケーション図



(2) ラッチオフ OCP タイミング図



(3) サイクル単位 OCP アプリケーション図



(4) サイクル単位 OCP タイミング図

図 57　PFC ゲートキル設定

図 57 (2)および図 57 (4)に示すように、過電流トリップ機構は内部コンパレータを使用しています。トリップレベルは参照電圧値によって駆動される分圧回路を使って外部的にプログラムできます。その出力は PFCREF ピンに連結されます。OCP トリップレベルの計算を以下に示します。

iTripLevel [V] = iTripCurrentLevel[Amps]*CurrentInputScale [V/Amps] + AmplifierOffset

CurrentInputScale= Rshunt*ExternalAmpliferGain

OCP 機構には、ラッチオフ OCP とサイクル単位 OCP の 2 つのオプションがあります。

ラッチオフ OCP オプションには、図 57 (1)に示すように、高周波ノイズを回避するための内部設定可能なデジタルフィルタがあります。'PFC_GateKillTime'パラメータの値を調節することにより、ゲートキル応答時間を設定できます。過電流フォールトを発生させるには、指定した GateKillFilterTime の期間、入力信号が安定している必要があります。このフォールトは無効にすることができません。

フィルタタイマは内部コンパレータ出力によってトリガされるレベルに設定されます。図 57 (2)に示すように、インダクタ電流が指定された PFC の OCP しきい値よりも高くなると、内部コンパレータ出力はロジック HIGH になります。その結果、タイマがカウントを開始します。コンパレータ出力電圧レベルが変化してロジックゼロまで下がると、タイマはリセットされます。タイマが'PFC_GateKillTime'までカウントしても過電流状態が継続する場合、PWM 出力はプログラムされたパッシブレベルに入ります。このフォールトはマスクすることができません。このフォールトは'PFC_SwFaults'PFC 変数に反映され、PFC ステートマシンは FAULT ステートへ移行して、PFC の駆動を停止します。

このフォールトは、過電流状態がなくなったときに、'PFC_FaultClear PFC'変数に 1 を書き込むことによってクリアできます。フォールトクリア操作が成功した場合、PFC ステートマシンは STOP ステートへ移行します。PFC OCP フォールトが確認された後、PFC ステートマシンが RUN ステートに戻って PFC 操作を再起動するように働きかけるには、ユーザが'PFC_Command' = 1 をセットする必要があります。

'PFC_GateKillTime'は静的 PFC パラメータの一種で、過電流フォールト検出に対するゲートキル応答時間を設定します。この値の有効範囲は 0～960 クロックサイクルです。値が 1 であれば 1/96MHz = 10.4167ns に相当します。デフォルト値は 48 で、これは 0.5µs に相当します。

図 57 (3)および図 57 (4)に示したように、サイクル単位 OCP オプションでは、インダクタ電流が指定した PFC の OCP しきい値よりも高くなると、内部コンパレータ出力はロジック HIGH になります。その結

果、PWM 出力は直ちにロジック LOW になり、インダクタ電流が PFC の OCP しきい値よりも低くなっても、PWM サイクルの終わりまで LOW のままになります。後続の PWM サイクルの初めにインダクタ電流が PFC の OCP しきい値よりも低い場合、PWM 出力は再開します。インダクタ電流が PFC の OCP しきい値よりもまだ高い場合は、PWM 出力はロジック LOW のままとなります。OCP フォールトが発生すると、内部 TRPF ビットが自動的にセットされます。これは手動でリセットする必要があります。50ms 毎に TRPF ビットの値が読み取られ、'PFC_FaultFlags'変数のビット 0 にコピーされます。その後、TRPF ビットはクリアされます。OCP フォールトがまだ残っている場合、TRPF ビットはうまくクリアされません。PFC OCP フォールトが 1 PWM サイクルのみ発生した場合は、50ms の期間'PFC_FaultFlags'変数のビット 0 が 1 にセットされます。サイクル単位 OCP オプションが選択された場合、PFC OCP フォールトが発生したら、PFC ステートマシンは RUN ステートに留まり、'PFC_SwFaults'変数のビット 0 はセットされません。ユーザは'PFC_FaultFlags'のビット 0 の値を読み取って、PFC OCP フォールトの発生を検出できます。

2.2.2.2 DC 過電圧/低電圧保護

DC バス電圧がしきい値以下になると低電圧がセットされ、DC バス電圧がしきい値を超えると過電圧がセットされます。

DC バス電圧は PFC スイッチングサイクル毎に測定されます。測定された DC バス電圧はローパスフィルタを通過して高周波ノイズを減衰させます。測定された DC バス電圧は'PFC_VdcRaw'PFC 変数から読み取ることができます。フィルタ処理された DC バス電圧は'PFC_VdcFilt'PFC 変数から読み取ることができます。LPF の時定数は PFC PWM 周波数によって決まります。

'PFC_VdcFilt'の値が'PFC_VdcLvLevel'よりも低い場合は、'PFC_FaultFlags'PFC 変数のビット 1 がセットされます。'PFC_FaultEnable'PFC ダイナミックパラメータのビット 1 がセットされると、このフォールトは'PFC_SwFaults'PFC 変数に反映され、PFC ステートマシンは FAULT ステートへ移行して、PFC の駆動を停止します。このビットがセットされない場合、'PFC_SwFaults'変数の対応するビットは'PFC_FaultEnable'パラメータによってマスクされ、このフォールトは'PFC_SwFaults'変数に反映されなくなります。この場合、PFC ステートマシンは FAULT ステートへ移行せず、PFC は駆動し続けます。

'PFC_VdcFilt'の値が'PFC_VdcOvLevel'よりも高い場合は、'PFC_FaultFlags'PFC 変数のビット 2 がセットされます。'PFC_FaultEnable'PFC ダイナミックパラメータのビット 2 がセットされると、このフォールトは'PFC_SwFaults'PFC 変数に反映され、PFC ステートマシンは FAULT ステートへ移行して、PFC の駆動を停止します。このビットがセットされない場合、'PFC_SwFaults'変数の対応するビットは'PFC_FaultEnable'パラメータによってマスクされ、このフォールトは'PFC_SwFaults'変数に反映されなくなります。この場合、PFC ステートマシンは FAULT ステートへ移行せず、PFC は駆動し続けます。

このフォールトは、DC バス過電圧状態または低電圧状態がなくなったときに、'PFC_FaultClear'PFC 変数に 1 を書き込むことによってクリアできます。フォールトクリア操作が成功した場合、PFC ステートマシンは STOP ステートへ移行します。

2.2.2.3 AC 過電圧/低電圧保護

PFC への AC 入力電圧がしきい値を超えると AC 過電圧フォールトがセットされ、PFC への AC 入力電圧がしきい値を下回ると AC 低電圧フォールトがセットされます。

AC 入力電圧は PFC スイッチングサイクル毎に測定されます。AC 入力電圧の RMS 値は PFC ステートマシンの更新ごとに計算されます（デフォルト値は 1ms）。

AC 過電圧フォールトは、VAC RMS の計算値を'PFC_VacOvLevel'値と比較することによって検出されます。VAC RMS の値が'PFC_VacOvLevel'よりも高い場合は、'PFC_FaultFlags'PFC 変数のビット 5 がセットされます。'PFC_FaultEnable'PFC ダイナミックパラメータのビット 5 がセットされると、このフォールトは'PFC_SwFaults'PFC 変数に反映され、PFC ステートマシンは FAULT ステートへ移行して、PFC の駆動を

停止します。このビットがセットされない場合、‘PFC_SwFaults’変数の対応するビットは‘PFC_FaultEnable’パラメータによってマスクされ、このフォールトは‘PFC_SwFaults’変数に反映されなくなります。この場合、PFC ステートマシンは FAULT ステートへ移行せず、PFC は駆動し続けます。

AC 低電圧フォールトは、VAC RMS の計算値を‘PFC_VacLvLevel’値と比較することによってチェックします。VAC RMS の値が‘PFC_VacLvLevel’を下回る場合は、‘PFC_FaultFlags’PFC 変数のビット 4 がセットされます。‘PFC_FaultEnable’PFC ダイナミックパラメータのビット 4 がセットされると、このフォールトは‘PFC_SwFaults’PFC 変数に反映され、PFC ステートマシンは FAULT ステートへ移行して、PFC の駆動を停止します。このビットがセットされない場合、‘PFC_SwFaults’変数の対応するビットは‘PFC_FaultEnable’パラメータによってマスクされ、このフォールトは‘PFC_SwFaults’変数に反映されなくなります。この場合、PFC ステートマシンは FAULT ステートへ移行せず、PFC は駆動し続けます。

このフォールトは、AC 入力過電圧状態または低電圧状態がなくなったときに、’PFC_FaultClear’PFC 変数に 1 を書き込むことによってクリアできます。フォールトクリア操作が成功した場合、PFC ステートマシンは STOP ステートへ移行します。

2.2.2.4 入力周波数保護

このフォールトは PFC への AC 入力周波数値が設定値と異なる場合にセットされます。

AC 入力周波数の最大制限値および最小制限値は、選択された公称 AC 入力周波数に基づいて MCEWizard によって自動的に設定されます。AC 入力周波数の公称値として 50Hz を選択した場合、実際の AC 入力周波数の有効範囲は 45～55Hz となります。AC 入力周波数の公称値として 60Hz を選択した場合、実際の AC 入力周波数の有効範囲は 55～65Hz となります。

AC 入力周波数の最小制限値は PFC ステートマシンの更新毎に確認されます（デフォルト値は 1ms）。測定された正または負の半サイクルが最小制限値よりも低い場合、‘PFC_FaultFlags’PFC 変数のビット 3 がセットされます。このフォールトはマスクすることができないため、フォールトは’PFC_SwFaults’PFC 変数に反映され、PFC ステートマシンは FAULT ステートへ移行して、PFC の駆動を停止します。

AC 入力周波数の最大制限値は、PFC PWM サイクル毎に実行されるゼロクロス検出のプロセス中に確認されます。各 PFC PWM サイクル毎に、カウンタは増分され、最大制限値と比較されます。カウンタの値が最大制限値を上回る場合、ゼロクロスは有効な半サイクル時間の最大値の中には見つからなかったということを意味します。このフォールトがセットされた場合、‘PFC_FaultFlags’PFC 変数のビット 3 がセットされます。このフォールトは‘PFC_SwFaults’PFC 変数に反映され、PFC ステートマシンは FAULT ステートへ移行して、PFC の駆動を停止します。

このフォールトは、AC 入力周波数フォールトがなくなったときに、’PFC_FaultClear’PFC 変数に 1 を書き込むことによってクリアできます。フォールトクリア操作が成功した場合、PFC ステートマシンは STOP ステートへ移行します。

注: モータ制御のどのフォールト中も、PFC は停止されます。

2.3 ユーザモード UART

ユーザモード UART 通信は、モータ制御アプリケーション用のシンプルで信頼できるスケーラブルな通信プロトコルとして設計されています。このプロトコルはシンプルなため低価格仕様のマイクロコントローラにも容易に実装することができ、モータ制御用のマスタとして機能させることができます。一部の産業用ファン/ポンプアプリケーションで必要とされるネットワーキング（同一ネットワーク上で最大 15 ノード）をサポートします。各 UART コマンドは 1ms 毎に処理されます。
ユーザが、カスタマイズされた UART 通信プロトコルの実装を検討中の場合は、セクション 2.6.11.3 に説明がある設定可能な UART ドライバの方法を使って行うことができます。

2.3.1 ボーレート

MCE はユーザモード UART で以下のボーレート設定をサポートしています。

2400bps、9600bps、19200bps、67500bps、115200bps、および 230400bps。

2.3.2 データフレーム

データフレームのフォーマットを図 58 に示します。リトルエンディアンフォーマットが使われている点に注意してください。

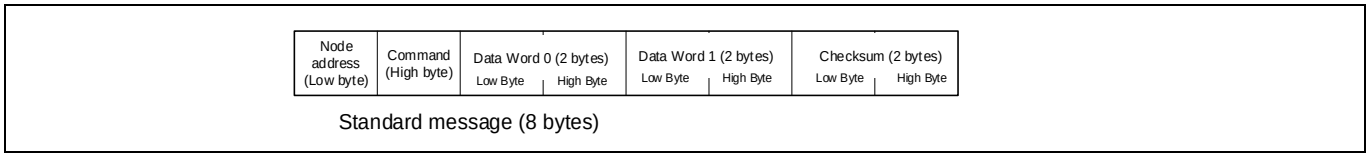


図 58 UART データフレーム

2.3.3 ノードアドレス

ノードアドレスはデータフレームの最初のバイトです。同一ネットワーク内で 1 台のマスタが複数のスレーブを制御できるように設計されています。スレーブノードにはそれぞれ一意のノード ID が付けられています。スレーブは同じ ID を持つメッセージのみを認識したり応答したりできます。異なる用途のために定義された 2 つの同報通信アドレスが用意されています（0x00 および 0xFF）。メッセージが address=0x00 で受信された場合、すべてのスレーブがコマンドを実行しますが、マスタへの応答は送信しません。これは複数のスレーブを持つネットワークの場合に便利です。マスタはすべてのスレーブを同時に制御することができるため、たとえば、メッセージを 1 件送るだけですべてのモータをオンにするといったことが可能になります。フレームが address=0xFF で受信された場合は、スレーブはコマンドを実行し、またマスタへの応答も送信します。これは、1 対 1 構成で、マスタがスレーブのノードアドレスを知らない場合や知る必要がない場合に便利です。

表 12 ノードアドレスの定義

ノードアドレス	コマンド
0x00	すべてのノードがコマンドを受信および実行しますが、応答はしません。
0x01～0x0F	同じアドレスを持つノードのみがコマンドを実行し、マスタに応答します。
0x10～0xFE	予約済み
0xFF	すべてのノードがコマンドを受信および実行し、マスタに応答します。1 対 1 構成の場合にのみ使用可能です。同一ネットワークに複数のノードが接続されている場合は、競合が生じます。

2.3.4 リンク切れ保護

リンク切れ保護とは、UART 通信が一定時間ない場合に、モータを停止させることです。一部のアプリケーションでは、メインコントローラはモータのコントローラとの通信を維持します。通信切断やライン切れの場合には、保護のためにモータを停止することが望ましいと言えます。この保護機能の有効/無効設定およびリンク切れタイムアウトの設定は MCEWizard で行います。

2.3.5 コマンド

UART コマンドはデータフレームの 2 番目のバイトです。ビット[6:0]はコマンドコードを指定します。ビット[7]はデータフレームの方向を示す指示ビットです。マスタが送信するすべてのデータフレームでは、ビット 7 がクリアされている（=0）必要があります。またスレーブが送信するすべての応答データフレームでは、ビット 7 がセットされている（=1）必要があります。

表 13 UART コマンド定義

コマンド（ビット[6:0]）	説明
0	ステータス読み取り
1	フォールトフラグのクリア要求
2	制御入力モードの選択
3	モータ制御目標速度の設定
4	使用しない。スレーブはマスタに応答しない。
5	レジスタ読み取り
6	レジスタ書き込み
7～31	使用しない。スレーブはマスタに応答しない。
32	パラメータセットの読み込みまたは保存
33～127	使用しない。スレーブはマスタに応答しない。

2.3.6 チェックサム

チェックサムは 16 ビットフォーマットで、以下のように計算されます。

[コマンド：ノードアドレス] + データワード 0 + データワード 1 + チェックサム = 0x0000

ユーザ UART インターフェイスにチェックサムワードを送信する際は、図 58 に示すように、リトルエンディアンフォーマットが使われる点に注意してください。

チェックサムの計算例：
入力：ノードアドレス = 1、コマンド = 2、データワード 0 = 0x1122、およびデータワード 1 = 0x3344

[コマンド：ノードアドレス] = 0x0201

チェックサム = -1 x (0x0201 + 0x1122 + 0x3344) = 0xB999

データフレーム： 0x01（ノードアドレスバイト）、0x02（コマンドバイト）、0x22（データワード 0 の下位バイト）、0x11（データワード 0 の上位バイト）、0x44（データワード 1 の下位バイト）、0x33（データワード 1 の上位バイト）、0x99（チェックサムワードの下位バイト）、0xB9（チェックサムワードの上位バイト）

2.3.7 UART メッセージ

2.3.7.1 ステータス読み取り：コマンド = 0x00

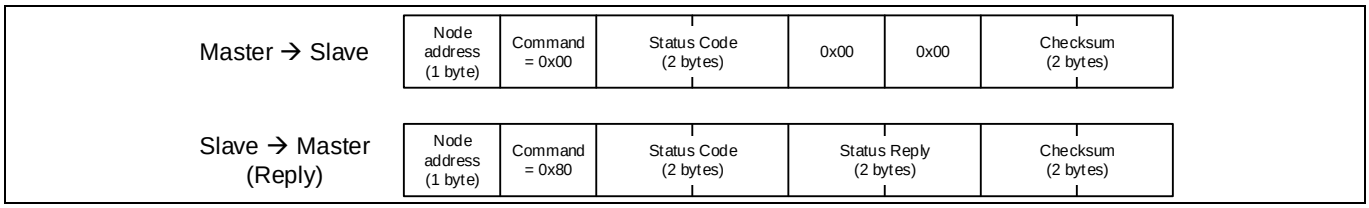


図 59 ステータス読み取りコマンド

表 14 ステータスコードおよびステータス応答

ステータスコード	ステータス応答
0x0000	Fault Flags（フォールトフラッグ）
0x0001	Motor Speed（モータ速度）
0x0002	Motor State（モータステート）
0x0003	Node ID（ノード ID）
0x0004～0xFFFF	0x0000

フォールトのクリア：コマンド = 0x01

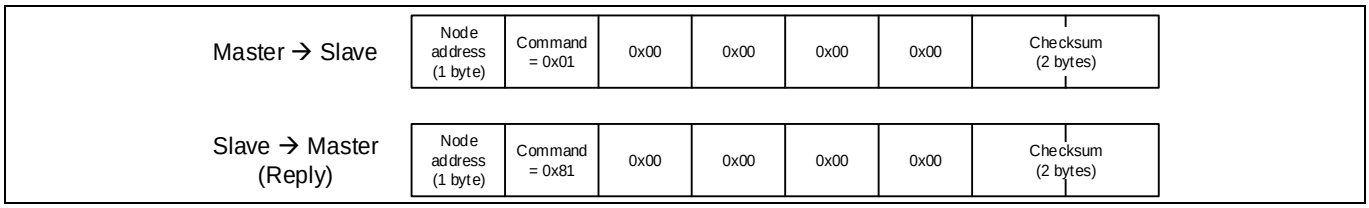


図 60 フォールトのクリアコマンド

2.3.7.2 制御入力モードの変更：コマンド = 0x02

図 61 制御入力モードコマンド

2.3.7.3 Motor Control: コマンド = 0x03

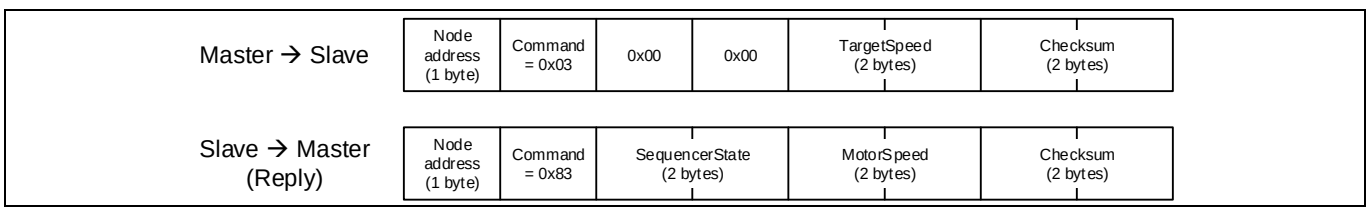


図 62 モータ制御コマンド

注： TargetSpeed=0: モータ停止、TargetSpeed≠0: モータ始動

2.3.7.4 レジスタ読み取り：コマンド = 0x05

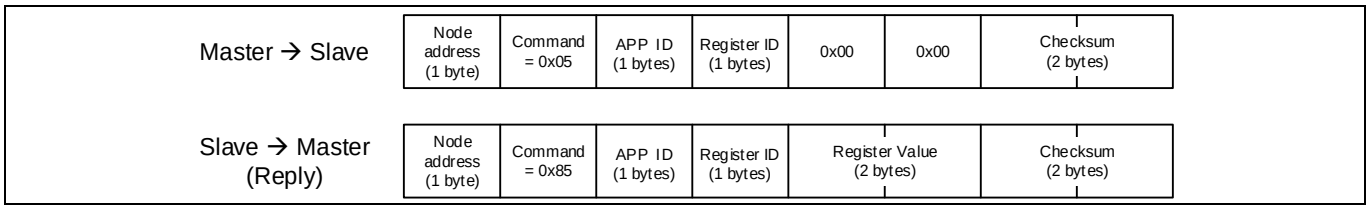


図 63 レジスタ読み取りコマンド

2.3.7.5 レジスタ書き込み：コマンド = 0x06

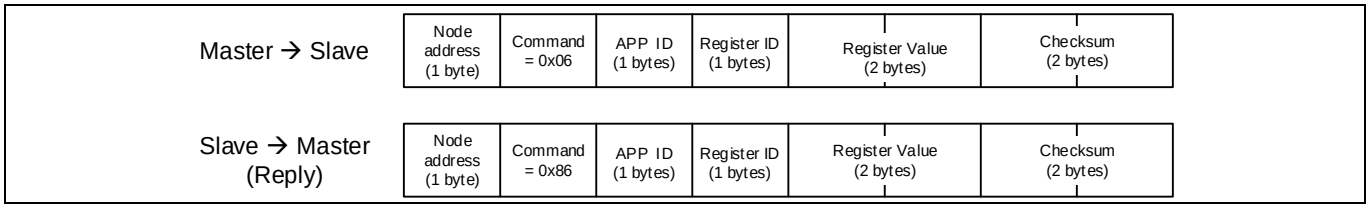


図 64 レジスタ書き込みコマンド

2.3.7.6 パラメータの読み込みおよび保存：コマンド = 0x20

パラメータ読み込みコマンドは、1 ページのすべてのパラメータを専用 RAM 位置に読み込みます。

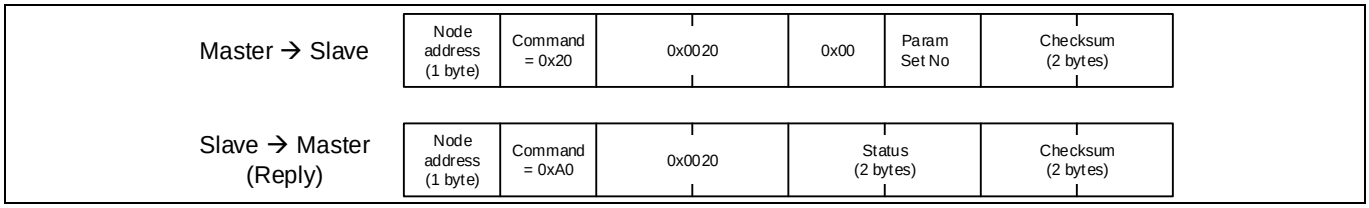


図 65 パラメータ読み込みコマンド

パラメータ保存コマンドはすべてのパラメータを 1 フラッシュページに保存します。

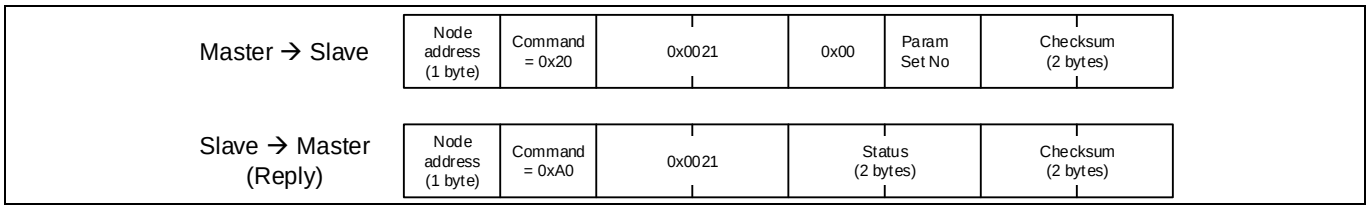


図 66 パラメータ保存コマンド

2.3.8 同一ネットワークに複数ノードを接続

複数の MCE を同一の UART ネットワークに接続できます。詳細は、図 67 を参照してください。
各 MCE ノードの TXD ピンは、同じワイヤに接続する前にショットキーダイオードに接続する必要があります。また、マスタコントローラ側には 4.7kOhm のプルアップ抵抗が必要です。

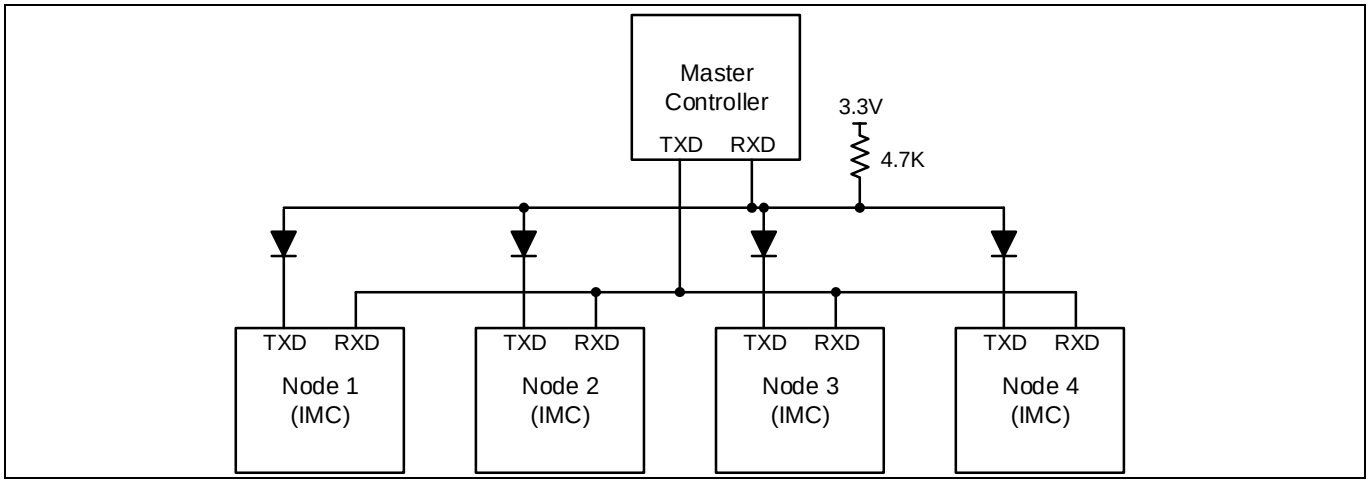


図 67　UART ネットワーク接続

2.3.9　UART 送信遅延

設定可能な遅延（‘InterfaceConf0’パラメータのビット[15:8]）を、ホストからのメッセージの受信と応答メッセージの送信の間に挿入できます。

2.4　JCOM チップ間通信

JCOM インターフェイスは、MCE を駆動するモータ制御コア（以降は T コアと表記）と統合 MCU（以降は A コアと表記）の間でデュアルコア製品のためのポイント間双方向通信を行うための手段を提供するように設計されています。JCOM インターフェイスは内部シリアルポートを使用します。JCOM プロトコルは通信中、1 台のマスタと 1 台のスレーブを使用することを想定しています。JCOM インターフェイスは‘InterfaceConf1’パラメータのビットフィールド[2:0]を使用して有効にできます。

2.4.1　操作モード

JCOM インターフェイスはマスタとスレーブの間で非同期モードをサポートします。

2.4.1.1　非同期モード

非同期モードでは、A コア（MCU）がマスタとして、T コア（MCE ベースのモータ制御）がスレーブとして動作します。すべての通信動作はマスタによって開始されます。

スレーブ側からは、T コアからの応答が最小限の遅延で処理されるように、JCOM インターフェイスドライバが割り込み駆動されます。受信 FIFO に十分なデータが蓄積されたら、JCOM 割り込みハンドラがトリガされます。受信されたフレームは解析されて、メッセージペイロードが抽出されます。メッセージオブジェクト（MO）番号に基づいて、コマンドおよび応答プロトコルによって関連する行動が実行されます。その後、応答フレームが構築されて、送信 FIFO に送られます。

2.4.2　ボーレート

JCOM インターフェイスのボーレートは起動時または駆動中に設定できます。有効範囲は 6.1Kbps～6Mbps です。デフォルトのボーレートは 1Mbps です。

T コアの JCOM インターフェイスが、A コアと T コアのボーレート構成の不一致によって 3 回以上フレームエラーを生じた場合、T コアの JCOM インターフェイスのボーレートは自動的にデフォルト値（1Mbps）にリセットされます。

2.4.3 メッセージフレーム構造

各 JCOM メッセージフレームは以下のフィールドで構成されています。送信シーケンスは左から右とします。以下の図 68 に、JCOM メッセージフレーム構造の詳細を示します。

Flag	Seq	Res	Message					CRC	Flag
			MO	Data[0]	Data[1]	Data[2]	Data[3]		
1 byte (0x7E)	2 bit	2 bit	4 bit	4 bytes				1 byte	1 byte (0x7E)

図 68 JCOM メッセージフレーム構造

Flag： フレームの始点および終点の表示。

Seq： このシーケンス番号は誤ったシーケンスフォールトを検出するために使われます。通常操作中、Seq 番号はフレーム毎に増加され、受信側でチェックされます。Seq 番号が一致しない場合は、フレーム全体が無視され、応答は送信されません。

Res： 将来使用するため予約済み。

MO： このメッセージオブジェクト番号はデータの解釈方法を定義します。

Data[x]： このデータフィールドにはメッセージのペイロードが入ります。

CRC: CRC バイトは MO 番号を含むメッセージフィールドに対して計算されます。CRC チェックが不合格の場合は、フレーム全体が無視され、応答は送信されません。

2.4.4 コマンドおよび応答プロトコル

コマンドおよび応答プロトコルは JCOM インターフェイスが非同期モードで作動するときに使用されます。メッセージにはメッセージオブジェクト番号と 4 データバイトが入っています。以下のメッセージ構造図の「方向」欄で、「DS」はマスタ（A コア）からスレーブ（T コア）への通信を、「US」はスレーブ（T コア）からマスタ（A コア）への通信を意味します。マスタから送信されたコマンドフレームがスレーブによって無事に受信され、CRC チェックに合格した場合、スレーブから対応する応答フレームが送信されます。マスタから送信されたコマンドフレームが Seq 番号不一致のため同期できなかった場合、または CRC チェックに不合格であった場合、スレーブはコマンドフレーム全体を無視し、応答を送信しません。これらのフォールトに対応するため、マスタ側に何らかのタイムアウト回復機構を設けることが推奨されます。以下の表 15 に、各 MO 番号に対応する機能を示します。

表 15 メッセージオブジェクト機能表

メッセージオブジェクト	機能
0	ステートマシン照会； 実行時間および CPU 負荷照会。
1	システム構成保護； T コアのリセット； 静的パラメータアクセス； ブートモード設定； JCOM ボーレート設定。

メッセージオブジェクト	機能
6	パラメータ取得。
7	パラメータ設定。
その他	将来使用するため予約済み。

2.4.4.1 メッセージオブジェクト：0

以下の図 69 に、MO を 0 にセットした場合のメッセージ構造の詳細を示します。MO = 0 のとき、data[0] にはステータスが要求されているオブジェクトのタイプを表すステータスバイトが入っています。

方向	Data[0]	Data[1]	Data[2]	Data[3]	コメント
DS	Status	x	x	x	Status = 0：SM0（モータ）および SM1（PFC）ステートマシンのステート番号を返します。 Status = 1：システムタスクおよび CPU_Load の実行時間を返します。
US	SM0 の状態	SM1 の状態	0xFF	0xFF	Status = 0
US	exe_sys		cpu_load		Status = 1

図 69　メッセージ構造（MO = 0）

2.4.4.1.1 ステートマシン照会

コマンドフレームで“status byte = 0“の場合、マスタによってモータおよび PC ステートマシンの関連するステート番号が要求されます。応答フレームには、データ[0]にモータステートマシンのステート番号（‘SequencerState’）を、またはデータ[1]に PFC ステートマシンのステート番号（‘PFC_SequencerState’）を含むこととします。

2.4.4.1.2 実行時間および CPU 負荷照会

コマンドフレームで“status byte = 1“の場合、systick ISR にスケジュールされたシステムタスクの実行時間（通常 1ms 毎）および CPU 負荷が要求されます。応答フレームには、システムタスクに対する実行時間ワード（1 カウントは 0.33µs）をデータ[0]（実行時間ワードの下位 8 ビット）およびデータ[1]（実行時間ワードの上位 8 ビット）に、また CPU_Load ワード（1 カウントは 0.1%）をデータ[2]（CPU_Load ワードの下位 8 ビット）およびデータ[3]（CPU_Load ワードの上位 8 ビット）に含むこととします。

2.4.4.2 メッセージオブジェクト：1

以下の図 70 に、MO を 1 にセットした場合のメッセージ構造の詳細を示します。MO = 1 のとき、コマンドフレームはデータ[0]とデータ[1]に Command ワードを、またデータ[2]とデータ[3]に Value ワードを含みます（該当する場合）。応答フレームには、受信が成功したことを確認するため、データ[0]とデータ[1]に同じ Command ワードを、またデータ[2]とデータ[3]に同じ Value ワードを含むこととします。

方向	Data[0] Data[1]	Data[2] Data[3]	コメント
	コマンド	値	
システム構成			
DS	0x0000	p	構成保護： p = 0：保護 0 < p < 3：次の p コマンドに対して保護なし
DS	0x0001	0	リセット（即時）
DS	0x0002	a	静的パラメータアクセス： a = 0：無効 a = 1：有効
DS	0x00BD	(~bmd<<8)+bmd	ブートモードを設定
JCOM 構成			
DS	0x0100	ボーレート	JCOM ボーレートを設定
パラメータハンドラコマンド			
DS	0x0200	x	コーヒレントなパラメータ処理を有効にします
DS	0x0201	x	コーヒレントなパラメータ処理を無効にします
DS	0x0202	x	パラメータをコーヒレントに設定
応答			
US	コマンド	値	スレーブからのアクノリッジ

図 70　メッセージ構造（MO = 1）

2.4.4.2.1　システム構成保護

システム構成を変更するには、安全確保のために 2 段階のアンロック手順を経る必要があります。この操作には、T コアのリセット、静的パラメータへのアクセス、およびブートモードの設定が含まれます。

最初に、マスタから Command = 0x0000 および Value = p のコマンドフレーム（MO = 1）を送信して、次の p コマンドの保護を解除します。p は 1 または 2 にセットできます。

次に、マスタからこれらのシステム構成関連コマンドのいずれかを持つコマンドフレーム（MO = 1）を送信して、システム構成を変更します。

2.4.4.2.2　T コアのリセット

A コアは以下の手順で T コアのリセット要求を実行できます。

最初に、マスタから Command = 0x0000 および Value = 1 のコマンドフレーム（MO = 1）を送信して、次の 1 コマンドの保護を解除します。

次に、マスタから Command = 0x0001 および Value = 0 のコマンドフレーム（MO = 1）を送信します。このフレームを受信すると、T コアは応答 US フレームなしに直ちに自分自身をリセットします。

2.4.4.2.3 静的パラメータへのアクセス

これらの静的タイプのパラメータへの書き込みは、デフォルトでは許可されていません。静的タイプのパラメータへの書き込みアクセスを行うには、2 段階のアンロック手順を経る必要があります。この手順を経ずに静的タイプのパラメータに書き込もうとしても、書き込むことはできません。

最初に、マスタから Command = 0x0000 および Value = 1 のコマンドフレーム（MO = 1）を送信して、次の 1 コマンドの保護を解除します。

次に、マスタから Command = 0x0002 および Value = 1 のコマンドフレーム（MO = 1）を送信し、これらの静的タイプのパラメータへの書き込みアクセス権を得られるようにします。

すると、マスタは MO = 7 のコマンドフレームを使ってこれらの静的タイプのパラメータに書き込む権利を持ちます。書き込み操作が終了したら、同じ 2 段階ロック手順を使って、これらの静的タイプのパラメータへの書き込みアクセスを無効にしておくことを推奨します。

最初に、マスタから Command = 0x0000 および Value = 1 のコマンドフレーム（MO = 1）を送信して、次の 1 コマンドの保護を解除します。

次に、マスタから Command = 0x0002 および Value = 0 のコマンドフレーム（MO = 1）を送信し、これらの静的タイプのパラメータへの書き込みアクセス権を無効にします。

2.4.4.2.4 ブートモード設定

デフォルトでは、T コア（MCE）はアプリケーションモードで作動します。A コアは以下の手順で、MCE を構成モード（BMD = 0xCD）またはブートローダモード（BMD = 0x5D）に変更することを要求できます。

最初に、マスタから Command = 0x0000 および Value = 1 のコマンドフレーム（MO = 1）を送信して、次の 1 コマンドの保護を解除します。

次に、マスタから Command = 0x00BD および Value = 0x32CD のコマンドフレーム（MO = 1）を送信してブートモードを構成モードにセットするか、Value = 0xA25D を送信してブートモードをブートローダモードにセットします。

2.4.4.2.5 JCOM ボーレート設定

マスタは、Command = 0x0100 および Value = 希望するボーレート（bps）/100 のコマンドフレーム（MO = 1）を送信して、スレーブの JCOM インターフェイスのボーレートの変更を要求できます。

2.4.4.3 メッセージオブジェクト：6

2.4.4.3.1 パラメータ取得

以下の図 71 に、MO を 6 にセットした場合のメッセージ構造の詳細を示します。各パラメータは、セクション 3 に示すように、一意の App ID および Index 番号を使ってアドレスできます。MO = 6 の場合、コマンドフレームには特定のパラメータまたは変数のデータ[0]に App ID バイトを、データ[1]に Index バイトを含めることができます。応答フレームには、データ[0]に要求したパラメータまたは変数と同じ App ID バイトを、データ[1]に Index バイト、データ[2]およびデータ[3]に Value ワードを含めることとします。

方向	Data[0]	Data[1]	Data[2]	Data[3]	コメント
DS	App ID	Index	0x0000		パラメータ取得
応答					
US	App ID	Index	Value		要求されたパラメータを送信

図 71　メッセージ構造（MO = 6）

2.4.4.4　メッセージオブジェクト：7

2.4.4.4.1　パラメータ設定

以下の図 72 に、MO を 7 にセットした場合のメッセージ構造の詳細を示します。MO = 7 の場合、コマンドフレームにはデータ[0]に特定のパラメータまたは変数の App ID バイトを、データ[1]に Index バイトを、データ[2]およびデータ[3]に Value ワードを含めることができます。応答フレームには、操作が成功したことを確認するため、データ[0]には要求したパラメータまたは変数と同じ App ID を、データ[1]には同じ Index バイトを、データ[2]およびデータ[3]には同じ Value ワードを含めることとします。

方向	Data[0]	Data[1]	Data[2]	Data[3]	コメント
DS	App ID	Index	Value		パラメータを設定
応答					
US	App ID	Index	Value		確認のためにパラメータを返送

図 72　メッセージ構造（MO = 7）

2.5　マルチパラメータプログラミング

2.5.1　パラメータページのレイアウト

iMOTION™ 製品では、4k バイトのフラッシュメモリを使用して制御パラメータデータを保存します。合計 16 個のパラメータブロックがあり、各パラメータブロックのサイズは 256 バイトです。最大 15 個のマルチパラメータブロックをプログラムして、様々なモータタイプやハードウエアをサポートします。1 個のブロックはシステムパラメータを保存するために予約されています。

アクティブなパラメータセットはパラメータセット番号によって指定されます。これは MCEWizard を使って指定できます。パラメータ値を含んだ MCEWizard の出力（*.txt）は MCEDesigner を使ってパラメータブロックにプログラムできます。MCEWizard 出力ファイルには指定したパラメータセット番号が入っています。MCEDesigner は、対応するパラメータブロックにパラメータ値を読み込みます。各パラメータブロックは複数回更新できます。

モータ制御機能のみのシステムの場合、各パラメータセットは 1 個のパラメータブロックを使用します。この場合、有効なパラメータセット ID は 0～14 の範囲になります。

モータ制御機能および PFC 機能を持つシステムの場合、各パラメータセットは 2 個の連続するパラメータブロックを使用します。モータ制御パラメータセットは選択されたパラメータブロックに保存され、

PFC パラメータセットは直後のパラメータブロックに保存されます。この場合、有効なパラメータセット ID は 0、2、4、6、8、10、12、および 14 になります。

2.5.2 パラメータブロック選択

MCE は 4 つの異なる方法でのパラメータブロック選択をサポートしています。

- 直接選択：：ParPageConf[3:0] =0
- UART 制御：ParPageConf[3:0] =1
- アナログ入力：ParPageConf[3:0] =2
- GPIO ピン：: ParPageConf[3:4] =3

パラメータブロック選択の入力設定は MCEWizard から“ParPageConf“パラメータを更新して行います。

注： ピンの使用状況が異なるため、すべての iMOTION™ デバイスで 4 つのパラメータブロック選択方法のすべてを使用できるわけではありません。使用可能なパラメータブロックの選択方法については、それぞれのデバイスのデータシートを参照してください。

2.5.2.1 直接選択

パラメータブロックの選択は“ParPageConf [7:4]“パラメータのビットフィールド値によって行われます。“ParPageConf [7:4]“パラメータのビットフィールド値は MCEWizard から更新できます。

2.5.2.2 UART 制御

フラッシュから RAM へのパラメータブロックの読み込みと、RAM からフラッシュへのパラメータセットの保存を行うための特定の UART メッセージが定義されています。メッセージのフォーマットについては、セクション 2.3.7.6 を参照してください。

2.5.2.3 アナログ入力

パラメータブロックはアナログ入力値に基づいて選択されます。MCE はパラメータセット選択に対するアナログ入力として“PARAM“ピンを使用します。アナログ入力に基づくパラメータページ選択間のマッピングは以下のとおりです。

$$ParameterBlock = Integer\left\{\left(\frac{AnalogInput}{Vadcref} * 15\right)\right\}$$

例： AnalogInput = 1.2V および V_{adcref} =3.3V のとき、ParameterBlock = 5

注： パラメータブロックの最大値は 14 です。

2.5.2.4 GPIO ピン

パラメータブロックは 4 本の GPIO ピンに基づいて選択されます。パラメータセットの選択に使用する GPIO ピンには、“PAR0“、“PAR1“、“PAR2“、および“PAR3“という名前が付けられています。GPIO ピンに基づくパラメータページのマッピング状況を表 16 に示します。

表 16　GPIO に対するパラメータページ選択

GPIO 入力				パラメータブロック
PAR3	PAR2	PAR1	PAR0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	14

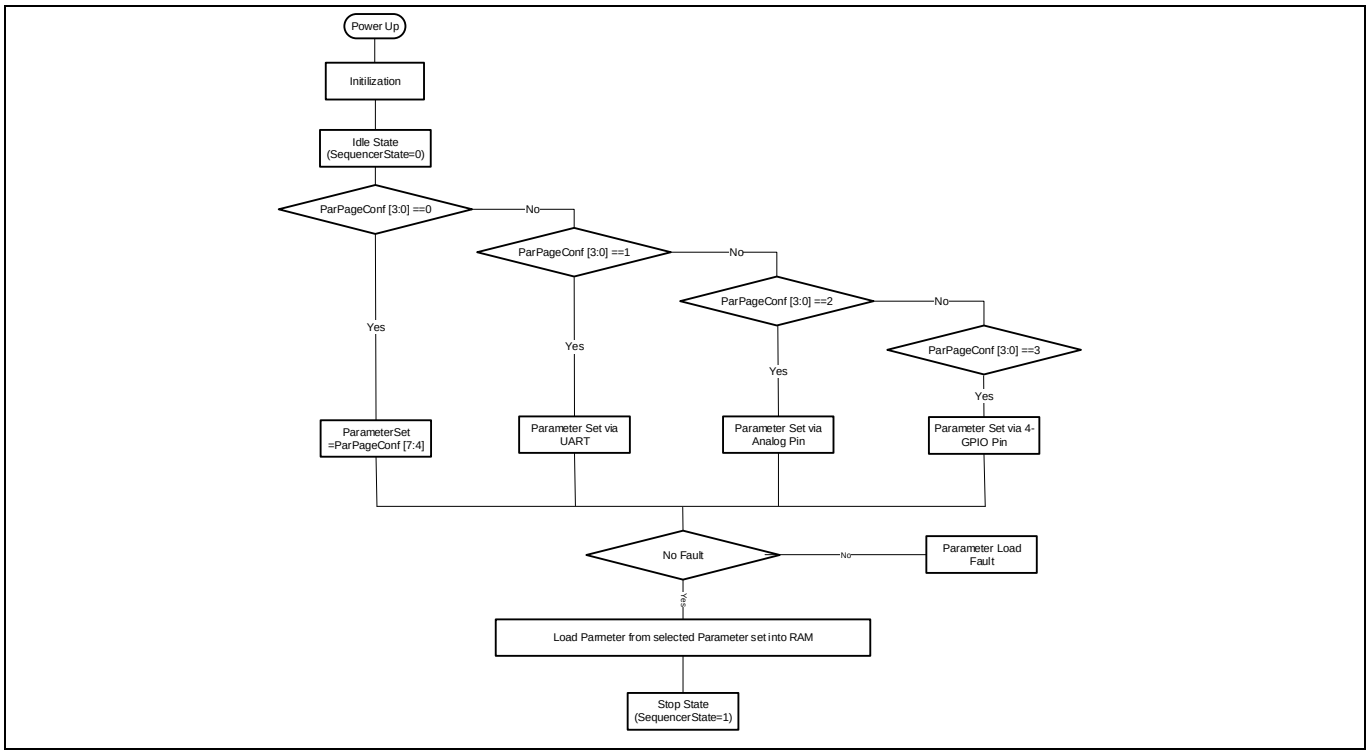


図 73　パラメータ読み込み手順

2.5.3 パラメータ読み込みフォールト

選択したパラメータブロックに使用できるパラメータデータがない場合、MCE は IDLE ステートに留まります。IDLE ステートからはモータを起動できません。選択したパラメータブロックに使用できる有効なパラメータデータがない場合、MCE はパラメータ読み込みフォールトを報告し、IDLE ステートに留まります。このステートでは、正しいパラメータデータを読み込むか、正しいパラメータブロックを選択する必要があります。

他にフォールトがない場合、MCE はパラメータの値を RAM に読み込み、そのあと STOP ステートへ移行して、モータの駆動準備状態になります。

2.6 スクリプトエンジン

スクリプトエンジンは MCE 内で動作する軽量なバーチャルマシンです。スクリプトエンジンは、ユーザがモータ制御および PFC のレベルを超えたシステムレベルの機能を実装できるようにします。スクリプトエンジンの主な利点は以下のとおりです。

- モータ制御および/または PFC によって使用されていないデジタルピンおよびアナログピンを使用可能にして、ターンキーデバイスの機能を拡張します。
- モータ制御および PFC を超えた将来の機能拡張のためのスケーラビリティ。
- すべてのモータ制御および PFC パラメータおよび変数の読み取りおよび書き込み。

スクリプト使用事例の一部を以下に示します。

- システム起動挙動のカスタマイズ
  - 外部センサまたは制御入力に基づいてモータおよび PFC を起動
  - モータおよび PFC を起動する前にシステムステータスを検証
  - モータ電流制限、電圧制限、速度増加レートなどを修正。
- 特定の速度プロファイルおよびパラメータ設定の定義
  - アナログ入力、DC バス電圧、スイッチリレー、または固定プロファイルに基づいて目標速度値を設定
  - 速度増加レートのランタイム調整
  - さまざまな速度範囲での PI 値のプロファイリング
  - PFC とモータ制御操作間の同期
- フォールト処理およびパラメータ設定
  - システム固有のフォールト処理の定義、いずれかのフォールト状態中の速度または電流抑制、およびフォールト後の回復スキーム
- 設定可能な UART ドライバを使ったカスタマイズされた UART プロトコルの実装

2.6.1 概要

スクリプトコードは「C 言語」のような構文を使用します。スクリプトエンジンは異なる優先順位を持った 2 つの異なるタスクからスクリプトコードを実行します。スクリプトエンジンは、算術、バイナリ論理演算子、判断ステートメント（If...else ステートメント）、およびループステートメント（FOR ステートメント）をサポートします。ユーザは変数をスクリプトコードで定義できます。これらの変数は MCEDesigner からモニタできます。iMOTION 製品では、16kB のメモリ領域がスクリプトコードの保存のために予約されています。そのため、最大許容されるスクリプトバイトコードのサイズは 16kB（約 1.5k ラインのコード）です。スクリプトコードの生成フローを図 74 に示します。

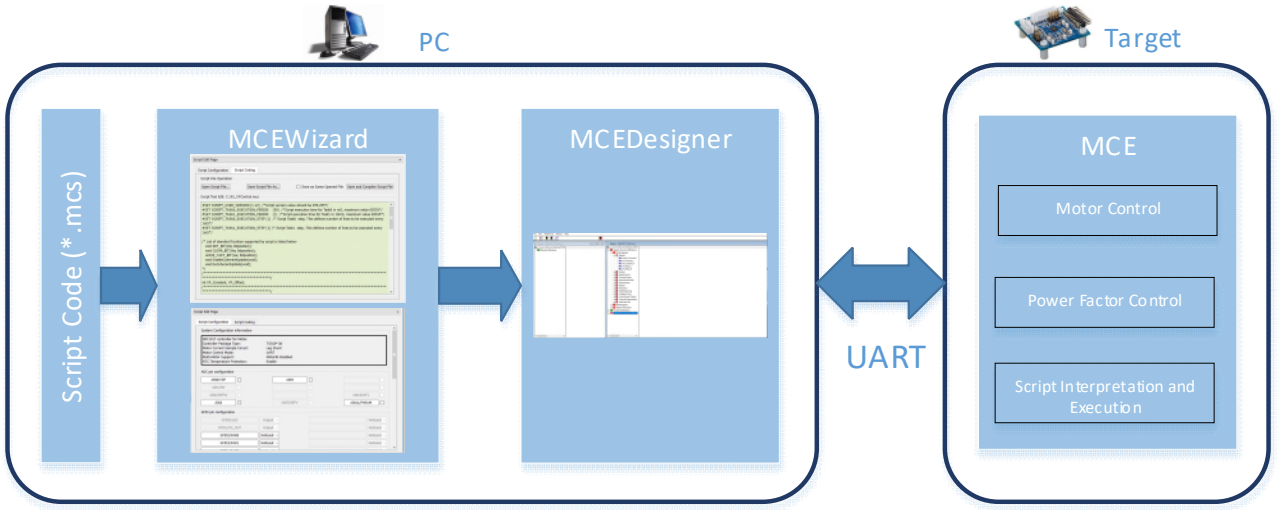


図 74　スクリプトコード生成フロー

2.6.2　スクリプトプログラム構成

スクリプトプログラムは以下の部分から構成されています。

- SET コマンド：スクリプトユーザーバージョンおよびスクリプトタスク実行間隔を定義
- 関数：スクリプトコードは以下の 4 つのあらかじめ定義された関数内で書かなければなりません：Script_Task0_init ()、Script_Task0 ()、Script_Task1_init ()、および Script_Task1 ()。
- 変数およびパラメータ
- ステートメントおよび式：各ステートメントには末尾にセミコロンを付けること。
- コメント：複数行のコメントはスラッシュアスタリスク/*で開始し、アスタリスクスラッシュ*/で終了します。1 行のコメントには接頭辞としてダブルスラッシュ//を付けます。

```
001       /*****************************************************************/
002       /*Script user version value, should be 255.255*/
003       #SET SCRIPT_USER_VERSION (1.12)
004       /*Script execution time for Task0 in ms, maximum value 65535*/
005       #SET SCRIPT_TASK0_EXECUTION_PERIOD    (500)
006       /*Script execution time for Task1 in 10ms, maximum value 65535*/
007       #SET SCRIPT_TASK1_EXECUTION_PERIOD    (1)
008       /*****************************************************************/
009       /* Global variable definition */
010       int Var1;
011       /*****************************************************************/
012       /*Task0 init function*/
013       Script_Task0_init()
014       {
015         /* local variable definition */
016         int Task0Var1;
017         Task0Var1 =0;  /*Initialize local variable*/
018         Var1 =0;       /*Initialize global variable*/
019       }
020       /*Task0 script function*/
021       Script_Task0()
022       {
```

```
023           Task0Var1 =  Task0Var1+1; /*Increment Task0Var1*/
024           Var1      =  Var1+1;      /*Increment Var1*/
025        }
026        /*************************************************************/
027        /*Task1 init function*/
028        Script_Task1_init()
029        {
030          /* local variable definition */
031          int Task1Var1;
032          Task1Var1 =0;    /*Initialize local variable*/
033        }
034        /*Task0 script function*/
035        Script_Task1()
036        {
037           Task1Var1 =  Task1Var1+1; //Increment Task1Var1
038           Var1      =  Var1+1;      /*Increment Var1*/
039        }
```

ソースコード一覧 1　ランタイムカウンタの使用例

2.6.3　スクリプトプログラムの実行

スクリプトエンジンは、Task0 および Task1 という名前の 2 つの独立したタスクからスクリプトコードを実行します。いずれのタスクも定期的に実行されます。タスク実行間隔は、各タスクに対して、スクリプト入力ファイル（*.mcs）の“SCRIPT_TASK0_EXECUTION_PERIOD“および“SCRIPT_TASK1_EXECUTION_PERIOD“パラメータを使って設定できます。各タスクは、スクリプト変数およびモータ/PFC パラメータを初期化するために別々の初期化関数（Script_Taskx_init ( )）を持ちます。また、初期化関数内でスクリプトコードを書くことも可能です。これらの初期化関数は起動中に一回のみ呼び出されます。Task0/Task1 スクリプト関数（Script_Taskx）はタスク実行間隔値に基づいて定期的に呼び出されます。

スクリプトタスクの優先順位はモータ制御または PFC 制御ループ機能よりも低くなります。スクリプトタスク内で Task0 の優先順位は Task1 よりも高いです。

Task0 については、デフォルトで実行ステップは 1、実行間隔は 50（50 x 1ms = 50ms）です。Task1 については、デフォルトで実行ステップは 10、実行間隔は 10（10 x 10ms = 100ms）です。そのため、Task0 はデフォルトで 1ms 毎にスクリプトコードやスクリプトインストラクションを 1 行実行し、スクリプトループ全体を 50ms 毎に最初から実行し直します。Task1 はデフォルトで 10ms 毎にスクリプトコードやスクリプトインストラクションを 10 行実行し、スクリプトループ全体を 100ms 毎に最初から実行し直します。

Task0 および Task1 の合計スクリプト実行時間はスクリプトコード内のスクリプトインストラクションの数に基づいて計算できます。たとえば、Task0 のスクリプトインストラクションの数が 20 である場合、デフォルトで、Task0 はスクリプトコード全体の実行を終了するのに 20ms を要します。残りの 30ms 間は、スクリプトコードは実行されません。50ms 経過後、Task0 は最初のスクリプトインストラクションを再び実行開始します。

ユーザは各タスクの実行ステップおよび実行間隔を自由に設定できます。Task0 の実行間隔を 100ms に設定（SCRIPT_TASK0_EXECUTION_PERIOD =100）すると、Task0 は 100ms 毎に繰り返し実行されます。

Task0 の実行間隔を 100ms に設定（SCRIPT_TASK0_EXECUTION_PERIOD =100）すると、Task0 のライン数は 150 になります。Task0 のスクリプト関数はスクリプトコード全体を 1 回実行するのに 150ms を要します。現在の実行終了後、直ちに最初からまた実行します。

2.6.3.1 実行時間の調整

先に述べた通り、Task0 はスクリプトコードまたはスクリプトインストラクションを 1ms 毎に 1 行実行し、Task1 はスクリプトコードまたはスクリプトインストラクションを 10ms 毎に 10 行実行します。スクリプトの実行速度を上げるために、Task0 または Task1 が実行するライン数を増加することが可能です。

Task0 が 1ms 毎に実行するライン数は、“SCRIPT_TASK0_EXECUTION_STEP“という名前のセットパラメータを使ってスクリプト入力ファイルで設定できます。Task0 の実行間隔が 100ms に設定されている場合（SCRIPT_TASK0_EXECUTION_PERIOD =100）、1ms 毎に実行される Task0 のライン数は 2 にセットされており（SCRIPT_TASK0_EXECUTION_STEP=2）、Task0 のライン数は 100 です。Task0 のスクリプト関数はスクリプトコード全体を 1 回実行するのに 50ms を要します。

同様に、Task1 で 10ms 毎に実行されるライン数は、“SCRIPT_TASK1_EXECUTION_STEP“という名前のセットパラメータを使ってスクリプト入力ファイルで設定できます。

```
001        /*************************************************************/
002        /*Script user version value, should be 255.255*/
003        #SET SCRIPT_USER_VERSION (1.12)
004        /*Script execution time for Task0 in ms, maximum value 65535*/
005        #SET SCRIPT_TASK0_EXECUTION_PERIOD    (500)
006        /*Script execution time for Task1 in 10ms, maximum value 65535*/
007        #SET SCRIPT_TASK1_EXECUTION_PERIOD    (1)
008        /*Defines number of lines to be executed every 1ms in Task0*/
009        #SET SCRIPT_TASK0_EXECUTION_STEP (2)
010        /*Defines number of lines to be executed every 10ms in Task1*/
011        #SET SCRIPT_TASK1_EXECUTION_STEP (10)
012
```

ソースコード一覧 2　ランタイムカウンタの使用例

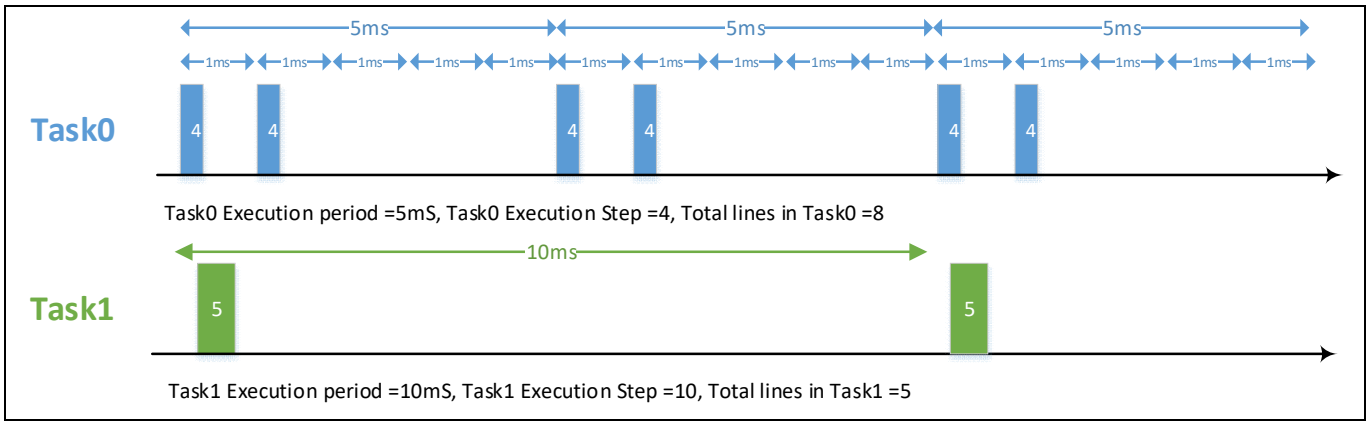


図 75　スクリプトタスクの実行

2.6.3.2 フリーランニングタイマ

スクリプトエンジンには 1ms の分解能のフリーランニングタイマが 1 つ搭載されており、定期的な操作をスケジュールできます。フリーランニングタイマ値（変数名：RunTimeCounter、サイズ：32 ビット、タイプ：読み取り専用）は、スクリプトコードから直接アクセスできます。RunTimeCounter の例をソースコード一覧 3 に示します。

```
001       /*Task0 init function*/
002       Script_Task0_init()
003       {
004         /* local variable definition */
005         int sSVar0,sSVar1;
006         sSVar0 = RunTimeCounter;
007       }
008       /*Task0 script function*/
009       Script_Task0()
010       {
011         /* sSVar1 value toggles for every 10 seconds*/
012         if((RunTimeCounter-sSVar0)>10000)
013         {
014           sSVar0 =RunTimeCounter;
015           if(sSVar1==0)
016           {
017             sSVar1 = 1;
018           }
019           else
020           {
021             sSVar1 = 0;
022           }
023         }
024       }
```

ソースコード一覧 3　ランタイムカウンタの使用例

2.6.4　定数

スクリプトは 10 進数および 16 進数の整定数のみをサポートします。16 進数値には「0x」という接頭辞が必要です。定数値には U や L といった接尾辞を入れることはできません。

変数に浮動小数点定数が割り当てられている場合、小数点以降の値はスクリプトトランスレータによって無視されます。

スクリプトトランスレータは最大 100 個の定数定義をサポートします。定数を定義するには、変数タイプキーワードの前に CONST または const 記述子を置きます。以下のソースコード一覧 4 に定数定義の例が記載されています。

```
001       /*************************************************************/
002       /* Constant definition */
003       CONST int Const1 = 1;
004       const int Const2 = 2;
005       /*************************************************************/
006       /*Task0 init function*/
007       Script_Task0_init()
008       {
009       /* local variable definition */
010             int Task0Var1, Task0Var2;
011             Task0Var1 = Const1; /*Initialize local variable*/
012             Task0Var2 = Const2; /*Initialize local variable*/
013       }
```

ソースコード一覧 4

2.6.5 変数のタイプおよび範囲

スクリプトエンジンは最大 30 個のグローバル変数をサポートします。これらの変数は両方のタスクからアクセス可能です。各タスクには最大 24 個の専用ローカル変数があります。これらの変数はそれぞれのタスク内でのみアクセス可能です。変数はすべて 32 ビットの符号付きです。グローバル変数のみ、MCEDesigner または User UART インターフェイスからアクセスできます。

ユーザはスクリプト変数に自由に名前を付与できます。スクリプト変数を宣言するには、'int'というキーワードを使用します。スクリプト変数名には英数字とアンダースコア（「_」）のみを使用します。変数名は大文字と小文字を区別します。すべての変数名は、グローバル変数とローカル変数を含めて一意でなければなりません。

Task0 または Task1 の関数の外側で宣言された変数はグローバル変数として扱われます。Task0 または Task1 の関数内で宣言された変数は Task0 または Task1 のローカル変数となります。

注: 宣言中は変数を初期化できません。

```
001       /*****************************************************************/
002       /* Global variable definition */
003       int Var1,Var2;
004       /*****************************************************************/
005       /*Task0 init function*/
006       Script_Task0_init()
007       {
008         /* local variable definition */
009         int Task0Var1;
010         Task0Var1 =0;  /*Initialize local variable*/
011         Var1 =0;       /*Initialize global variable*/
012       }
013       /*Task0 script function*/
014       Script_Task0()
015       {
016          Task0Var1 =  Task0Var1+1; /*Increment Task0Var1*/
017          Var1      =  Var1+1;      /*Increment Var1*/
018       }
019       /*****************************************************************/
020       /*Task1 init function*/
021       Script_Task1_init()
022       {
023         /* local variable definition */
024         int Task1Var1;
025         Task1Var1 =0;    /*Initialize local variable*/
026       }
027       /*Task0 script function*/
028       Script_Task1()
029       {
030          Task1Var1 =  Task1Var1+1; //Increment Task1Var1
031          Var2      =  Var1+1;
032       }
```

ソースコード一覧 5　スクリプトのグローバル変数およびローカル変数

2.6.6 モータおよび PFC パラメータアクセス

表 27、表 28、表 29、および表 30 にリストしたモータ制御および PFC のパラメータおよび変数はすべてスクリプトからアクセスできます。パラメータおよび変数は宣言なしで直接スクリプトコードで使用できます。DYNAMIC タイプのパラメータおよび READWRITE タイプの変数のみ、スクリプトコードから書き込むことができます。このパラメータまたは変数を書き込む間、パラメータまたは変数を更新する前に範囲チェックが実行されます。値が範囲外の場合、パラメータ/変数は更新されず、エラービットが 0x13 にセットされます。スクリプトコードからエラーフラグ（変数名：“ErrorFlag“）を読み取ることができます。

STATIC タイプのパラメータまたは READONLY タイプの変数に書き込み操作が行われた場合、パラメータまたは変数は更新されず、エラービットが 0x10 にセットされます。このエラーフラグはスクリプトコードから直接クリアできます。

パラメータおよび変数のセットはコーヒレントな更新方法を使って同時に更新できます。2 つの方法（EnableCoherentUpdate ()および DoCoherentUpdate ()）は、パラメータおよび変数の同時更新を行うようにスクリプトで定義されています。

コーヒレントな更新を有効にした場合（呼び出した EnableCoherentUpdate ()メソッドにより）、書き込み操作をしてもパラメータおよび変数の値は直ちには更新されません。値はすべてまずバッファに保存し、DoCoherentUpdate ()方法を呼びだしてから、すべてのパラメータおよび変数を同時に更新します。スクリプトは最大 32 個のパラメータおよび変数の同時更新をサポートしています。（2.6.11.2 を参照）

2.6.7 演算子

演算子はスクリプトに特定の数学的または論理的関数を実行するように通知する記号です。スクリプト関数でサポートされている演算子のリストを表 17～表 21 に示します。

表 17 算術演算子

演算子	説明
+	2 つの被演算子を加算します
-	最初の被演算子から二番目の被演算子を減算します
*	両方の被演算子を乗算します
/	分子を分母によって除算します
%	剰余演算子および整数除算後の余り

表 18 バイナリ演算子

演算子	説明
|	バイナリ OR(ビット単位論理和)演算子は、2 つの被演算子の対応するビットのどちらかが 1 であれば 1 を、ともに 0 であれば 0 を対応するビットに返します。
&	バイナリ AND(ビット論理積)演算子は、2 つの被演算子の対応するビットの両方が 1 であれば 1 を、それ以外は 0 を対応するビットに返します。
^	バイナリ XOR(ビット単位排他的論理和)演算子は、2 つの被演算子の対応するビットの値が異なれば 1 を、等しければ 0 を対応するビットに返します。
~	バイナリ補数(ビット単位補数)演算子は単項演算子で、ビットを「反転させる」効果を持つ。
<<	バイナリ左シフト演算子。左の被演算子の値が右の被演算子によって指定されたビット数だけ左に移動される。

演算子	説明
>>	バイナリ右シフト演算子。左の被演算子の値が右の被演算子によって指定されたビット数だけ右に移動される。

表 19　代入演算子

演算子	説明
=	単純代入演算子。右側の被演算子の値を左側の被演算子に代入する。

表 20　関係演算子

演算子	説明
==	2 つの被演算子の値が等しいかどうかを確認する。等しい場合、条件は真となる。
>	左側の被演算子の値が右側の被演算子の値よりも大きいかどうかを確認する。大きい場合、条件は真となる。
<	左側の被演算子の値が右側の被演算子の値よりも小さいかどうかを確認する。小さい場合、条件は真となる。

表 21　論理演算子

演算子	説明
&&	論理 AND 演算子は 2 つまたはそれ以上の条件を組み合わせるために使う。演算子は検討対象の両方の条件が満たされたとき、真となる。そうでない場合は、偽を返す。
||	論理 OR 演算子は 2 つまたはそれ以上の条件を組み合わせるために使う。演算子は検討対象のいずれかの条件が満たされたとき、真となる。そうでない場合は、偽を返す。

2.6.8　数式

数式には数字、スクリプト変数、モータ制御/PFC 変数およびパラメータを含めることができます。左括弧および右括弧を使用できます。

例：　TargetSpeed = ADCResult10*(InputScale+10)

数式では、すべての演算子の優先順位に差はなく、左から右に実行されます。そのため、実行の順序を強制するには、左括弧と右括弧を使用する必要があります。

2.6.9　分岐選択構造

分岐選択構造は条件分岐のために使用します。スクリプトエンジンは条件分岐の決定のために If ステートメントを提供しています。If ステートメントは、その後にオプションで Else ステートメントをつなげることができます。この文は論理式が偽であった場合に実行されます。論理式は関係演算子と論理演算子によって構成できます。スクリプト言語での If...else ステートメントの構文は以下に示すとおりです。

```
001        if(boolean_expression)
002        {
003          /*Statement(s) will execute if the expression is true*/
004        }
005        else
```

```
  006       {
  007         /*Statement(s) will execute if the expression is false*/
  008       }

If and else statement should be followed by curly braces
```

ソースコード一覧 6　If...else ステートメントの構文

スクリプトプログラミングではゼロ以外および NULL 以外の値を真、ゼロまたは NULL の値を偽とみなします。

```
  001       /*Task0 init function*/
  002       Script_Task0_init()
  003       {
  004         /* local variable definition */
  005         int InputVal,OutputVal;
  006         InputVal =1;
  007       }
  008       /*Task0 script function*/
  009       Script_Task0()
  010       {
  011         /*Check the boolean condition*/
  012         if(InputVal)
  013         {/* if condition is true then assign OutputVal =10 */
  014           OutputVal=10;
  015         }
  016         else
  017         {/* if condition is false then assign OutputVal =100 */
  018           OutputVal=100;
  019         }
  020       }
  021

  Result : OutputVal =10
```

ソースコード一覧 7　If...else ステートメントの例

以下に、スクリプトによってサポートされる If ステートメントの論理式のその他の例を示します。

表 22　If ステートメントの論理式の例

論理式	説明
If(InputVal1 ==1)	Inputval1 が 1 と等しい場合、条件は真
if(InputVal1)	Inputval1 が 1 と等しい場合、条件は真
if(InputVal1 &0x01)	Inputval1 の最初のビットが SET(1)の場合、条件は真
if(InputVal1 |0x01)	条件は常に真
if(InputVal1 ^0x01)	Inputval1 が 1 と等しくない場合、条件は真
if(~InputVa1l)	Inputval1 が 0 と等しい場合、条件は真
if((InputVal1 ==1)&&(TargetSpeed==0))	Inputval1 が 1 と等しく、かつ TargetSpeed が 0 と等しい場合、条件は真
if((InputVal1 ==1)||(TargetSpeed==0))	Inputval1 が 1 と等しいか、または TargetSpeed が 0 と等しい場合、条件は真

論理式	説明
if((InputVal1 ==1) && ((TargetSpeed==0)||(InputVal2 ==0)))	Inputval1 が 1 と等しく、かつ、TargetSpeed または Inputval2 が 0 と等しい場合、条件は真
if((InputVal1 ==1)||((TargetSpeed==0)||(InputVal2 ==1)))	Inputval1 が 1 と等しいか、または TargetSpeed が 0 と等しいか、または Inputval2 が 1 と等しい場合、条件は真

ネストされた If 条件を書くことができます。ネストされた If 条件の深さは 15 に制限されています。

```
001       if(boolean_expression1)
002       {
003         /*Statement(s) will execute if the expression1 is true*/
004         if((boolean_expression2)
005         {
006           /*Statement(s) will execute if the expression2 is true*/
007           if((boolean_expression3)
008           {
009             /*Statement(s) will execute if the expression3 is true*/
010           }
011         }
012       }
```

ソースコード一覧 8　ネストされた If...ステートメントの構文

スクリプトコードは“if“および“else“のキーワードを決定構造でのみサポートします。ソースコード一覧 9 に“if…elseif..else“ステートメントの構文を示します。

```
001       if(boolean_expression1)
002       {
003         /*Statement(s) will execute if the expression1 is true*/
004       }
005       else
006       {
007         if(boolean_expression)
008         {
009           /*Statement(s) will execute if the expression is true*/
010         }
011         else
012         {
013           /*Statement(s) will execute if the expression is false*/
014         }
015
016       }
```

ソースコード一覧 9　If...の構文 Elseif...else ステートメント

2.6.10　ループ構造

繰り返しプロセスに対してはループ構造を使用します。繰り返しプロセスに対しては FOR ステートメントがサポートされています。

スクリプト言語での FOR ステートメントの構文は以下のとおりです。

```
001       for(<ScriptVariable> = <Startvalue> : <Endvalue>)
002       {
003         /*Statement(s) will execute for defined loop time*/
004       }
```

ソースコード一覧 10 For ステートメントの構文

For ループ内のステートメントは“Endvalue- Startvalue+1“回繰り返して実行されます。

```
001       /*Task0 init function*/
002       Script_Task0_init()
003       {
004         /* local variable definition */
005         int InputVal,OutputVal;
006         OutputVal=0;
007       }
008       /*Task0 script function*/
009       Script_Task0()
010       {
011         if(OutputVal==0)
012         {
013           for(InputVal =1 : 10)
014           {/* for loop executed for 10 times*/
015             OutputVal= OutputVal+1;
016           }
017         }
018       }

  Result : OutputVal =10
```

ソースコード一覧 11 For ステートメントの例

FOR ステートメントはカウントダウンモードではありません。開始値は終了値よりも必ず小さくなければなりません。

2.6.11 メソッド

特定の操作に対して、あらかじめ定義されたメソッドが用意されています。スクリプト関数でサポートされているメソッドについては、以下のセクションで説明します。

2.6.11.1 ビットアクセスメソッド

スクリプト変数またはモータ制御/PFC 関連の変数またはパラメータの特定のビットを読み取りまたは書き込みするため、スクリプトには 3 つのメソッドが定義されています。

表 23 ビットアクセスメソッド

方法	説明
void SET_BIT(<Var>, <bitposition>)	変数の特定のビットをセット
void CLEAR_BIT(<Var>, <bitposition>)	変数の特定のビットをクリア
uint32_t GET_BIT(<Var>, <bitposition>)	変数の特定のビットを読み取る

注: ビットの位置値は 0～15 とします。

```
001       /*Task0 init function*/
002       Script_Task0_init()
003       {
004         /* local variable definition */
005         int InputVal,OutputVal1, OutputVal2;
006         InputVal =0;
007       }
008       /*Task0 script function*/
009       Script_Task0()
010       {
011         SET_BIT(InputVal,15);/*Set 15 bit of InputVal, InputVal
   =0x8000*/
012         /*Read 15 bit of InputVal and assign to OutputVal1*/
013         OutputVal1=GET_BIT(InputVal,15); /*OutputVal1=1*/
014         CLEAR_BIT(InputVal,15);/*clear 15 bit of InputVal, InputVal
   =0*/
015         /*Read 15 bit of InputVal and assign to OutputVal1*/
016         OutputVal2=GET_BIT(InputVal,15);/*OutputVal2=0*/
017       }

  Result : OutputVal1 =1 and OutputVal2=0
```

ソースコード一覧 12 ビットアクセスメソッドの例

2.6.11.2 コーヒレントな更新方法

これらのメソッドはモータ制御および/または PFC のパラメータおよび変数を同時に更新するために使用されます。

表 24 コーヒレントな変数更新メソッド

方法	説明
void EnableCoherentUpdate(void)	パラメータ/変数の同時更新を有効にする
void DoCoherentUpdate(void)	パラメータ/変数の同時更新を実行する

注: 最大 32 個の変数を同時に更新できます。

コーヒレントな更新を有効にした場合、値はパラメータおよび変数に直ちには更新されません。値はまずバッファに保存され、コーヒレントな更新を実行した後に実際の変数/パラメータが更新されます。

```
001       /*Task1 init function*/
002       Script_Task0_init()
003       {
004         EnableCoherentUpdate();
005         AngleSelect =0; //Set to open loop
006         CtrlModeSelect =0;// voltage control mode
007         TargetSpeed = 0x258; //Set speed value
008         DoCoherentUpdate();
009       }
```

ソースコード一覧 13 コーヒレントな変数更新メソッドの例

2.6.11.3 設定可能な UART ドライバを使用するメソッド

これらの方法は UART プロトコルのカスタマイズをサポートするためのものです。

表 25 設定可能な UART を使用するメソッド

方法	説明
void UART_DriverInit(channel, rxInvert, txInvert, baudrate, dataBits, parity, stopBits)	UART ドライバを初期化します。 channe：有効な UART チャンネルを設定（0：channel0、1：channel1） rxInvert / txInvert：データ解釈論理を設定（0：非反転論理、1：反転論理） baudrate：希望するボーレートを bps 単位で設定 dataBits：希望するデータビットの長さ（5、6、7、8）を設定 parity：希望するパリティチェックを設定（0：なし、1：偶数、2：奇数） stopBits：希望するストップビット（1、2）を設定
void UART_DriverDeinit(void)	UART ドライバを初期化解除します。
void UART_FifoInit(rxFifoSize, txFifoSize)	UART ハードウエア FIFO を初期化します。
void UART BufferInit(halfDuplex , rxTimeout, txDelay, txByteDelay, rxFlag, txFlag, rxDataLength, txDataLength)	UART ソフトウェアバッファを初期化します。 halfDuplex：全二重通信または半二重通信モードを設定（0：全二重、1：半二重） rxTimeout：フレーム 1 個の受信にかかると予想される最長時間を ms で設定。 txDelay：フレーム受信からフレーム送信開始までの遅延時間を ms で設定。 txByteDelay：1 個のフレームで受信される各バイト間の予想される遅延時間を設定。 rxFlag：受信フレームのスタートフラグバイトを設定。(0～255：有効なフラグ、256～65535：無効なフラグ/フラグなしのバイト使用） txFlag：送信フレームのスタートフラグバイトを設定。(0～255：有効なフラグ、256～65535：無効なフラグ/フラグなしのバイト使用） rxDatalength：受信フレームからスタートフラグバイトを引いた長さを設定。 txDataLength：送信フレームからスタートフラグバイトを引いた長さを設定。
uint32_t UART_GetStatus(void)	ステータスワードを返します。ビットフィールドの表示方法は以下に示すとおりです。 FIFO のステータス： [0]：IsRxFIFOEmpty（0：受信 FIFO は空ではない、1：受信 FIFO は空） [1]：IsRxFIFOFull（0：受信 FIFO は一杯ではない、1：受信 FIFO は一杯）

方法	説明
	[2]：IsTxFIFOEmpty（0：送信 FIFO は空ではない、1：送信 FIFO は空） [3]：IsTxFIFOFull（0：送信 FIFO は一杯ではない、1：送信 FIFO は一杯） バッファのステータス： [4]：IsRxBufferFull（0：受信バッファは一杯ではない、1：受信バッファは一杯） [5]：予約済み [6]：IsTxBufferEmpty（0：送信バッファは空ではない、1：送信バッファは空） [7]：予約済み ハンドラのステータス： [8]：IsRxTimeout：（0：受信フレームはタイムアウトしていない、1：受信フレームはタイムアウトしている） [9]：IsCollision：（0：衝突は検出されていない、1：衝突が検出されている） [11:10]：BufferState（0：FRAME_START、1：FRAME_RECEIVE、2：FRAME_DELAY、3：FRAME_TRANSMIT） ドライバのステータス： [12]：IsHalfDuplex（0：全二重モード、1：半二重モード） [13]：IsBufferMode（0：FIFO モード、1：バッファモード） [14]：予約済み [15]：IsInitialized（0：初期化されていない、1：初期化されている）
uint32_t UART GetRxDelay(void)	受信フレームの最後のバイトを受信してから次の受信フレームの最初のバイトを受信するまでの遅延時間を ms で返します。
void UART_Control(command)	制御ワードを書き込みます。制御ワードのビットフィールドの表示方法は以下のとおりです。 FIFO 制御： [0]：予約済み [1]：ClrRxFIFO（0：該当なし、1：受信 FIFO をクリア） [2]：予約済み [3]：ClrTxFIFO（0：該当なし、1：送信 FIFO をクリア） バッファ制御： [4]：ClrRxBufferFlag（0：該当なし、1：受信バッファフラグをクリア）

方法	説明
	[5]：予約済み [6]：SendTxBuffer（0：該当なし、1：指定した UART チャンネルを通じて送信バッファでバイトの送信を開始） [7]：予約済み ハンドラ制御： [8]：ClrRxTimeoutFlag（0：該当なし、1：受信タイムアウトフラグをクリア） [9]：ClrCollisionFlag（0：該当なし、1：衝突検出フラグをクリア） [10]：RstBufferControl（0：該当なし、1：バッファ制御ステートマシンをリセット） [11]：予約済み ドライバ制御： [15:12]：予約済み
uint32_t UART_RxFifo(void)	受信 FIFO から 1 バイトを返します。
void UART_TxFifo(uint32_t data)	送信 FIFO に 1 バイトを入れます。
uint32_t UART_RxBuffer(uint32_t idx)	受信バッファから idx 番号によって指定された場所に 1 バイトを返します。
Void UART_TxBuffer(uint32_t idx, uint32_t data)	送信バッファの idx 番号によって指定された場所に 1 バイトを入れます。

以下のコードリストに、上に示した設定可能な UART ドライバを使用したメソッドの一部を使った例を示します。

```
001       UART_DriverInit(
002       1 /*channel*/, 1 /*rxInvert*/, 1 /*txInvert*/,
003       600 /*baudrate*/, 8 /*dataBits*/, 0 /*parity*/, 1 /*stopBits*/
004       );
005       UART_BufferInit(
006       1 /*halfDuplex*/, 2000 /*rxTimeout*/, 2 /*txDelay*/, 0
   /*txByteDelay*/,
007       0xA5 /*rxFlag*/, 0x5A /*txFlag*/, 4 /*rxDataLength*/, 4
   /*txDataLength*/
008       );
009       …
010       if( UART_GetStatus() & UART_IsRxBufferFull )
011       {
012       checksum = -(0xA5 + UART_RxBuffer( 0 ) + UART_RxBuffer( 1 ) +
   UART_RxBuffer( 2 )) & 0xFF;
013       if( checksum == (UART_RxBuffer( 3 ) & 0xFF) )
014       {
015       /* process command */
016       if( 0xC3 == UART_RxBuffer( 0 ) )
017       { /* speed command */
018       TargetSpeed = UART_RxBuffer( 1 ) + (UART_RxBuffer( 2 ) << 8);
019       /* prepare response */
```

```
020       UART_TxBuffer( 0, 0x83 );
021       UART_TxBuffer( 1, status & 0xff );
022       UART_TxBuffer( 2, status >> 8 );
023       checksum = -(0x5A + 0x83 + (status & 0xff) + (status >> 8));
   /* no truncation to 8bit required */
024       UART_TxBuffer( 3, checksum );
025       UART_Control( 0x0050 /* UART_SendTxBuffer |
   UART_ClrRxBufferFlag */ );
026       }
027       }
028       }
```

ソースコード一覧 14 設定可能な UART ドライバを使用するメソッド

2.6.12 ユーザ GPIO

スクリプトは、ユーザが使用できるデジタルピン（モータ制御または PFC によって使用されていないデジタルピン）からの読み取りや書き込みを可能にします。また、ユーザが使用できるアナログピンの値を読み取ることも可能です。

2.6.12.1 デジタル入力および出力ピン

ユーザが使用できるデジタルピンは、MCEWizard を使って入力ピンまたは出力ピンとして設定できます。設定されたデジタル入出力ピンの値はすべて 1ms 毎に MCE によって読み取り/書き込みされます。その値はスクリプトコードによって読み取ることができます。

MCE には、デジタル入出力ピンを読み取りまたは書き込みするための 4 つの専用変数が定義されています。

変数名	タイプ	説明
GPIO_IN_L	READONLY（読み取り専用）	デジタル入出力（GPIO0～GPIO15）ピンの値を保持。
GPIO_IN_H	READONLY（読み取り専用）	デジタル入出力（GPIO16～GPIO29）ピンの値を保持。
GPIO_OUT_L	READWRITE（読み取り/書き込み）	デジタル出力ピンをセットまたはリセット（GPIO0～GPIO15）
GPIO_OUT_H	READWRITE（読み取り/書き込み）	デジタル出力ピンをセットまたはリセット（GPIO16～GPIO29）

GPIO ピンの論理レベルは読み取り専用レジスタ GPIO_IN_L および GPIO_IN_H 経由で読み取ることができます。GPIO_IN_L および GPIO_IN_H レジスタを読み取ると、これらのピンが入力として選択されていても出力として選択されていても、必ず GPIO ピンの現在の論理値を返します。

GPIO_IN_L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO15_IN	GPIO14_IN	GPIO13_IN	GPIO12_IN	GPIO11_IN	GPIO10_IN	GPIO9_IN	GPIO8_IN	GPIO7_IN	GPIO6_IN	GPIO5_IN	GPIO4_IN	GPIO3_IN	GPIO2_IN	GPIO1_IN	GPIO0_IN

GPIO_IN_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	予約済み	GPIO29_IN	GPIO28_IN	GPIO27_IN	GPIO26_IN	GPIO25_IN	GPIO24_IN	GPIO23_IN	GPIO22_IN	GPIO21_IN	GPIO20_IN	GPIO19_IN	GPIO18_IN	GPIO17_IN	GPIO16_IN

GPIOx_IN(x=0:29)変数は、スクリプトから直接アクセスして特定のピンの論理レベルを読み取ることができます。

GPIO_OUT_L および GPIO_OUT_H レジスタは、MCEWizard によって出力として選択されているとき、デジタルピンの値を決定します。ビット位置に 0 を書き込むことにより、対応する出力ピンで低レベル出力が得られます。高レベル出力は対応するビットに 1 を書き込んだ場合に得られます。

GPIO_OUT_L

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO15_OUT	GPIO14_OUT	GPIO13_OUT	GPIO12_OUT	GPIO11_OUT	GPIO10_OUT	GPIO9_OUT	GPIO8_OUT	GPIO7_OUT	GPIO6_OUT	GPIO5_OUT	GPIO4_OUT	GPIO3_OUT	GPIO2_OUT	GPIO1_OUT	GPIO0_OUT

GPIO_OUT_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
予約済み	予約済み	GPIO29_OUT	GPIO28_OUT	GPIO27_OUT	GPIO26_OUT	GPIO25_OUT	GPIO24_OUT	GPIO23_OUT	GPIO22_OUT	GPIO21_OUT	GPIO20_OUT	GPIO19_OUT	GPIO18_OUT	GPIO17_OUT	GPIO16_OUT

GPIOx_OUT(x=0:29)変数は、スクリプトから直接アクセスして特定のピンの論理レベルを書き出すことができます。

```
001       /*Task0 script function*/
002       Script_Task0()
003       {
004         /*Toggle the GPIO2 pin using bit field*/
005         if(GPIO2_IN)
006         {
007           GPIO2_OUT=0;
008         }
009         else
010         {
011           GPIO2_OUT =1;
012         }
013         /*Toggle the GPIO2 pin using variable*/
014         if(GPIO_IN_L&0x2)
015         {
016           SET_BIT(GPIO_OUT_L,2);
```

```
017          }
018          else
019          {
020            CLEAR_BIT(GPIO_OUT_L,2);
021          }
022          /*Toggle the GPIO2 pin using variable*/
023          GPIO_OUT_L = GPIO_OUT_L^0x4;
024        }
```

ソースコード一覧 15 デジタル IO アクセスの例

2.6.12.2 アナログピン

ユーザが使用できるアナログピンは MCEWizard を使って有効にできます。有効にされたアナログピンの値はすべて 1ms 毎に MCE によって読み取られます。その値はスクリプトコードによって読み取ることができます。

MCE にはアナログ入力ピンの値を読み取るための 12 個の専用変数が定義されています。

変数名	タイプ	説明
ADC_Result0	READONLY（読み取り専用）	AIN0 アナログ入力値（12 ビット値）を保持。
ADC_Result1	READONLY（読み取り専用）	AIN1 アナログ入力値（12 ビット値）を保持。
ADC_Result2	READONLY（読み取り専用）	AIN2 アナログ入力値（12 ビット値）を保持。
ADC_Result3	READONLY（読み取り専用）	AIN3 アナログ入力値（12 ビット値）を保持。
ADC_Result4	READONLY（読み取り専用）	AIN4 アナログ入力値（12 ビット値）を保持。
ADC_Result5	READONLY（読み取り専用）	AIN5 アナログ入力値（12 ビット値）を保持。
ADC_Result6	READONLY（読み取り専用）	AIN6 アナログ入力値（12 ビット値）を保持。
ADC_Result7	READONLY（読み取り専用）	AIN7 アナログ入力値（12 ビット値）を保持。
ADC_Result8	READONLY（読み取り専用）	AIN8 アナログ入力値（12 ビット値）を保持。
ADC_Result9	READONLY（読み取り専用）	AIN9 アナログ入力値（12 ビット値）を保持。
ADC_Result10	READONLY（読み取り専用）	AIN10 アナログ入力値（12 ビット値）を保持。
ADC_Result11	READONLY（読み取り専用）	AIN11 アナログ入力値（12 ビット値）を保持。

注： ユーザアナログ入力が MCEWizard で使用可能にされていない場合、ADC_Result 変数は値 0 を保持します。

```
001       /*Task0 script function*/
002       Script_Task0()
003       {
004         /*Start the motor if ADC value is more than 100 count*/
005         if(ADC_Result10>100)
006         {
007           /*Set Target speed value based on ADC input*/
008           TargetSpeed = ADC_Result10<<2;
009           /*Motor start command*/
010           Command=1;
011         }
012         Else /*stop the motor*/
013         {
014           TargetSpeed=0;
015           /*Motor stop command*/
016           Command=0;
017         }
018       }
019
```

ソースコード一覧 16 ユーザアナログピン読み取りの例

2.6.13 スクリプトコードの例

本セクションにはシンプルなスクリプトの例が説明されています。この例でのプロジェクト要件以下示す通りです。

- モータをオープンループモードと電圧制御で駆動します。
- アナログ入力で TargetSpeed コマンドを設定します。
- 目標速度に基づいて電圧コマンドを計算します。電圧 = A*TargetSpeed+B.
- GPIO 入力によって始動/停止指示を行います。GPIO 入力が High の場合はモータを始動し、GPIO 入力が Low の場合はモータを停止します。

2.6.13.1 スクリプト実装

- AIN0 ピンを使って速度コマンドを読み取ります。AIN0 ピンは MCEWizard で使用可能にされます。
- GPIO3 ピンは始動/停止コマンドに使用します。GPIO3 ピンは MCEWizard で入力ピンとして設定されます。
- 電圧 = A*TargetSpeed+B、このとき A および B はスクリプトで（Q23.8 固定小数点フォーマット）として表されます（A=1.5 はスクリプトでは 384 と表される）。
- Task0 から 50ms 毎にスクリプトコードを実行します。

```
001       /*************************************************************/
002       #SET SCRIPT_USER_VERSION (1.0)
003       #SET SCRIPT_TASK0_EXECUTION_PERIOD    (50)
004       /*************************************************************/
005       int A_Const,B_Const;          /* Global variable definition */
006       /*************************************************************/
007       Script_Task0_init()           /*Task0 init function*/
008       {
009         int volt, Max_Limit;        /*Local variable definition */
010         Max_Limit = 4095;
011         A_Const = 150;
012         B_Const = 150;
013         EnableCoherentUpdate();
014           AngleSelect =0;           /*Set to open loop mode*/
015           CtrlModeSelect =0;        /*voltage control mode */
016         DoCoherentUpdate();
017       }
018       Script_Task0()                /*Task0 script function*/
019       {
020        if(GPIO3_IN==0)              /*Check GPIO3 input level*/
021        {
022          TargetSpeed=0;             /*Set Target Speed to zero*/
023          if(SpdRef<100)             /*Wait until motor rampdown */
024          {
025            Command=0;               /*Motor Stop Command*/
026          }
027        }
028        else
029        {
030          TargetSpeed = ADC_Result0;  /*Set Target Speed from ADC0*/
031          /*Calculate voltage set value*/
032          volt = ((A_Const* SpdRef) + B_Const)>>8;
033
034          if(volt> Max_Limit)         /*Limit Check*/
035          {
036            volt = Max_Limit;
037          }
038          Vd_Ext = volt;              /*Set Vd value*/
039          Command=1;                 /*Motor Start Command*/
040        }
041       }
```

ソースコード一覧 17 スクリプト例

2.7 内部発振器の温度校正

2.7.1 概要

MCE の内部発振器の周波数は温度変化によって変動します。内部発振器の精度は、温度変化に対する校正プロセスによって改善できます。MCE は、オンチップ温度センサを使ってダイの温度測定を行うランタイム校正ルーチンを実装しており、オフセット値を使って内部発振器を調整することにより、より高い精度を獲得します。この校正ルーチンは 20ms 毎に自動的に実行されます。

この内部発振器校正機能は‘SysTaskConfig’パラメータの 6 番目のビットをセットすることによって使用可能にできます。達成可能な精度の詳細については、個々の製品のデータシートを参照してください。

3 Register Description

This chapter describes the registers used in MCE. Parameters and variables are scaled within the 16 bit fixed point data range to represent floating-point quantities of the physical value (e.g.: in SI units).

There are two types of parameters used in MCE:

- STATIC : These type of parameters only can be modified/configured from MCEWizard and read from MCEDesigner
- DYNAMIC: These types of parameters can be modified/configured from MCEWizard and read/ write from MCEDesigner

Each parameter or variable can be addressed by using its unique App ID and Index. This section describes each parameter and variable in details.

The Table 26 below describes the scaling from register count values to the corresponding real variable values. The same scaling applies to limit setting or threshold level parameter registers. Notice this assumes DC bus compensation is enabled, and current sensing is centered in the middle of the measurement range. The motor Nominal flux is the motor back EMF constant in Volts per rad/s.

Table 26 Variable Scaling Table

Variable	Count to real scaling	Units	Variable registers
Motor ▯▯▯ current	$\frac{Maximum\_I_{sense}}{2048}$	A	Iu, Iv, Iw, I_Alpha, I_Beta
DC bus voltage	$\frac{Maximum\_Vdc_{sense}}{4096}$	V	VdcRaw, VdcFilt
Rotor Flux Magnitude	$\frac{Motor\ Nominal\ Flux}{2048}$	Wb	Flx_M, Flx_Q
Rotor angle	$\frac{180^o}{32768}$	°	FluxAngle, HallAngle, RotorAngle
Rotor ▯▯▯ Flux	$\frac{Motor\ Nominal\ Flux}{1244}$	Wb	FluxAlpha, FluxBeta
Motor ▯▯▯ Voltage	$\frac{1}{3}\frac{Max\_Vdc_{sense}}{8191}$	V	V_Alpha, V_Beta
Motor d-q current	$\frac{Motor\ Rated\ Current}{4096}$	A rms	Id, Iq, IdFilt, IqFilt, TrqRef, MotorCurrent, TrqRef_Comp
Motor d-q voltage	$\frac{\sqrt{2}}{3}\frac{Max\_Vdc_{sense}}{4974}$	V rms	Vd, Vq, MotorVoltage
Motor Speed	$\frac{Maximum\ Speed}{16384}$	RPM	TargetSpeed, MotorSpeed, abs_MotorSpeed, SpeedError, SpdRef, HallMotorSpeed, FluxMotorSpeed
ADC voltage input	$\frac{V_{DD}}{4096}$	V	ADC_Result0..11
PFC current	$\frac{Max\_I_{sense}}{4096}$	A	PFC_IpfcRaw, PFC_IpfcAvg,
PFC rms current	$\frac{Max\_I_{sense}}{4096}$	A rms	PFC_IpfcRMS

Variable	Count to real scaling	Units	Variable registers
PFC AC voltage	$\frac{Max\_VAC_{sense}}{4096}$	V	PFC_AbsVacRaw, PFC_VacRaw
PFC AC rms voltage	$\frac{Max\_VAC_{sense}}{4096}$	V rms	PFC_VacRMS
PFC DC bus voltage	$\frac{Max\_VDC_{sense}}{4096}$	V	PFC_TargetVolt, PFC_VdcRaw, PFC_VdcFilt
PFC power	$\frac{PFC\_Current\_Scaling \cdot PFC\_VAC\_Scaling}{128}$	W	PFC_ACPower

3.1 System Control Register (App ID =0)

3.1.1 ParPageConf

Index	0		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:0xFFFF	Default: 0

Scaling or Notation: Bit field defined

Description: This parameter defines parameter page selection method and default parameter page

[3:0] Parameter page selection
- 0- No Selection
- 1- Parameter page selection via UART
- 2- Parameter page selection via Analog input
- 3- Parameter page selection via digital input

[7:4] Default parameter page number

[15:8] Reserved

3.1.2 InterfaceConf0

Index	2		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:0xFFFF	Default: 0
Scaling or Notation:	Bit field defined		

Description:

This parameter defines UART0 and UART1 configuration.

[2:0] UART0 function
000: UART0 not used
001: UART0 is used for User UART
111: UART0 is used for MCEDesigner communication
Others: reserved

[3] UART0 TxD output configuration
0: push-pull output
1: open-drain output

[6:4] UART1 function
000: UART1 not used
001: UART1 is used for User UART
111: UART1 is used for MCEDesigner communication
Others: reserved

[7] UART1 TxD output configuration
0: push-pull output
1: open-drain output

[15:8] Transmission delay configuration
1 = 1 ms
Min: 0; Max: 255; Default: 0

3.1.3 InterfaceConf1

Index	3		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:0xFFFF	Default: 0
Scaling or Notation:	Bit field defined		

Description:

This parameter configures JCOM interface.

[2:0] UART0 function
000: JCOM interface is not used by MCE
011: JCOM interface is used for inter-core communication by MCE
Others: reserved

[15:3] Reserved

3.1.4 SysTaskTime

Index	62		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:1000	Default: 1
Scaling or Notation:	1 count = 1ms		
Description:	This parameter defines the execution rate of state machine. The default value is 1 and it is not recommended to be changed by the users.		

3.1.5 CPU_Load

Index	80		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max:1000	Default: 0
Scaling or Notation:	CPU load is represented in %. 1 = 0.1%		
Description:	CPU load is calculated in real-time. This parameter holds the value of CPU load value.		

3.1.6 CPU_Load_Peak

Index	84		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:1000	Default: 0
Scaling or Notation:	CPU load is represented in %. 1 = 0.1%		
Description:	CPU load peak is calculated in real-time. This parameter holds the peak value of CPU load value. This value can be reset to start over the tracking of CPU load peak value.		

3.1.7 FeatureID_selectH

Index	61		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:0xFFFF	Default: 0

Scaling or Notation: Bit field defined

Description: This parameter defines enable or disable motor control and PFC

[0] Enable PFC : 0-Disable, 1 -Enable

[7:1] Reserved

[8] Enable Motor control : 0-Disable, 1 -Enable

[15:9] Reserved

3.1.8 GKConf

Index	22		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:0xFFFF	Default: 0

Scaling or Notation: Bit field defined

Description: This parameter defines gate kill input source for motor control and pfc.

Refer section 2.1.16.3

[1:0] Comparator 0 usage : 0- used by motor, 1- used by PFC

[2] Comparator 0 enable : 0-Disable, 1 -Enable

[4:3] Comparator 1 usage : 0- used by motor, 1- used by PFC

[5] Comparator 1 enable : 0-Disable, 1 -Enable

[7:6] Comparator 2 usage : 0- used by motor, 1- used by PFC

[8] Comparator 2 enable : 0-Disable, 1 -Enable

[10:9] Comparator 3 usage : 0- used by motor, 1- used by PFC

[11] Comparator 3 enable : 0-Disable, 1 -Enable

[12] Motor control gate kill pin enable :0- Disable, 1 -Enable

[15:13] Reserved

3.1.9 SW_Version

Index	82		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max:0xFFFF	Default: 0

Scaling or Notation: Bit field defined

Description: SW version scheme - 4:6:6 Bit Coding

[5:0] Test version

[11:6] Minor version

[15:12] Major version

3.1.10 InternalTemp

Index	81		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	InternalTemp is represented in Kelvin. Ex: 300 = 300K.		
Description:	This parameter holds the sampled value of the internal temperature sensor. It is updated every $50 \cdot SysTaskTime\ (ms)$.		

3.1.11 SysTaskConfig

Index	63		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max: 65535	Default: Set by MCEWizard
Scaling or Notation:	Bit field defined		
Description:	This parameter controls certain system functions as described below. [5:0] Reserved [6] Enable Internal Oscillator Temperature Compensation 0: Disable 1: Enable [15:7] Reserved		

3.2 Motor Control Register (App ID =1)

Complete list of parameter and variables are listed in the Table 27 and Table 28 and find description in the following chapters.

Table 27 Motor control Parameter list

App ID	Index	Parameter Name	Type	Description
1	1	HwConfig	STATIC	Application hardware configuration parameter
1	2	SysConfig	STATIC	System configuration parameter
1	3	AngleSelect	DYNAMIC	Angle selection from flux or open loop or external
1	4	CtrlModeSelect	DYNAMIC	Control mode : Speed control, Current control or Voltage control
1	5	PwmFreq	STATIC	Motor PWM frequency
1	6	PwmDeadtimeR	STATIC	PWM dead time during leading edge (raising edge of high side switch output)
1	7	PwmDeadtimeF	STATIC	PWM dead time during trailing edge (falling edge of high side switch output)

App ID	Index	Parameter Name	Type	Description
1	8	SHDelay	DYNAMIC	Switch delay from PWM output to ADC sample time to avoid ADC sample during switching
1	9	TMinPhaseShift	DYNAMIC	Minimum time of an active vector for single shunt current measurement in phase shift PWM mode
1	10	TCntMin	DYNAMIC	Minimum time of an active vector for single shunt current measurement (minimum pulse width method)
1	11	PwmGuardBand	DYNAMIC	Minimum time of null vector for phase current measurement
1	12	FaultEnable	DYNAMIC	Enable or disable fault condition handling. When a fault bit is not set, the fault condition is ignored
1	13	VdcOvLevel	DYNAMIC	DC bus over voltage trip level
1	14	VdcUvLevel	DYNAMIC	DC bus under voltage trip level
1	15	CriticalOvLevel	DYNAMIC	DC bus critical over voltage trip level
1	16	RotorLockTime	DYNAMIC	Rotor lock fault detection time
1	18	FluxFaultTime	DYNAMIC	PLL out of synchronous fault detection time
1	19	GatekillFilterTime	STATIC	Persistence filter time for PWM gate kill input
1	20	CompRef	STATIC	Overcurrent trip level for gate kill input
1	21	BtsChargeTime	DYNAMIC	Bootstrap capacitor charging time
1	22	TCatchSpin	DYNAMIC	Catch spin synchronization time duration before engaging motor start acceleration
1	23	DirectStartThr	DYNAMIC	Catch Spin threshold speed limit for free running motor that decides whether to run directly in close loop FOC or configured startup mode
1	24	ParkTime	DYNAMIC	Total parking time of the rotor during startup
1	25	ParkAngle	DYNAMIC	Rotor alignment angle during parking
1	26	OpenloopRamp	DYNAMIC	Open loop speed acceleration rate
1	27	IS_Pulses	DYNAMIC	Number of PWM cycles for each inductor sensing pulse under nominal DC bus voltage
1	28	IS _Duty	DYNAMIC	PWM duty cycle during ANGLE_SENSING
1	29	IS_IqInit	DYNAMIC	Initial torque been applied after done ANGLE_SENSING stage and before entering MOTOR_RUN.
1	30	KpSreg	DYNAMIC	Proportional gain of the speed regulator
1	31	KxSreg	DYNAMIC	Integral gain of the speed regulator
1	32	MotorLim	DYNAMIC	Maximum allowable motor current (d axis and q axis)
1	33	RegenLim	DYNAMIC	Maximum allowable motor current (d axis and q axis)while motor is running in regenerative mode
1	34	RegenSpdThr	DYNAMIC	Switch over speed threshold between RegenLim and MotorLim motor current limits
1	35	LowSpeedLim	DYNAMIC	Maximum allowable motor current (d axis and q axis) at low speed

App ID	Index	Parameter Name	Type	Description
1	36	LowSpeedGain	STATIC	Increment rate of Motor current limit after Minspd Threshold
1	37	SpdRampRate	DYNAMIC	close loop speed acceleration rate
1	38	MinSpd	DYNAMIC	Minimum allowed drive operating speed
1	39	Rs	STATIC	Per phase winding resistance of the motor
1	40	L0	STATIC	Per phase winding inductance of the motor at rated current
1	41	LSlncy	STATIC	Saliency inductance of the motor
1	42	VoltScl	STATIC	Internal scaling factor between voltage and flux
1	43	PllKp	DYNAMIC	Angle Frequency Generator tracking proportional gain
1	44	PllKi	DYNAMIC	Angle Frequency Generator tracking integral gain
1	45	PllFreqLim	DYNAMIC	Frequency limit of the PLL integral gain output
1	46	AngMTPA	DYNAMIC	Angle compensation
1	47	FlxTau	STATIC	Define by flux estimator time constant. Used to adjustment for the flux estimator bandwidth.
1	48	AtanTau	DYNAMIC	Angle compensation for the phase shift introduced by flux integration time constant
1	49	SpeedScalePsc	STATIC	Speed scale prescaler value. Used for internal scaling between rotor frequency and motor speed
1	50	SpeedScale	STATIC	Internal scaling factor between rotor frequency and motor speed. Convert rotor frequency to motor speed
1	51	SpeedScaleRcp	STATIC	Internal scaling factor between rotor frequency and motor speed. Convert motor speed to rotor frequency
1	52	SpdFiltBW	DYNAMIC	Low pass filter time constant for motor Speed estimation
1	53	PGDeltaAngle	STATIC	PG output configuration, defines number of pulses per motor revolution
1	54	IfbkScl	STATIC	Internal scaling factor between alpha/beta current 1to d/q current. Current scale to represent d axis and q axis current rated motor current.
1	55	KpIreg	DYNAMIC	Proportional gain of q-axis current regulator
1	56	KpIregD	DYNAMIC	Proportional gain of d-axis current regulator
1	57	KxIreg	DYNAMIC	Integral gain of d- and q-axis current regulator
1	58	FwkLevel	DYNAMIC	Modulation threshold to start field weakening
1	59	FwkKx	DYNAMIC	Gain of field weakening control
1	60	FwkCurRatio	DYNAMIC	–Id current limit for field weakening
1	61	VdqLim	DYNAMIC	Current regulator output limit
1	62	AngDel	DYNAMIC	Gain adjustment for current angle advancement
1	63	AngLim	DYNAMIC	Maximum limit on the current angle phase advancement

App ID	Index	Parameter Name	Type	Description
1	64	IdqFiltBW	DYNAMIC	low pass filter time constant for Id and Iq
1	65	Pwm2PhThr	DYNAMIC	Switch over speed from 3 phase PWM to 2 phase PWM
1	66	TDerating	STATIC	Reserved for future use.
1	67	TShutdown	DYNAMIC	Over-temperature shutdown threshold
1	68	CmdStop	STATIC	Motor stop threshold value for Vsp/frequency/duty cycle control inputs. If the input value is less than threshold, motor will be stopped.
1	69	CmdStart	STATIC	Motor start threshold value for Vsp/frequency/duty cycle control inputs. If the input value is more than threshold, motor will be started.
1	70	CmdGain	STATIC	Slope of set speed value for Vsp/frequency/duty cycle control inputs.
1	71	AppConfig	DYNAMIC	Application configuration Parameter
1	72	NodeAddress	STATIC	Node Address
1	73	PrimaryControlLoop	DYNAMIC	Primary control loop
1	74	PhaseLossLevel	DYNAMIC	Phase loss detection current level
1	75	TrqCompGain	DYNAMIC	Torque Compensation gain factor
1	76	TrqCompAngOfst	DYNAMIC	Torque compensation angle offset value
1	77	TrqCompLim	DYNAMIC	Compensation torque reference limit value
1	78	TrqCompOnSpeed	DYNAMIC	Torque compensation ON speed threshold value
1	79	TrqCompOffSpeed	DYNAMIC	Torque compensation OFF speed threshold value
1	80	PolePair	DYNAMIC	Motor pole pair value
1	81	FaultRetryPeriod	DYNAMIC	Retry period after fault if control input is frequency, duty or VSP
1	85	HallAngleOffset	DYNAMIC	Hall sensor alignment offset value.
1	86	Hall2FluxThr	DYNAMIC	Switch-over speed threshold from using Hall angle to using flux angle.
1	87	Flux2HallThr	DYNAMIC	Switch-over speed threshold from using flux angle to using Hall angle.
1	88	HallSampleFilter	STATIC	Hall sample filter de-bounce time.
1	89	HallSpdFiltBW	DYNAMIC	Time constant of the digital low-pass filter for Hall frequency.
1	94	HallTimeoutPeriod	DYNAMIC	Hall sensor time-out fault period.
1	100	KpHallPLL	DYNAMIC	Proportional gain for Hall_FrequencyAdjust.

Table 28 Motor control Variable list

App ID	Index	Variable Name	Type	Description
1	120	Command	READWRITE	Controls the system state - Stop/ start the motor

App ID	Index	Variable Name	Type	Description
1	121	TargetSpeed	READWRITE	Target speed of the motor, when the drive is in speed control mode.
1	122	Iu	READONLY	Reconstructed motor phase U current
1	123	Iv	READONLY	Reconstructed motor phase V current
1	124	Iw	READONLY	Reconstructed motor phase W current
1	125	MotorSpeed	READONLY	Filtered motor running speed
1	126	I_Alpha	READONLY	Ialpha current
1	127	I_Beta	READONLY	Ibeta current
1	128	IdRef_Ext	READWRITE	Current command on d axis, when the drive is in current control mode.
1	129	IqRef_Ext	READWRITE	Current command on q axis, when the drive is in current control mode.
1	130	Vd_Ext	READWRITE	Vd command when the drive is in voltage control mode.
1	131	Vq_Ext	READWRITE	Vq command when the drive is in voltage control mode.
1	132	SwFaults	READONLY	Drive fault status based on fault condition and fault mask
1	133	SequencerState	READONLY	Current state of the drive
1	134	FaultClear	READWRITE	Fault clear
1	135	FaultFlags	READONLY	Drive fault status based on fault condition
1	136	VdcRaw	READONLY	DC bus voltage
1	137	VdcFilt	READONLY	DC bus filtered voltage
1	138	FluxAngle	READONLY	Estimated rotor Angle
1	139	Flx_M	READONLY	Fundamental flux amplitude
1	140	abs_MotorSpeed	READONLY	Absolute motor speed
1	141	IdFilt	READONLY	Id current filter value
1	142	IqFilt	READONLY	Iq current filter value
1	143	IdFwk	READONLY	Id field weakening current value
1	144	VTH	READONLY	NTC temperature value
1	145	FluxAlpha	READONLY	Flux alpha component value
1	146	FluxBeta	READONLY	Flux beta component value
1	147	Flx_Q	READONLY	Net Flux on Q axis value
1	148	TrqRef	READONLY	Torque Reference, Speed PI output
1	149	Id	READONLY	Id current value
1	150	Iq	READONLY	Iq current value
1	151	V_Alpha	READONLY	Voltage alpha component value
1	152	V_Beta	READONLY	Voltage beta component value
1	153	SpeedError	READONLY	Speed PI error value
1	154	MotorCurrent	READONLY	Motor current value
1	155	OpenLoopAngle	READWRITE	Open loop Angle

App ID	Index	Variable Name	Type	Description
1	156	Vd	READONLY	Motor Vd voltage value
1	157	Vq	READONLY	Motor Vq voltage value
1	158	MotorVoltage	READONLY	Motor voltage value
1	159	TrqRef_Ext	READONLY	Desired compensation torque reference value from torque compensation function block
1	162	SpdRef	READONLY	Speed Reference output from Speed ramp function. Input to Speed PI controller
1	164	ControlFreq	READONLY	Input signal frequency measurement result
1	165	ControlDuty	READONLY	Input signal duty cycle measurement result
1	167	HallAngle	READONLY	Hall sensor-based rotor angle.
1	168	HallMotorSpeed	READONLY	Hall sensor-based motor speed.
1	169	FluxMotorSpeed	READONLY	Flux estimator based motor speed.
1	170	RotorAngle	READONLY	Rotor angle.
1	171	MotorStatus	READONLY	Angle type, PWM modulation type, and phase shift PWM scheme.
1	175	PositionCounter	READONLY	Lower 16 bit of the Hall position counter.
1	176	PositionCounter_H	READONLY	Higher 16 bit of the Hall position counter.
1	181	HallStatus	READONLY	Hall driver status.
1	182	Hall_FrequencyOut	READONLY	
1	183	HallPLL_FrequencyAdjust	READONLY	
1	184	Hall_Atan_Angle	READONLY	Estimated rotor angle using Atan calculation method for analog Hall sensors
1	185	HallU	READONLY	This variable provides the subtraction result of analog Hall U+ input ADC sample value and analog Hall U- input ADC sample value.
1	186	HallV	READONLY	This variable provides the subtraction result of analog Hall V+ input ADC sample value and analog Hall V- input ADC sample value.
1	187	Ipeak	READONLY	Peak amplitude of the motor phase current.
1	188	CurrentAmpOffset0	READONLY	Phase U current sensing offset value.
1	189	CurrentAmpOffset1	READONLY	Phase V current sensing offset value.
1	190	TrqRef_Total	READONLY	Total torque reference value that includes TrqRef from speed PI output and TrqRef_Ext from the torque compensation function.
1	191	TrqCompBaseAngle	READONLY	Base angle value that is used in synthesizing a sinusoidal compensation torque reference.
1	194	TrqCompStatus	READONLY	Torque compensation function status.

3.2.1 Control Register Group

3.2.1.1 HwConfig

Index	1		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 0xFFFF	Default: 0x0120

Scaling or Notation: Bit field definitions are mentioned in description

Description: Motor control application hardware configuration parameter

[0] Current Shunt Type
- 0- Single Shunt current sensing
- 1- Leg Shunts current sensing (2 Phase current sensing)

[2:1] Reserved

[4:3] PWM Mode
- 0- 3 Phase PWM only
- 3- 2 Phase Type 3 PWM

[5] Minimum Pulse (single shunt only)
- 0- Use phase shift for narrow pulse
- 1- Use minimum pulse for narrow pulse

[6] Active polarity for Low side PWM outputs
- 0- Active level is high
- 1- Active level is low

[7] Active polarity for High side PWM outputs
- 0- Active level is high
- 1- Active level is low

[8:9] Internal gain for current measurement
- 0- Internal gain is 1
- 1- Internal gain is 3
- 2- Internal gain is 6
- 3- Internal gain is 12

[12] Low noise phase shift PWM enable
- 0- Disable low noise phase shift PWM
- 1- Enable low noise phase shift PWM

[15:13] CurrentOffsetCal_Psc value (Refer to section 2.1.3 for details.)

Attention: *In MCEWizard current shunt type shall be configured as per actual hardware. Wrong configuration may leads to damage of switches due to over current.*

Attention: *In MCEWizard active polarity of high side and low side switches shall be configured as per actual hardware. Wrong configuration may lead to short circuits in switches.*

3.2.1.2 SysConfig

Index	2		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 0xFFFF	Default: Set by MCEWizard

Scaling or Notation: Bit field definitions are mentioned in description

Description: Motor control system configuration parameter

[0] DC bus voltage compensation
- 0- Disabled
- 1- Enabled

[1] Reserved

[5:2] Execution rate for current control loop (Fast Control Rate).
- 1- Current control loop executed every PWM period
- 2- Current control loop executed every 2 PWM period
- …
- 15- Current control loop executed every 15 PWM period

[10:8] Hall input, hold number of hall inputs
- 000_b – No hall sensor
- 011_b - 2 Hall sensor input
- 111_b – 3 Hall sensor input
- Other values are reserved.

[12:11] Hall Type, Digital or Analog hall sensor
- 0 – Digital hall interface
- 2 – Analog hall interface

[13] Enable Hall Atan angle calculation method
- 0 – Disabled
- 1 – Enabled

[15:14] Analog Hall sensor interface comparator hysteresis configuration
- 0 – 20 mV
- 1 – 15 mV
- 2 – 5 mV
- 3 – 0 mV

3.2.1.3 AngleSelect

Index	3		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 2	Default: Set by MCEWizard
Scaling or Notation:	See description		
Description:	This parameter used to select the rotor angle 0- Open loop angle. Rotating speed is configured by parameter “TargetSpeed“, if Targetspeed=0, open loop angle is fixed can be changed by writing a value to parameter “OpenLoopAngle“ 1- Hall angle. Rotor angle is provided by Hall angle. 2- Flux angle. Rotor angle is provided by Flux estimator. 3- Hybrid angle. Rotor angle switches between Hall angle and flux angle. The switch-over threholds are determined by parameters named ‘Hall2FluxThr’ and ‘Flux2HallThr’.		

3.2.1.4 CtrlModeSelect

Index	4		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 2	Default: Set by MCEWizard
Scaling or Notation:	See description		
Description:	This parameter used to select one of three control modes: 0- Open loop voltage control mode, voltage command is Vd_Ext and Vq_Ext 1- Current control mode, current command is IdRef_Ext and IqRef_Ext 2- Speed control mode, speed command is TargetSpeed		

3.2.1.5 APPConfig

Index	71		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 0xFFFF	Default: 0

Scaling or Notation: Bit field definitions are mentioned in description

Description: Motor control system configuration parameter

[2:0] Control input selection (Set target speed value)
0: UART control
1: Vsp analog input
2: Frequency input
3: Duty cycle input

[3] Enable Restart after Fault: Vsp, frequency or duty cycle control input mode, if there is fault condition, motor will stop and start a 10seconds counter. After 10 seconds, the control will try to clear the fault for 10 times. If the fault condition still exists, control will stay in fault condition.
This feature is not available in UART control mode.
0: Disable restart after fault
1: Enable restart after fault

[4] Enable torque compensation
0: Disable
1: Enable

[5] Reserved

[6] Enable control input measurement
0: Disable
1: Enable

[7] Reserved

[15:8] Hall Atan angle active period: this higher 8-bit word defines the number of sectors for which Hall Atan angle is being used as rotor angle during start-up if Hall angle or hybrid angle is selected and Hall Atan angle calculation method is enabled.
0: Atan angle is not used as rotor angle during start-up if Hall angle or hybrid angle is selected and Hall Atan angle calculation method is enabled.
k = 1~254: Atan angle is used as rotor angle for k number of sectors during start-up if Hall angle or hybrid angle is selected and Hall Atan angle calculation method is enabled.
255: Atan angle is always used as rotor angle if Hall angle or hybrid angle is selected and Hall Atan angle calculation method is enabled.

3.2.1.6 PrimaryControlRate

Index	73		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 1	Max: 16	Default: 2
Scaling or Notation:	See description		
Description:	This parameter defines the execution rate of speed control loop. Speed control loop executed every Fast control rate * Primary control rate * PWM period. Speed ramp rate is calculated based on this value.		

3.2.1.7 Command

Index	120		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max: 1	Default: 0
Scaling or Notation:	See description		
Description:	This variable controls the system state with the following values: 0- Stop the motor 1- Start the motor		

3.2.1.8 SequencerState

Index	133		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max: 9	Default: 0
Scaling or Notation:	See description		
Description:	This variable contains the current sequence state of the drive 0- Power on state 1- Stop state 2- Calculate offset current 3- Charging boot strap capacitors 4- Motor running 5- Fault state 6- Catch spin 7- Parking 8- Open loop acceleration 9- Angle Sensing (Initial rotor angle detection)		

3.2.1.9 MotorStatus

Index	171		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	See description.		
Description:	This variable contains the status of angle type, PWM modulation type, and phase shift PWM scheme. [0] Angle type 0: using Hall angle 1: using flux angle [1] Phase shift PWM scheme 0: using normal phase shift PWM scheme 1: using low noise phase shift PWM scheme [3:2] PWM modulation type 0: using 3-phase PWM modulation 3: using 2-phase PWM modulation		

3.2.2 PWM Register Group

3.2.2.1 PwmFreq

Index	5		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 20	Max:800	Default:160
Scaling or Notation:	1 = 0.1 kHz F_{PWM} ; 160 = 16kHz F_{PWM}		
Description:	This parameter configures the motor PWM frequency in 0.1 kHz increment. PWM Period value = 96,000,000/(2*PWMFreq[Hz])		

3.2.2.2 PWMDeadtimeR

Index	6		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:240	Default:48
Scaling or Notation:	$1 = 20.8333ns$		
Description:	PWM dead time during leading edge (raising edge of high side switch output)		

3.2.2.3 PWMDeadtimeF

Index	7		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:240	Default:48
Scaling or Notation:	$1 = 20.8333ns$		
Description:	PWM dead time during trailing edge (falling edge of high side switch output)		

3.2.2.4 SHDelay

Index	8		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: -192	Max:960	Default: 0
Scaling or Notation:	$1 = 10.4167ns$		
Description:	SHDelay specifies the time delay from PWM output to ADC sample time for current sensing. The delay time is depending on the hardware design; usually it should consider propagation delay of gate driver circuit and turn on (turn off) delay of switching devices. In Phase Shift PWM mode, in order to avoid sample the shunt resistor signal while device is switching, SHDelay should be configured smaller than actual hardware delay. Some board design may allow bigger SHDelay value without causing much current sensing noise (bigger SHDelay value may help for a smaller TMinPhaseShift value). In Minimum Pulse PWM mode, SHDelay should be configured same as actual hardware delay.		

3.2.2.5 TMinPhaseShift

Index	9		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:960	Default: 0
Scaling or Notation:	$1 = 10.4167ns$		
Description:	In Phase Shift PWM mode, TMinPhaseShift configure the minimum time of an active vector for single shunt current sensing.		

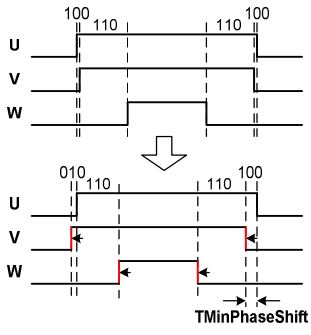


Figure 76 TminphaseShift PWM

3.2.2.6 TCntMin

Index	10		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:960	Default: 0
Scaling or Notation:	$1 = 10.4167ns$		
Description:	This parameter specifies the minimum PWM pulse width if minimum pulse width method is being used.		

3.2.2.7 PwmGuardBand

Index	11		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:960	Default: 0
Scaling or Notation:	$1 = 10.4167ns$		
Description:	In leg shunt configuration, this parameter provides a guard band such that PWM switching at high modulation cannot migrate into the beginning and end of a PWM cycle. The guard band insertion can improve feedback noise immunity for signals sampled near the beginning and end of a PWM cycle. Guard band insertion will reduce the maximum achievable inverter output voltage.		

3.2.2.8 Pwm2PhThr

Index	65		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	16383= Motor Max RPM		
Description:	Switch over speed from 3 phase PWM to 2 phase PWM.		

3.2.3 Speed Control Register Group

3.2.3.1 KpSreg

Index	30		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: Set by MCEWizard
Scaling or Notation:	U8.8		
Description:	This parameter specifies the proportional gain of the speed regulator.		

3.2.3.2 KxSreg

Index	31		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default:12
Scaling or Notation:	U0.16		
Description:	This parameter specifies the integral gain of the speed regulator		

3.2.3.3 MotorLim

Index	32		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default:4095
Scaling or Notation:	4095 = 100% motor rated current		
Description:	This parameter specifies the maximum allowable total motor current (d axis and q axis)		

3.2.3.4 RegenLim

Index	33		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default:409
Scaling or Notation:	4095= 100% motor rated current		
Description:	This parameter specifies the maximum total motor current (d axis and q axis) while motor is running in regenerative mode. RegenLim should be set to a low value if the drive has no break resistor otherwise regenerative current will raise the DC bus voltage and cause fault condition.		

3.2.3.5 RegenSpdThr

Index	34		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default:600
Scaling or Notation:	16383 = Motor Max RPM		
Description:	This parameter specifies the switch over speed threshold between RegenLim and MotorLim.		

3.2.3.6 LowSpeedLim

Index	35		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default:2047
Scaling or Notation:	4095 = 100% motor rated current.		
Description:	This parameter specifies the maximum allowable motor current (d axis and q axis) at low speed (motor speed value less than or equal to Minspd value). Refer section 2.1.7.2.3 for more information.		

3.2.3.7 LowSpeedGain

Index	36		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	U16.0		
Description:	This parameter specifies the increment rate of Motor current limit between MinSpd and low speed threshold. Refer section 2.1.7.2.3 for more information.		

3.2.3.8 SpdRampRate

Index	37		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	See description		
Description:	This parameter (Q11) specifies the ramp rate of target speed reference.		

$$SpdRampRate = \frac{Speed\ Ramp\ Rate(RPM/s) \cdot 16383}{Motor\ Max\ Speed(RPM)} \cdot \frac{Primary\ Control\ Rate \cdot Fast\ Control\ Rate}{F_{PWM}(Hz)} \cdot 2^{11}$$

3.2.3.9 MinSpd

Index	38		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default: 600
Scaling or Notation:	16383 = Motor Max RPM		
Description:	This parameter configures the minimum motor speed. Motor will run at MinSpd when the target motor speed is below MinSpd.		

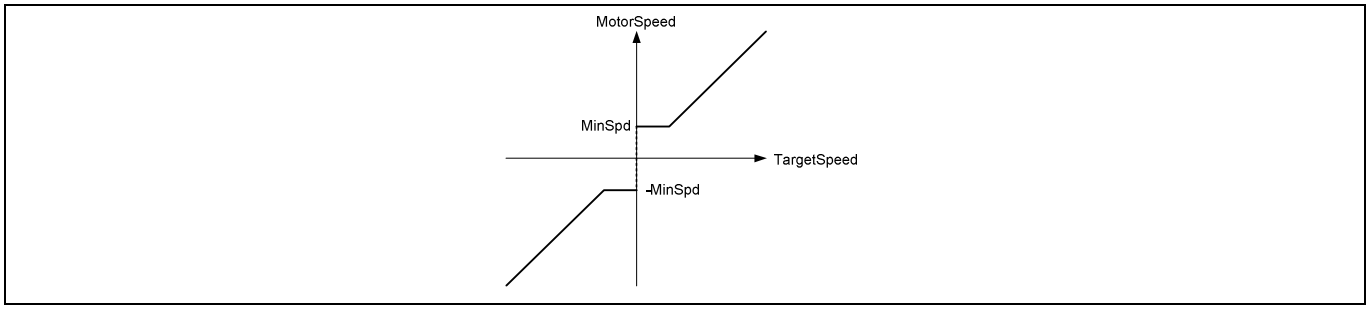


Figure 77 Minimum Speed

3.2.3.10 TargetSpeed

Index	121		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32767	Max:32767	Default: 0
Scaling or Notation:	16383 = Motor Max RPM		
Description:	This variable sets the target speed of the motor, when the drive is in speed control mode. If the motor is running in Vsp analog input, frequency input or duty cycle control, this variable will be updated by software and writing to it has no effect.		

3.2.3.11 TrqRef

Index	148		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -4095	Max:4095	Default: 0
Scaling or Notation:	4095= 100% motor rated RMS current		
Description:	This variable holds the value of Speed PI output value.		

3.2.3.1 SpdRef

Index	162		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32767	Max:32767	Default: 0
Scaling or Notation:	16383 = Motor Max RPM		
Description:	Speed Reference output from Speed ramp function. Input to Speed PI controller		

3.2.3.2 RotorAngle

Index	170		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	90° = 16384		
Description:	This variable represents the rotor angle being used by speed control loop.		

3.2.4 Hall Sensor InterfaceRegister Group

3.2.4.1 HallAngleOffset

Index	85		
Size	Signed 16 bit		
Parameter Type	DYNAMIC		
Range	Min: -32768	Max: 32767	Default: Set by MCEWizard
Scaling or Notation:	90° = 16384		
Description:	This parameter specifies the offset value that the MCE uses to compensate for the angle difference between the rotor position and the Hall sensor (DHALL1 or AHALL1) mounting position. Refer to Section 2.1.14.7 for details.		

3.2.4.2 Hall2FluxThr

Index	86		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 32767	Default: Set by MCEWizard
Scaling or Notation:	16383 = motor max. RPM		
Description:	This parameter specifies the switch-over speed threshold from using Hall angle to using flux angle in hybrid angle mode.		

3.2.4.3 Flux2HallThr

Index	87		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 16383	Default: Set by MCEWizard
Scaling or Notation:	16383 = motor max. RPM		
Description:	This parameter specifies the switch-over speed threshold from using flux angle to using Hall angle in hybrid angle mode.		

3.2.4.4 HallSampleFilter

Index	88		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 32767	Default: Set by MCEWizard
Scaling or Notation:	1 = 10.417 ns		
Description:	This parameter specifies the de-bounce time used by the Hall sample noise filter.		

3.2.4.5 HallSpdFiltBW

Index	89		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 32767	Default: Set by MCEWizard
Scaling or Notation:	See description		
Description:	This parameter specifies the time constant of the digital low-pass filter (LPF) for Hall frequency. The following pseudo code shows the LPF implementation in SW. $$filter(n) = filter(n-1) + \big(input(n) - output(n-1)\big) \cdot HallSpdFiltBW$$ $$output(n) = filter(n) \gg 16$$ Since Hall frequency is updated every motor PWM cycle, the sampling frequency F_s is the same as the motor PWM frequency. The time constant of the LPF for Hall frequency can be calculated as follows. $$T_{decay} = -\frac{Fast\_Control\_Rate}{F_{PWM} \cdot Ln(1 - \frac{HallSpdFiltBW}{2^{16}})}$$		

3.2.4.6 HallTimeoutPeriod

Index	94		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 65565	Default: Set by MCEWizard
Scaling or Notation:	1 = 10 ms		
Description:	This parameter specifies the Hall timeout fault detection time. If the Zero Frequency Flag bit in parameter 'HallStatus' is asserted for the duration of HallTimeoutPeriod x 10 ms, then Hall timeout fault bit in parameter 'FaultFlags' is asserted.		

3.2.4.7 KpHallPLL

Index	100		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 65535	Default: Set by MCEWizard
Scaling or Notation:	U4.12		
Description:	This variable specifies the proportional gain for parameter 'HallPLL_FrequencyAdjust'. It is recommended to set this variable to a value between 0 and 4096. 0: Hall PLL function is disabled. ≠0: Hall PLL function is enabled.		

3.2.4.8 HallAngle

Index	167		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	90° = 16384		
Description:	This variable represents the rotor angle based on Hall sensors.		

3.2.4.9 HallMotorSpeed

Index	168		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	16383 = motor max. RPM		
Description:	This variable represents the motor speed based on Hall sensors.		

3.2.4.10 PositionCounter

Index	175		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max: 65535	Default: NA
Scaling or Notation:	6 = 1 electrical revolution		
Description:	This variable represents the lower 16 bit of the Hall position counter, which is updated during each Hall event.		

3.2.4.11 PositionCounter_H

Index	176		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max: 65535	Default: NA
Scaling or Notation:	1 = PositionCounter overflows from 65535		
Description:	This variable represents the higher 16 bit of the Hall position counter, which is updated during each Hall event.		

3.2.4.12 HallStatus

Index	181		
Size	Unsigned 16 bit		
Parameter Type	Read only		
Range	Min: 0	Max: 65535	Default: NA
Scaling or Notation:	Bit field definitions are mentioned in description		
Description:	Hall sensor interface driver status parameter. [0] Actual motor direction 0: Clockwise direction 1: Counter-clockwise direction [1] Motor direction change 0: No direction change 1: direction changed [2] Zero frequency flag 0: no zero frequency fault 1: zero frequency fault [3] Reserved [6:4] Sector number 0~5: valid pattern 7: invalid pattern [7] Wide sector flag 0: The last sector just before the latest capture is a normal sector (60°). The current sector after the latest capture is a wide sector (120°). 1: The last sector just before the latest capture is a wide sector (120°). The current sector after the latest capture is a normal sector (60°). [10:8] Reserved [11] Target direction 0: Clockwise direction 1: Counter-clockwise direction [15:12] Reserved		

3.2.4.13 Hall_FrequencyOut

Index	182		
Size	Signed 16 bit		
Parameter Type	STATIC		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	1 = 0.0055° / PWM cycle		
Description:	This variable represents the low-pass filtered Hall frequency, which is defined as angle change per motor PWM cycle.		

3.2.4.14 HallPLL_FrequencyAdjust

Index	183		
Size	Signed 16 bit		
Parameter Type	STATIC		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	1 = 0.0055° / PWM cycle		
Description:	This variable represents the compensation term for Hall frequency when calculating Hall angle. It is defined as angle change per motor PWM cycle.		

3.2.4.15 Hall_Atan_Angle

Index	184		
Size	Signed 16 bit		
Parameter Type	Read only		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	90° = 16384		
Description:	This variable represents the estimated rotor angle using Atan calculation method for analog Hall sensors.		

3.2.4.16 HallU

Index	185		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max:2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the subtraction result of analog Hall U+ input ADC sample value and analog Hall U- input ADC sample value.		

3.2.4.17 HallV

Index	186		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max:2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the subtraction result of analog Hall V+ input ADC sample value and analog Hall V- input ADC sample value.		

3.2.5 Flux Estimator PLL Register Group

3.2.5.1 Rs

Index	39		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description.		
Description:	This parameter is the motor winding resistance term in the flux integrator function. MCEWizard calculates this parameter from the motor per phase resistance (motor + cable), motor flux, feedback gains and the control loop sample rate. It is proportional to but is not independently related to motor resistance.		

3.2.5.2 L0

Index	40		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description.		
Description:	This parameter is the motor winding inductance term in the flux integrator function. MCEWizard calculates this parameter from the average per phase inductance, motor flux, feedback gains and the control loop sample rate. It is proportional to but is not independently related to $\frac{Ld+Lq}{2}$.		

3.2.5.3 LSIncy

Index	41		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description.		
Description:	This parameter is the motor winding inductance saliency term in the flux integrator function. MCEWizard calculates this parameter from the var per phase inductance, motor flux, feedback gains and the control loop sample rate. It is proportional to but is not independently related to $\frac{Lq-Ld}{2}$.		

3.2.5.4 VoltScl

Index	42		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	See description.		
Description:	This parameter defines the internal scaling factor between voltage and flux. The value of this parameter is calculated by MCEWizard from user input (PWM frequency, motor poles, DC bus scaling and back EMF Ke). This parameter may need to be adjusted when DC bus compensation is disabled.		

3.2.5.5 PllKp

Index	43		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	U0.16		
Description:	This parameter specifies the Angle Frequency Generator tracking proportional gain. The Angle Frequency Generator is mainly a phase lock loop (PLL). A larger value of PllKp will increase tracking bandwidth at the expense of increasing speed or frequency ripple.		

3.2.5.6 PllKi

Index	44		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	U0.16		
Description:	This parameter specifies the Angle Frequency Generator tracking integral gain. The Angle Frequency Generator is mainly a phase lock loop (PLL). The Figure 78 shows both a simplified and the detailed PLL architecture. PllKi relates internal PLL tracking error (q) to frequency (Rtr_Freq). A larger value of PllKi will increase tracking bandwidth at the expense of increasing speed or frequency ripple.		

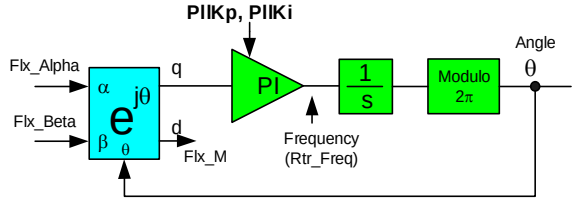


Figure 78 Simplified Block diagram of a FLUX PLL

3.2.5.7 PllFreqLim

Index	45		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	See description		
Description:	This parameter specifies the frequency limit of the PLL integral gain output. The relationship between the actual frequency in Hz and this parameter is given by: $A = \frac{PllFreqLim * PwmFreq}{8192}$, *in Hz* where: A - Actual frequency in Hz PwmFreq -Inverter pwm frequency in Hz		

3.2.5.8 FlxTau

Index	47		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description		
Description:	Motor flux is calculated by integration of estimated voltages. Pure (ideal) integrator cannot be used due to dc offset problem. The integration is done using non-ideal integrator (low pass filter) as shown in the Figure 79. The flux integration time constant (Tau) is an entry of the MCEWizard. Typical range of non-ideal integrator time constant is in the range of 0.01 to 0.025 sec. This parameter provides the adjustment for the flux estimator bandwidth. FlxTau is inversely proportional to the “Flux estimator time constant“ entered in MCEWizard. The relationship of the Flux estimator time constant and FlxTau is given by: $$Flux\ estimator\ time\ constant = \frac{2^{18} \times T_{PWM}}{FlxTau} - T_{PWM}, in\ seconds$$ where FlxTm - the Flux estimator time constant[S] T_{PWM} - 1/(PWM switching frequency) [S] This parameter is also used as low pass filter time constant for rotor frequency (Rtr_Freq, which is the output of flux PLL) as well as some internal filtering.		

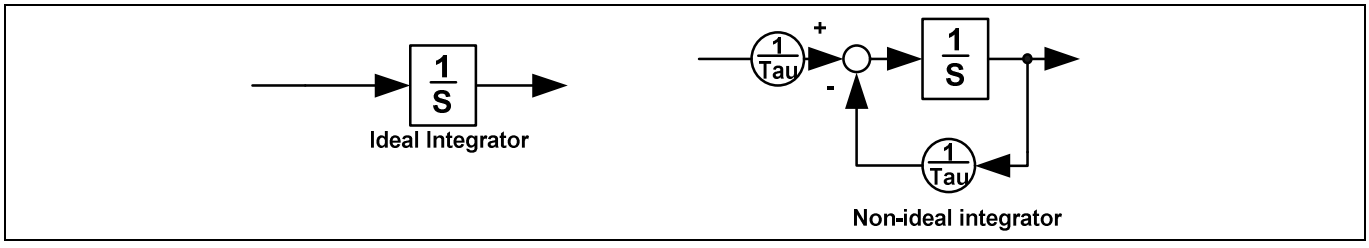


Figure 79 Ideal and Non-ideal Integrator

3.2.5.9 AtanTau

Index	48		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description		
Description:	This parameter provides angle compensation (frequency dependent) for the phase shift introduced by flux integration time constant. Rotor frequency (Rtr_Freq) is multiplied by a time constant (AtanTau) to form a compensating angle. This angle represents the phase shift introduced by the non-ideal flux integrators (low pass filter). Pure (ideal) integrator cannot be used due to dc offset problem. The flux integration time constant is an entry of the MCEWizard. Typical range of integrator time constant is in the range of 0.01 to 0.025 sec.		

3.2.5.10 AngMTPA

Index	46		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 16383
Scaling or Notation:	0°→+360° is represented as 0 to 65535		
Description:	This parameter defines the angle compensation for rotor angle calculation		

3.2.5.11 SpdFiltBW

Index	52		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default: 0
Scaling or Notation:	U2.14, $\tau = \frac{16384}{SpdFiltBw}, in\ T_{PWM}$		
Description:	This parameter configures the low pass filter time constant for motor Speed. Please note that the input of motor speed calculation low pass filter is rotor frequency (Rtr_Freq), which has already been filtered by FreqBW.		

3.2.5.12 SpeedScale

Index	50		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	See Description		
Description:	This parameter is the internal scaling factor between rotor frequency and motor speed. The value of this parameter is calculated by MCEWizard tool from user input (PWM frequency, motor poles and motor maximum speed).		

3.2.5.13 MotorSpeed

Index	125		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32767	Max:32767	Default: 0
Scaling or Notation:	16383 = Motor Max RPM		
Description:	Filtered motor running speed. Filter timer constant is set by parameter SpdFiltBW. Its value will be reset to 0 when the control is not in RUN state.		

3.2.5.14 FluxAngle

Index	138		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32768	Max:32767	Default: 0
Scaling or Notation:	90° = 16384		
Description:	This is the estimated rotor angle. It is used for the Field-Oriented control reference frame.		

3.2.5.15 Flx_M

Index	139		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	2048 = 100% rated flux		
Description:	This variable represents the fundamental flux amplitude.		

3.2.5.16 Abs_MotorSpeed

Index	140		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	16383 = Motor Max RPM		
Description:	Absolute value of motor Speed.		

3.2.5.17 OpenLoopAngle

Index	155		
Size	Signed 16 bit		
Variable Type	Read Write		
Range	Min: -32768	Max:32767	Default: 0
Scaling or Notation:	90° = 16384		
Description:	If Angle Select is set to 0, use this variable to specify the internal open loop angle.		

3.2.5.18 FluxMotorSpeed

Index	169		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -32768	Max: 32767	Default: NA
Scaling or Notation:	16383 = motor max. RPM		
Description:	This variable represents the motor speed based on flux estimator.		

3.2.6 FOC Register Group

3.2.6.1 IfbkScl

Index	54		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description		
Description:	This parameter provides current gain such that 4095 digital counts of d-axis or q-axis current represents rated motor current. IfbkScl is calculated in MCEWizard and is a function of motor rated Amps and analog current scaling.		

$$IfbkScl = \frac{4095 * 1024}{1.647 * AiBiScale * RatedMotorCurrent * \sqrt{2}}$$

Where :
RatedMotorCurrent is in rms Amps

$$AiBiScale = \frac{CurrentInput * InternalADCGain * 4095}{Vadcref}, in\ count/Amps$$

3.2.6.2 KpIreg

Index	55		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	U4.12		
Description:	This parameter specifies the proportional gain of the Q-axis current regulator. The value of this parameter is calculated by MCEWizard from user input (PWM frequency, motor phase inductance).		

3.2.6.3 KpIregD

Index	56		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	U4.12		
Description:	This parameter specifies the proportional gain of the d-axis current regulator. The value of this parameter is calculated by MCEWizard from user input (PWM frequency, motor phase inductance, Bandwidth).		

3.2.6.4 KxIreg

Index	57		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	U0.16		
Description:	This parameter specifies the integral gain of the d-axis and q-axis current regulator. The scaling depends on the current regulator execution rate which is directly related to the pwm frequency. The value of this parameter is calculated by MCEWizard from user input (PWM frequency, motor phase resistance, Bandwidth).		

3.2.6.5 FwkVoltLvl

Index	58		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:PWMPeriod	Default:32
Scaling or Notation:	PWMPeriod = 100% PWM duty cycle.		
Description:	This parameter specifies the modulation threshold to start field weakening. It must be set below PWMPeriod and it's also recommended to set this value below SVPWM linear range (PWMPeriod*0.95). Lower threshold gives more voltage margin which provides better control performance but it will enter field weakening mode earlier. Where : PWMPeriod is 96,000,000/(2*PWMFreq[Hz])		

3.2.6.6 FwkKx

Index	59		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default:32
Scaling or Notation:	See description		
Description:	This parameter configures the gain of field weakening.		

3.2.6.7 FwkCurRatio

Index	60		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default:32
Scaling or Notation:	4096 = 100% MotorLim.		
Description:	This parameter limits the –Id current for field weakening.		

3.2.6.8 VdqLim

Index	61		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4974	Default:32
Scaling or Notation:	4974 = 100 [% modulation]		
Description:	This parameter specifies the current regulator output limit. Refer to Section 2.1.10 for details.		

3.2.6.9 AngDel

Index	62		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See Description		
Description:	This parameter provides gain adjustment for current angle advancement. The current angle advancement is added to a fixed defaulted phase (90 Deg) and the rotor angle to form the relative phasing of the current vector. Current angle advancement is required for Permanent Magnet motor with rotor saliency (Interior Permanent Magnet Motors). A value of zero represents zero angle advancement and therefore the current vector is placed at 90 degrees with respect to the rotor angle. Diagram below shows the implementation of the angle advancement function and the related controller parameters.		

$$Angle\ advancement = AngDel \times 0.35156 \times \frac{I\_Motor}{Rated\ Motor\ Amps}, in\ degree$$

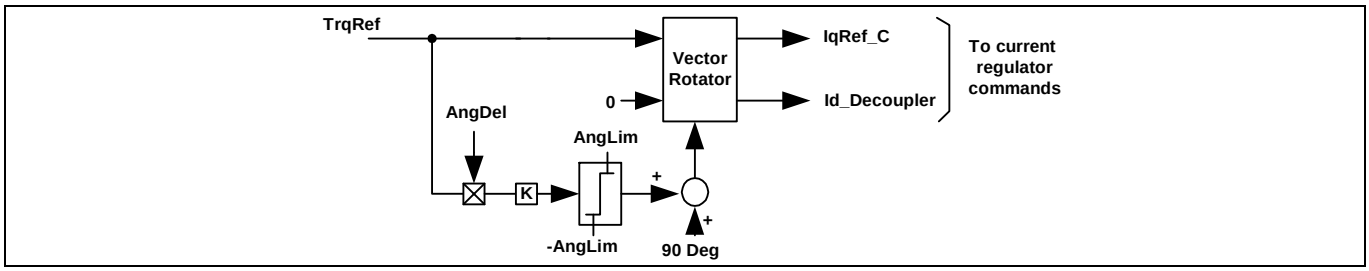


Figure 80 Angle Del

3.2.6.10 AngLim

Index	63		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	$1 = 0.17578\ degree$		
Description:	This parameter provides the maximum limit on the current angle phase advancement specified by parameter AngDel. (See also AngDel.)		

3.2.6.11 IdqFiltBw

Index	64		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:16383	Default:4096
Scaling or Notation:	U2.14 ; $\tau = \frac{16384}{\text{IdqFiltBw}}$, in T_{PWM}		
Description:	This parameter configures the low pass filter time constant for Id and Iq.		

3.2.6.12 IdRef_Ext

Index	128		
Size	Signed 16 bit		
Variable Type	Read Write		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4095 = 100% motor rated RMS current		
Description:	This is the reference input of current regulator on D axis. In speed control mode, this variable has no influence. In current control mode, this variable is used as current input.		

3.2.6.13 IqRef_Ext

Index	129		
Size	Signed 16 bit		
Variable Type	Read Write		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4095 = 100% motor rated RMS current		
Description:	This is the reference input of current regulator on D axis. In speed control mode, this variable has no influence. In current control mode, this variable is used as current input.		

3.2.6.14 IdFilt

Index	141		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4096 = 100% motor rated RMS current		
Description:	Id current after low pass filter. LPF gain is set by IdqFiltBw parameter.		

3.2.6.15 IqFilt

Index	142		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4096 = 100% motor rated RMS current		
Description:	Iq current after low pass filter. LPF gain is set by IdqFiltBw parameter.		

3.2.6.16 IdFwk

Index	143		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4096 = 100% motor rated RMS current		
Description:	IdFwk represents the -Id current during field weakening.		

3.2.6.17 Id

Index	149		
Size	Signed 16 bit		
Variable Type	Read Only		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4096 = 100% motor rated RMS current		
Description:	This variable holds the motor Id component current		

3.2.6.18 Iq

Index	150		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4096 = 100% motor rated RMS current		
Description:	This variable holds the motor Iq component current		

3.2.6.19 MotorCurrent

Index	154		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -16383	Max:16383	Default: 0
Scaling or Notation:	4096 = 100% motor rated RMS current		
Description:	Motor current (IdqFilt) is actual motor phase RMS current. It is calculated as below: $$IdqFilt = (\sqrt{Idfilt^2 + Iqfilt^2})$$		

3.2.7 Measurement Register Group

3.2.7.1 Iu

Index	122		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max:2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides reconstructed motor phase U current (offset eliminated and ADC compensated). This current is calculated from the DC bus link current feedback (single shunt configuration) or U phase shunt resistor (leg shunt configuration). It is sampled every PWM cycle.		

3.2.7.2 Iv

Index	123		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max:2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides reconstructed motor phase V current (offset eliminated and ADC compensated). This current is calculated from the DC bus link current feedback (single shunt configuration) or V phase shunt resistor (leg shunt configuration). It is sampled every PWM cycle.		

3.2.7.3 IW

Index	124		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max:2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides reconstructed motor phase W current (offset eliminated and ADC compensated). This current is calculated from Iu and Iv by equation$Iw = -(Iu + Iv)$. Its value is updated on every PWM cycle.		

3.2.7.4 I_Alpha

Index	126		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max:2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides reconstructed motor alpha component current. Its value is updated on every PWM cycle.		

3.2.7.5 I_Beta

Index	127		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max:2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides reconstructed motor beta component current. Its value is updated on every PWM cycle.		

3.2.7.6 VdcRaw

Index	136		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the measured DC bus voltage value. This value is updated every PWM cycle.		

3.2.7.7 VdcFilt

Index	137		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the filtered DC bus voltage value.		

3.2.7.8 VTH

Index	144		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides NTC temperature input value.		

3.2.7.9 Ipeak

Index	187		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: -2047	Max: 2047	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable keeps track of the peak amplitude of the motor phase current when MCE is in peak current tracking mode with single-shunt configuration by setting TminPhaseShift = 0. Its value is updated every PWM cycle.		

3.2.7.10 CurrentAmpOffset0

Index	188		
Size	Unsigned 16 bit		
Variable Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the offset value for phase U current sensing input pin. Its value is updated every PWM cycle.		

3.2.7.11 CurrentAmpOffset1

Index	189		
Size	Signed 16 bit		
Variable Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the offset value for phase V current sensing input pin. Its value is updated every PWM cycle.		

3.2.8 Protection Register Group

3.2.8.1 FaultEnable

Index	12		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:0xFFFF	Default: 0
Scaling or Notation:	Bit field definitions are mentioned in description. For each bit, 0 – Ignore the associated fault; 1 – enable processing of the associated fault.		
Description:	This parameter specifies enable/disable of faults are mentioned below [1:0] Reserved, must be set to “0“ [2] Enable DC bus overvoltage fault [3] Enable DC bus under voltage fault [4] Enable Flux PLL out of control fault [5] Reserved, must be set to “0“ [6] Enable Over temperature fault [7] Enable rotor lock fault [8] Enable Phase loss fault [12:9] Reserved, must be set to “0“ [13] Enable UART link break fault [14] Enable Hall timeout fault [15] Enable Hall invalid fault When a fault is disabled (bit set to “0“), the fault condition is ignored and the motor keeps running. However, even when a fault is disabled, its occurrence is reported in the FaultFlags variable, until the condition that caused the fault disappears.		

Note: Phase loss fault is only detected in “PARKING“ state.

3.2.8.2 VdcOvLevel

Index	13		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the dc bus over voltage trip level. A dc bus over voltage fault will be generated if dc bus voltage exceeds this threshold.		

3.2.8.3 VdcLvLevel

Index	14		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the dc bus under voltage trip level. A dc bus under trip voltage fault will be generated if dc bus voltage falls below this threshold.		

3.2.8.4 CriticalOvLevel

Index	15		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	Detection level for Critical Overvoltage. If this threshold is exceeded, all low side switches are clamped (zero-vector-braking) to protect the drive and to brake the motor. The Zero-vector is held until fault is cleared.		

3.2.8.5 RotorLockTime

Index	16		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default:1000
Scaling or Notation:	1 = 0.01 Second		
Description:	User can change the value of this parameter to customize the rotor lock detect time (default = 10 seconds). Please note if rotor lock detect time is configured too short, it may trigger the fault during acceleration or momentary high load condition.		

3.2.8.6 FluxFaultTime

Index	18		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 65535	Default:800
Scaling or Notation:	1 = 0.01 Second		
Description:	User can change the value of this parameter to customize the PLL out of synchronous detect time (default = 8 seconds).		

3.2.8.7 GateKillFilterTime

Index	19		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 4	Max:960	Default:96
Scaling or Notation:	$1 = 10.4167ns$		
Description:	Persistence filter time for PWM gate kill input (in clock cycles)		

3.2.8.8 CompRef

Index	20		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	ADC count (4095 = V_{adcref})		
Description:	This parameter value is derived from current trip level and current input offset value. $$CompRef = \frac{((iTripLevel \times CurrentScale) + Offset)}{Vadcref} * 4095$$ Example : iTripLevel = 2A, Current input scale 0.5V/A, Offset =0.55V, Vadcref =3.3V CompRef value is 1924 counts		

3.2.8.9 Tshutdown

Index	67		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the over-temperature shutdown threshold. If actual temperature input value is less than this threshold, trigger over temperature fault.		

3.2.8.10 PhaseLossLevel

Index	74		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: MCEWizard
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the phase current threshold value for phase loss protection function. Refer to Section 2.1.16.6 for details.		

3.2.8.11 SwFaults

Index	132		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	See description.		
Description:	This variable is derived from FaultFlags by the following bitwise logical operation: $SwFaults = FaultFlags \cdot FaultEnable$ SwFaults is cleared by FaultClear. For bit field definition, refer to Faultflags.		

3.2.8.12 FaultClear

Index	134		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:1	Default: 0
Scaling or Notation:	See Description		
Description:	Writing 1 to this variable clears all faults. Once clear has been done, the variable will be cleared. If fault condition doesn't exist, fault clear will be successful and the drive will enter STOP state. If fault condition still exists, the drive will remain in fault state.		

3.2.8.13 FaultRetryPeriod

Index	81		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:0xFFF	Default:0x6403
Scaling or Notation:	Bit field definitions are mentioned in description		
Description:	This parameter enables to restart the motor after fault condition when control input signal is from frequency, duty or VSP [7:0] This field defines how many times restart MCE after fault condition. If value is 0, restart after fault is disabled. If value is 255, restart always after fault. [15:8] This field defines waiting time to restart MCE after fault condition. This value is represented in 0.1 seconds. If value is 100, MCE restarted 10S after fault.		

3.2.8.14 FaultClear

Index	134		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:1	Default: 0
Scaling or Notation:	See Description		
Description:	Writing 1 to this variable clears all faults. Once clear has been done, the variable will be cleared. If fault condition doesn't exist, fault clear will be successful and the drive will enter STOP state. If fault condition still exists, the drive will remain in fault state.		

3.2.8.15 FaultFlags

Index	135		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:0xFFFF	Default: 0

Scaling or Notation: Bit field definitions are mentioned in description

Description: This variable provides drive fault status. Most faults are handled by a fault handling routine operating at the PWM inverter switching frequency with the exception of Gate Kill faults. Gate Kill is handled within the Faults module and will instantly initiate inverter and regulator shutdown. The FaultFlags variable indicates currently pending fault conditions. The FaultClear variable is used to reset fault conditions.

For all bit fields defined below, a value of 1 indicates that the corresponding fault condition has occurred.

[0] Motor gatekill fault
[1] DC bus Critical overvoltage fault
[2] DC bus overvoltage fault
[3] DC bus under voltage fault
[4] Flux PLL out of control fault
[5] Reserved
[6] Over Temperature fault
[7] Rotor lock fault
[8] Phase Loss fault
[9] Reserved
[10] Execution fault (CPU load is more than 95%)
[11] Reserved
[12] Parameter load fault
[13] UART link break fault
[14] Hall timeout fault
[15] Hall invalid fault

Note: DC bus critical overvoltage and Gatekill fault cannot be masked by FaultEnable.

3.2.9 Start Control Register Group

3.2.9.1 BtsChargeTime

Index	21		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 150

Scaling or Notation: 1 = one PWM period

Description: User can change the value of this parameter to configure the boot strap capacitor charging time (default = 150 PWM period).

3.2.9.2 ParkAngle

Index	25		
Size	Signed 16 bit		
Parameter Type	DYNAMIC		
Range	Min: -32768	Max:32767	Default:5461

Scaling or Notation: 90° = 16384

Description: This parameter configures the current angle during parking state. Reset value of parking angle is set to 5461 (30°) which is the center of sector 0.

3.2.9.3 ParkTime

Index	24		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default:1000
Scaling or Notation:	1 = 0.001 Second		
Description:	This parameter configures total parking time. During parking state, parking current increases linearly from 0 to low speed current limit. If ParkTime=0, parking state will be skipped.		

3.2.9.4 OpenLoopRamp

Index	26		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default:1000
Scaling or Notation:	See description.		
Description:	This parameter configures the open loop acceleration rate. During open loop state, motor current is regulated at low speed current limit; rotation speed accelerates linearly from 0 to MinSpd. Total duration of open loop: $T_{OpenLoop} = \frac{MinSpd*1024*10}{OpenLoopRamp}$, in 1 millisecond If OpenLoopRamp=0, open loop state will be skipped		

3.2.9.5 IS_Pulses

Index	27		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:1000	Default:14
Scaling or Notation:	1 = 1 sensing pulse under nominal DC bus voltage (DcBusVolts=2048)		
Description:	This parameter specifies the number of PWM cycles for each angle sensing pulse under nominal DC bus voltage. Actual number of PWM cycles is DC bus compensated in order to keep constant volt-second which in turns keep the same peak sensing current. Initial angle sensing function measures the current of last 2 PWM cycles, if the parameter is configured at 1, only one PWM cycle will actually carry current. So it is advised to configure this parameter ≥ 2. Write 0 to this parameter disables the initial angle sensing function.		

3.2.9.6 IS_Duty

Index	28		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:8191	Default:4095
Scaling or Notation:	8192 = 100% PWM duty cycle		
Description:	This parameter specifies the PWM duty cycle during ANGLE_SENSING state. For better current sensing quality, in single shunt current sensing, duty cycle of angle sensing should not be too low otherwise active vector will be too short to sense the current. In leg shunt current sensing, duty cycle should not be too high otherwise there will not be enough time to sense the current during zero vector.		

3.2.9.7 IS _IqInit

Index	29		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:8191	Default:14
Scaling or Notation:	4096 = 100% motor rated RMS current		
Description:	This parameter specifies the initial torque that would be applied after having completed ANGLE_SENSING state and before entering MOTOR_RUN state. Right after having completed ANGLE_SENSING state, the flux PLL has not locked onto the rotor angle. It takes some time to stabilize itself and requires the motor speed to be sufficiently high. This means at the beginning of MOTOR_RUN state, flux PLL is not working properly and it relies on initial torque to accelerate the motor in order for the PLL to lock. To achieve reliable and smooth start, some tuning to flux estimator and flux PLL is required.		

3.2.10 Catch Spin Register Group

3.2.10.1 TCatchSpin

Index	22		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default:1000
Scaling or Notation:	1 = 0.001 Second		
Description:	This parameter specifies the catch spin synchronization time duration before engaging motor start acceleration. During this period, the internal controller tries to sync rotor position with zero torque current reference.		

3.2.10.2 DirectStartThr

Index	23		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default:1000
Scaling or Notation:	16384 = Motor Max RPM		
Description:	At the end of catch spin state, this parameter is the absolute motor speed threshold that decides whether to go through Angle_Sensing + Parking + OpenLoop start or directly go to closed loop run state. If DirectStartThr=0, after catch spin, it will directly go to closed loop run state.		

3.2.11 Control Input Register Group

3.2.11.1 PGDeltaAngle

Index	53		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	512 = 1 PG pulse every 360 electrical degree (every electrical cycle)		
Description:	This parameter configures the PG output. $PGDeltaAngle = 256 * \frac{Motor\ poles}{PPR}$ Write 0 to PGDeltaAngle will disable the PG output. PPR is expected PG Pulses Per Revolution. For example, 4 PPR for an 8 poles motor (1 pulse per electrical cycle), then: $PGDeltaAngle = 256 * \frac{8}{4} = 512$ PG output is updated every PWM cycle, so the maximum PG output frequency is ½ Fpwm. The maximum value for PGDeltaAngle is 16383, which means 1 PG pulse take 32 electrical cycle (16384/512=32), on an 8 poles motor, the PG output will be 0.125PPR. If PGDeltaAngle is 2^n (2,4,8,16….8192,16384), PG pulse will be synchronized with rotor angle. For example, if PGDeltaAngle=512 for an 8 poles motor (4PPR). There are 4 PG pulses every 4 electrical cycles and the PG transition (high to low or low to high) will happen at 0 and 180 electrical degree.		

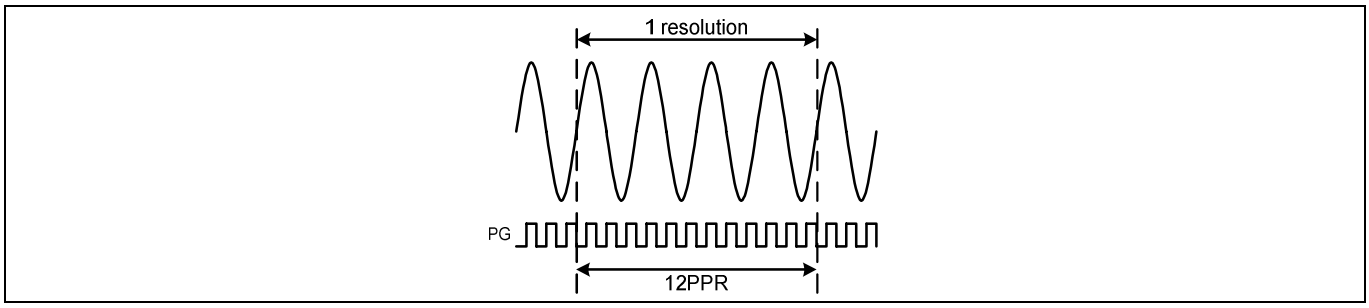


Figure 81 PG Output

3.2.11.2 CmdStart

Index	69		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	See description.		
Description:	In Vsp/Frequency/Duty Cycle input mode, this parameter specifies the input threshold for motor start.		

3.2.11.3 CmdStop

Index	68		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	See description.		
Description:	In Vsp/Frequency/Duty Cycle input mode, this parameter specifies the input threshold for motor stop.		

3.2.11.4 CmdGain

Index	70		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 65535	Default: 0
Scaling or Notation:	See description.		
Description:	In Vsp/Frequency/Duty Cycle input mode, this parameter specifies the slope between the input threshold for motor start and threshold for MaxRPM.		

3.2.11.5 ControlFreq

Index	164		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max: 50000	Default: 0
Scaling or Notation:	1 count = 0.1Hz		
Description:	When control input measurement function is enabled, this variable represents the measured frequency of the input signal which is updated every 10 ms.		

3.2.11.6 ControlDuty

Index	165		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max: 1000	Default: 0
Scaling or Notation:	1 count = 0.1%		
Description:	When control input measurement function is enabled, this variable represents the measured duty cycle of the input signal which is updated every 10 ms.		

3.2.12 Voltage Control Register Group

3.2.12.1 Vd_Ext

Index	130		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:4974	Default: 0
Scaling or Notation:	$\text{Output duty cycle} = \frac{\text{Vd\_Ext} \times 2048}{4974 \times \text{DcBusVoltsFilt}} \times 100\%$		
Description:	Vd command when the drive is working in voltage control mode.		

3.2.12.2 Vq_Ext

Index	131		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:4974	Default: 0
Scaling or Notation:	$Output\ duty\ cycle = \frac{Vq\_Ext \times 2048}{4974 \times DcBusVoltsFilt} \times 100\%$		
Description:	Vq command when the drive is working in voltage control mode.		

3.2.12.3 V_Alpha

Index	151		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -8191	Max:8191	Default: 0
Scaling or Notation:	8191 = 100%		
Description:	This variable provides Alpha motor phase voltage.		

3.2.12.4 V_Beta

Index	152		
Size	Signed 16 bit		
Variable Type	Read only		
Range	Min: -8191	Max:8191	Default: 0
Scaling or Notation:	8191 = 100%		
Description:	This variable provides Beta motor phase voltage.		

3.2.12.5 Vd

Index	156		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4974	Default: 0
Scaling or Notation:	4974 = 100%		
Description:	Motor Vd voltage component. This variable holds the value of Id PI output value in case of speed control or current control mode.		

3.2.12.6 Vq

Index	157		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4974	Default: 0
Scaling or Notation:	4974 = 100%		
Description:	Motor Vq voltage component. This variable holds the value of Iq PI output value in case of speed control or current control mode.		

3.2.12.7 MotorVoltage

Index	158		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4974	Default: 0
Scaling or Notation:	4974 = 100%		
Description:	This variable holds motor applied voltage. $Vdq = (\sqrt{Vd^2 + Vq^2})$		

3.2.13 Torque Compensation Register Group

3.2.13.1 TrqCompGain

Index	75		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default:N/A
Scaling or Notation:	U8.8		
Description:	This parameter specifies the gain factor that is used to adjust the amplitude of the sinusoidal compensation torque reference.		

3.2.13.2 TrqCompAngOfst

Index	76		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 65535	Default: Set by MCEWizard
Scaling or Notation:	90° = 16384		
Description:	This parameter specifies the angle offset value that the MCE uses in synthesizing a sinusoidal compensation torque reference. Refer to Section 2.1.15 for details.		

3.2.13.3 TrqCompLim

Index	77		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max: 16383	Default: 2048
Scaling or Notation:	4095 = 100% motor rated current		
Description:	This parameter specifies the maximum allowable value for internal variable ‘TrqRefFilt’ that is the low-pass filtered result of the variable ‘TrqRef’. Refer to Section 2.1.15 for details.		

3.2.13.4 TrqCompOnSpeed

Index	78		
Size	Unsigned 16 bit		
Variable Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	16383 = Motor Max RPM		
Description:	This parameter set torque compensation ON speed threshold value. Torque compensation is active if motor speed reference value is less than this parameter value.		

3.2.13.5 TrqCompOffSpeed

Index	79		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:32767	Default: 0
Scaling or Notation:	16383 = Motor Max RPM		
Description:	This parameter set torque compensation OFF speed threshold value. Torque compensation is inactive if motor speed reference value is greater than this parameter value.		

3.2.13.6 TrqRef_Ext

Index	159		
Size	Signed 16 bit		
Parameter Type	Read only		
Range	Min: -16383	Max:16383	Default: N/A
Scaling or Notation:	4095 = 100% motor rated current		
Description:	This variable represents the synthesized sinusoidal compensation torque reference.		

3.2.13.7 TrqRef_Total

Index	190		
Size	Signed 16 bit		
Parameter Type	Read only		
Range	Min: -16383	Max:16383	Default: N/A
Scaling or Notation:	4095 = 100% motor rated current		
Description:	This variable represents the summed up torque reference that includes ‘TrqRef’ from speed PI output as well as ‘TrqRef_Ext’ from the torque compensation function.		

3.2.13.8 TrqCompBaseAngle

Index	191		
Size	Signed 16 bit		
Parameter Type	Read only		
Range	Min: -32768	Max: 32767	Default: Set by MCEWizard
Scaling or Notation:	90° = 16384		
Description:	This variable represents the base angle that is used in synthesizing a sinusoidal compensation torque reference.		

3.2.13.9 TrqCompStatus

Index	194		
Size	Unsigned 16 bit		
Parameter Type	Read only		
Range	Min: 0	Max: 65535	Default: NA
Scaling or Notation:	Bit field definitions are mentioned in description		
Description:	Torque compensation function status variable. [0] Torque compensation function active status 0: inactive 1: active [1] Mechanical cycle synchronization status 0: Not synchronized to mechanical cycle 1: Synchronized to mechanical cycle		

3.3 PFC Control Register (App ID =3)

Complete list of parameter and variables are listed in the Table 29 and Table 30 and find description in the following chapters.

Table 29 PFC Parameter list

App ID	Index	Parameter Name	Type	Description
3	1	PFC_HwConfig	STATIC	Application hardware configuration parameter
3	2	PFC_SysConfig	STATIC	System configuration parameter
3	3	PFC_PwmFreq	STATIC	PFC PWM frequency
3	4	PFC_TMinOff	DYNAMIC	Minimum PWM off time
3	5	PFC_Deadtime	STATIC	PWM dead time
3	6	PFC_SHDelay	DYNAMIC	Delay time from PWM output to ADC sample time
3	7	PFC_IRectLim	DYNAMIC	Rectifying current limit value
3	8	PFC_IGenLim	DYNAMIC	Generating current limit value
3	9	PFC_VdcRampRate	DYNAMIC	Voltage reference ramp up/down rate
3	10	PFC_KpVreg	DYNAMIC	Proportional gain of the voltage regulator
3	11	PFC_KxVreg	DYNAMIC	Integral gain of the voltage regulator
3	12	PFC_KpIreg	DYNAMIC	Proportional gain of the current regulator
3	13	PFC_KxIreg	DYNAMIC	Integral gain of the current regulator
3	15	PFC_TrackingCycle	DYNAMIC	Used for voltage reference in tracking mode
3	16	PFC_TrackingGain	DYNAMIC	Used for voltage reference in tracking mode
3	17	PFC_HalfCycleMin	STATIC	AC voltage minimum limit for input frequency check
3	18	PFC_HalfCycleMax	STATIC	AC voltage maximum limit for input frequency check
3	19	PFC_VacZCThr	DYNAMIC	AC voltage threshold to detect zero crossing
3	20	PFC_VacOvLevel	DYNAMIC	AC voltage input overvoltage trip level

App ID	Index	Parameter Name	Type	Description
3	21	PFC_VacUvLevel	DYNAMIC	AC voltage input under voltage trip level
3	22	PFC_VdcOvLevel	DYNAMIC	DC bus overvoltage trip level
3	23	PFC_VdcUvLevel	DYNAMIC	DC bus under voltage trip level
3	24	PFC_AcDcScale	DYNAMIC	Ratio of feedforward component added to the duty output
3	25	PFC_LFactor	DYNAMIC	Used for average current calculation
3	26	PFC_FaultEnable	DYNAMIC	Enable or disable fault condition handling. When a fault bit is not set, the fault condition is ignored
3	27	PFC_GateKillTime	STATIC	Persistence filter time for PWM gate kill input
3	28	PFC_TargetDCVolt	STATIC	Target DC bus voltage for fixed-voltage mode

Table 30 PFC Variable list

App ID	Index	Variable Name	Type	Description
3	81	PFC_SequencerState	READONLY	Current state
3	82	PFC_Command	READWRITE	Controls the system state - Stop/ start the PFC
3	85	PFC_FaultClear	READWRITE	Fault clear
3	87	PFC_SwFaults	READONLY	Fault status based on fault condition and fault mask
3	89	PFC_TargetVolt	READWRITE	Voltage set point value
3	90	PFC_VoltagePIoutput	READONLY	Voltage PI controller output value
3	92	PFC_VdcRaw	READONLY	DC bus voltage
3	93	PFC_IpfcRaw	READONLY	PFC Current
3	94	PFC_AbsVacRaw	READONLY	AC voltage absolute value
3	98	PFC_VacRMS	READONLY	AC voltage RMS value
3	99	PFC_VdcFilt	READONLY	DC bus filtered voltage
3	103	PFC_VacRaw	READONLY	AC voltage value
3	104	PFC_Fault Flag	READONLY	Fault status based on fault condition
3	105	PFC_IpfcAvg	READONLY	PFC current average value
3	106	PFC_IpfcRMS	READONLY	PFC current RMS value
3	107	PFC_ACPower	READONLY	PFC input power
3	110	PFC_CurrentPIoutput	READONLY	Current control PI output

3.3.1 Control Register Group

3.3.1.1 PFC_HwConfig

Index	1		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 0xFFFF	Default: 0

Scaling or Notation: Bit field definitions are mentioned in description

Description: PFC application hardware configuration parameter

[0] PFC Topology
- 0- Boost Mode PFC
- 1- Totem Pole PFC

[4:1] Reserved

[5] Active polarity for Low side PWM output
- 0- Active level is low
- 1- Active level is high

[6] Active polarity for High side PWM output
- 0- Active level is low
- 1- Active level is high

[8:7] Internal gain for current measurement
- 0- Internal gain is 1
- 1- Internal gain is 3
- 2- Internal gain is 6
- 3- Internal gain is 12

[9] Current sensing polarity
- 0- Non-Inverting
- 1- Inverting

[10] AC Voltage Sensing
- 0- Single ended sensing (external op-amp required)
- 1- Differential sensing (no op-amp, use two ADC channels)

[15:11] Reserved

3.3.1.2 PFC_SysConfig

Index	2		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 0xFFFF	Default: 0

Scaling or Notation: Bit field definitions are mentioned in description

Description: PFC system configuration parameter

[3:0] Execution rate for current control loop.

1- Current control loop executed every PWM period

2- Current control loop executed every 2 PWM period

…

15 - Current control loop executed every 15 PWM period

[4] Control mode selection

0- Voltage Control Mode

1- Tracking Control Mode

[5] Enable cycle-by-cycle over-current protection for boost mode PFC

0: Disable

1: Enable

[7:6] Configure PFC OCP comparator hysteresis

0: 0 mV

1: 10 mV

2: 15 mV

3: 20 mV

[15:8] Reserved

3.3.1.3 PFC_ SequencerState

Index	81		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max: 5	Default: 0

Scaling or Notation: See description

Description: This variable contains the current sequence state of the drive

0- Power on state

1- Stop state

2- Measuring offset current

4- PFC running

5- Fault state

3.3.1.4 PFC_Command

Index	82		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max: 1	Default: 0

Scaling or Notation: See description

Description: This variable controls the system state with the following values:

0- Stop the PFC

1- Start the PFC

3.3.2 PWM Register Group

3.3.2.1 PFC_PwmFreq

Index	3		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:1000	Default:6000
Scaling or Notation:	1 = 0.1 kHz F_{PWM} ; 1600 = 16kHz F_{PWM}		
Description:	This parameter configures the PWM frequency in 0.1 kHz increment.		

3.3.2.2 PFC_TMinOff

Index	4		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:PWMPeriod	Default: 0
Scaling or Notation:	$1 = 10.4167ns$		
Description	This parameter configures minimum PWM off time. Where : PWMPeriod is 96,000,000/(2*PWMFreq[Hz])		

3.3.2.3 PFC_Deadtime

Index	5		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max: 255	Default: 0
Scaling or Notation:	$1 = 10.4167ns$		
Description	This parameter configures PWM dead time value. This parameter is reserved for future use and should be always written 0.		

3.3.2.4 PFC_SHDelay

Index	6		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:960	Default: 0
Scaling or Notation:	$1 = 10.4167ns$		
Description:	SHDelay specifies the time delay from PWM output to ADC sample time for current sensing. The delay time is depending on the hardware design; usually it should consider propagation delay of gate driver circuit and turn on (turn off) delay of switching devices.		

3.3.3 Voltage Control Register Group

3.3.3.1 PFC_IRectLim

Index	7		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	4095 = 100% maximum measureable current		
Description:	This parameter specifies the maximum voltage PI positive output value which means allowable PFC rectifying current. Rectifying current is the energy direction from AC to DC. This limit should be set higher than maximum possible PFC current considering load condition, lowest AC voltage as well as some margin. The goal is never entering current limit unless there is hardware issue.		

3.3.3.2 PFC_IGenLim

Index	8		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	4095 = 100% maximum measureable current		
Description:	This parameter specifies the maximum voltage PI negative output value which means allowable PFC generating current. Generating current is the energy direction from DC to AC. This parameter is reserved for future PFC release and not actually working in current PFC release. Set this parameter to 0, or set to a small value (<100).		

3.3.3.3 PFC_VdcRampRate

Index	9		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description		
Description:	This parameter specifies the ramp rate of voltage reference. $VdcRampRate = \frac{PFC\_VdcRampRate * 0.001 * VadcRef}{4095 * VdcVoltageDividerRatio * 2^{16}}$, $in\ V/s$ Where: $VdcVoltageDividerRatio = \frac{Vdc\ Sensing\ Low\ Resister}{Vdc\ Sensing\ Low\ Resister + Vdc\ Sensing\ High\ Resister}$		

3.3.3.4 PFC_KpVreg

Index	10		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	U4.12		
Description:	This parameter specifies the proportional gain of the voltage regulator		

3.3.3.5 PFC_KxVreg

Index	11		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	U4.12		
Description:	This parameter specifies the integral gain of the voltage regulator		

3.3.3.6 PFC_TargetVoltInit

Index	28		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	4095 = Maximum measureable voltage		
Description:	This parameter defines the initial DC bus target voltage for fixed voltage mode. This parameter is copied to PFC_TargetVolt variable during startup.		

3.3.3.7 PFC_TargetVolt

Index	89		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	4095 = Maximum measureable voltage		
Description:	This is the reference input of voltage regulator. In tracking mode, this variable has no influence. $TargetVolt = \frac{PFC\_TargetVolt * VacRef * (Vdc\ Sensing\ High\ Resister + Vdc\ Sensing\ Low\ Resister)}{Vdc\ Sensing\ Low\ Resister * 4095}, V$		

3.3.3.8 PFC_VoltagePIoutput

Index	90		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	4095 = 100% maximum measureable current		
Description:	Voltage regulator output value		

3.3.4 Current Control Register Group

3.3.4.1 PFC_KpIreg

Index	12		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	U4.12		
Description:	This parameter specifies the proportional gain of the current regulator. Higher proportional gain improves PFC current waveform, but it may crease current oscillation if its value is too high, and/or AC voltage and PFC current sensing in PFC hardware has high noise.		

3.3.4.2 PFC_KxIreg

Index	13		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	U4.12		
Description:	This parameter specifies the integral gain of the current regulator		

3.3.4.3 PFC_AcDcScale

Index	24		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description		
Description:	This parameter defines the ratio of feedforward component added to the duty output and scaling between AC and DC voltage measurement. $PFC_{AcDCScale} = \text{AcDcScaleAdjustent} * \frac{(\text{Vac Sensing High Resister} + \text{Vac Sensing Low Resister}) * \text{Vdc Sensing Low Resister} * 2048}{\text{Vac Sensing Low Resister} * (\text{Vdc Sensing High Resister} + \text{Vdc Sensing Low Resister})}$ If high resistor and low resistor are the same value for AC voltage sensing and DC voltage sensing, PFC_AcDcScale=2048 represents 100% feedforward ratio. The ratio should be adjusted accordingly to achieve best PFC current waveform.		

3.3.4.4 PFC_LFactor

Index	25		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	See description		
Description:	This parameter is used to calculate the average current as current measurement is done at every peak current period. Average current calculation helps improve the PFC current waveform. Although the MCEWizard create the value for this parameter, due to many factors which affect the actual result, it may still need to be fine-tuned to achieve best PFC current waveform.		

3.3.4.5 PFC_CurrentPIoutput

Index	110		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max:PWM Period	Default: 0
Scaling or Notation:	PWMPeriod= 100% duty cycle		
Description:	Sum of output from current regulator and feed forward output values Where : PWMPeriod is 96,000,000/(2*PWMFreq[Hz])		

3.3.5 Protection Register Group

3.3.5.1 PFC_GateKillTime

Index	19		
Size	Unsigned 16 bit		
Parameter Type	STATIC		
Range	Min: 0	Max:960	Default:48
Scaling or Notation:	$1 = 10.4167ns$		
Description:	Persistence filter time for PWM gate kill input (in clock cycles)		

3.3.5.2 PFC_VacOvLevel

Index	20		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the AC over voltage trip level. AC over voltage fault will be generated if AC input voltage exceeds this threshold.		

3.3.5.3 PFC_VacLvLevel

Index	21		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the AC under voltage trip level. AC bus under trip voltage fault will be generated if AC input voltage falls below this threshold.		

3.3.5.4 PFC_VdcOvLevel

Index	22		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the dc bus over voltage trip level. A dc bus over voltage fault will be generated if dc bus voltage exceeds this threshold.		

3.3.5.5 PFC_VdcLvLevel

Index	23		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts.		
Description:	This parameter defines the dc bus under voltage trip level. A dc bus under trip voltage fault will be generated if dc bus voltage falls below this threshold.		

3.3.5.6 PFC_FaultEnable

Index	26		
Size	Unsigned 16 bit		
Parameter Type	DYNAMIC		
Range	Min: 0	Max:0xFFFF	Default: 0
Scaling or Notation:	Bit field definitions are mentioned in description. For each bit, 0 – Ignore the associated fault; 1 – enable processing of the associated fault.		
Description:	This parameter specifies enable/disable of faults are mentioned below [0] Reserved, must be set to “0“ [1] Enable DC bus under voltage fault [2] Enable DC bus over voltage fault [3] Reserved, must be set to “0“ [4] Enable AC under voltage fault [5] Enable AC over voltage fault [15:6] Reserved, must be set to “0“ When a fault is disabled (bit set to “0“), the fault condition is ignored and the motor keeps running. However, even when a fault is disabled, its occurrence is reported in the FaultFlags variable, until the condition that caused the fault disappears.		

3.3.5.7 PFC_FaultClear

Index	85		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:1	Default: 0
Scaling or Notation:	See Description		
Description:	Writing 1 to this variable clears all faults. Once clear has been done, the variable will be cleared. If fault condition doesn’t exist, fault clear will be successful and the drive will enter STOP state. If fault condition still exists, the drive will remain in fault state.		

3.3.5.8 PFC_SwFaults

Index	87		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:255	Default: 0
Scaling or Notation:	See description.		
Description:	This variable is derived from FaultFlags by the following bitwise logical operation: $PFC\_SwFaults = PFC\_FaultFlags \cdot PFC\_FaultEnable$ SwFaults is cleared by PFC_FaultClear. For bit field definition, refer to PFC_Faultflags.		

3.3.5.9 PFC_FaultFlags

Index	104		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:0xFFFF	Default: 0
Scaling or Notation:	Bit field definitions are mentioned in description		
Description:	This variable provides drive fault status. Most faults are handled by a fault handling routine operating at the PWM inverter switching frequency with the exception of Gate Kill faults. Gate Kill is handled within the Faults module and will instantly initiate inverter and regulator shutdown. The FaultFlags variable indicates currently pending fault conditions. The FaultClear variable is used to reset fault conditions. For all bit fields defined below, a value of 1 indicates that the corresponding fault condition has occurred. [0] PFC gate kill fault [1] DC bus under voltage fault [2] DC bus over voltage fault [3] Vac frequency fault [4] Vac under voltage fault [5] Vac overvoltage fault [11:6] Reserved [12] Parameter load fault [15:13] Reserved Gate kill fault and Vac Frequency fault cannot be masked by PFC_FaultEnable		

3.3.6 Measurement Register Group

3.3.6.1 PFC_VdcRaw

Index	92		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the measured DC bus voltage value. This value is updated every PWM cycle.		

3.3.6.2 PFC_VdcFilt

Index	99		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the filtered DC bus voltage value.		

3.3.6.3 PFC_IpfcRaw

Index	93		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the Ipfc current raw value.		

3.3.6.4 PFC_IpfcAvg

Index	105		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the Ipfc average current value.		

3.3.6.5 PFC_IpfcRMS

Index	106		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the Ipfc current RMS value.		

3.3.6.6 PFC_VacRaw

Index	103		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the Vac raw voltage value.		

3.3.6.7 PFC_AbsVacRaw

Index	94		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the Vac absolute voltage value.		

3.3.6.8 PFC_VacRMS

Index	98		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:4095	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the Vac RMS voltage value.		

3.3.6.9 PFC_ACPower

Index	107		
Size	Unsigned 16 bit		
Variable Type	Read only		
Range	Min: 0	Max:65535	Default: 0
Scaling or Notation:	In ADC counts		
Description:	This variable provides the AC input power value.		

3.4 Script Register (App ID = 4)

Note: To access script related registers from MCEDesigner, users shall use App ID = 0, and relevant index numbers shall be offseted by 128. To access script related registers from script code, JCOM or user UART interfaces, users shall use App ID = 4.

Complete list of variables are listed in the Table 31 and find description in the following chapters.

Table 31 Script Variable list

App ID	Index	Variable Name	Type	Description
4	0	Script_UserVersion	Read Only	Holds script user version configured in script input file
4	1	Script_Command	Read Write	Controls the script state – Stop or start
4	98	ADC_Result0	READONLY	Holds AIN0 analog input value (12 bit value)
4	99	ADC_Result1	READONLY	Holds AIN1 analog input value (12 bit value)
4	100	ADC_Result2	READONLY	Holds AIN2 analog input value (12 bit value)
4	101	ADC_Result3	READONLY	Holds AIN3 analog input value (12 bit value)
4	102	ADC_Result4	READONLY	Holds AIN4 analog input value (12 bit value)
4	103	ADC_Result5	READONLY	Holds AIN5 analog input value (12 bit value)
4	104	ADC_Result6	READONLY	Holds AIN6 analog input value (12 bit value)
4	105	ADC_Result7	READONLY	Holds AIN7 analog input value (12 bit value)
4	106	ADC_Result8	READONLY	Holds AIN8 analog input value (12 bit value)
4	107	ADC_Result9	READONLY	Holds AIN9 analog input value (12 bit value)
4	108	ADC_Result10	READONLY	Holds AIN10 analog input value (12 bit value)
4	109	ADC_Result11	READONLY	Holds AIN11 analog input value (12 bit value)
4	110	GPIO_IN_L	READONLY	Holds digital input/output (GPIO0 to GPIO15) pins values.
4	111	GPIO_IN_H	READONLY	Holds digital input/output (GPIO16 to GPIO29) pins values.
4	112	GPIO_OUT_L	READWRITE	Set or reset digital output pin (GPIO0 to GPIO15)
4	113	GPIO_OUT_H	READWRITE	Set or reset digital output pin (GPIO16 to GPIO29)

3.4.1 Script_UserVersion

Index	0		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max:0xFFFF	Default: 0

Scaling or Notation: Bit field defined

Description: Script user version scheme – 8:8 Bit Coding. This variable only can be read from MCEDesigner or User UART interface.

[7:0] Minor version

[15:8] Major version

3.4.2 Script_Command

Index	1		
Size	Unsigned 16 bit		
Variable Type	Read Write		
Range	Min: 0	Max:3	Default: 3
Scaling or Notation:	See description		
Description:	This variable controls the system state with the following values: 0x00 : Stop Task0 and Task1 script function 0x01 : Start Task0 script function and stop Task1 script function 0x02 : Stop Task0 script function and start Task1 script function 0x03 : Start Task0 and Task1 script function		

3.4.3 ADC_Resultx [x: 0 to 11]

Index	98 – 109		
Size	Unsigned 16 bit		
Variable Type	Read Only		
Range	Min: 0	Max:0xFFF	Default: 0
Scaling or Notation:	In ADC counts, InputVoltage[v] = ADC_Result*VDD[v]/0xFFF		
Description:	These variable holds the configured user ADC pin value. These variables value are read by MCE every 1 mS		

3.4.4 GPIO_IN_L

Index	110		
Size	Unsigned 16 bit		
Parameter Type	Read Only		
Range	Min: 0	Max: 0xFFFF	Default: 0

Scaling or notation: Bitfield

Description: This variable holds the configured user GPIO value. This variable value are read by MCE every 1 mS

[0] Holds GPIO0 pin value, 0- input is low, 1- input is high
[1] Holds GPIO1 pin value, 0- input is low, 1- input is high
[2] Holds GPIO2 pin value, 0- input is low, 1- input is high
[3] Holds GPIO3 pin value, 0- input is low, 1- input is high
[4] Holds GPIO4 pin value, 0- input is low, 1- input is high
[5] Holds GPIO5 pin value, 0- input is low, 1- input is high
[6] Holds GPIO6 pin value, 0- input is low, 1- input is high
[7] Holds GPIO7 pin value, 0- input is low, 1- input is high
[8] Holds GPIO8 pin value, 0- input is low, 1- input is high
[9] Holds GPIO9 pin value, 0- input is low, 1- input is high
[10] Holds GPIO10 pin value, 0- input is low, 1- input is high
[11] Holds GPIO11 pin value, 0- input is low, 1- input is high
[12] Holds GPIO12 pin value, 0- input is low, 1- input is high
[13] Holds GPIO13 pin value, 0- input is low, 1- input is high
[14] Holds GPIO14 pin value, 0- input is low, 1- input is high
[15] Holds GPIO15 pin value, 0- input is low, 1- input is high

3.4.5 GPIO_IN_H

Index	111		
Size	Unsigned 16 bit		
Parameter Type	Read Only		
Range	Min: 0	Max: 0x3FFF	Default:0

Scaling or notation: Bitfield

Description: This variable holds the configured user GPIO value. This variable value are read by MCE every 1 mS

[0]	Holds GPIO16 pin value, 0- input is low, 1- input is high
[1]	Holds GPIO17 pin value, 0- input is low, 1- input is high
[2]	Holds GPIO18 pin value, 0- input is low, 1- input is high
[3]	Holds GPIO19 pin value, 0- input is low, 1- input is high
[4]	Holds GPIO20 pin value, 0- input is low, 1- input is high
[5]	Holds GPIO21 pin value, 0- input is low, 1- input is high
[6]	Holds GPIO22 pin value, 0- input is low, 1- input is high
[7]	Holds GPIO23 pin value, 0- input is low, 1- input is high
[8]	Holds GPIO24 pin value, 0- input is low, 1- input is high
[9]	Holds GPIO25 pin value, 0- input is low, 1- input is high
[10]	Holds GPIO26 pin value, 0- input is low, 1- input is high
[11]	Holds GPIO27 pin value, 0- input is low, 1- input is high
[12]	Holds GPIO28 pin value, 0- input is low, 1- input is high
[13]	Holds GPIO29 pin value, 0- input is low, 1- input is high
[15:14]	Reserved

3.4.6 GPIO_OUT_L

Index	112		
Size	Unsigned 16 bit		
Parameter Type	Read Write		
Range	Min: 0	Max :0xFFFF	Default: 0

Scaling or Notation: Bitfield

Description: This variable holds the configured user GPIO value. This variable value are read by MCE every 1 mS

[0]	Set or Reset GPIO0 pin, 0- Reset (low), 1- Set(High)
[1]	Set or Reset GPIO1 pin, 0- Reset (low), 1- Set(High)
[2]	Set or Reset GPIO2 pin, 0- Reset (low), 1- Set(High)
[3]	Set or Reset GPIO3 pin, 0- Reset (low), 1- Set(High)
[4]	Set or Reset GPIO4 pin, 0- Reset (low), 1- Set(High)
[5]	Set or Reset GPIO5 pin, 0- Reset (low), 1- Set(High)
[6]	Set or Reset GPIO6 pin, 0- Reset (low), 1- Set(High)
[7]	Set or Reset GPIO7 pin, 0- Reset (low), 1- Set(High)
[8]	Set or Reset GPIO8 pin, 0- Reset (low), 1- Set(High)
[9]	Set or Reset GPIO9 pin, 0- Reset (low), 1- Set(High)
[10]	Set or Reset GPIO10 pin, 0- Reset (low), 1- Set(High)
[11]	Set or Reset GPIO11 pin, 0- Reset (low), 1- Set(High)
[12]	Set or Reset GPIO12 pin, 0- Reset (low), 1- Set(High)
[13]	Set or Reset GPIO13 pin, 0- Reset (low), 1- Set(High)
[14]	Set or Reset GPIO14 pin, 0- Reset (low), 1- Set(High)
[15]	Set or Reset GPIO15 pin, 0- Reset (low), 1- Set(High)

3.4.7 GPIO_OUT_H

Index	113		
Size	Unsigned 16 bit		
Parameter Type	Read Write		
Range	Min: 0	Max: 0x3FFF	Default: 0

Scaling or Notation: Bit field

Description: This variable holds the configured user GPIO value. This variable value are read by MCE every 1 ms

[0]	Set or Reset GPIO16 pin, 0- Reset (low), 1- Set(High)
[1]	Set or Reset GPIO17 pin, 0- Reset (low), 1- Set(High)
[2]	Set or Reset GPIO18 pin, 0- Reset (low), 1- Set(High)
[3]	Set or Reset GPIO19 pin, 0- Reset (low), 1- Set(High)
[4]	Set or Reset GPIO20 pin, 0- Reset (low), 1- Set(High)
[5]	Set or Reset GPIO21 pin, 0- Reset (low), 1- Set(High)
[6]	Set or Reset GPIO22 pin, 0- Reset (low), 1- Set(High)
[7]	Set or Reset GPIO23 pin, 0- Reset (low), 1- Set(High)
[8]	Set or Reset GPIO24 pin, 0- Reset (low), 1- Set(High)
[9]	Set or Reset GPIO25 pin, 0- Reset (low), 1- Set(High)
[10]	Set or Reset GPIO26 pin, 0- Reset (low), 1- Set(High)
[11]	Set or Reset GPIO27 pin, 0- Reset (low), 1- Set(High)
[12]	Set or Reset GPIO28 pin, 0- Reset (low), 1- Set(High)
[13]	Set or Reset GPIO29 pin, 0- Reset (low), 1- Set(High)
[15:14]	Reserved

4 Motor Tuning

MCEWizard で、システム構成に関する入力情報に基づき、ハードウエア変数、モータパラメータ、制御パラメータおよび特性、保護機能のパラメータおよび特性およびシステム全体の特性を計算します。MCEWizard でパラメータファイルを生成することが、ユーザがモータを駆動する前に行う最初のステップです。

モータパラメータを正しく設定することは、センサレスベクトル制御により定常状態でモータを駆動するために重要です。MCE はモータを容易に起動させることが可能な、先進的な磁束ベースのセンサレスアルゴリズムを使用しています。モータが起動できたとしても、アプリケーションの要求によっては、実際の負荷条件でモータの起動特性と動的特性を調整する必要があります。

この後、いくつかの主な課題と、それに対する MCE を用いたチューニング方法について記述します。

4.1 電流計測設定が良好かを判断する方法

電流計測設定を確認するには、無負荷状態でモータシステムの設定を行い、モータを始動してモータがスムーズに回る速度を設定します。オシロスコープを用いてモータの RMS 電流値を計測します。MCEDesigner で、出力電流の表示は計測ノイズのために若干高めに出る傾向があります。しかし、MCEDesigner に示される値は、測定されたモータ電流に可能な限り近くする必要があります。

電流計測ノイズが良好でない場合は、下記の可能性について確認してください。

- PCB レイアウトが適正でない
- パワー素子のスイッチングが早すぎて、多くのノイズを発生させている
- 電流計測の変数がハードウエアに対して合っていない。関係する変数は下記になります。
    1. デッドタイム
    2. PwmGuardBand (3 シャントの場合)
    3. TcntMin (1 シャントの場合)
    4. SHDelay
    5. TMinPhaseShift (1 シャントの場合)

1 シャント構成においては、位相シフト PWM 方式を用いることで制御特性が改善します。TMinPhaseShift と SHDelay が、位相シフト PWM 方式で良好な電流計測を行うための重要な変数となります。

良好な 1 シャント電流計測信号を得るために、TMinPhaseShift と SHDelay は下記のガイドラインに従って調整される必要があります。

$$TMinPhaseShift > Dead\ time + Ringing + ADC\ sampling\ time$$

$$SHDelay < Hardware\ delay\ time - ADC\ sampling\ time$$

$$TMinPhaseShift + SHDelay > Hardware\ delay\ time + Dead\ time + Ringing$$

TMinPhaseShift は音響ノイズを発生させる可能性があるため、出来るだけ小さい値に設定してください。

図 82　位相シフト PWM、1 シャント構成での電流測定タイミング

上の数式には 3 つのタイミング変数があります。デッドタイム、ハードウエア遅延、リンギング時間です。デッドタイムは MCEWizard に設定されている既知の値です。ハードウエア遅延とリンギング時間については、実際のハードウエアボードを用いて計測する必要があります。

適切な TMinPhaseShift と SHDelay の設定例

下記に、ハードウエアの測定方法と、これら 2 つの変数の設定方法の事例を示します。

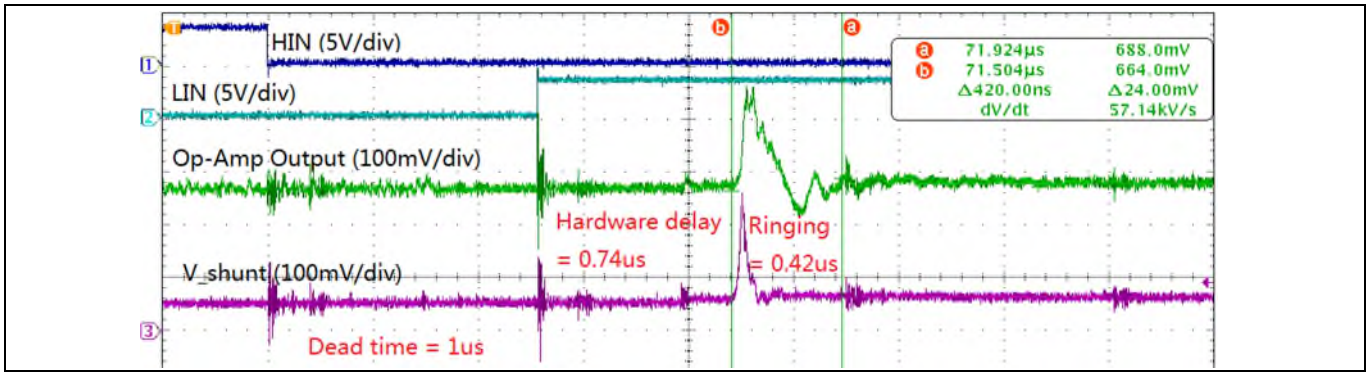


図 83　ハードウエア遅延とリンギング時間測定結果

$$TMinPhaseShift > 1us + 0.42us = 1.42us$$

$$SHDelay < 0.74us$$

$$TMinPhaseShift + SHDelay > 0.74us + 1us + 0.42us = 2.16us$$

上記の数式を満足させるために、TMinPhaseShift=2.2 us，SHDelay=0 と容易に設定することも可能です。しかし、位相シフト PWM 方式で音響ノイズを最小化するためには、TMinPhaseShift の値を出来る限り小さくする必要があります。従って、下記のように値を設定するのも一つの方法です。

$$TMinPhaseShift = 1.6us$$

$$SHDelay = 0.6us$$

4.2　電流コントローラのチューニング

MCE 電流コントローラはベクトル制御方式を用いた、同期回転フレーム型コントローラを採用しています。ベクトル制御は電流ループの動的制御性を大幅に簡略化できます。電流制御系には 2 つのコントローラ(d チャネルと q チャネル)があります。q チャネル(トルク)と d チャネル(磁束)の制御構造は同一です。d チャネルの電流制御方式を図 84 に示します。モータ巻線は時定数 L/R をもつ一次遅れ系で表現

されます。この時定数はモータのインダクタンスと等価抵抗(ケーブルと巻線の抵抗値の和)より求められます。表面磁石同期モータ(SPM, surface mounted permanent magnet motor)では、d チャネルと q チャネルのインダクタンスはほぼ等しくなります。埋込磁石同期モータ(IPM, interior permanent magnet motor)では、通常 q チャネルのインダクタンスは d チャネルよりも大きくなります。

図 84 に示される電流制御の連続時間領域モデルでは、フォワードゲイン A はデジタルコントローラ出力変換(インバータゲインを含む)をモデル化したものであり、フィードバックゲイン B は電流フィードバック値(アンペア)を A/D コンバータ経由で内部のデジタル値へ変換することをモデル化したものです。PI 補償器のゲイン(KIIreg, Kpireg_D)計算は、図 84 に示される極零相殺を用いています。電流コントローラは伝達関数ブロック C(s)を与えるよう再配置されます。C(s)の KpIreg_D および KIIreg がモータの時定数($\tau = L/R$)と等しくなるよう設定することで、コントローラのゼロ極がモータの極を相殺します(極零相殺)。そのため、コントローラモデルは図 86 のように更に簡略化できます。図 86 の等価伝達関数は、時定数τ_cをもつ一次遅れ系になります。対象となるアプリケーションに対して適切な電流コントローラ応答(通常 1~5 ms)を選択することにより、電流コントローラのゲインを容易に得ることが出来ます。極零相殺を使用するに当たっては、モータのインダクタンスを比例ゲイン計算に、モータ抵抗を積分ゲイン計算に使用する必要があります。

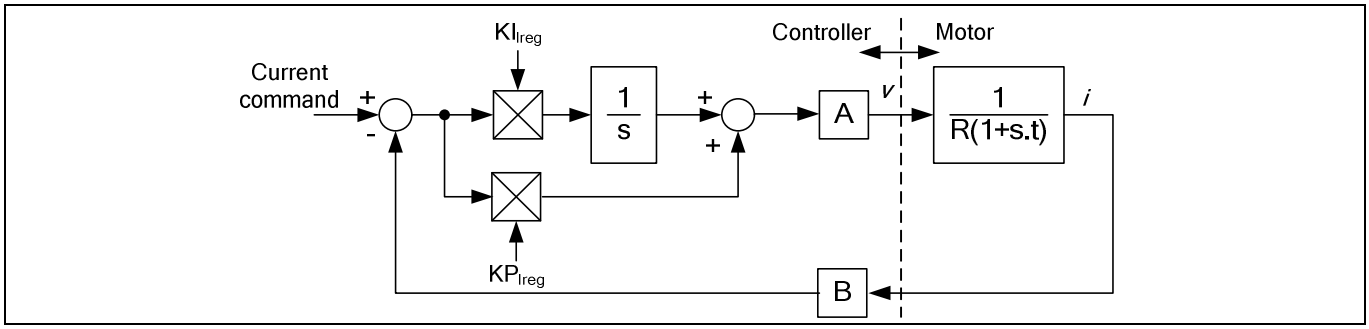


図 84　電流制御モデル

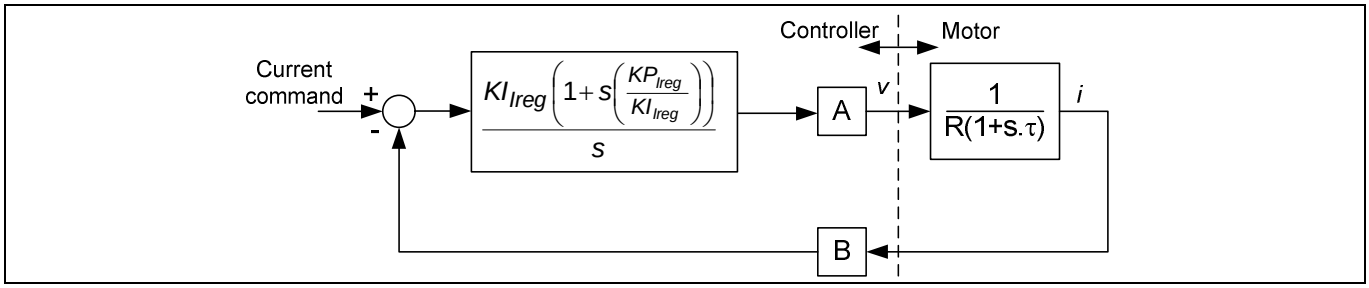


図 85　極零相殺

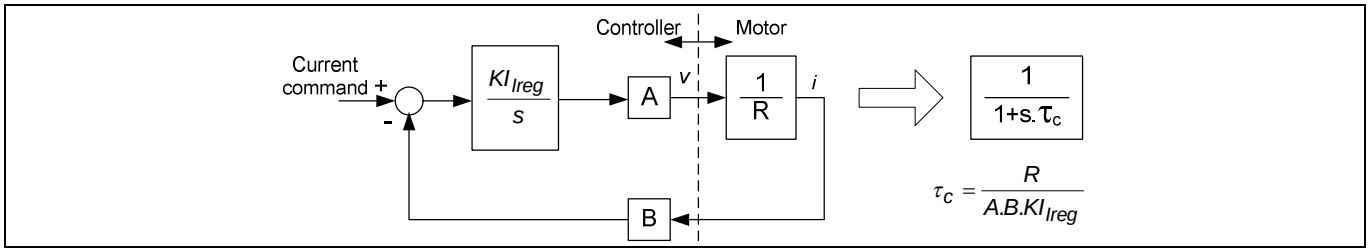


図 86　極零相殺後の簡略化された電流制御モデル

極零相殺に基づき、連続時間領域モデルにおけるコントローラのゲインは下記のようになります。

$$Kp_{Ireg} = \frac{L_q \cdot CurrentRegBW}{A \cdot B}$$

$$KI_{Ireg} = \frac{R \cdot CurrentRegBW}{A \cdot B}$$

ここで A および B は電圧および電流のスケーリング値となります。

デジタル制御への実装に当たっては、積分器はデジタルアキュムレータであり、PI 補償器に対する離散時間領域モデルを積分器に用いる必要があります。この場合、デジタル積分器のゲイン KXIreg は積分器のサンプリング時間に対するスケーリング係数を含む必要があります。

$$Kx_{Ireg} = KI_{Ireg} \cdot T$$

ここで、T はコントローラのサンプリング時間であり、この場合は PWM 周期と等しくなります。

電圧スケーリング値 A は正転および空間ベクトル変調器のゲインを考慮する必要があります。3 相インバータは DC バス電圧 Vdc に等しいピークライン電圧を生成するため、デューティ比 100%の場合の rms 相電圧は$\frac{V_{dc}}{\sqrt{2}\sqrt{3}}$となります。変調器が 100%のデューティを出力するときのデジタル入力値は 8192 であり、正転機能は 1.64676 のゲインを持ちます。従って、電流ループでの電圧スケーリング値 A は下記の数式で与えられます。

$$A = \frac{V_{dc}/\sqrt{6}}{8192/1.64676}\ (in\ V_{rms}/cts)$$

電流ループフィードバックのスケーリング値 B は、シャント抵抗、アンプのゲイン、A/D コンバータゲイン、電流フィードバックのスケーリングパラメータ IfbksScl より求められます。しかしながら、MCEWizard は 4096 カウントがモータの定格 rms 電流となるように IfbScl を計算します。従って、電流ループフィードバックのスケーリング値は下記のように簡略化されます。

$$B = \frac{4096}{I_{RATED}}\ (in\ cts/A_{rms})$$

電流制御ループの制御ゲインは通常 1 より小さい値を取るので、電流制御ループの PI コントローラはコントローラのゲインの精度向上のため Kp と Kx 入力に逓倍後のスケーリングを含んでいます。Kp 入力に対しては 14 bit のシフト乗算、Kx に対しては 19 bit のシフト乗算処理が行われます。そのため、このデジタル制御系のゲインは、次式にて与えられます。

$$Kp_{Ireg} = \frac{L_q \cdot CurrentRegBW \cdot 2^{14}}{A \cdot B}$$

$$Kx_{Ireg} = \frac{R \cdot CurrentRegBW \cdot T \cdot 2^{19}}{A \cdot B}$$

電流コントローラのステップ応答は電流制御モードを用いることで測定可能です。電流制御モードへ移行するために、下記のステップを実行してください。

Step1 ロータを 0°の位置へ停止させます

a. モータを接続し、オシロスコープで U 相電流を測定してください。
b. AngleSelect = 0: センサレスの角度推定器から切り離し、内部のオープンループ角度を使用します。
c. CtrlModeSelect = 1: これにより電流制御モードに設定され、スピードコントローラは無効化されます。
d. TargetSpeed = 0: オープンループ角度の回転速度を 0 に設定し、試験中角度は 0 に固定されます。
e. Idref = 1024: D 軸に定格電流の 25%を与えます。
f. Command = 1: モータが起動します。電流が 0°の状態に制御され、ロータは 0°に停止します。電流はは U 相からモータに流れこみ、V 相および W 相から戻ってきます。

ロータが動かない状態でステップ応答を観測します。Step 1 でロータを特定の角度に停止しているので、次のステップではロータが動かないようにしなければなりません。負荷の慣性が大きい(ファンブレード等)場合、ロータは停止位置で振動し、振動が収まるまでに長い時間を要します。可能であれば、手で振動を止め、0°で停止するようにしてください。

Step 2 10%の Id 電流を設定

a. Idref = 410: D 軸に定格電流の 10%を与えます。

Step 3 50%の Id 電流を設定

a. Idref = 2048: Id を 50%にステップ状に変化させます。

このステップ応答が観察対象です。U 相電流波形をオシロスコープで取得してください。

a. Command = 0: モータを停止させます。
b. AngleSelect =2: センサレスモードへ移行します。
c. CtrlModeSelect = 2: 速度制御モードへ設定します。

図 87 に、異なる電流コントローラのバンド幅設定におけるステップ応答の測定結果を示します。ステップ応答の時定数は、電流がゼロから最終値(オーバーシュートは含まない)の 63.2%($1 - 1/e$)に達するまでの時間で定義されます。

電流コントローラのバンド幅が小さい場合、実際のステップ応答の時定数は理論値に近くなります(実測 9.88 ms/理論値 10 ms, 実測 4.84 ms/理論値 5 ms, 実測 2.4 ms/理論値 2.5 ms)。電流コントローラのバンド幅が大きくなると、実際の時定数は理論値よりもはるかに小さくなり(実測 1.02 ms/理論値 1.25 ms、実測 0.428 ms/理論値 0.625 ms)、オーバーシュートが発生するようになります。より良好なステップ応答特性を得るために、電流コントローラのバンド幅が大きい場合にはKx_{Ireg}を小さくすることを推奨します。

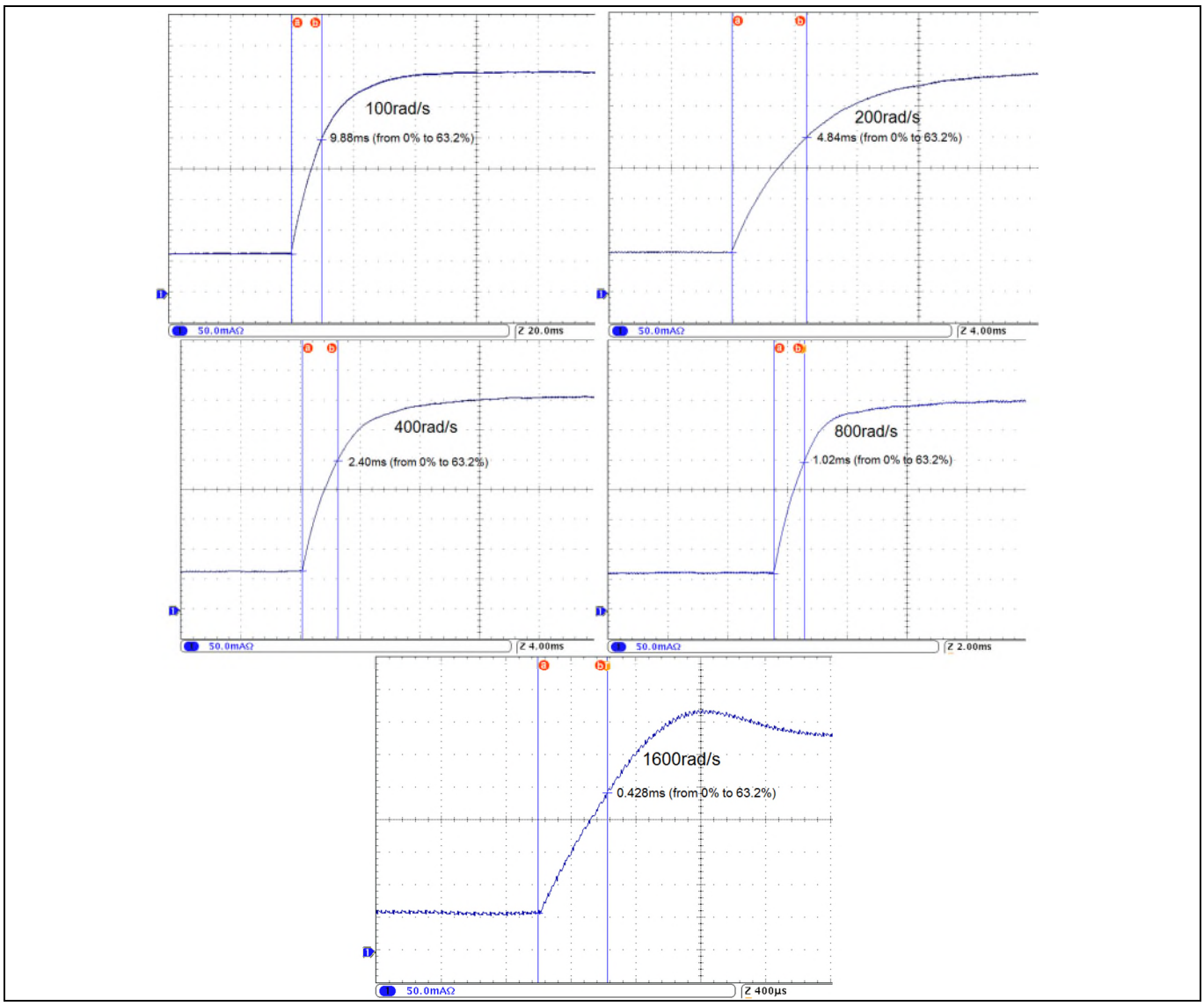


図 87　電流コントローラのステップ応答 (100/200/400/800/1600rad/s)

4.3　モータが起動しない

- 電流計測が良好かを確認してください
- モータパラメータが正しいかを確認してください。
- 速度制御系の PI ゲインと、速度フィードバック系のフィルタ時定数(speed feedback filter time constant)を確認してください。
- 最低速度(Minimum speed)を調整してください。
- 加速/減速レート(ramp rate)を調整してください。
- 磁束推定器の時定数(flux estimator time constant)を調整してください。
- モータ電流のリミット値を上げてください。

4.4　モータ速度が安定しない

- 速度が低速で不安定な場合は、電流計測が良好かを確認してください。
- モータ速度が振動する場合は、速度コントローラの PI ゲイン(特に I ゲイン)を減らしてください
- 負荷変動に対してモータ速度の変化が急激すぎる場合は、速度コントローラの PI ゲイン(特に P ゲイン)を増加してください。

- 2 相 PWM を有効にしている場合は、3 相/2 相の変化点(速度)を出来るだけ上げるか、一時的に 2 相 PWM を無効にしてください。

4.5 弱め界磁使用時にモータ電流が安定しない

- 速度コントローラの PI ゲインと、FwkKx を調整してください。
- 電流コントローラの PI ゲインを調整してください。弱め界磁モードでは、D 軸電流コントローラのバンド幅を Q 軸より大きくし、KpIregD を KpIreg の 2 倍以上にしてください。

4.6 音響ノイズの低減

音響ノイズには様々な要因があります。主要な要因とそれに対する対策をいくつか示します。

- 要因: 電流計測回路起因のノイズ。
  対策: PCB レイアウトの最適化、オペアンプのロードキャパシタの調整、電流検出回路のパラメータの調整等、電流検出回路を改善してください。
- 要因: 電流コントローラのバンド幅が高い。電流測定系は常にノイズの発生要因となります。不適切なバンド幅設定はノイズを増幅させます。
  対策: 電流コントローラの PI ゲインを減らしてください。この際に、制御特性(特に起動と高負荷時)が良好化を確認してください。
- 要因: PWM 周波数が低い、あるいは 2 相 PWM 時のノイズ。
  対策: PWM 周波数を上げてください。ハードウエアが高い PWM 周波数に対応していない場合は、2 相 PWM を無効化して常に 3 相 PWM を用いてください。
- 要因: 最小パルス幅 PWM 方式あるいは位相シフト PWM 方式時のノイズ(1 シャント構成の場合)
  対策: 最小パルス幅方式で発生するノイズは、TCntMin の値を小さくすることで抑制可能です。位相シフト PWM 方式で発生するノイズは、TMinPhaseShift の値を小さくすることで抑制可能です。どちらの場合も、SHDelay を最適化する必要があります。アプリケーションが非常に小さいノイズレベルを要求する場合は、1 シャント構成ではノイズを削減することが出来ませんが、3 シャント構成にすることで問題を解決できる可能性があります。
- 要因: 過変調に起因するノイズ。モータが高速運転しているときには、過変調を利用することで DC バスの利用率を最大化できます。
  対策: 過変調を無効化してください。

5 改訂履歴

版数	リリース日	変更内容
1.3	2020/04/26	‘MotorStatus’および‘AppConfig’パラメータの説明を更新。 ‘SysTaskConfig’パラメータの説明を追加。 セクション 2.1.8.7（制御入力のカスタマイズ）を追加。 セクション 2.1.6.3（モータ過電流保護）を更新。 セクション 2.2.2.1（PFC 過電流保護）を更新。 レジスタグループ（セクション 3.2.1.1）を修正。 セクション 3.4（スクリプトレジスタ）を更新。 セクション 2.6.11.3（設定可能な UART ドライバの方法）を追加。 セクション 2.6（スクリプトエンジン）を更新。 セクション 3.2.6（磁束推定器 PLL レジスタグループ）を更新。 新しいレジスタグループ（セクション 3.1.3、セクション 3.2.8.9、セクション 3.2.8.10、セクション 3.2.8.11、セクション 3.2.5.12、セクション 3.2.5.13、セクション 3.2.5.14、セクション 3.2.5.15、セクション 3.2.5.16、セクション 3.2.5.17、およびセクション 3.2.14）を追加。 セクション 2.1.6（電流測定）を修正し、電流センシングタイミングの説明とセクション 2.1.4.2.4（位相シフトウインドウ幅が無い場合のピーク電流トラッキング）を含めた。 セクション‘デューティモード制御’を削除。 セクション‘デューティモード制御レジスタグループ’を削除。 セクション 2.1.13（ホールセンサインターフェイス）を修正して、ホール PLL および Atan 角度の説明を含めた。 セクション 3.2.5.1（HallAngleOffset）を修正。 セクション 2.1 を修正。 セクション 2.1.6（DC バス補償）を追加。 セクション 2.3.1（ボーレート）を追加。 セクション 2.1.8（モータ電流制限プロファイル）を修正。 セクション 2.1.9（初期角度センシング）を追加。 セクション 2.1.10（過変調）を追加。 セクション 2.1.11（2 相変調）を追加。 セクション 2.1.16.1 を修正。 セクション 2.1.16.2 を修正。 セクション 2.1.16.4 を修正。 セクション 2.1.16.5 を修正。 セクション 2.1.16.6 を修正。 セクション 2.1.15（トルク補償）を追加。 セクション 2.1.3（電流センシングオフセット測定）を追加。 セクション 2.1.1（可変スケーリング）を追加。 可変スケーリング表を追加。

改訂履歴

版数	リリース日	変更内容
1.2	2019/06/05	セクション 2.1.4.1.3（ローノイズ位相シフト PWM）を追加。 セクション 2.1.9（デューティモード制御）を追加。 セクション 2.1.10（ホールセンサインターフェイス）を追加。 セクション 2.4（JCOM チップ間通信）を追加。 新しいレジスタグループ（セクション 3.2.4、セクション 3.2.5 など）を追加。
1.1	2018/08/01	スクリプトエンジンの関数の説明を追加 (セクション 2.5 とセクション 3.4)。 モータ制御と PFC 制御の説明を追加 (セクション 2.1.6 とセクション 2.2.2)。 下記の 2 つのパラメータを追加。FaultRetryPeriod と PFC_TargetDCVolt (セクション 3.2 とセクション 3.3) IdFwk 変数タイプを READONLY（読み取り専用）に変更。 PFC_HalfCycleMin パラメータと PFC_HalfCycleMax パラメータのタイプを静的パラメータに変更。
1.0	2018/02/09	バージョン-のリリース。センサレスベクトルおよび PFC の機能およびレジスタの説明。

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-4-26

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

E メール：erratum@infineon.com

Document reference

ifx1

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.