



## TMC5241

65V 2ARMS のスマート集積ステッピング・  
モーター・ドライバおよびコントローラ

## 概要

TMC5241 は、スマートな高性能ステッピング・モーター・コントローラおよびドライバ IC で、シリアル通信インターフェース (SPI、UART) および広範な診断機能を備えています。このデバイスは、ジャークを最適化した柔軟なランプ・ジェネレータと、256 マイクロステップ、組み込みインデクサ、65V、3.0AMAX の全機能内蔵 H ブリッジ、および電力散逸のない内蔵電流検出 (ICS) 機能をベースとした業界最先端のステッピング・モーター・ドライバを組み合わせることで、自動ターゲット・ポジショニングを実現します。

ADI-Trinamic の精巧な StealthChop2 チョッパーにより、最大効率と最大モーター・トルクを両立した超低ノイズ動作が可能となります。

高集積、高エネルギー効率、小さなフォーム・ファクタという長を備えているため、コスト効率の高いソリューションに適した小型でスケラブルなシステムを実現します。このフル機能ソリューションにより、学習曲線を最小限に抑えたと共に、最高クラスの性能を可能にできます。H ブリッジ電界効果トランジスタ (FET) は、インピーダンスが非常に低いため、駆動効率が高く、発生する熱を最小限に抑えます。代表的な合計  $R_{ON}$  (ハイサイド+ローサイド) は  $0.31\Omega$  です。

H ブリッジあたりの最大実効値電流は、4層 PCB を前提として、室温で電源電圧  $V_S = 24V$  のときに  $I_{RMS} = 2A_{RMS}$ 、 $V_S = 48V$  のときに  $I_{RMS} = 1.7A_{RMS}$  です。

熱による制限を受けるため、実際の最大実効値電流は、PCB のグラウンド・プレーン、ヒートシンク、換気など、アプリケーションの熱特性に依存します。

H ブリッジあたりの最大フルスケール電流は  $I_{FS} = 3.0AMAX$  で、 $I_{REF}$  に接続された外付け抵抗で設定できます。この電流は、エンベデッド電流駆動レギュレーション回路の最大電流設定値として定義されます。

H ブリッジあたりの最大出力電流は、過電流保護 (OCP) により  $I_{OCP} = 5.0A$  に制限されます。

電力散逸のない ICS を備えていることから、大きな外付けパワー抵抗が不要となり、外部センス抵抗を用いる主流のアプリケーションと比べて同じ全体精度を実現しながら、スペースと消費電力を大幅に削減できます。

TMC5241 は、豊富な診断機能と、短絡保護/OCP、サーマル・シャットダウン、低電圧ロックアウト (UVLO) などの保護機能を備えています。

サーマル・シャットダウン・イベントと UVLO イベントの際には、ドライバはディスエーブルされます。更に、TMC5241 には、ドライバ温度の測定、モーター温度の推定、1 つの外部アナログ入力の測定を行う機能が備わっています。

TMC5241 は、露出パッド付き  $5mm \times 7mm$  小型 TQFN38 パッケージを採用しています。

## アプリケーション

- テキスタイル、ミシン、編み機
- ラボ・オートメーションおよびファクトリ・オートメーション
- ID プリンタ/カード・プリンタ
- リキッド・ハンドリング、医療用アプリケーション
- オフィス・オートメーションおよびペーパー・ハンドリング
- POS、マッサージ・チェア
- ATM、現金回収機、紙幣識別機、現金自動支払機
- CCTV、セキュリティ
- ポンプおよびバルブ制御
- ヘリオスタットやアンテナのポジショニング
- 舞台照明

## 機能と利点

- 電圧範囲 :  $4.5V \sim 65V$  DC
- 低  $R_{ON}$  (HS + LS) :  $0.31\Omega$  (代表値、 $T_A = 25^\circ C$ )
- H ブリッジあたりの電流定格 ( $25^\circ C$  での代表値) :
  - $V_S = 24V$  時に  $I_{RMS} = 2A_{RMS}$  (2.8A のサイン波ピーク)
  - $V_S = 48V$  時に  $I_{RMS} = 1.7A_{RMS}$  (2.4A のサイン波ピーク)
- 全機能内蔵無損失電流検出
- ジャークを最小限に抑えるための 8 点モーション・コントローラ
- SPI および単線式 UART
- エンコーダ・インターフェースおよび 2 個のリファレンス・スイッチ入力
- フルステップあたり 256 マイクロステップの最大分解能
- 柔軟な波形テーブルと位相シフトによりモーターとのマッチングが容易
- StealthChop2 によるサイレント・モーター動作
- SpreadCycle 高ダイナミック・モーター制御チョッパー
- ジャークのない StealthChop2 と SpreadCycle の組み合わせ
- StallGuard2 および StallGuard4 によるセンサー不要のモーター負荷検出
- 最大 75% のエネルギー節減を実現する CoolStep 電流制御
- 受動ブレーキおよびフリーホイール・モード
- モーター相の温度の推定
- チップ温度の測定
- 汎用アナログ入力
- すべてを網羅する保護および診断機能
- 過電圧保護出力
- $5mm \times 7mm$  の小型 TQFN38 パッケージ

※こちらのデータシートには正誤表が付属しています。当該資料の最終ページ以降をご参照ください。

## 目次

概要.....	1
アプリケーション.....	1
機能と利点.....	1
パッケージ情報.....	8
絶対最大定格.....	8
電気的特性.....	9
端子説明.....	13
ピン配置.....	15
機能図.....	16
詳細説明.....	17
動作原理.....	17
フル機能モーション・コントローラおよびドライバ.....	17
SPI ステッピング・モーター・ドライバ.....	17
主要コンセプト.....	18
制御インターフェース.....	19
内蔵 8 点モーション・コントローラ.....	19
自動停止パワー・ダウン.....	19
StealthChop2 および SpreadCycle ドライバ.....	20
利点.....	20
StallGuard2/StallGuard4 – 機械的負荷検出.....	20
CoolStep – 負荷適応型電流制御.....	20
利点.....	20
エンコーダ・インターフェース.....	20
シリアル・ペリフェラル・インターフェース (SPI).....	20
SPI データグラム構造.....	20
書き込み/読み出し選択 (WRITE_notREAD).....	21
各データグラム・リード・バックで転送される SPI ステータス・ビット.....	22
データ・アライメント.....	22
SPI の信号.....	22
SPI のタイミング.....	22
UART 単線式インターフェース.....	23
データグラム構造.....	23
書き込みアクセス.....	23

読出しアクセス .....	24
CRC の計算 .....	24
CRC 計算の C コード例 .....	25
UART の信号 .....	25
複数ノードのアドレス指定 .....	26
StealthChop2 .....	26
自動調整 .....	27
ヒント：自動調整用の最適条件を決定するには、評価用ボードを uses。 .....	27
注意： .....	27
StealthChop2 のオプション .....	27
StealthChop2 電流レギュレータ .....	29
下限電流値 .....	31
速度ベースのスケーリング .....	32
モーターのバック EMF 定数の解釈 .....	33
StealthChop2 と SpreadCycle の結合 .....	34
SpreadCycle へのジャークのない切り替え .....	34
StealthChop2 でのフラグ .....	35
開放負荷フラグ .....	35
モーターの状態を示す PWM_SCALE_SUM .....	35
フリーホイールリングおよび受動ブレーキング .....	35
StealthChop2 を制御するパラメータ .....	35
SpreadCycle および一般的なチョッパー .....	37
SpreadCycle チョッパー .....	39
一般的な定オフ時間チョッパー .....	41
内蔵電流検出機能 .....	42
モーター電流の設定 .....	42
フルスケール電流レンジの設定 .....	43
速度ベースのモード制御 .....	44
ランプ・ジェネレータ .....	47
現実の単位への変換 .....	47
動作プロファイル .....	48
ランプ・モード .....	48
8 点ランプの始動速度および停止速度 .....	50
速度モード .....	52
ランプの早期終了 .....	52
アプリケーション例：ジョイスティック制御 .....	52
速度閾値 .....	53
リファレンス・スイッチ .....	53

ホームिंग・プロシージャの実行 .....	54
3つ目のスイッチによるホームिंग .....	54
仮想リファレンス・スイッチ .....	55
ランプ・ジェネレータの応答時間 .....	55
外部 STEP/DIR ドライバ .....	55
位置比較機能 .....	56
StallGuard2 の負荷測定 .....	56
StallGuard2 の閾値 SGT の調整 .....	57
StallGuard SGT を調整するための初期手順 .....	57
TCOOLTHRS および THIGH を制限する可変速度 .....	58
高トルク・リップルと共振を伴う小型モーター .....	59
モーター・コイル抵抗の温度依存性 .....	59
StallGuard2 測定の正確さと再現性 .....	59
StallGuard2 の更新レートおよびフィルタ .....	59
モーター・ストールの検出 .....	59
StallGuard2 を使用したホームिंग .....	60
StallGuard2 動作の制限 .....	60
StallGuard4 の負荷測定 .....	60
StallGuard4 の調整 .....	62
StallGuard4 の更新レート .....	62
モーター・ストールの検出 .....	62
StallGuard4 動作の制限 .....	63
CoolStep の負荷適応型電流スケーリング .....	63
CoolStep 用の設定 .....	63
CoolStep の調整 .....	65
応答時間 .....	65
低速度およびスタンバイ動作 .....	65
診断出力 .....	65
DcStep .....	66
DcStep のデザインイン .....	66
DcStep とモーション・コントローラの統合 .....	67
DcStep モードにおけるストール検出 .....	68
DcStep 動作での実際のモーター速度の測定 .....	69
サイン波ルックアップ・テーブル .....	69
マイクロステップ・テーブル .....	69
ABN インクリメンタル・エンコーダ・インターフェース .....	71
モーターの分解能に合うエンコーダ設定 .....	73
リセット、ディスエーブル/停止、パワー・ダウン .....	73

緊急停止.....	73
外部リセットおよびスリープ・モード.....	73
保護およびドライバ診断.....	74
過電流保護.....	74
過熱保護およびサーマル・シャットダウン.....	74
温度測定.....	74
チップ温度の測定.....	74
モーター温度の測定.....	75
過電圧保護と OV ピン.....	75
短絡保護 (GND への短絡および $V_s$ への短絡).....	75
開放負荷診断.....	76
低電圧ロックアウト保護.....	76
ESD 保護.....	76
外部アナログ入力 AIN のモニタリング.....	76
クロック発振器とクロック入力.....	77
内部クロックの使用.....	77
外部クロックの使用.....	77
クイック設定ガイド.....	77
電流設定.....	78
StealthChop2 の設定.....	79
SpreadCycle 設定.....	80
CoolStep を StealthChop2 と組み合わせてイネーブル.....	81
CoolStep を SpreadCycle と組み合わせてイネーブル.....	82
モーション・コントローラを用いたモーターの作動.....	83
モーション・ランプ・パラメータの設定.....	84
DcStep 動作のイネーブル.....	85
DcStep でストール検出を使用.....	86
一般的なレジスタ・マッピングおよびレジスタ情報.....	87
標準アプリケーション回路.....	88
標準的なアプリケーション回路.....	88
大モーター電流.....	88
勾配コントロール.....	88
ドライバ保護および EME 回路.....	89
レジスタ・マップ.....	91
GCR レジスタの概要.....	91
型番.....	143

## 図一覧

図 1. ブロック図	16
図 2. 代表的な外部コンポーネントを使用したブロック図	17
図 3. 代表的な外部コンポーネントを使用したブロック図	18
図 4. 停止時およびランプアップ時の自動モーター電流制御	19
図 5. SPI のタイミング図	23
図 6. UART のデジタイゼーション例	26
図 7. StealthChop2 の自動調整手順	28
図 8. StealthChop2 : PWM_REG の設定は良好	30
図 9. StealthChop2 : AT#2 時の PWM_REG の設定が過小	30
図 10. 正しく決定された PWM_GRAD(_AUTO) および PWM_OFS(_AUTO)	31
図 11. 過小な PWM_GRAD 設定の例	31
図 12. 速度ベースの PWM スケーリング (pwm_autoscale = 0)	33
図 13. SpreadCycle への切り替えオプションのための TPWMTHRS	34
図 14. 代表的なチョッパー減衰フェーズ	38
図 15. 1 チョッパー・サイクルの間のコイル電流を示す、SpreadCycle チョッパーの概略図	40
図 16. オフセットを持つ一般的な定オフ時間チョッパーが示すコイル電流	41
図 17. 一般的なチョッパーのゼロ交差とサイン波オフセットを使用した補正	41
図 18. 速度依存モードの選択	45
図 19. 2 回目の動作が負の方向に向かうランプ・ジェネレータ速度パターン	49
図 20. ランプ・ジェネレータを用いた最適なモーター・トルク使用法を示す図	49
図 21. 距離が短いために VMAX に到達できない 8 点ランプ	50
図 22. 距離が短いために V2 に達せず AMAX フェーズと DMAX フェーズがないプロファイル	51
図 23. 距離が短いために V1 に達しないプロファイル	51
図 24. 距離が短いために TVMAX を維持できないプロファイル	51
図 25. オンザフライの目標ポジション変更があった場合の 8 点ランプの例	52
図 26. ランプ・ジェネレータの速度依存モーター制御	53
図 27. リファレンス・スイッチの使用 (例)	54
図 28. 仮想停止スイッチと制限の図解	55
図 29. StallGuard2 の機能原理	57
図 30. 例: サンプル・モーターを用いた SGT の最適設定値と StallGuard2 の読出し値	59
図 31. StallGuard4 の動作モード	61
図 32. モーター電流を負荷に適合させる CoolStep	64
図 33. 診断出力構成のオプション	66
図 34. DcStep により拡大されるアプリケーション動作領域	67
図 35. 過負荷状態時の DcStep の速度プロファイル	68
図 36. LUT プログラミングの例	70
図 37. OFFSET_SIN90 を使用したコサイン波のシフト	71
図 38. インクリメンタル・エンコーダの ABN 信号の概要	72
図 39. ブレーキ・チョッパー回路の例	75
図 40. 電流設定のためのクイック設定ガイド	78
図 41. StealthChop2 のクイック設定ガイド	79
図 42. SpreadCycle のクイック設定ガイド	80
図 43. StealthChop2 との組み合わせによる CoolStep のクイック設定ガイド	81
図 44. SpreadCycle との組み合わせによる CoolStep のクイック設定ガイド	82
図 45. 速度モードでモーターを作動させるためのクイック設定ガイド	83
図 46. 目標ポジションへモーターを作動させるためのクイック設定ガイド	83
図 47. モーション・ランプ・パラメータを設定するためのクイック設定ガイド	84
図 48. DcStep のクイック設定ガイド	85
図 49. DcStep でストール検出を使用するためのクイック設定ガイド	86
図 50. 標準的なアプリケーション回路	88
図 51. 簡単な ESD 対策	89
図 52. モーター出力保護の強化	90

## 表一覧

表 1. SPI データグラム構造 .....	21
表 2. SPI 読出し／書込みフローの例 .....	22
表 3. SPI_STATUS - ビット 39～32 の各 SPI アクセスで転送されるステータス・フラグ .....	22
表 4. UART 書込みアクセス・データグラム構造 .....	23
表 5. UART 読出しアクセス要求データグラムの構造 .....	24
表 6. UART 読出しアクセス返答データグラムの構造 .....	24
表 7. TMC5241 UART のインターフェース信号 .....	25
表 8. 最大 255 ノードをアドレス指定する UART 例 .....	26
表 9. StealthChop2 自動調整 AT#1 および AT#2 の制約事項と条件 .....	27
表 10. StealthChop2 の PWM 周波数の選択 .....	29
表 11. StealthChop2 を制御するパラメータ .....	36
表 12. SpreadCycle および一般的な定オフ時間チョッパを制御するパラメータ .....	38
表 13. SpreadCycle モードのパラメータ .....	40
表 14. 定オフ時間チョッパ・モードを制御するパラメータ .....	41
表 15. モーター電流を制御するパラメータ .....	42
表 16. I <sub>FS</sub> フルスケール・ピーク・レンジの設定値 (R <sub>REF</sub> = 12kΩ の場合の例) .....	43
表 17. DRV_CONF のビット 1 および 0 の設定と様々な R <sub>REF</sub> に基づく、フルスケール実効値電流 I <sub>FS</sub> (単位: アンペア (A <sub>RMS</sub> )) .....	44
表 18. 速度ベースのモード制御パラメータ .....	46
表 19. ランプ・ジェネレータのパラメータと単位の関係 .....	47
表 20. X_COMPARE_REPEAT オプションと周期的なパルス動作 .....	56
表 21. StallGuard2 関連のパラメータ .....	57
表 22. StallGuard4 関連のパラメータ .....	61
表 23. CoolStep の重要パラメータ .....	63
表 24. CoolStep のその他のパラメータとステータス情報 .....	64
表 25. DcStep の重要パラメータ .....	68
表 26. 200 フルステップのモーターに対して 256 マイクロステップを使用するエンコーダの設定例 .....	73
表 27. フルスケール電流設定に基づく過電流保護閾値 .....	76
表 28. レジスタ・マップの概要 .....	87

## パッケージ情報

TQFN38 5mm x 7mm	
Package Code	T3857+1C
Outline Number	<a href="#">21-0172</a>
Land Pattern Number	<a href="#">90-0076</a>
Thermal Resistance, Single Layer Board:	
Junction to Ambient ( $\theta_{JA}$ )	38°C/W
Junction to Case ( $\theta_{JC}$ )	1°C/W
Thermal Resistance, Four Layer Board:	
Junction to Ambient ( $\theta_{JA}$ )	28°C/W
Junction to Case ( $\theta_{JC}$ )	1°C/W

最新のパッケージ外形図とランド・パターン（フットプリント）に関しては、<https://www.analog.com/jp/design-center/packaging-quality-symbols-footprints/package-index.html> で確認してください。パッケージ・コードの「+」、「#」、「-」は RoHS 対応状況のみを示します。パッケージ図面は異なる末尾記号が示されている場合がありますが、図面は RoHS 状況に関わらず該当のパッケージについて図示しています。

パッケージの熱抵抗は、JEDEC 規格 JESD51-7 に記載の方法で 4 層基板を使用して求めたものです。パッケージの熱に対する考慮事項の詳細については、<https://www.analog.com/jp/technical-articles/thermal-characterization-of-ic-packages.html> を参照してください。

## 絶対最大定格

$V_S \sim GND$ .....	-0.3V~70V	$I_{REF}$ , $A_{IN} \sim GND$ .....	-0.3V~Min. (2.2, $V_{DD1V8} + 0.3$ )V
$V_{DD1V8} \sim GND$ .....	-0.3V~Min. (2.2, $V_S + 0.3$ )V	$V_{CC10} \sim GND$ .....	-0.3V~6V
$AGND \sim GND$ .....	-0.3V~+0.3V	ロジック入力/出力電圧~GND .....	-0.3V~ $V_{CC10} + 0.3$ V
OUT1A, OUT2A, OUT1B, OUT2B .....	-0.3V~ $V_S + 0.3$ V	OV~GND .....	-0.3V~6V
$V_{CP} \sim GND$ .....	$V_S - 0.3$ V~Min. (74, $V_S + 6$ )V	動作温度範囲 .....	-40°C~125°C
CPO~GND .....	$V_S - 0.3$ V~ $V_{CP} + 0.3$ V	ジャンクション温度 .....	+165°C
CPI~GND .....	-0.3V~ $V_S + 0.3$ V	保管温度範囲 .....	-65°C~+150°C
SLEEPN~GND .....	-0.3V~ $V_S + 0.3$ V	ハンダ付け処理温度（リフロー） .....	+260°C

上記の絶対最大定格を超えるストレスを加えるとデバイスに恒久的な損傷を与えることがあります。これらの規定はストレス定格のみを定めたものであり、この仕様の動作セクションに記載する規定値以上でデバイスが正常に動作することを意味するものではありません。デバイスを長時間絶対最大定格状態に置くとデバイスの信頼性に影響を与えます。

## 電气的特性

( $V_S = 4.5V \sim 65V$ 、 $V_{CC\_IO} = 2.2V \sim 5.5V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 。代表値は  $T_A = +25^\circ C$  および  $V_S = 48V$  を前提としています。限界値は、 $T_A = +25^\circ C$  で 100%テストされています。動作温度範囲および対応する電源電圧範囲にわたる限界値は、設計と特性評価によって裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>POWER SUPPLY</b>						
Supply Voltage Range	$V_S$		4.5		65	V
Sleep Mode Current Consumption	$I_{VS}$	$V(\text{SLEEPN}) = 0$		4	25	$\mu A$
Quiescent Current Consumption	$I_{VS}$	$V(\text{SLEEPN}) = 1$ , $V(\text{DRV\_ENN}) = 1$		3.5	6	mA
1.8V Regulator Output Voltage	$V_{VDD}$	$V_S = 4.5V$		1.8		V
$V_{DD}$ Current Limit	$I_{V18LIM}$		20			mA
Charge Pump Voltage	$V_{CP}$			$V_S + 2.7$		V
Logic I/O Supply Voltage Range	$V_{CC\_IO}$		2.2		5.5	V
Sleep Mode Current Consumption	$I_{VCC\_IO}$	$V(\text{SLEEPN}) = 0$		5	10	$\mu A$
Quiescent Current Consumption	$I_{VCC\_IO}$	$V(\text{SLEEPN}) = 1$		35	60	$\mu A$
<b>LOGIC LEVEL INPUTS-OUTPUTS</b>						
Input Voltage Level - High	$V_{IH}$		$0.7 \times V_{CC\_IO}$			V
Input Voltage Level - Low	$V_{IL}$				$0.3 \times V_{CC\_IO}$	V
Input Hysteresis	$V_{HYS}$			$0.15 \times V_{CC\_IO}$		V
Internal Pullup/Pulldown Resistance	$R_{PULL}$	To GND or to $V_{CC\_IO}$	60	100	140	k $\Omega$
Input Leakage	$I_{nLeak}$	Inputs without pullup/pulldown resistance	-1		+1	$\mu A$
Output Logic-Low Voltage	$V_{OL}$	$I_{LOAD} = 5mA$			0.4	V
Push-Pull Output LogicHigh Voltage	$V_{OH}$	$I_{LOAD} = 5mA$	$V_{CC\_IO} - 0.4V$			
Open-Drain Output Logic High Leakage Current	$I_{OH}$	$V(\text{PIN}) = 5.5V$	-1		+1	$\mu A$
SLEEPN Voltage Level High	$V_{IHSLEEPN}$		0.9			V
SLEEPN Voltage Level Low	$V_{ILSLEEPN}$				0.6	V
SLEEPN Pulldown Input Resistance	$R_{PDSLEEPN}$		0.8	1.5		M $\Omega$
<b>OUTPUT SPECIFICATIONS</b>						
Output ON-Resistance Low Side	$R_{ONLS}$	Full-scale bits = 10		0.15	0.3	$\Omega$
		Full-scale bits = 01		0.21	0.4	
Output ON-Resistance Low Side	$R_{ONLS}$	Full-scale bits = 00		0.37	0.75	$\Omega$
Output ON-Resistance High Side	$R_{ONLS}$			0.16	0.3	$\Omega$
Output Leakage	$I_{LEAK}$		-10		+10	$\mu A$

( $V_S = 4.5V \sim 65V$ 、 $V_{CC\_IO} = 2.2V \sim 5.5V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 。代表値は  $T_A = +25^\circ C$  および  $V_S = 48V$  を前提としています。限界値は、 $T_A = +25^\circ C$  で 100%テストされています。動作温度範囲および対応する電源電圧範囲にわたる限界値は、設計と特性評価によって裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Slew Rate	SR	Slew-rate bits = 00		100		V/ $\mu$ s
		Slew-rate bits = 01		200		
		Slew-rate bits = 10		400		
		Slew-rate bits = 11		800		
<b>PROTECTION CIRCUITS</b>						
Overcurrent Protection Threshold	OCP	Full-scale bits = 10	5.0			A
		Full-scale bits = 01	3.33			
		Full-scale bits = 00	1.67			
Overcurrent Protection Blanking Time	TOCP		0.9	1.5	2.3	$\mu$ s
UVLO Threshold on VS	UVLO	VS falling	3.75	3.9	4.05	V
UVLO Threshold on VS Hysteris	UVLOHYS			0.12		V
UVLO Threshold on $V_{CC\_IO}$	UVLO	$V_{CC\_IO}$ falling	0.9	1.5	1.95	
$V_{CC\_IO}$ UVLO Hysteresis	UVLOVCCCH			100		mV
Thermal Protection Threshold Temperature	TSD			165		$^\circ C$
Thermal Protection Temperature Hysteresis				20		$^\circ C$
<b>CURRENT REGULATION</b>						
IREF Pin Resistor Range	RREF		12		60	k $\Omega$
IREF Output Voltage	VREF		0.882	0.9	0.918	V
Full-Scale Current Constant	KIFS	IFS = 1A		11.75		A $\times$ k $\Omega$
		IFS = 2A		24		
		IFS = 3A		36		
Regulation Accuracy	DITRIP1	Output current from 7% to 100% FS, $R_{REF} = 12k\Omega$	-7		+7	%
Phase-to-Phase Current Regulation Mismatch	IMATCH	Output currents from 7% to 100% FS, $R_{REF} = 12k\Omega$ One Sigma		0.7		%
<b>FUNCTIONAL TIMINGS</b>						
SLEEP Time	$t_{SLEEP}$	SLEEPN = 0 to OUT_ three state			50	$\mu$ s
Wake-Up Time from Sleep	$T_{WAKE}$	SLEEPN = 1 to normal operation			2.5	ms
Enable Time	$T_{EN}$	Time from DRV_ENN pin falling edge to driver on			1.5	$\mu$ s
Disable Time	$T_{EN}$	Time from DRV_ENN pin rising edge to driver off			6	$\mu$ s
<b>CLOCK</b>						
Internal Clock Frequency	$f_{CLKOSC}$		11.9	12.5	13.2	MHz
External Clock Frequency	$f_{CLK}$		8	16	20	MHz

( $V_S = 4.5V \sim 65V$ 、 $V_{CC\_IO} = 2.2V \sim 5.5V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 。代表値は  $T_A = +25^\circ C$  および  $V_S = 48V$  を前提としています。限界値は、 $T_A = +25^\circ C$  で 100%テストされています。動作温度範囲および対応する電源電圧範囲にわたる限界値は、設計と特性評価によって裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	
External Clock Duty Cycle	$t_{CLKL}$		40		60	%	
External Clock Detection in Cycles			4		8		
External Clock Timeout Detection in Cycles of Internal fCLKOSC			12		16		
External Clock Detection Lower Frequency Threshold	$f_{CLKLO}$		4			MHz	
<b>SPI TIMINGS</b>							
SCK Valid before or after Change of CSN	$t_{CC}$		TSCK			ns	
CSN High Time	$t_{CSH}$		4 x TSCK			ns	
SCK Low Time	$t_{CL}$		20			ns	
SCK High Time	$t_{CH}$		20			ns	
SCK Frequency	$f_{SCK}$	$V_{CC\_IO} = 3V$			8	MHz	
		$V_{CC\_IO} = 2.2V$			5		
SDI Setup Time Before SCK Rising Edge	$t_{DU}$		10			ns	
SDI Hold Time After SCK Rising Edge	$t_{DH}$		10			ns	
Data Out Valid Time After SCK Falling Edge	$t_{DO}$	$V_{CC\_IO} = 3V$			33	50	ns
		$V_{CC\_IO} = 2.2V$			50	80	ns
SDI, SCK, and CSN Filter Delay Time	$t_{FILT}$	Rising and falling edge			10	ns	
<b>ENCODER TIMING</b>							
Encoder Counting Frequency	$f_{CNT}$		< 2/3 fCLK		fCLK		
A/B/N Input Low Time	$t_{ABNL}$		3 x TCLK + 20			ns	
A/B/N Input High Time	$t_{ABNH}$		3 x TCLK + 20			ns	
A/B/N Spike Filtering Time	$t_{FILTABN}$	Rising and falling edge	3 x CLK				
<b>ADC/Analog Input/Temperature</b>							
ADC Resolution		12 bit + sign	13			Bit	
Analog Input Voltage Range	$V_{AIN}$		0		1.25	V	
Analog Input Leakage	$I_{AIN,IEAK}$		-1		+1	uA	
Analog Input Frequency	$f_{AIN}$	Assuming undersampling at AIN is accepted, the AIN input frequency must be lower than the given max. value for a meaningful ADC conversion for a single ADC channel.			70	kHz	

( $V_S = 4.5V \sim 65V$ 、 $V_{CC\_IO} = 2.2V \sim 5.5V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 。代表値は  $T_A = +25^\circ C$  および  $V_S = 48V$  を前提としています。限界値は、 $T_A = +25^\circ C$  で 100%テストされています。動作温度範囲および対応する電源電圧範囲にわたる限界値は、設計と特性評価によって裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Temperature Accuracy	$T_{DRIVER}$			$\pm 10$		$^\circ C$
Supply Voltage Measurement Accuracy		$V_M = 4.5V$	-7		+7	%
		$V_M \geq 12V$	-4		+4	
ADC Sample Rate	$f_{SAMPLE, ADC}$			$\frac{f_{CLK}}{2048}$		

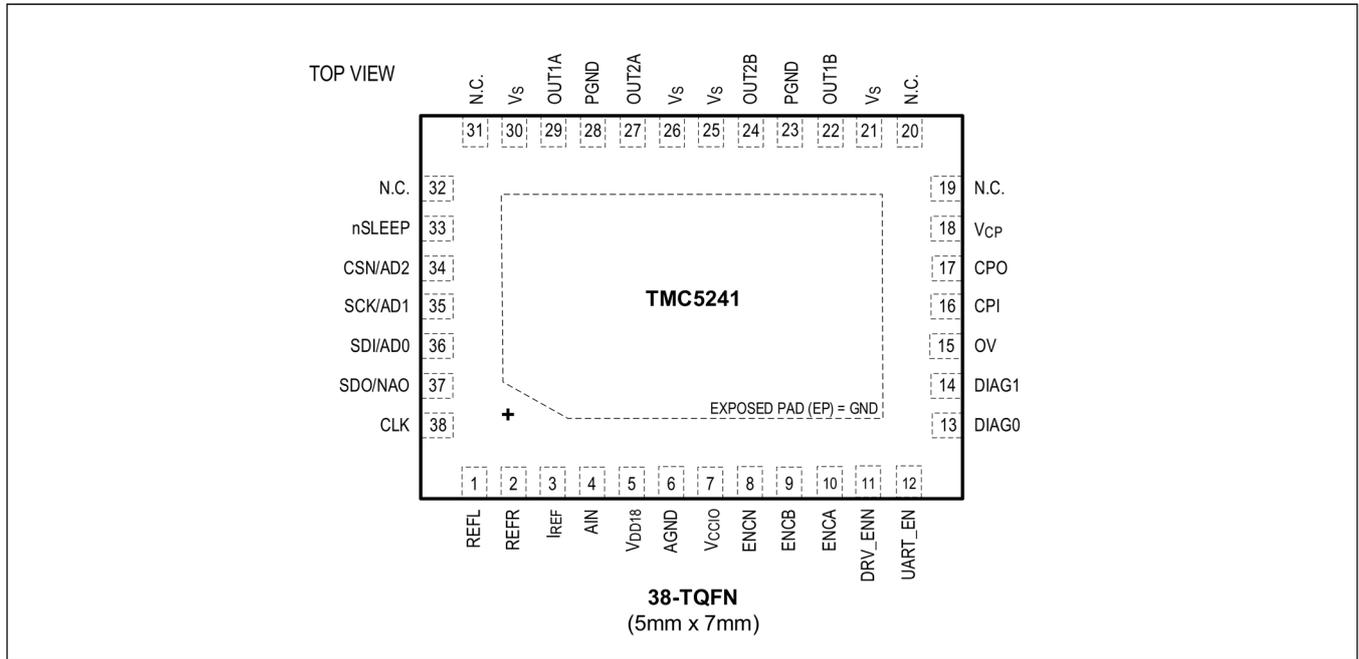
## 端子説明

ピン	名称	説明	リファレンス電源	タイプ
6	AGND	アナログ・グラウンド。グラウンド・プレーンに接続します。		GND
23, 28	PGND	電源グラウンド。グラウンド・プレーンに接続します。		GND
21, 25, 26, 30	V <sub>S</sub>	モーターの電源電圧。GND プレーン／露出パッドへ最短のループを形成してピン近くでのフィルタによる除去機能を提供します。		Supply
5	V <sub>DD1V8</sub>	内蔵 1.8V レギュレータの出力。最高性能を発揮させるには、AGND との間のピン近くに 2.2μF 以上のセラミック・コンデンサを接続します。		Supply
18	V <sub>CP</sub>	チャージ・ポンプ電圧。1.0μF のコンデンサを用いて V <sub>S</sub> に接続します。誘導性ピークを避けるために、コンデンサの正側端子を V <sub>S</sub> ピンの近くに接続します。		Analog Output
7	V <sub>CC_IO</sub>	回路の IO レベルを定義するために外部電源から供給されるデジタル IO 電源電圧。出力ピンに適切な電圧レベルを設定するために必要です。		Supply
17	CPO	チャージ・ポンプ・コンデンサ出力。		Analog Output
16	CPI	チャージ・ポンプ・コンデンサ入力。22nF、50V のコンデンサを用いて CPO に接続します。		Analog Output
38	CLK	CLK 入力。内部クロック用に短い配線を用いて GND に接続します。あるいは外部クロックを供給します。外部クロック信号が失われた場合には内蔵のクロック・フェイル・オーバー回路が保護します。	VCC_IO	Digital Input
1	REFL	内蔵ランプ・ジェネレータ用の左側リファレンス入力。	VCC_IO	Digital Input
2	REFR	内蔵ランプ・ジェネレータ用の右側リファレンス入力。	VCC_IO	Digital Input
34	CSN/AD2	SPI チップ・セレクト入力（ロー・アクティブ）（UART_EN = 0）または UART モードのアドレス入力 2（+4）（UART_EN = 1）。	VCC_IO	Digital Input (Pullup)
35	SCK/AD1	SPI シリアル・クロック入力（UART_EN = 0）または UART モードのアドレス入力 1（+2）（UART_EN = 1）。	VCC_IO	Digital Input (Pullup)
36	SDI/AD0	SPI データ入力（UART_EN = 0）または UART モードのアドレス入力 0（+1）（UART_EN = 1）。	VCC_IO	Digital Input (Pullup)
37	SDO/NAO	SPI データ出力（スリーステート）（UART_EN = 0）または UART モードの次のアドレス出力（NAO）（UART_EN = 1）。	VCC_IO	Digital Output
3	I <sub>REF</sub>	電流スケール用のアナログ・リファレンス電流。外付け抵抗を介して GND に接続します。	VDD_18	Analog Input
12	UART_EN	インターフェース選択ピン。ローに接続された場合、SPI インターフェースがイネーブルされます。ハイに接続された場合、UART インターフェースがイネーブルされます。内蔵プルダウン抵抗。	VCC_IO	Digital Input (Pulldown)
9	ENCB	エンコーダの B チャンネル入力。	VCC_IO	Digital Input (Pullup)
10	ENCA	エンコーダの A チャンネル入力。	VCC_IO	Digital Input (Pullup)
8	ENCN	エンコーダの N チャンネル入力。	VCC_IO	Digital Input (Pullup)
11	DRV_ENN	イネーブル入力。このピンがハイ・レベルに駆動されると、電力段のスイッチがオフ（すべてのモーター出力がフロート状態）になります。	VCC_IO	Digital Input (Pullup)
13	DIAG0	診断出力 DIAG0。内蔵モーション・コントローラから外部ドライバへの割込みまたは STEP 出力。オープン・ドレイン・モードでは外部プルアップ抵抗を使用します。 システムがリセット状態の場合は、このピンがアクティブにローになり、リセット状態であることを外部コントローラに通知します。	VCC_IO	Digital Output

14	DIAG1/SW	診断出力 DIAG1。 内蔵モーション・コントローラから外部ドライバへの位置比較または DIR 出力。 オープン・ドレイン・モードでは外部プルアップ抵抗を使用します。 UART モードでは単線式 I/O となります。	VCC_IO	Digital IO
33	nSLEEP	ロー・アクティブのパワー・ダウン入力/リセット入力。 デバイスをスリープ・モードにするには、連続的なロー・レベルを印加します。 SLEEPN はプルダウンを内蔵しています。 使用しない場合は、V <sub>S</sub> または V <sub>CC_IO</sub> (これは高電圧ピンです) に接続します。 IC がスリープ・モード/リセットから回復した後は、再度使用する前に再設定を行う必要があります。スリープ・モードの間、レジスタの内容は保存されません。 IC の再設定の間、DRV_ENN を用いてブリッジ・ドライバをディスエーブルしたままにすることを推奨します。 モーター速度が高い場合には決して使用しないでください。	VS	Analog Input (Pulldown)
24	OUT2B	モーター・コイル B の出力 2	VS	Analog Output
22	OUT1B	モーター・コイル B の出力 1	VS	Analog Output
27	OUT2A	モーター・コイル A の出力 2	VS	Analog Output
29	OUT1A	モーター・コイル A の出力 1	VS	Analog Output
EP	GND	露出ダイ・パッド。露出ダイ・パッドは GND プレーンに接続します。GND プレーンとの間には熱伝導のためにできるだけ多くのビアを使用してください。電力段および内部回路には GND ピンとして機能します。		GND
19, 20, 31, 32	N.C.	内部接続なし。このピンは開放のままにするか、冷却効率を向上するために GND に接続します。		N.C.
15	OV	プログラマブルな閾値電圧を備えた過電圧インジケータ出力 (オープン・ドレイン)。電源電圧を制限するために、外部 MOSFET を負荷抵抗と共に接続します。 外部プルアップ抵抗が必要です。ADC により $f_{clk}/2048$ ごとに更新されます。	VCC_IO	Digital Output (Open-Drain)
4	AIN	汎用アナログ入力。内部 ADC により $f_{clk}/2048$ ごとに計測されます。 入力範囲は 0~1.25V です。 値は SPI/UART を通じて得られます。	VDD_18	Analog Input

ピン配置

TMC5241 TQFN のピン配置



機能図

TMC5241

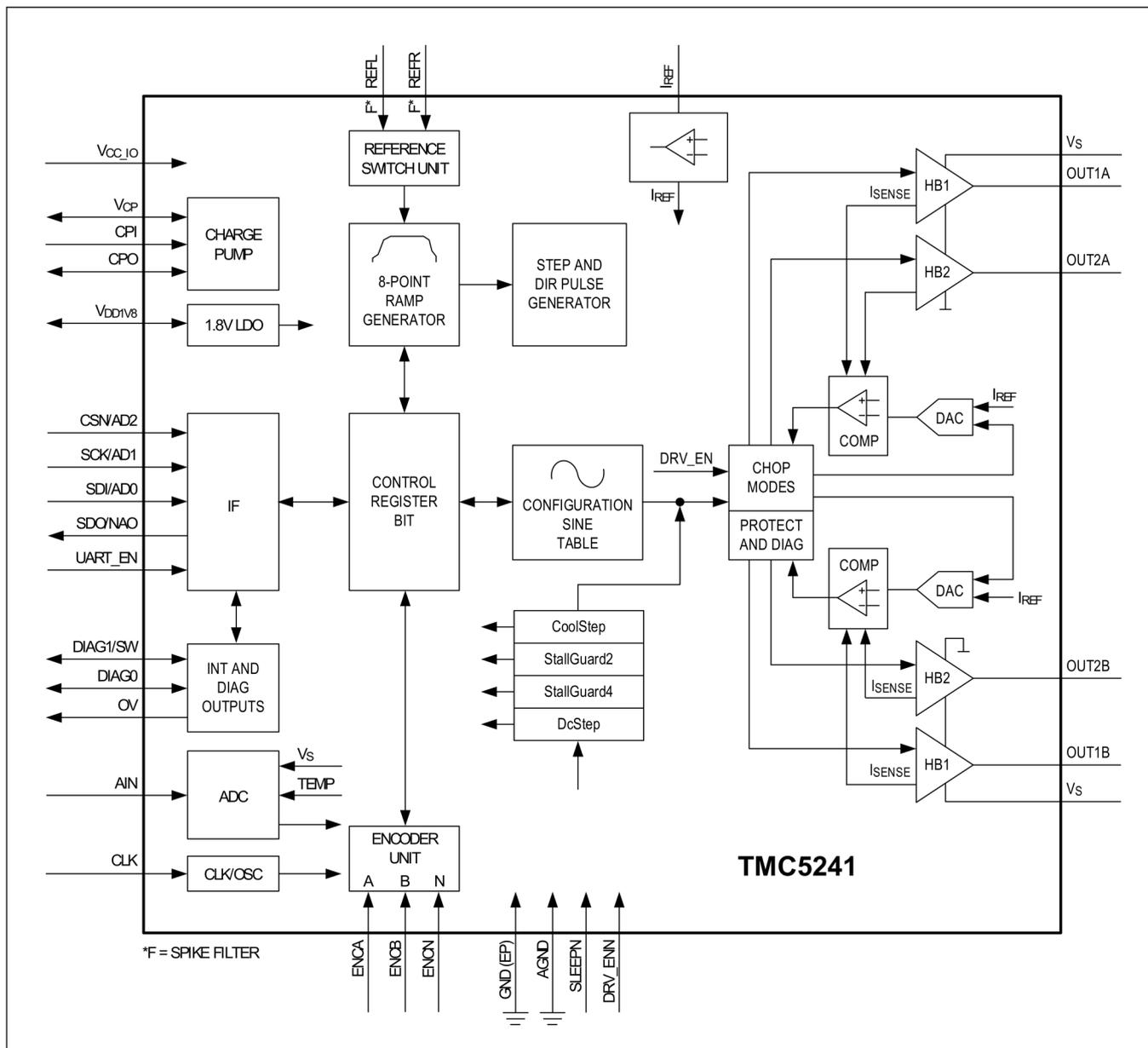


図 1. ブロック図

## 詳細説明

## 動作原理

## フル機能モーション・コントローラおよびドライバ

TMC5241 モーション・コントローラおよびドライバ・チップは、CPU とステップング・モーターの間のインターフェースとなるインテリジェントな電源コンポーネントです。すべてのステップング・モーター・ロジックが TMC5241 に完全に内蔵されています。モーターを制御するソフトウェアは不要です。単に目標ポジションを指定するだけです。TMC5241 は、ドライバおよびコントローラのシステムオンチップ集積により可能となっている、数多くのユニークな強化機能を備えています。TMC5241 の 8 点ランプ・ジェネレータは、StealthChop、CoolStep、StallGuard、DcStep、SpreadCycle を自動で使用し、あらゆるモーター動作を最適化します。

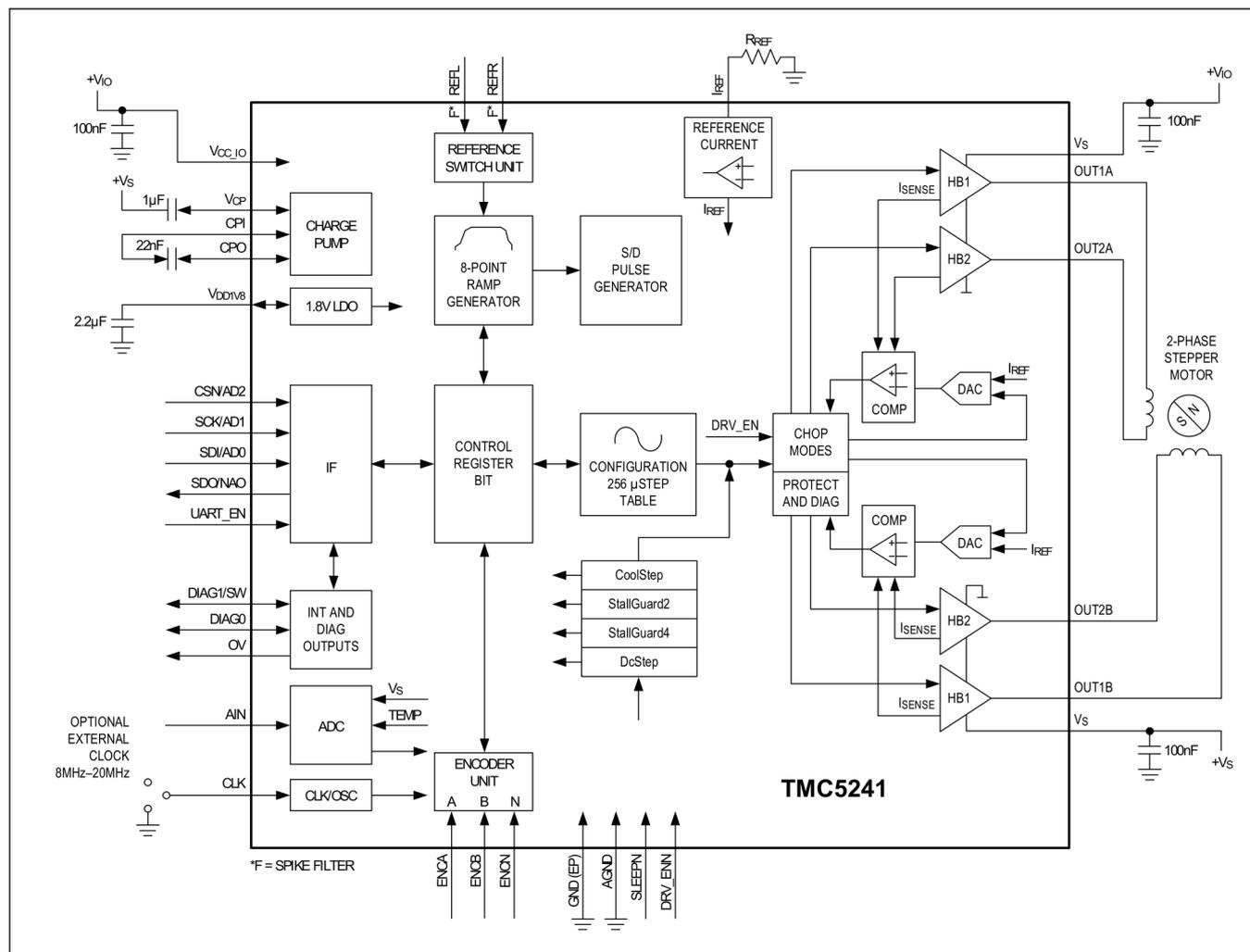


図 2. 代表的な外部コンポーネントを使用したブロック図

## SPI ステッピング・モーター・ドライバ

TMC5241 はインテリジェントなステップング・モーター・ドライバ・チップです。このスマートな電力変換部品は、2 相のステップング・モーターを直接駆動し、CPU とのインターフェースとして設定、コントロール、診断フィードバックに使用します。ステップング・モーターを駆動するためのすべての電流制御機能が内蔵されています。TMC5241 は、アプリケーション内でのモーター動作をサポートする多くのユニークな機能を備えています。

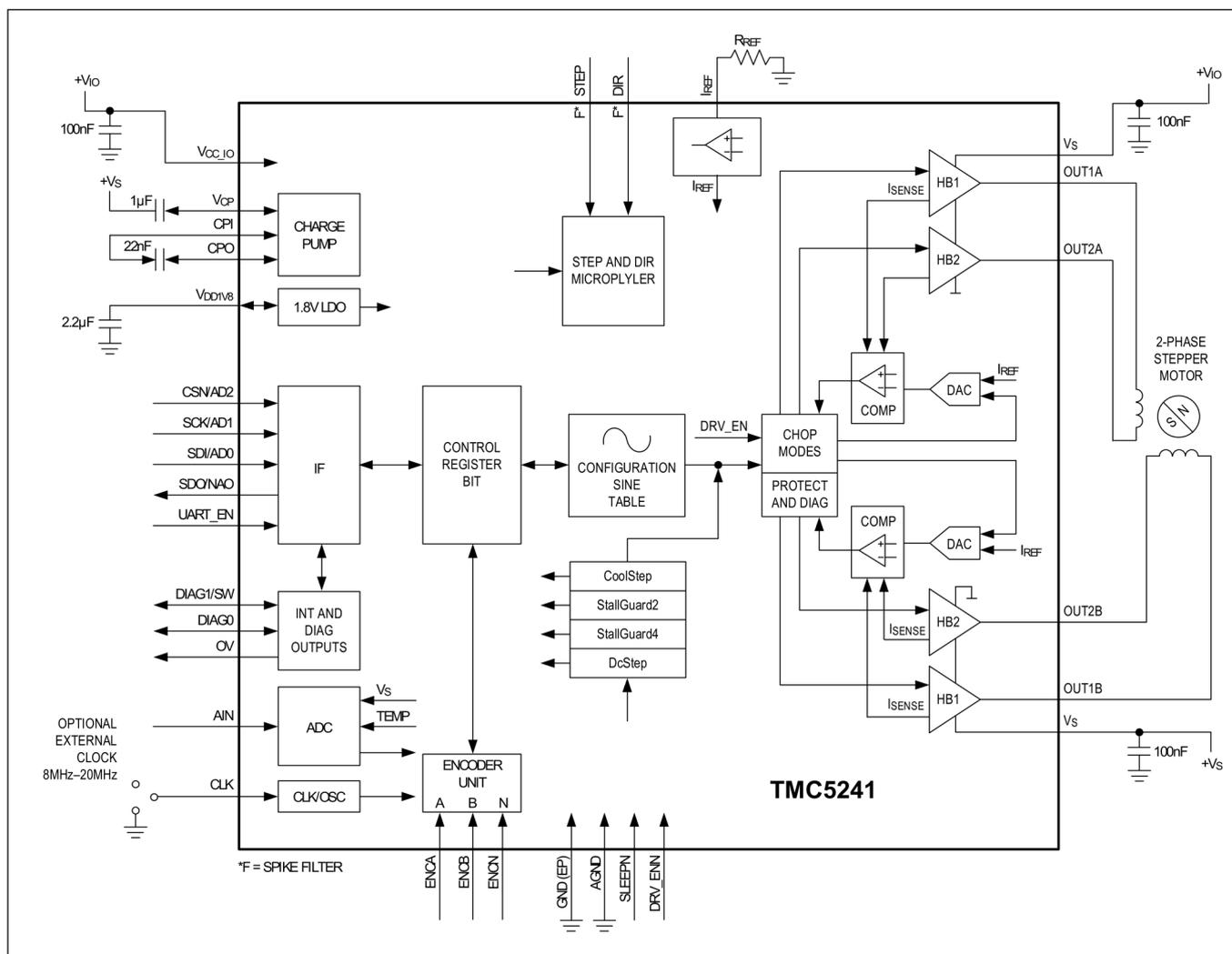


図 3. 代表的な外部コンポーネントを使用したブロック図

主要コンセプト

TMC5241 は、ADI-Trinamic 製品独自の高度な機能を各種備えています。これらの機能は、多くのステップング・モーター・アプリケーションにおいて、精度、エネルギー効率、信頼性、動作の滑らかさを向上することや、動作による温度上昇を抑えることに貢献しています。

StealthChop2	モーターの無音動作および無音停止を実現する、無騒音高精度チョッパ・アルゴリズム。StealthChop よりも急速なモーター加減速を可能にし、StealthChop を小停止モーター電流まで拡張します。
SpreadCycle	動的動作を最大限に高める、サイクルごとの高精度な電流制御。
StallGuard2	SpreadCycle 用のセンサーレス・ストール検出および機械的負荷測定。
StallGuard4	StealthChop 用のセンサーレス・ストール検出および機械的負荷測定。
CoolStep	モーターやドライバの効率を最大にして、温度上昇を最小限に抑えるようモーター電流を適応させるために、StallGuard 測定を使用します。

これらの性能向上機能に加え、ADI-Trinamic のモーター・ドライバは、短絡出力、出力オープンサーキット、過熱、低電圧などの状態を検出して保護するための防護機能を備え、安全性および機器の誤動作からの回復をより確実なものにします。

## 制御インターフェース

TMC5241 は、SPI および UART ベースの単線式インターフェースのどちらにも対応し、また、CRC チェック機能も備えています。実際のインターフェースの組み合わせは、UART\_EN ピンを通じて選択します。このピンは、目的のインターフェース選択に応じて GND または Vcc<sub>IO</sub> にハードワイヤ接続します。

SPI は、バス・クロックに同期するビットシリアル・インターフェースです。バス・コントローラからバス・ペリフェラルに送られるビットごとに、同時に別のビットがペリフェラルからコントローラに送り返されます。SPI コントローラ (MCU など) とペリフェラルとの間の通信は、1つの 40 ビット・コマンド・ワードの送信と 1つの 40 ビット・ステータス・ワードの受信で常に構成されます。

単線式インターフェースは、双方向の単線インターフェース接続が可能です。これは標準的な UART で駆動できます。ボー・レートの設定は不要です。

## 内蔵 8 点モーション・コントローラ

内蔵の 32 ビット・モーション・コントローラは、モーターを自動で目標位置に駆動したり、目標速度まで加速したりします。すべてのモーション・パラメータはオンザフライで変更できます。モーション・コントローラは直ちに再計算を行います。設定データの最小セットは、加速度値と減速度値および最大運動速度で構成されます。始動速度と停止速度が、速度閾値によって選択される 2 番目と 3 番目の加減速設定と合わせてサポートされ、それによって 8 点の速度プロファイルが形成されます。これらの設定値により、モーション・プロファイルをもーター・トルク・プロファイルに適応させることや、S 字に近い形状のランプ性能を実現するためのジャーク除去を行うことができます。この内蔵モーション・コントローラは、機械式リファレンス・スイッチやセンサーレス・ストール検出 StallGuard2 および StallGuard4 に対し、即時対応が可能です。

## 利点

- 柔軟なランプ・プログラミング
- 加減速に対してモーター・トルクを効率的に利用できるため、機械のスループット向上が可能。
- ジャーク除去のための擬似 S 字形状ランプ。
- 停止およびストール状態への即時反応。

## 自動停止パワー・ダウン

自動的な電流低減機能により、アプリケーションの消費電力条件および冷却条件が大幅に軽減されます。動作電流を半分に低減することで、停止に要する消費電力はおよそ 25%に減少します。停止電流、遅延時間、減衰の各パラメータは、シリアル制御インターフェースを通じて設定できます。

自動フリーホイーリングおよび受動モーター・ブレーキングは、停止用オプションとして提供されています。受動ブレーキングにより、効果的な減衰およびブレーキングを維持したまま、モーター停止の消費電力をゼロに減少できます。

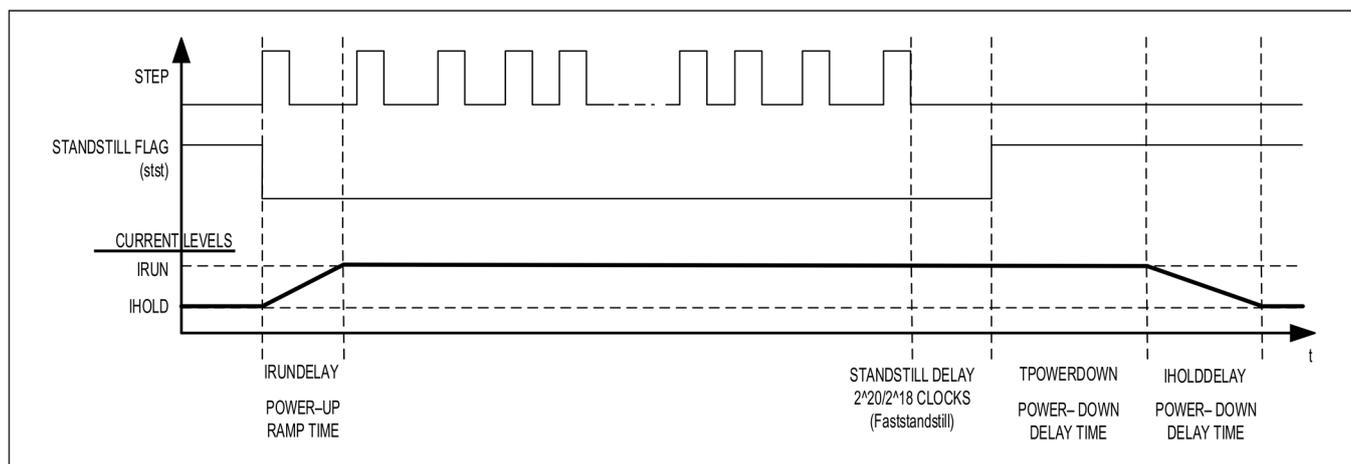


図 4. 停止時およびランプアップ時の自動モーター電流制御

## StealthChop2 および SpreadCycle ドライバ

StealthChop2 の原理は電圧チョッパーに基づいています。停止時および低速動作時には、ボール・ベアリングで発生するノイズを除き、モーターに絶対的な静音が確保されます。

他の電圧モード・チョッパーとは異なり、StealthChop2 はいかなる設定も不要です。起動後最初の動作時に最適設定を自動的に学習し、その後の動作を通じて更に設定を最適化します。

学習には最初のホーミング・シーケンスで十分です。また、最初の学習パラメータをレジスタ・セットにロードすることもできます。StealthChop2 では、モーター速度の変化に対し一度の応答で高いモーター運動性能を実現できます。

更に速度の高いアプリケーションに対しては、SpreadCycle が StealthChop2 に替わるオプションとなります。StealthChop2 および SpreadCycle は、組み合わせることで以下の両者の利点をどちらも生かすことが可能です。StealthChop2 は無騒音の停止、静音、円滑な性能を実現するのに適しており、SpreadCycle はより高い速度で用いた場合に高運動性能と最高ピーク速度を低振動で実現するのに適しています。

SpreadCycle はサイクルごとの高度なチョッパー・モードです。広い速度範囲および負荷範囲にわたり、円滑な動作と良好な共振減衰が可能です。SpreadCycle のチョッパー方式では、高速減衰サイクルを自動的に統合・調整できるため、滑らかなゼロ交差性能を確実に実現します。

### 利点

- 低コスト・モーターでのマイクロステッピング性能が大幅に向上。
- モーターは滑らかで静かに動作。
- スタンバイ時のノイズが完全に無音。
- 機械的な共振が抑えられているためトルク出力が向上。

## StallGuard2/StallGuard4 – 機械的負荷検出

StallGuard2 および StallGuard4 では、モーターに加わる負荷を正確に測定できます。これは、ストール検出や、モーターをストールさせる下限未満の負荷の場合におけるその他の用途（CoolStep 負荷適応型電流低減など）で用いることができます。

これにより駆動について得られる情報が多くなるため、センサーレス・ホーミングや駆動機能の診断などの機能が可能になります。StallGuard2 は SpreadCycle チョッパーと組み合わせられますが、StallGuard4 を StealthChop2 と組み合わせるには別の原理を用います。

## CoolStep – 負荷適応型電流制御

CoolStep は、最適な電流でモーターを駆動します。これは StallGuard2 または StallGuard4 の負荷測定情報を用いて、実際の負荷状態に必要な最低限の量となるようモーター電流を調整します。

CoolStep を用いることでエネルギーを節約でき、部品の発熱が抑えられます。CoolStep では最適な電流でモーターを駆動するため、トルク・リザーブが約 50%の標準的な動作と比較して、モーター効率が向上します。

### 利点

- 最高のエネルギー効率、消費電力は最大 75%低減。
- モーターの発熱が減少。
- 機械的精度が向上。
- 冷却が軽減または不要。
- 信頼性が向上。
- より小型のモーターを使用可能、必要なトルク・リザーブが軽減。
- モーターの励振エネルギーが低減するためモーター・ノイズが減少。

## エンコーダ・インターフェース

TMC5241 は、外部インクリメンタル・エンコーダ用にエンコーダ・インターフェースを備えています。エンコーダは、モーション・コントローラのホーミングに（リファレンス・スイッチの代替として）使用したり、エンコーダ・ポジションとランプ・ジェネレータ・ポジションの整合性チェックをオンザフライで行うために使用したりできます。プログラマブル・プリスケアラを用いると、エンコーダの分解能をモーターの分解能に合わせるすることができます。32 ビットのエンコーダ・カウンタが備わっています。

## シリアル・ペリフェラル・インターフェース（SPI）

### SPI データグラム構造

TMC5241 は、40 ビットの SPI データグラムを用いてマイクロコントローラと通信します。ハードウェア SPI を備えたマイクロコントローラは、通常、8 の整数倍のビットを用いて通信できます。TMC5241 は、シングル・チェーン内で 5 バイトの倍数を使用して IC をカスケード接続できます。シングル・バイトを使用してサード・パーティのペリフェラルをカスケード接続することが可能です。この場合、ダミー・バイトを挿入して 5 バイトの倍数にする必要があります。データグラムの伝送期間中、デバイスの CSN ラインはアクティブ（ロー）状態を維持する必要があります。

デバイスに送られる各データグラムは、アドレス・バイトとそれに続く 4 データ・バイトで構成されます。これにより、レジスタ・セットとの間で直接、32 ビットのデータ・ワード通信を行うことができます。各レジスタは、使用するデータ・ビットが 32 ビット未満であっても、32 データ・ビットを通じてアクセスできます。

簡単化のため、各レジスタは以下のように 1 バイト・アドレスで指定されます。

- 読出しアクセスでは、アドレス・バイトの最上位ビットが 0 です。
- 書込みアクセスでは、アドレス・バイトの最上位ビットが 1 です。

全レジスタが読出し可能で、その大半は読書き可能、一部は読出し専用です。また、一部のレジスタをクリアするには 1 を書き込みます (例: GSTAT レジスタ)。

表 1. SPI データグラム構造

MSB (TRANSMITTED FIRST)		40-BIT				LSB (TRANSMITTED LAST)			
39 ... 0									
Write: 8 bit address Read: 8 bit SPI status		Read/write 32-bit data							
39 ... 32		31 ... 0							
Write to RW + 7-bit address		8-bit data		8-bit data		8-bit data		8-bit data	
Read from 8-bit SPI status		8-bit data		8-bit data		8-bit data		8-bit data	
39/38 ... 32		31 ... 24		23 ... 16		15 ... 8		7 ... 0	
W	38...32	31...28	27...24	23...20	19...16	15...12	11...8	7...4	3...0

### 書込み／読出し選択 (WRITE\_notREAD)

読出しと書込みの選択は、アドレス・バイトの MSB (SPI データグラムのビット 39) で制御されます。このビットは読出しアクセスの場合は 0、書込みアクセスの場合は 1 です。そのため、W と記されたビットは WRITE\_notREAD 制御ビットです。アクティブ・ハイの書込みビットはアドレス・バイトの MSB です。したがって、書込みアクセスではアドレスに 0x80 を追加する必要があります。SPI は、W ビットに関わらず、データを常にコントローラに送り返します。直前のアクセスが読出しアクセスであった場合、返されるデータは、直前のデータグラムで伝送されたアドレスで読み出されるデータです。直前のアクセスが書込みアクセスであった場合は、リード・バックされるデータはその前に受信した書込みデータを反映したものになります。したがって、読出しアクセスと書込みアクセスの違いは、読出しアクセスではアドレス指定されたレジスタにはデータを伝送せずにアドレスのみを伝送し、その 32 データ・ビットはダミーであるという点にあります。また、更に続けて読出しまたは書込みアクセスを行うと、その前の読出しサイクルで伝送されたアドレスからのデータ読出しが返されます。

読出しアクセス要求のデータグラムはダミーの書込みデータを使用します。読出しデータは、その後の読出しアクセスまたは書込みアクセスで再びコントローラに返されます。したがって、パイプライン方式で複数のレジスタを読み出せます。

TMC5241 との間でデータの読出しや書込みが行われる場合は必ず、返される MSB に SPI のステータスが含まれます。SPI\_STATUS は 8 個のうちから選択されたいくつかのステータス・ビットです。

#### 例：

アドレス 0x21 のレジスタ (XACTUAL) への読出しアクセスの場合、読出しアクセスの直前のアクセス時にアドレス・バイトを 0x21 に設定する必要があります。レジスタ (VACTUAL) への書込みアクセスの場合、アドレス・バイトは 0x80 + 0x22 = 0xA2 に設定する必要があります。読出しアクセスの場合、データ・ビットは任意の値 (-) を持つことが可能です。そのため、0 に設定することもできます。

表 2. SPI 読み出し/書き込みフローの例

ACTION	DATA SENT TO TMC5241	DATA RECEIVED FROM TMC5241
Read TSTEP	0x1200000000	0xST and unused data*
Read TSTEP	0x1200000000	0xST and TSTEP
Write X_ENC = 0x00ABCDEF	0xB900ABCDEF	0xST and TSTEP

\*SS はステータス・ビット SPI\_STATUS のためのブレースホルダです。

### 各データグラム・リード・バックで転送される SPI ステータス・ビット

新たなステータス情報は、各アクセスの最後にラッチされ、次の SPI 転送で使用できます。

表 3. SPI\_STATUS - ビット 39~32 の各 SPI アクセスで転送されるステータス・フラグ

ビット	名称	コメント
7	Status_stop_r	RAMP_STAT[1] - 1: 停止右側スイッチのステータスを示す (モーション・コントローラ専用)
6	Status_stop_l	RAMP_STAT[0] - 1: 停止左側スイッチのステータスを示す (モーション・コントローラ専用)
5	Position_reached	RAMP_STAT[9] - 1: 目標ポジションに到達したことを示す (モーション・コントローラ専用)
4	Velocity_reached	RAMP_STAT[8] - 1: 目標速度に到達したことを示す (モーション・コントローラ専用)
3	Standstill	DRV_STATUS[31] - 1: モーターが停止状態であることを示す
2	Sg2	DRV_STATUS[24] - 1: StallGuard フラグがアクティブであることを示す
1	Driver_error	GSTAT[1] - 1: ドライバ・エラーを示す (GSTAT のビットをリセットすることでクリア)
0	Reset_flag	GSTAT[0] - 1: リセットが発生したことを示す (GSTAT のビットをリセットすることでクリア)

### データ・アライメント

すべてのデータは右揃えです。一部のレジスタは符号なし (正) の値を表し、一部のレジスタは整数値 (符号付き) を 2 の補数で表します。1 つのビットまたはビットのグループは、それぞれ単一ビットまたは整数グループで表されます。

### SPI の信号

TMC5241 の SPI バスには次の 4 種類の信号があります。

- SCK - バス・クロック入力
- SDI - シリアル・データ入力
- SDO - シリアル・データ出力
- CSN - チップ・セレクト入力 (アクティブ・ロー)

チップ・セレクト入力 CSN をローにすることで、SPI トランザクションの SPI ペリフェラルがイネーブルされます。ビット転送は、バス・クロック SCK に同期します。ペリフェラルは、SCK の立上がりエッジで SDI からのデータをラッチし、立下がりエッジの後 SDO にデータを送ります。最初に最上位ビットが送られます。TMC5241 でバス・トランザクションを行うには、最低でも 40 SCK クロック・サイクルが必要です。

40 より多くのクロックを駆動する場合、SDI にシフト・インされた追加ビットは、内部シフト・レジスタを介して 40 クロックの遅延後に SDO にシフト・アウトされます。これを用いると、複数チップをデジタイズチェーン接続できます。

バス・トランザクション全体を通じて、CSN はローにする必要があります。CSN がハイになると、内部シフト・レジスタの内容は内部制御レジスタにラッチされ、SPI コントローラから SPI ペリフェラルへのコマンドとして認識されます。40 を超えるビットが送られた場合、CSN の立上がりエッジの前に受信された最後の 40 ビットのみが、コマンドとして認識されます。

### SPI のタイミング

SPI の最大周波数は 8MHz です。SCK はシステムのクロック周波数とは無関係ですが、クロック周波数に依存する唯一のパラメータが最小 CSN ハイ時間です。すべての SPI 入力には内部でフィルタ処理が行われるため、10ns より短いパルスでトリガされることはありません。以下の図には、SPI バス・トランザクションのタイミング・パラメータが示されています。タイミング値は電氣的特性の表に記載されています。

SPI は SPI MODE 3 を使用します。

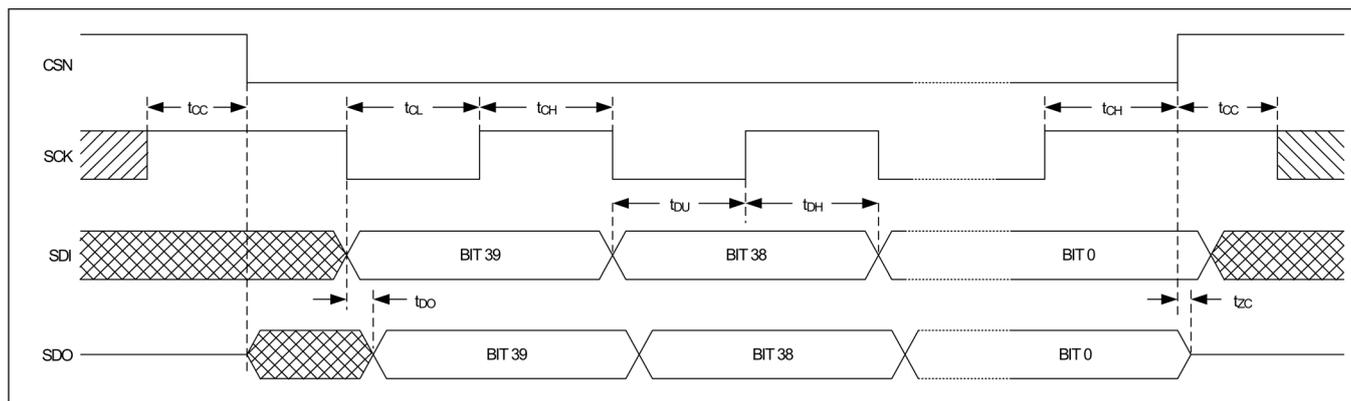


図 5. SPIのタイミング図

### UART 単線式インターフェース

UART 単線式インターフェースでは、任意のマイクロコントローラの UART を用いて TMC5241 を制御できます。これは、RS485 に基づくインターフェースと同様に、送信ラインと受信ラインを共有します。データ伝送は巡回冗長検査を用いて安全性が確保されるため、電磁的な擾乱が発生しても誤ったコマンドが伝送されたりコマンドが消失したりするおそれがなく、長いインターフェース距離（2 つの PCB 間のケーブルを介する場合など）を接続できます。自動ボー・レート検出機能を備えているため、このインターフェースは容易に導入可能です。

### データグラム構造

### 書込みアクセス

表 4. UART 書込みアクセス・データグラム構造

EACH BYTE IS LSB...MSB, HIGHEST BYTE TRANSMITTED FIRST																				
0 ... 63																				
Sync + Reserved					8-bit node address				RW + 7-bit register address				32-bit data				CRC			
0...7					8...15				16...23				24...55				56...63			
1	0	1	0	Reserved (don't cares but included in CRC)				NODEADDR				Register address		1	Data bytes 3, 2, 1, 0 (high to low byte)				CRC	
0	1	2	3	4	5	6	7	8	...	15	16	...	23	24	...	55	56	...	63	

同期ニブルが TMC5241 との間の伝送の前に置かれ、最初の伝送バイトに組み込まれた後にアドレス指定バイトが置かれます。伝送ごとに、内部ボー・レート分周器と UART ホスト・クロックの同期が可能で、実際のボー・レートは適応型であり、内部クロック周波数の変化は補償されます。そのため、ボー・レートは有効な範囲内で自由に選択できます。伝送される各バイトは、開始ビット (DIAG1/SW のロジック 0、ロー・レベル) で始まり、停止ビット (DIAG1/SW のロジック 1、ハイ・レベル) で終了します。ビット時間は、開始ビットの始まり (1 から 0 への遷移) から同期フレームの終了 (ビット 2 およびビット 3 の 1 から 0 への遷移) までの時間を計測して計算されます。すべてのデータはバイト単位で送信されます。32 ビット・データ・ワードは最上位バイト・ファーストで伝送されます。

20MHz のクロックを仮定した場合、9000 ボーの最小ボー・レートが可能です (低ボー・レートの最も厳しい場合)。最大ボー・レートは、ボー・クロックの必要な安定性を確保するため、fclk/16 です。

最初のペリフェラル・アドレス NODEADDR は、0~7 の範囲の CSN\_AD2、SCK\_AD1、および SDI\_AD0 で選択できます。

ペリフェラル・アドレスはレジスタ NODEADDR と上記ピン選択の和で決まります。これは、SDI がハイ・レベルになる (および CSN がロー、SCK がローになる) ことによって NODEADDR の設定が 1 ずつインクリメントされることを意味します。

レジスタ・アドレスのビット 7 は読出し (0) アクセスまたは書込み (1) アクセスを識別します。例: アドレス 0x10 は書込みアクセスでは 0x90 に変更されます。

2 つの連続バイトの開始ビットの間隔が 63 ビット分より長い休止時間になった場合、通信はリセットされます。このタイミングは最後に正しく受信されたデータグラムに基づきます。この場合、バス・アイドル時間である 12 ビット分以上の故障回復時間が経過してから、伝送を再開する必要があります。この方式をとることで、伝送エラーの場合に UART ホストが通信をリセットできます。16 クロック・サイクルに満たないアイドル・データ・ラインのパルスはグリッチとみなされ、12 ビット分の時間のタイムアウトの原因となります。この間はデータ・ラインがアイドル状態になる必要があります。CRC の誤りなど、その他のエラーも同様に処理されます。これにより、エラー状態の後の伝送を安全に再同期できます。このメカニズムにより、ボー・レートが直前の値の 15%未滿に突然減少することはあり得ないことに注意してください。

受け入れられた各書き込みデータグラムは、レシーバによって内部のサイクリック・データグラム・カウンタ（8 ビット）をインクリメントすることでアクノレッジされます。UART ホストは、データグラム・カウンタを読み出すことで、初期化シーケンスまたは単一書き込みアクセスが成功したかどうかをチェックできます。読出しアクセスではカウンタは変更されません。

## 読出しアクセス

表 5. UART 読出しアクセス要求データグラムの構造

EACH BYTE IS LSB...MSB, HIGHEST BYTE TRANSMITTED FIRST																
Sync + Reserved				8-bit node address				RW + 7-bit register address				CRC				
0...7				8...15				16...23				24...31				
1	0	1	0	Reserved (don't cares but included in CRC)				NODEADDR				Register address		0	CRC	
0	1	2	3	4	5	6	7	8	...	15	16	...	23	24	...	31

読出しアクセス要求データグラムの構造は、書き込みアクセスのデータグラム構造と同じですが、使用するユーザ・ビットの数が少なくなっています。その機能は、UART ノードのアドレス指定と、読出しアクセスに必要なレジスタ・アドレスの伝送です。TMC5241 は、UART ホストが読出し要求のために用いるボー・レートと同じレートで応答します。ホストからノードへのクリーンなバス遷移を確保するために、TMC5241 は読出しアクセスに対する返答をすぐには送信せず、プログラマブルな遅延時間を用い、その時間が経過した後、読出し要求に続いて最初の返答バイトを送信します。

この遅延は、UART ホストの必要に応じて、SENDDELAY 時間設定を用い 8 ビットの倍数分の時間で設定できます（デフォルトは 8 ビット分の時間）。複数ノードのシステムでは、すべてのノードに対して SENDDELAY を最小でも 2 に設定します。そうしない場合、異なるノードへの読出しアクセス時に、アドレス指定されていないノードが伝送エラーを検出する可能性があります。

表 6. UART 読出しアクセス返答データグラムの構造

EACH BYTE IS LSB...MSB, HIGHEST BYTE TRANSMITTED FIRST																					
0 ..... 63																					
Sync + Reserved								8-bit node address				RW + 7-bit register address				32-bit data				CRC	
0...7								8...15				16...23				24...55				56...63	
1	0	1	0	Reserved (0)				0xFF				Register address		0		Data bytes 3, 2, 1, 0 (high to low byte)				CRC	
0	1	2	3	4	5	6	7	8	...	15	16	...	23	24	...	55	56	...	63		

読出し応答は、アドレス・コード%11111111 を用いて UART ホストに送信されます。最後のビットが送信された後、トランスミッタは、4 ビット分の時間、インアクティブに切り替わります。

アドレス%11111111 は、UART ホストへの読出しアクセス用に予約されています。ノードではこのアドレスを使用することはできません。

## CRC の計算

8 ビット CRC 多項式を用いて、読出しおよび書き込み両方のアクセスをチェックします。これにより、最大 8 個のシングルビット・エラーを検出できます。初期値がゼロの CRC8-ATM 多項式を、同期バイトとアドレス指定バイトを含む LSB～MSB に適用します。同期ニブルは常に正しいと仮定されています。TMC5241 は、固有のノード・アドレスを含む、正しく伝送されたデータグラムに対してのみ、応答します。書き込みアクセス・データグラムが正しく受信されるごとに、データグラム・カウンタが増加します。

$$\text{CRC} = x^8 + x^2 + x^1 + x^0$$

シリアル計算例：

$$\text{CRC} = (\text{CRC} \ll 1) \text{ OR } (\text{CRC}.7 \text{ XOR } \text{CRC}.1 \text{ XOR } \text{CRC}.0 \text{ XOR } [\text{new incoming bit}])$$

### CRC 計算の C コード例

```

void swuart_calcCRC(uint8_t* datagram, uint8_t datagramLength)
{
    // Initialization
    int i,j;
    uint8_t* crc = &datagram[datagramLength-1]; // CRC located in last byte of message
    uint8_t currentByte;
    *crc = 0;

    for (i = 0; i < (datagramLength-1); i++)
    {
        // Execute for all bytes of a message
        currentByte = datagram[i]; // Retrieve a byte to be sent from Array
        for (j = 0; j < 8; j++)
        {
            if ((*crc >> 7) ^ (currentByte & 0x01))
            {
                // update CRC based result of XOR operation
                *crc = (*crc << 1) ^ 0x07;
            }
            else
            {
                *crc = (*crc << 1);
            }
            currentByte = currentByte >> 1;
        } // for CRC bit
    } // for message byte
}

```

### UART の信号

TMC5241 の UART インターフェースは 5 種類の信号で構成されます。UART モードでは、各ノードは、その固有アドレスを持つデータグラムを正しく受信したかどうかを確認するために、単線ピン DIAG1/SW を連続的にチェックします。ピンはこの期間、入力に切り替わります。前述のように、このピンは同期エッジに基づくボー・レートに対応します。読出しアクセスの場合、DIAG1/SW の出力ドライバをオンに切り替え、同じボー・レートで応答を送信します。

表 7. TMC5241 UART のインターフェース信号

信号	説明
DIAG1/SW	データの入出力
CSN/AD2	UART アドレスのビット 2 をインクリメント (+4)
SCK/AD1	UART アドレスのビット 1 をインクリメント (+2)
SDI/AD0	UART アドレスのビット 0 をインクリメント (+1)、チェーン接続している前段の IC の NAO に接続
SDO/NAO	チェーン・シーケンシャル・アドレス指定スキームの NAO ピン (リセット後のデフォルト=ハイ)

## 複数ノードのアドレス指定

1~8 個の TMC5241 を単一の UART バス・インターフェースを用いてホストがアドレス指定する場合、単純なハードウェア・アドレス選択を使用できます。個々の UART ノード・アドレスの設定は、UART アドレス・ピン (SDI、SCK、CSN) を V<sub>CC\_10</sub> および GND に接続することで行います。

8 個を超えるノードを同じ UART バスに接続する必要がある場合は、異なる手法を用いる必要があります。この手法では、出力 NAO (SDO) を次のデバイスのビット 0 アドレス・ピン用の選択ピンとして用いることで、最大 255 個のノードのアドレスを指定できます。手順は次のとおりです。

- すべてのアドレス・ピンおよび最初の TMC5241 の SDI/AD0 を GND に接続します。
- 最初の TMC5241 の SDO/NAO の出力を次のノードのアドレス[0]ピン (SDI/AD0) に接続します。同様に他のノードを接続します。
- これで、最初のノードがアドレス 0 に応答します。次のノードはアドレス 1 に設定します。
- 最初の TMC5241 をその特定のノード・アドレスにプログラムします。**注**：ノードがそのノード・アドレスで初期化された後は、次のノードのアドレス[0]ピン (SDI/AD0) に接続されているその SDO/NAO 出力をロジック 0 にプログラムして、次のノードをすべての後続のノードと区別できるようにする必要があります。
- これで、2 番目のノードがアクセス可能となり、特定のノード・アドレスを持つことができます。引き続きこれ以降のノードもそれぞれの特定ノード・アドレスにプログラムできます。

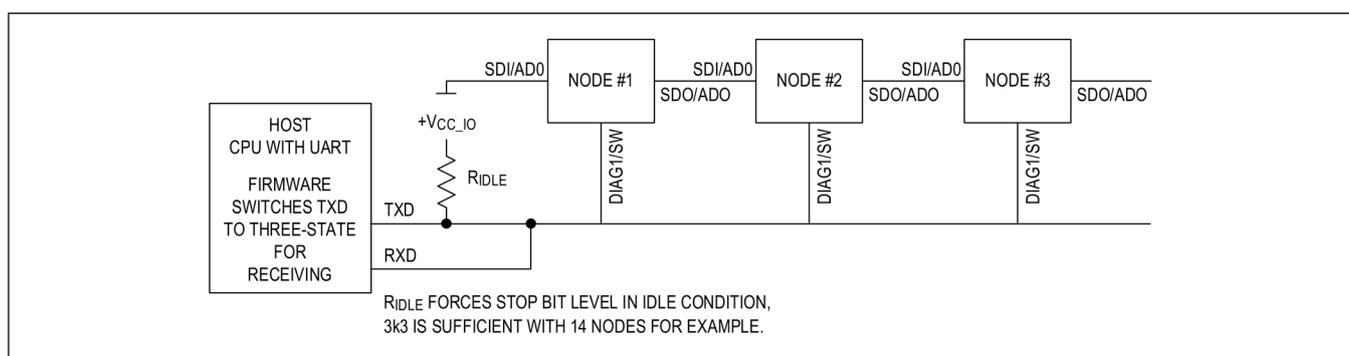


図 6. UART のデジーチェーン例

表 8. 最大 255 ノードをアドレス指定する UART 例

PHASE	NODE #1	NODE #2	NODE #3
Addressing phase 1	Address 0, NAO is high	Address 1	Address 1
Addressing phase 2	Program to address 254 and set NAO low	Address 0, NAO is high	Address 1
Addressing phase 3	Address 254	Program to address 253 and set NAO low	Address 0
Addressing phase 4	Address 254	Address 253	Program to address 252 and set NAO low
Addressing phase x	Continue procedure		

## StealthChop2

StealthChop2 は、ステップング・モーター用の極めて静かな動作モードです。これは電圧モードのパルス幅変調 (PWM) をベースとします。停止時や低速時、モーターは絶対的に無雑音です。したがって、StealthChop2 で動作するステップング・モーター・アプリケーションは、屋内や家庭での使用に最適です。モーターは、低速時にはまったくの無振動で動作します。

StealthChop を用いる場合には、電圧モードの PWM を用いて特定の実効電圧をコイルに印加することでモーター電流を供給します。機能強化された StealthChop2 では、ドライバはアプリケーションが最高性能を発揮できるように自動的に適応します。新たな設定は不要です。オプションの設定を行うと、特別な場合の設定を調整したり、自動適応アルゴリズムの初期値を設定したりできます。高速駆動では、SpreadCycle を StealthChop2 と組み合わせることを検討する必要があります。

StealthChop2 を試してみる場合には、現在使用しているアプリケーション内でモーターを動作させてください。モーター性能は多くの場合、機械的な負荷があるほうが良好です。負荷がない場合には機械的な振動が生じ、それによってモーターがストールしてしまう可能性があります。負荷があればそれを防止できるためです。

## 自動調整

StealthChop2 には自動調整 (AT) プロシージャが内蔵されており、極めて重要な動作パラメータを自動的にモーターに適用します。したがって、StealthChop2 では、高いモーター・ダイナミクスが可能で、モーターを非常に小さな電流までパワー・ダウンできます。最良の結果を得るために考慮すべきステップはわずか 2 つです。まず、停止状態からモーターを始動します。ただし、公称稼働電流を供給します (AT#1)。次に、モーターを例えばホーミング・プロシージャの一部として、中間速度にします (AT#2)。図 7 のフローチャートに、この調整プロシージャを示します。

表 9. StealthChop2 自動調整 AT#1 および AT#2 の制約事項と条件

ステップ	パラメータ	条件	必要な経過時間
AT#1	PWM_OFS_AUTO	モーターは停止状態にあり、実際の電流スケール (CS) は稼働時の電流 (IRUN) と同じ。 停止をなくすことが可能な場合は、初期ステップ・パルスが駆動を稼働電流に戻すか、IHOLD を IRUN に設定するよう切り替え。 ピン $V_s$ は動作レベル。	$\leq 220 + 2 \times 2^{18} t_{CLK}$ , $\leq 130\text{ms}$ (内部クロック使用)
AT#2	PWM_GRAD_AUTO	大量のバック EMF が生じ、最大稼働電流に達する可能性のある速度でモーターを駆動。 <b>条件:</b> $1.5 \times \text{PWM\_OFS\_AUTO} \times (\text{IRUN} + 1)/32 < \text{PWM\_SCALE\_SUM} < 4 \times \text{PWM\_OFS\_AUTO} \times (\text{IRUN} + 1)/32$ $\text{PWM\_SCALE\_SUM} < 255$ <b>ヒント:</b> 代表的な範囲は 60RPM~300RPM です。	$\pm 1$ の変更には 8 個のフルステップが必要です。 PWM_GRAD_AUTO の最適値が 50 以下の代表的なモーターの場合、デフォルト値 0 から開始するには最大 400 フルステップが必要です。

**ヒント:** 自動調整用の最適条件を決定するには、評価用ボードを用います。

初期調整パラメータを求めるには、ファームウェア初期化用の PWM\_GRAD と PWM\_OFS にアプリケーション固有のパラメータを使用します。

AT#2 の調整時には、一定の速度フェーズの間に PWM\_SCALE\_AUTO がゼロに近付いて行くのをモニタします。これは調整がうまくいったことを示します。

### 注意:

適切な調整を行わずに StealthChop2 で動作させると、特に、低抵抗モーターで高速の減速設定を行う場合、減速ランプ中にモーター電流が増加する原因となります。自動調整手順に従い、評価用ボードを用いて最適な調整条件をチェックしてください。モーター・タイプごとに定める PWM\_OFS および PWM\_GRAD の設定値には、初期値を用いることを推奨します。

GLOBALSCALER や  $V_s$  電圧を変更すると、自動調整プロセスの結果は無効になります。モーターの電流レギュレーションでは、次の AT#1 フェーズになるまで大きな変化を補償できません。AT#1 条件と AT#2 条件がその後の動作で満たされれば必ず、自動調整は変更された条件に適応します。

## StealthChop2 のオプション

モーター電流を特定のレベルに一致させるために、実際のモーター速度に応じて実効的な PWM 電圧がスケールリングされます。目標電流でモーターを駆動するために必要な電圧レベルには、モーター抵抗、そのバック EMF (例えば、その速度に正比例)、電源電圧の実際のレベルといった、いくつかの要素が影響します。PWM レギュレーションには 2 つのモードがあります。1 つは、電流帰還を用いる自動調整 (AT) モード (`pwm_autoscale = 1`, `pwm_autograd = 1`)、もう 1 つは、フィードフォワード速度制御モード (`pwm_autoscale = 0`) です。

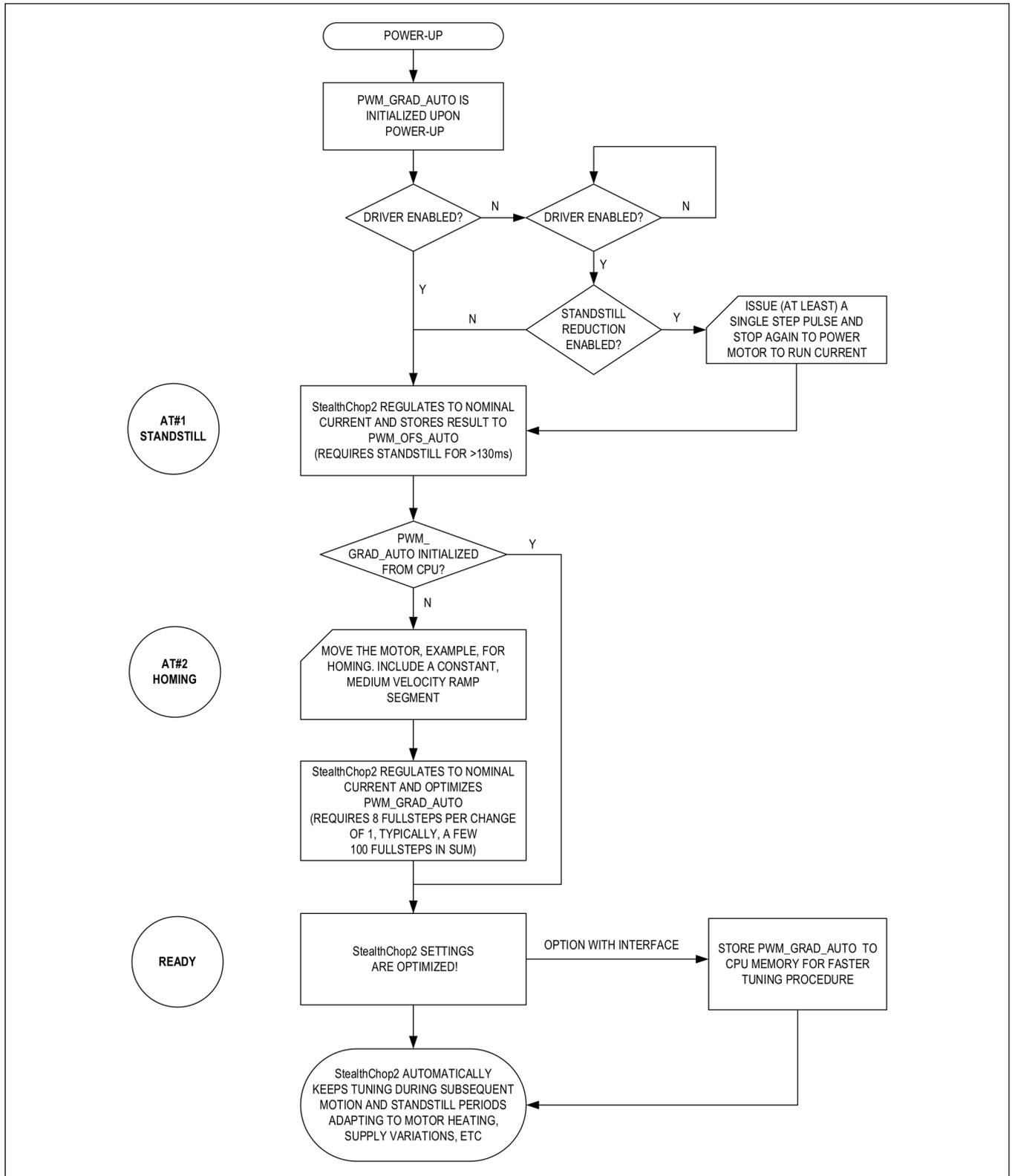


図 7. StealthChop2 の自動調整手順

フィードフォワード速度制御モードは、電源電圧の変動や、モーター・ストールなどのイベントには反応しませんが、非常に安定した振幅が可能です。電流測定手段は使用せず、また、必要でもありません。このモードは、モーターのタイプや電源電圧がよくわかっている場合に最適です。そのため、与えられた動作条件での電流レギュレーションに問題がある場合でない限り、自動モードを推奨します。

モーターのタイプと電源電圧に合ったアプリケーション固有の初期調整パラメータを用いることを推奨します。また、モーターの過熱や電源電圧の変動などによるパラメータの変化に対応するためには、自動調整モードで動作させてください。

非自動モード (`pwm_autoscale = 0`) を考慮するのは、モーターと動作条件がよくわかっている場合のみにしてください。この場合、インターフェースを使用した注意深いプログラミングが必要です。PWM\_GRAD と PWM\_OFS の 2 つの動作パラメータは、最初に自動調整モードで決定できます。

StealthChop2 の PWM 周波数を 4 ステップで選択することで、周波数分周器をクロック・ソースの周波数に合わせるすることができます。ほとんどのアプリケーションでは、20kHz~50kHz の範囲での設定が適しています。これにより、低電流リップルおよび高速時の良好な性能と動的な消費電力とのバランスをとることができます。

表 10. StealthChop2 の PWM 周波数の選択

CLOCK FREQUENCY $f_{CLK}$	PWM_FREQ = %00 $f_{PWM} = 2/1024 f_{CLK}$	PWM_FREQ = %01 $f_{PWM} = 2/683 f_{CLK}$	PWM_FREQ = %10 $f_{PWM} = 2/512 f_{CLK}$	PWM_FREQ = %11 $f_{PWM} = 2/410 f_{CLK}$
20MHz	39.1kHz	58.1kHz	78.1kHz	97.6kHz
18MHz	35.2kHz	52.7kHz	70.3kHz	87.8kHz
16MHz	31.3kHz	46.9kHz	62.5kHz	78.0kHz
12.5MHz (internal)	24.4kHz	36.6kHz	48.8kHz	61.0kHz
10MHz	19.5kHz	29.3kHz	39.1kHz	48.8kHz
8MHz	15.6kHz	23.4kHz	31.2kHz	39.0kHz

## StealthChop2 電流レギュレータ

StealthChop2 電圧 PWM モードでは、自動スケーリング機能 (`pwm_autoscale = 1`, `pwm_auto_grad = 1`) がモーター電流を目的の電流設定値に安定化します。自動スケーリングは、AT プロセスの一部として用いられ、また、その後のモーター・パラメータの変化をトラッキングするために用いられます。ドライバはチョッパーのオン時間中にモーター電流を測定し、モーター電流が目標電流に一致するよう、比例レギュレータを用いて PWM\_SCALE\_AUTO を安定化します。PWM\_REG は、このレギュレータの比例係数です。基本的に、安定した穏やかなレギュレーション特性を得るために、比例係数はできる限り小さくする必要がありますが、ドライバがアプリケーション・パラメータの変動による変化（モーターの負荷や電源電圧の変化など）に素早く反応できるだけの大きさを持っていないとなりません。

（このデバイス・ファミリでは  $V_{REF}$  は変動しません。）初期調整ステップ AT#2 では、PWM\_REG はモーター速度の変化の補償も行っています。そのため、AT#2 で高加速度を実現するには、PWM\_REG の設定を大きくする必要があります。ホーミングの速度と加速度を注意深く選択することで、多くの場合、レギュレーション勾配は最小設定 (PWM\_REG=1) にすれば十分です。PWM\_REG の設定は、必要な最高速の加速ランプや減速ランプについて最適化する必要があります（次の 2 つの図を比較してください）。

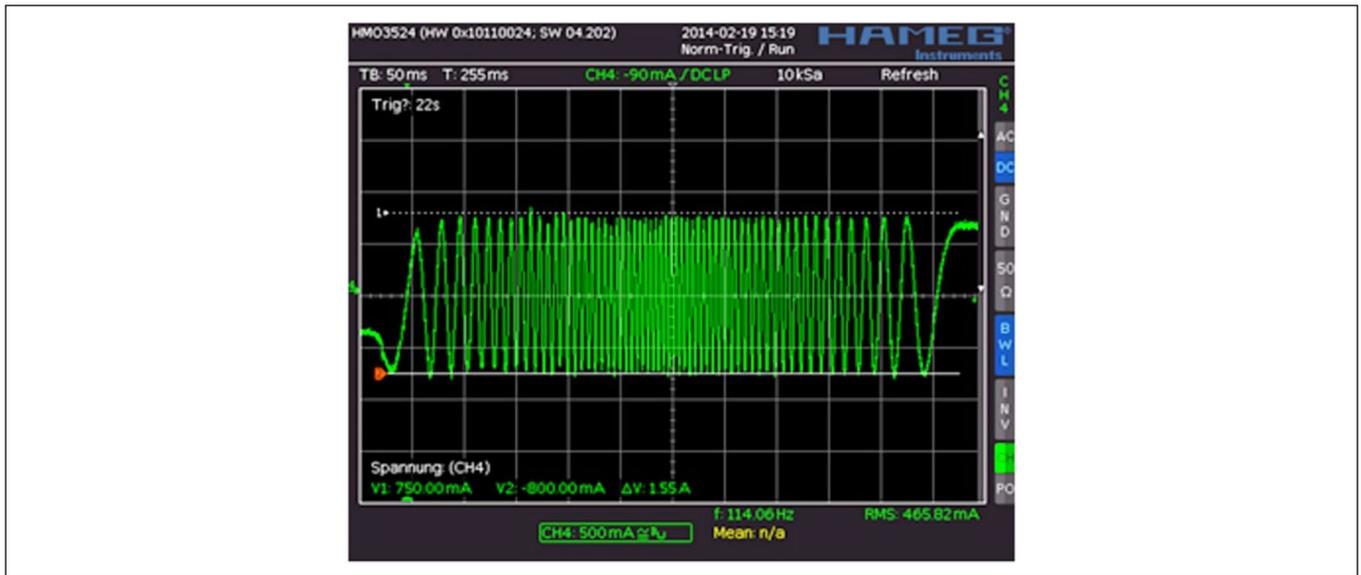


図 8. StealthChop2 : PWM\_REG の設定は良好

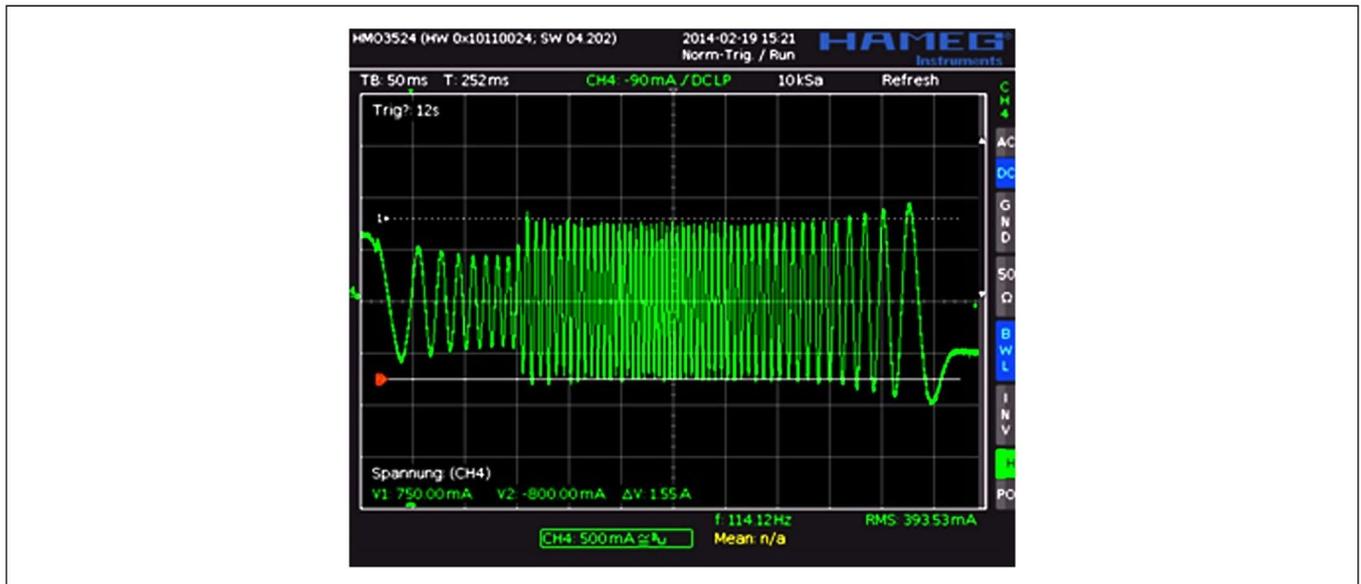


図 9. StealthChop2 : AT#2 時の PWM\_REG の設定が過小

AT#2 フェーズおよび完了した自動調整プロシージャでの PWM\_REG 設定値（または PWM\_OFS および PWM\_GRAD に対する非自動設定値）の質は、次の図に示すように、加速フェーズでのモーター電流のモニタリング時に調べることができます。

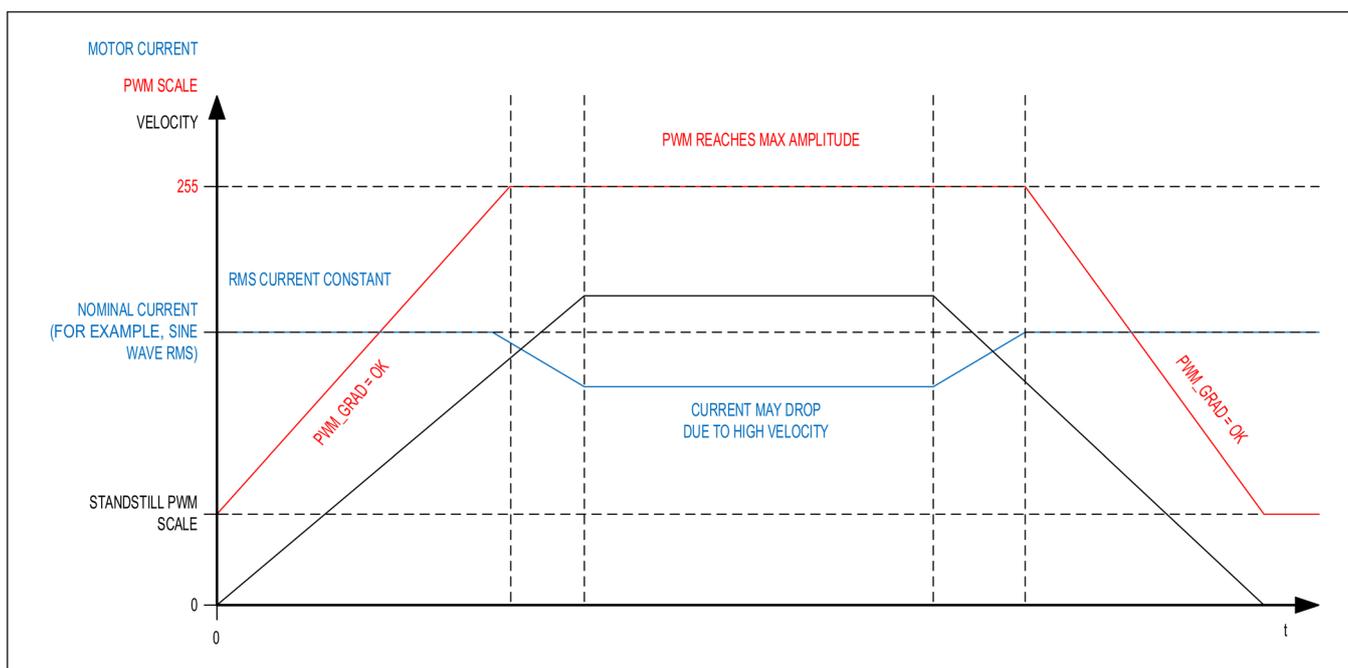


図 10. 正しく決定された PWM\_GRAD(\_AUTO)および PWM\_OFS(\_AUTO)

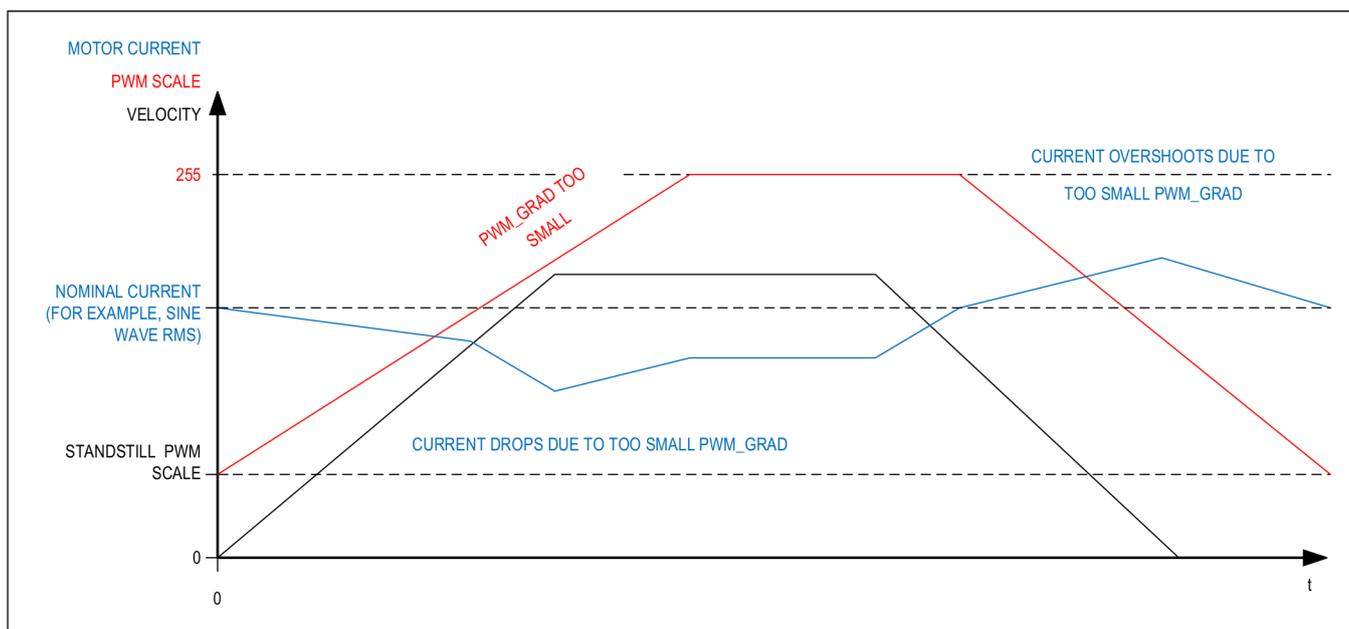


図 11. 過小な PWM\_GRAD 設定の例

## 下限電流値

pwm\_meas\_sd\_enable の設定に応じて、StealthChop2 の電流レギュレータ原理では、モーター電流レギュレーションに下限値を設けます。チョッパ・オンフェーズ時 (pwm\_meas\_sd\_enable = 0) にのみコイル電流を測定する場合、コイル電流のレギュレーションが可能な最小チョッパ・デューティサイクルは、TBL とチョッパ周波数設定値によって設定されるブランキング時間で与えられます。そのため、StealthChop2 自動スケーリング・モードでのモーター固有の最小コイル電流は、電源電圧およびチョッパ周波数と共に増加します。ブランキング時間が短いほど、電流制限値は小さくなります。PWM\_OFS\_AUTO を正しく定めるには、AT#1 時に、稼働電流、GLOBALSCALER、IRUN がレギュレーション範囲内に十分収まっていることが重要です。電流の低下（例えば停止パワー・ダウンのため）は、PWM\_OFS\_AUTO および PWM\_GRAD\_AUTO に基づき自動で行われます。これらは非自動電流スケーリングでの、PWM\_OFS および PWM\_GRAD に対応するものです。フリーホイーリング・オプションを使えば、モーター電流をゼロにできます。

StealthChop2 自動調整でのモーター・コイル下限電流値 (`pwm_meas_sd_enable=0`) は、次式で表せます。

$$I_{LowerLimit} = t_{BLANK} \times f_{PWM} \times \frac{V_S}{R_{COIL}}$$

ここで、 $V_S$  はモーターの電源電圧、 $R_{COIL}$  はモーターのコイル抵抗です。

$I_{LowerLimit}$  は、最小公称 IRUN モーター電流設定値のおおよその目安として扱うことができます。必要な設定に達するのにこの下限値では不十分な場合は、`pwm_meas_sd_enable=1` と設定されていることを確認してください。

$f_{PWM}$  は、PWM\_FREQ の設定で決まるチョッパー周波数です。

**例：**モーターのコイル抵抗が  $5\Omega$ 、電源電圧が  $24V$  とします。TBL=%01 および PWM\_FREQ=%00 の場合、 $t_{BLANK}$  は 24 クロック・サイクル、 $f_{PWM}$  は  $2/(1024 \text{ クロック} \cdot \text{サイクル})$  となり、次式が成立します。

$$I_{LowerLimit} = 24t_{clk} \times \frac{2}{1024t_{CLK}} \times \frac{24V}{5\Omega} = \frac{24}{512} \times \frac{24V}{5\Omega} = 225mA$$

これは、関連する設定をすべて考慮すると、自動調整のモーター目標電流値を  $225mA$  以上としなくてはならないことを意味します。この下限電流値は、GLOBALSCALER を通じてモーター電流を変更する場合にもあてはまります。

#### 注意：

自動調整の場合、下記のコイル電流下限値が適用されます。

IRUN  $\geq 8$  : IRUN が 8 未満の電流設定では、自動調整が機能しません。

**I<sub>LOWERLIMIT</sub> :** 自動調整用ビット `pwm_meas_sd_enable` (レジスタ PWM\_CONF[22]内) の設定に応じてコイル電流下限値が適用されます。自動調整フェーズ AT#1 でのモーター電流はこの下限値より大きくなくてはなりません。I<sub>LOWERLIMIT</sub> を計算するか、電流プローブを用いてそれを測定してください。IRUN および IHOLD を変更することで、動作中にモーターの稼働電流やホールド電流を電流下限値未満に設定することは、自動調整が正常に行われた後に可能です。

電流下限値は、GLOBALSCALER の変化へのドライバの対応能力も制限します。

電流下限値による制限を無効にするには、`pwm_meas_sd_enable` を 1 に設定します。これにより、低速減衰フェーズ時にコイル電流を IC で別途測定できます。

### 速度ベースのスケールリング

速度ベースのスケールリングは、クロック・サイクル単位で測定した 1 ステップおきの時間に基づき (例えば、TSTEP に基づき)、StealthChop2 振幅をスケールリングします。このコンセプトは、基本的に、電流測定を必要としません。レギュレーション・ループが不要であるためです。純粋な速度ベースのスケールリングは、`pwm_autoscale` を 0 に設定した場合のみ、プログラミングを通じて可能です。基本的な考え方は、モーターに目標電流を供給するのに必要な電圧の線形近似を得ることです。ステッピング・モーターには一定のコイル抵抗があり、そのため  $I = U/R$  の基本公式に基づき、目標電流を生じさせるには一定の電圧振幅が必要です。つまり、コイルの抵抗  $R$  と、PWM 値に比例する電源電圧  $U$  から、電流  $I$  が求まります。PWM\_OFS の初期値は次式で計算できます。

$$PWM\_OFS = \frac{374 \times R_{COIL} \times I_{COIL}}{V_S}$$

ここで、 $V_S$  はモーターの電源電圧、 $I_{COIL}$  は目標の実効値電流です。

実効的な PWM 電圧  $U_{PWM}$  ( $1/\sqrt{2}$  × ピーク値) は、8 ビットの分解能と PWM\_SCALE で示される実際の PWM 振幅の 248 サイン波ピークを考慮して、次のように求まります。

$$U_{PWM} = V_S \times \frac{PWM\_SCALE}{256} \times \frac{248}{256} \times \left( \frac{CS\_ACTUAL + 1}{32} \right) + PWM\_GRAD \times \frac{PWM\_SCALE}{374}$$

モーター速度の増加時には、モーターのバック EMF の電圧が増加します。バック EMF の電圧は、モーター速度に比例します。これにより、コイル抵抗に有効な PWM 電圧が低下し、これにより電流も減少します。TMC5241 は、これを補償するために、2 つ目の速度依存係数 (PWM\_GRAD) を備えています。このモードにおける全体的な実効 PWM 振幅 (PWM\_SCALE\_SUM) は、次式のように、マイクロステップ周波数に応じて自動的に計算されます。

$$PWM\_SCALE\_SUM = PWM_{OFS} \times \left( \frac{CS\_ACTUAL + 1}{32} \right) + PWM\_GRAD \times \frac{256}{TSTEP}$$

CS\_ACTUAL には、IHOLD および IRUN の定義に従い、あるいは CoolStep により個別に、実際の電流スケールリングが考慮されています。

$f_{STEP}$  は、相当する 256 マイクロステップ分解能と等価なマイクロステップ周波数、 $f_{CLK}$  はドライバに共有されるクロック周波数または実際の内部周波数です。

第一近似として、バック EMF が電源電圧から差し引かれ、その結果実効的な電流振幅は減少します。このようにして、PWM\_GRAD 設定の第一近似が次のように計算されます。

$$PWM\_GRAD = C_{BEMF} \left| \frac{V}{s} \right| \times 2\pi \times \frac{f_{clk} \times 1.46}{V_M \times MSPR}$$

$C_{BEMF}$  は、モーターのバック EMF 定数で、単位は V/ラジアン/秒です。

MSPR は、1/256 マイクロステップ分解能を基準とする 1 回転あたりのマイクロステップ数です。例えば、1.8° のモーターの場合、256 マイクロステップに 200 フルステップを乗じて 51200 になります。

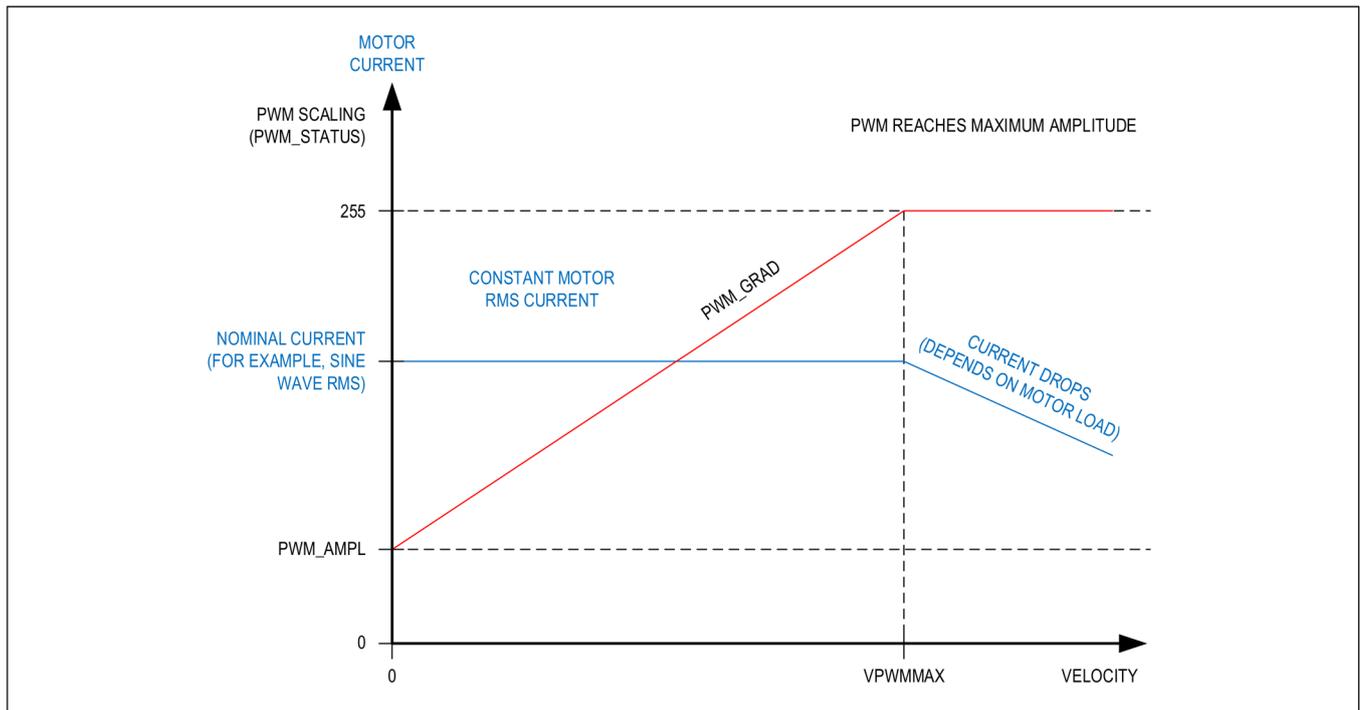


図 12. 速度ベースの PWM スケーリング (pwm\_autoscale = 0)

PWM\_OFS および PWM\_GRAD の値は、オシロスコープの電流プローブを用いてモーター電流を追跡することで、容易に最適化できます。あるいは、自動調整によりこれらの値を決定し、それを PWM\_OFS\_AUTO および PWM\_GRAD\_AUTO から読み出すこともできます。

### モーターのバック EMF 定数の解釈

バック EMF 定数は、モーターが特定の速度に調整されている場合に発生する電圧です。多くの場合、モーターのデータシートでは、この値は仕様規定されていません。モーターのトルクとコイル電流定格から計算できるためです。SI 単位系では、バック EMF 定数  $C_{BEMF}$  の数値は、トルク定数の数値と同じ値です。例えば、トルク定数が 1Nm/A のモーターの  $C_{BEMF}$  は 1V/rad/s です。そのようなモーターを 1rps (1rps = 毎秒 1 回転 = 6.28rad/s) で回転させると、発生するバック EMF の電圧は 6.28V となります。したがって、バック EMF 定数は次のように計算できます。

$$C_{BEMF} = \left| \frac{V}{s} \right| = \frac{HoldingTorque[Nm]}{2 \times I_{COILNOM}[A]}$$

$I_{COILNOM}$  は、指定保持トルクに対するモーターの定格実効値相電流です。

HoldingTorque はモーター固有の保持トルクで、例えば、両方のコイルでのトルクは  $I_{COILNOM}$  に達します。トルクの単位は[Nm]で、 $1Nm = 100Ncm = 1000mNm$  です。

電圧は、コイルごとの実効値電圧として有効です。したがって、公称電流値は 2 つのコイルが動作するフルステップ位置を前提とするため、この式における公称電流値には 2 が乗じられています。

### StealthChop2 と SpreadCycle の結合

高速度運動が必要なアプリケーションでは、SpreadCycle の方が、高速レンジにおいてより安定な動作をもたらす場合があります。無騒音動作と最大の動的性を組み合わせるため、TMC5241 では、速度閾値に基づき StealthChop2 と SpreadCycle を組み合わせることができません。この場合、StealthChop2 は低速時のみ有効です。

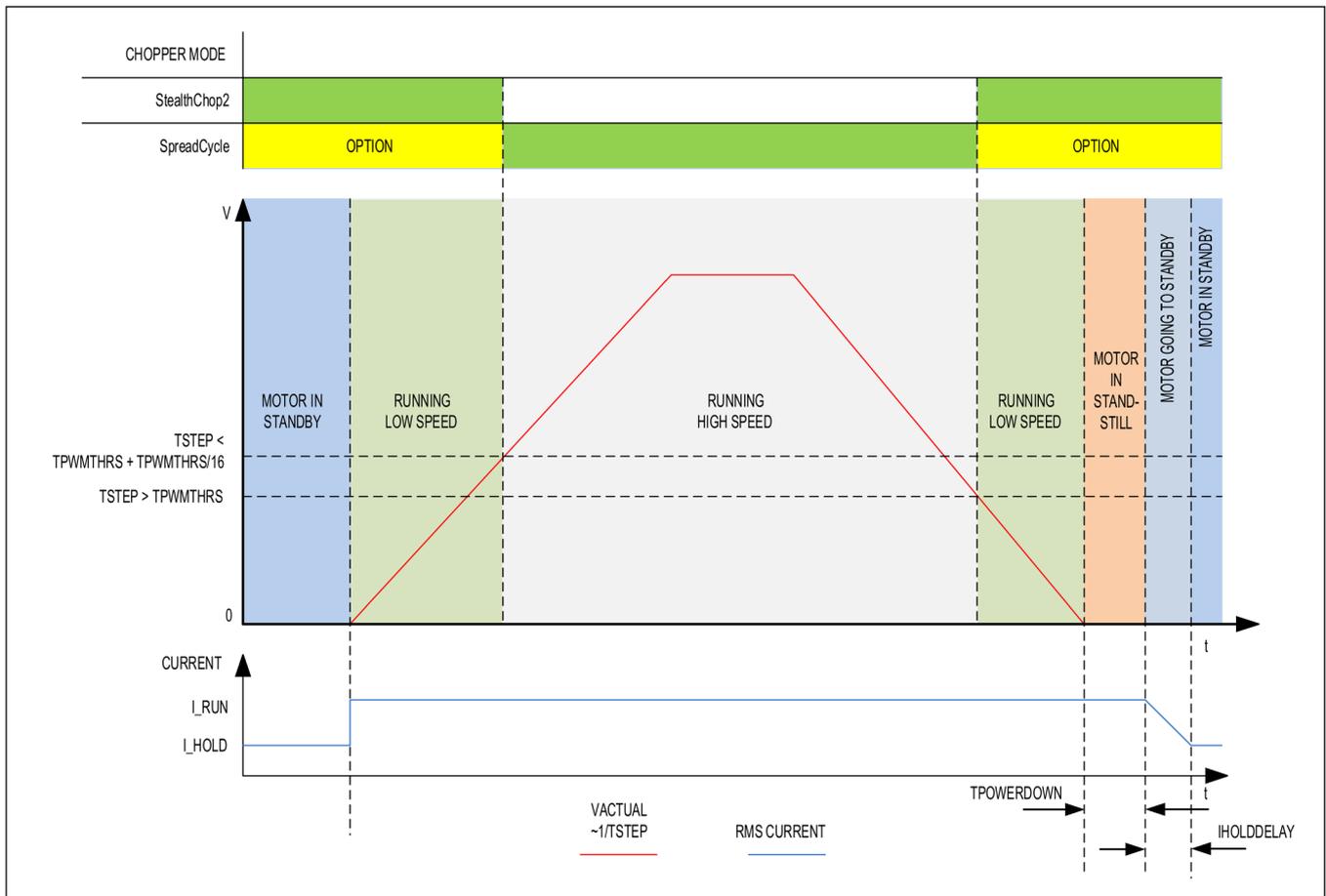


図 13. SpreadCycle への切り替えオプションのための TPWMTHRS

最初のステップとして、両方のチョッパー原理をパラメータ化し個別に最適化します。

次のステップでは、切り替え速度を定義する必要があります。例えば、StealthChop2 動作を用いて低速の位置設定を正確に行い、その一方で極めて動的な運動には SpreadCycle を用いることができます。TPWMTHRS がこの遷移速度を決定します。目的の速度で運動するときの TSTEP を読み出し、その結果の値を TPWMTHRS にプログラムします。低移動速度を用いると切り替え点でのジャークを回避できます。

### SpreadCycle へのジャークのない切り替え

ジャークが発生するのは高速で切り替えを行う場合です。それは、モーターのバック EMF (速度と共に増加) がモーター電圧とモーター電流の間の位相を最大で  $90^\circ$  シフトさせるためです。そのため、電圧 PWM モードと電流 PWM モード間での切り替え速度が速い場合、このジャークが高い強度で発生します。ジャークが大きい場合、一時的な過電流状態 (モーターのコイル抵抗に依存) が生じる場合さえあります。低速度時 (例えば 1RPM~数 10RPM) には、ほとんどのモーターでジャークは完全に無視できます。そのため、SpreadCycle と StealthChop2 の間でドライバを切り替える場合には、ジャークを考慮してください。TPWMTHRS で制御された自動切り替えでは、ドライバは StallGuard4 を用いて位相シフトを定めることにより自動的にジャークを除去できます。ドライバは、速度が切り替え閾値未満に戻るまで、SpreadCycle に対して同じ位相シフトを適用します。この機能を有効にするには、SG4\_THRS の sg\_angle\_offset にフラグを設定します。

StealthChop2 のみで動作させる場合には、TPWMTHRS をゼロに設定します。

自動電流レギュレーションを用いて初めて StealthChop2 モードを有効化する場合、適切な電流レギュレーションを可能にするために、モーターは停止状態になくはなりません。ドライブが高速で SpreadCycle に切り替わる場合、StealthChop2 のロジックは、モーターが再度低速に戻るまで、最後の電流レギュレーション設定を保存します。したがって、レギュレーションには低速状態に戻る場合の既知の開始点があり、ここから StealthChop2 が再度イネーブルになります。そのため、チョッパーが別のモードに切り替わるフェーズの間に、速度閾値も電源電圧も大きく変化してはなりません。変化すると、モーターはステップを喪失するか、あるいは瞬時電流が過大あるいは過小になるためです。

モーターがストールしたりモーター速度が突然変化したりした場合、ドライブが短絡を検出したり、自動電流レギュレーションの状態になったりする可能性があります。その場合は回復ができません。この状況から回復するには、エラー・フラグをクリアし、モーターを速度ゼロから再始動します。

初めて StealthChop2 に切り替える場合、StealthChop2 が初期停止電流制御を行うことができるよう、モーターを停止状態にし、その状態を 128 チョッパー周期以上継続します。

### StealthChop2 でのフラグ

StealthChop2 は電圧モード駆動を用いるため、電流測定に基づくステータス・フラグの応答は遅くなり、モーター停止のようなバック EMF の突然の変化に対してドライブは遅れて反応します。

StealthChop2 での動作時にモーターのストールあるいは動作の突然の停止が生じると、過電流状態になる場合があります。直前のモーター速度、およびモーターのコイル抵抗に応じて、モーター電流は数 10ms の時間にわたり大きく増加します。低速時、バック EMF は電源電圧に比べてごくわずかにすぎないため、短絡検出がトリガされるおそれはありません。

ドライブ段を、モーターをサポートする最低電流レンジ (DRV\_CONF の current\_range) に切り替えてください。これは過電流閾値を 3 つのステップで自動的に適応させ、それによって突然モーターがストールした場合のピーク電流を低減します。

### 開放負荷フラグ

StealthChop2 モードでは、OLA および OLB のフラグのステータス情報が、サイクルごとに安定化を行う SpreadCycle モードとは異なります。

- OLA と OLB がセットされない場合、電流レギュレーションが両コイルで公称電流値に達していることを示します。
- OLA フラグおよび OLB フラグが常にセットされる場合は、モーター・コイルが不通となっています。
- OLA と OLB が時々セットされる場合は、モーター・コイルの抵抗値が約 5%以上変動しています。
- 一方または両方のフラグがアクティブになるのは、(モーターが取り付けられていないため、あるいは、PWM 制限値を超える高速度になったため) 電流レギュレーションが、最後の数フルステップ以内に最終目標電流に達しなかった場合です。

必要に応じて、SpreadCycle チョッパーを用いてオンデマンド開放負荷テストを行ってください。この方法が最も安全な結果をもたらします。StealthChop2 では、PWM\_SCALE\_SUM をチェックするとコイル抵抗が適切かどうか確認できます。

### モーターの状態を示す PWM\_SCALE\_SUM

自動スケーリングで PWM\_SCALE\_SUM を読み出すことで、モーターの状態に関する情報を得られます。このパラメータは、モーターに目標電流を供給するのに必要な実際の電圧を反映するため、モーター負荷、コイル抵抗、電源電圧、電流設定値など、いくつかの要素に依存します。そのため、PWM\_SCALE\_SUM の値を評価することで、モーターの動作点をチェックできます。制限値 (1023) に達すると、電源電圧が永久にあるいは一時的に低下するなどの理由で、電流レギュレータはモーターのフル電流を持続できません。

### フリーホイールリングおよび受動ブレーキング

StealthChop2 はモーター停止に対する様々なオプションを備えています。これらのオプションは、停止電流 IHOLD をゼロに設定し、FREEWHEEL 設定を用いて目的のオプションを選択することでイネーブルできます。目的のオプションがイネーブルされるのは、TPOWERDOWN および IHOLDDELAY で指定された時間が経過した後です。短時間で始動できるようにするため、モーターの目標電流がゼロになると電流レギュレーションはフリーズします。フリーホイールリング・オプションを用いると、フリーホイールリングと受動ブレーキングの両方を実行できます。受動ブレーキングは、渦電流による有効なモーター・ブレーキングで、アクティブな電流がコイルに供給されないため、消費エネルギーは最小限に抑えられます。ただし、連続的なトルクが加えられている場合、受動ブレーキングではモーターが低速回転を続ける場合があります。

### StealthChop2 を制御するパラメータ

以下の表に、StealthChop2 チョッパー・モードに関連したすべてのパラメータを示します。

表 11. StealthChop2 を制御するパラメータ

パラメータ	説明	設定	コメント
en_pwm_mode	StealthChop2 の使用を全般的にイネーブル（レジスタ GCONF）。 デフォルト= 0	0	StealthChop2 は無効。 SpreadCycle アクティブ。
		1	StealthChop2 イネーブル（速度 閾値に依存）。 停止時および IHOLD が公称 IRUN 電流時にのみイネーブル。
pwm_meas_sd_enable	低速減衰フェーズ時の電流測定の制御。 デフォルト= 0	0	オン・フェーズ時にのみ電流を 測定。電流下限値を適用。
		1	電流下限値を克服するために低 速減衰フェーズ時に電流を追加 測定。
pwm_dis_reg_stst	このオプションは、停止時のレギュレーション・ノイズを抑制し ます。 デフォルト= 0	0	電流レギュレーションを常時オ ン。
		1	モーターの停止時および電流低 下（IRUN 未満）時に電流レ ギュレーションをディスエーブル 。
TPWMTHRS	StealthChop2 での上限動作速度を指定します。目標の閾値速度 で動作している場合は、TSTEP の読出し値（2 つのマイクロス テップ間の時間）を入力します。 デフォルト= 0	0 ... 1048575	TSTEP が TPWMTHRS 未満に なると StealthChop2 はディス エーブルされます。
PWM_LIM	SpreadCycle から StealthChop2 への切り替え時に電流ジャーク を制限するための制限値。電流ジャークを低下させるにはこの値 を減じます。 デフォルト= 12	0 ... 15	8 ビットの振幅制限の上位 4 ビット
pwm_autoscale	電流測定を用いる自動電流スケーリングをイネーブル。オフの場 合、フォワード制御速度ベースのモードを使用します。 デフォルト= 1	0	フォワード制御モード
		1	電流レギュレータを使用する自 動スケーリング
pwm_autograd	PWM_GRAD_AUTO の自動調整をイネーブル。 デフォルト= 1	0	ディスエーブル。代わりにレジ スタの PWM_GRAD を使用。
		1	イネーブル
PWM_FREQ	PWM 周波数の選択。最小設定にすると良い結果が得られます。 チョッパー出力ごとに測定される周波数は、実効的なチョッパー 周波数 $f_{PWM}$ の半分です。 デフォルト= 0	0	$f_{PWM} = 2/1024 f_{CLK}$
		1	$f_{PWM} = 2/683 f_{CLK}$
		2	$f_{PWM} = 2/512 f_{CLK}$
		3	$f_{PWM} = 2/410 f_{CLK}$
PWM_REG	ユーザ定義による PWM 振幅レギュレーション・ループの P 係 数。pwm_autoscale = 1 の場合に、値が高いほど適応速度も高く なります。 デフォルト= 4	1 ... 15	フルステップあたりの PWM_SCALE_AUTO レギュ レータに対する結果は 0.5~7.5 ステップになります。

PWM_OFS	ユーザ定義の PWM 振幅 (オフセット)。PWM_OFFS_AUTO の自動調整に対して、速度ベースのスケールリングおよび初期化の値を与えます。  デフォルト= 0x1D	0 ... 255	PWM_OFS = 0 に設定すると、電流設定に基づく線形電流スケールリングがディセーブルされます。
PWM_GRAD	ユーザ定義の PWM 振幅 (勾配)。PWM_GRAD_AUTO の自動調整に対し、速度ベースのスケールリングおよび初期化の値を与えます。  デフォルト= 0	0 ... 255	
PWM_SCALE_SUM	実際の設定値によって決定される実際の PWM スケールリング。この値は、PWM_GRAD/OFS_AUTO 値用の 8 ビットよりも高い精度 (10 ビット) で示されます。  デフォルト= 0	0 ... 1023	
FREEWHEEL	モーター電流設定値がゼロ (I_HOLD = 0) の場合の停止オプション。StealthChop2 がイネーブルされている場合のみ使用できます。このフリーホイールリング・オプションにより、モーターの可動性が高まります。これに対して、両コイルの短絡オプションにより受動ブレーキングが実現できます。  デフォルト= 0	0	通常動作
		1	フリーホイールリング
		2	LS ドライバを用いたコイル短絡
		3	HS ドライバを用いたコイル短絡
PWM_SCALE_AUTO	電流レギュレータによって決定される、実際の StealthChop2 電圧 PWM スケールリング補正のリード・バック。調整時にはほぼ 0 に安定化します。  デフォルト= 0	-255 ... 255	(読出し専用) SpreadCycle での動作時、スケールリング値はフリーズされます。
PWM_GRAD_AUTO PWM_OFS_AUTO	自動調整のモニタリングと PWM_OFS および PWM_GRAD の初期値の決定が可能です。  デフォルト= 0	0 ... 255	(読出し専用)
TOFF	モーター・ドライバの全般的なイネーブル。実際の値は StealthChop2 には影響しません。  デフォルト= 0	0	ドライバ・オフ
		1 ... 15	ドライバ・イネーブル
TBL	コンパレータのブランキング時間。標準的なアプリケーションでは 1 または 2 の設定を選択します。負荷の容量がより大きい場合には 3 が必要となる場合もあります。設定値を下げると、StealthChop2 はより小さなコイル電流値に安定化できます。  デフォルト= 2	0	16 t <sub>CLK</sub>
		1	24 t <sub>CLK</sub>
		2	36 t <sub>CLK</sub>
		3	54 t <sub>CLK</sub>

## SpreadCycle および一般的なチョッパー

StealthChop2 は電圧モード PWM 制御のチョッパーであるのに対して、SpreadCycle はサイクルごとに電流制御を行います。そのため、モーター速度やモーター負荷の変化に対して、極めて高速に反応します。両方のモーター・コイルを流れる電流はチョッパーを用いて制御されます。チョッパーは互いに独立に動作します。次の図に、様々なチョッパー・フェーズを示します。

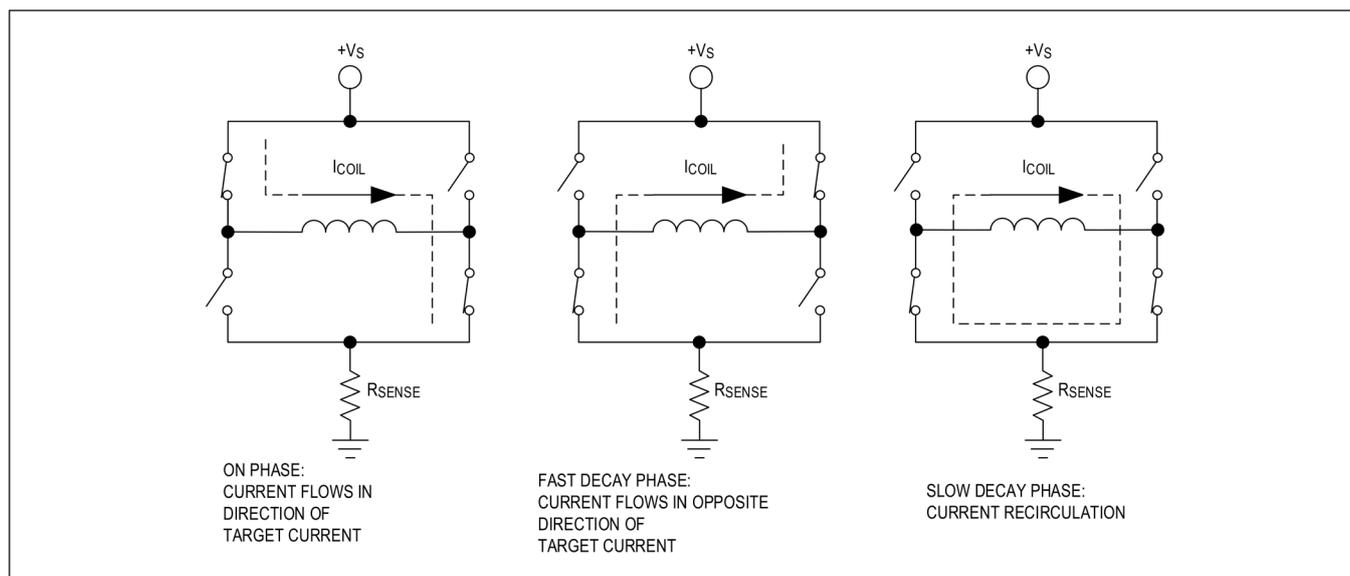


図 14. 代表的なチョッパー減衰フェーズ

電流はオン・フェーズと高速減衰フェーズのみを用いて安定化できますが、低速減衰フェーズを挿入することは、モーターの電氣的な損失や電流リップルを低減する上で重要です。低速減衰フェーズに要する時間は、制御パラメータで指定され、これによってチョッパー周波数の上限が定まります。電流コンパレータは、電流が厳密に 1 個のローサイド・トランジスタを流れるようなフェーズにあるときにはコイル電流を測定しますが、低速減衰フェーズ時には測定は行いません。低速減衰フェーズの終了はタイマーによって行われます。オン・フェーズの終了は、コイルを流れる電流が目標電流に達したときにコンパレータが行います。高速減衰フェーズの終了は、コンパレータまたは別のタイマーのどちらかによって行われます。

コイル電流が切り替わるとき、寄生容量の充放電により  $R_{DS(ON)}$  ベースの電流測定でスパイクが生じます。この期間（通常 1~2 マイクロ秒）、電流は測定できません。ブランキングとは、これらのスパイクをブロックするためにコンパレータへの入力がかき消されている期間のことです。

使用できるサイクルごとのチョッパー・モードは 2 つあります。SpreadCycle と呼ばれる新しい高性能チョッパー・アルゴリズムと、定評のある定オフ時間チョッパー・モードです。定オフ時間モードは、オン、高速減衰、低速減衰の 3 つのフェーズを順番に繰り返します。SpreadCycle モードは、オン、低速減衰、高速減衰、第二低速減衰の 4 つのフェーズを順番に繰り返します。

チョッパー周波数は、チョッパー・モーター・ドライバにとって重要なパラメータです。周波数が低すぎると、可聴ノイズを発生する可能性があります。周波数を上げると、モーターの電流リップルは減りますが、周波数が高すぎると磁気損失が生じる場合があります。また、ドライバの消費電力も、周波数の増加によって上昇します。動的消費電力の原因となるスイッチング勾配の影響が高まるためです。そのため、妥協点を見つける必要があります。ほとんどのモーターは 25kHz~40kHz の周波数範囲で最適に動作します。チョッパー周波数は、多くのパラメータ設定値とモーターのインダクタンスおよび電源電圧の影響を受けます。

**ヒント**：25kHz~40kHz の範囲のチョッパー周波数は、SpreadCycle を用いた場合にほとんどのモーターに対して良好な結果をもたらします。周波数がこれより高いとスイッチング損失が増大します。

表 12. SpreadCycle および一般的な定オフ時間チョッパーを制御するパラメータ

パラメータ	説明	設定	コメント
TOFF	低速減衰時間（オフ時間）を設定します。この設定により、最大チョッパー周波数も制限されます。	0	チョッパー・オフ
	StealthChop2 での動作の場合、このパラメータは使用しませんが、モーターをイネーブルすることが必要です。StealthChop2 で動作させる場合のみ、任意の設定が可能です。  このパラメータをゼロに設定すると、すべてのドライバ・トランジスタが完全にディスエーブルされ、モーターはフリーホイーリングが可能になります。  デフォルト=0	1...15	オフ時間の設定値 $N_{CLK} = 24 + 32 \times TOFF$ (1 に設定すると 24 クロックの最小ブランキング時間で動作します)

TBL	コンパレータのブランキング時間を選択します。この時間は、スイッチング・イベントとリングの時間を余裕を持ってカバーする必要があります。ほとんどのアプリケーションでは、1または2の設定で十分です。フィルタ・ネットワークを用いるなど、負荷の容量が大きい場合には、設定値を2または3にする必要があります。  デフォルト=2	0	16 t <sub>CLK</sub> <b>制限事項:</b> この設定を使用できるのは、8MHz以下の外部クロック発振器を組み合わせた場合のみです。
		1	24 t <sub>CLK</sub> <b>制限事項:</b> 内部クロックまたは13MHz以下の外部クロックを適用する場合に使用できます。
		2	36 t <sub>CLK</sub>
		3	54 t <sub>CLK</sub>
chm	チョッパー・モードの選択。  デフォルト=0	0	SpreadCycle
		1	一般的な定オフ時間

### SpreadCycle チョッパー

SpreadCycle (特許取得済み) チョッパー・アルゴリズムは、高速減衰フェーズの最適な長さを自動的に決定する、高精度で使いやすいチョッパー・モードです。SpreadCycle は、デフォルト設定でも、優れたマイクロステッピング品質を提供します。チョッパーをアプリケーションに応じて最適化するために、いくつかのパラメータが使用できます。

各チョッパー・サイクルは、オン・フェーズ、低速減衰フェーズ、高速減衰フェーズ、第二低速減衰フェーズで構成されます。2つの低速減衰フェーズおよびチョッパーあたり2つのブランキング時間によって、チョッパー周波数の上限が設定されます。低速減衰フェーズは、通常、チョッパー・サイクルの約30%~70%を停止状態にし、モーターやドライバを低消費電力化するのに重要です。

**例:** 低速減衰時間 TOFF の開始値の計算:

目標チョッパー周波数: 25kHz

$$\text{TOFF} = (1\text{kHz}/25\text{kHz}) \times (50/100) \times 1/2 = 10\mu\text{s}$$

**前提条件:** 2つの低速減衰サイクルが全チョッパー・サイクル時間の50%を使用。

TOFF の設定に関しては、次のようになります。TOFF = (TOFF × f<sub>CLK</sub> - 24)/32。

クロックが 12MHz の場合は、TOFF = 3.0 という結果が得られるため、TOFF は 3 に設定する必要があります。

クロックが 16MHz の場合は、TOFF = 4.25 という結果が得られるため、TOFF は 4 に設定する必要があります。

**ヒント:** モーター速度を最大にする場合は、TOFF を 1 または 2 に設定すると共に短めの TBL 設定を用いると、良い結果が得られることがあります。

ヒステリシスの開始設定値では、ドライバが最小限の量の電流リップルをモーター・コイルに導入します。最適なマイクロステッピングの結果を得るためには、この電流リップルが、モーターの抵抗性損失を原因とする電流リップルより大きいことが必要です。これによって、チョッパーは、立上がりおよび立下がりのどちらの目標電流に対しても正確に電流を安定化できます。モーター・コイルに電流リップルを導入するために必要な時間は、チョッパー周波数の減少ももたらします。そのため、ヒステリシスの設定値を高くすると、チョッパー周波数は低下します。モーターのインダクタンスは、チョッパーがモーター電流の変化に追従する能力を制限します。また、オン・フェーズの時間と高速減衰の時間は、ブランキング時間より長くなくてはなりません。これは、ブランキング時には電流コンパレータがディセーブルされるためです。

最適な設定値を得るための最も簡単な方法は、低ヒステリシス設定 (例えば、HSTRT = 0、HEND = 0) から始め、モーターが低速度設定で円滑に回転するようになるまで、HSTRT を増加させることです。これは、電流プローブを用いてモーター電流を測定すると、最もよく確認できます。ヒステリシスの設定値が小さすぎる場合、サイン波の波形をゼロ遷移付近でチェックすると、両半波の間に小さな棚形状が見られます。中程度の速度 (毎秒 100 フルステップ~400 フルステップ) では、ヒステリシスの設定値が低すぎるとモーターのハミングや振動が増加する原因となります。ヒステリシスの設定値が高すぎる場合は、チョッパー周波数が低下し、チョッパー・ノイズが増加しますが、波形には何ら好影響を及ぼしません。

いくつかの実験を行うとわかるように、設定はモーターにはまったく依存しません。大電流のモーターは、通常、コイル抵抗も低いからです。そのため、ヒステリシスのデフォルト値として低い値から中程度の値を選択すると (例えば、実効ヒステリシス=4)、通常、ほとんどのアプリケーションに適します。この設定は、モーターを用いて実際に試すことで最適化できます。設定が低すぎるとマイクロステップの精度が下がり、設定が高すぎるとチョッパー・ノイズとモーターの消費電力が増加します。高速減衰時間をブランキング時間よりわずかに長くすると、設定は最適なものとなります。これを実現するのが難しい場合は、オフ時間の設定値を減らしてください。

電源電圧に比較してコイル抵抗が高い場合など、場合によっては、このヒステリシスの原理はチョッパ周波数が低くなりすぎる原因となります。これは、ヒステリシスの設定を開始設定 (HSTRT + HEND) と終了設定 (HEND) に分割することで回避できます。自動ヒステリシス・デクリメンタ (HDEC) は、ヒステリシス値を段階的に 16 システム・クロックごとにデクリメントすることで、両方の設定の間の補間を行います。ヒステリシスは、各チョッパ・サイクルの開始時に、開始値と終了値の和の値 (HSTRT + HEND) で始まり、サイクルの間デクリメントを行い、チョッパ・サイクルの終了するまで、あるいはヒステリシスの最終値 (HEND) に達するまで続きます。このようにして、チョッパ周波数は、周波数が低すぎるようになっても、大振幅かつ低電源電圧の状態で安定化します。これにより、周波数が可聴範囲に達するのが防止されます。

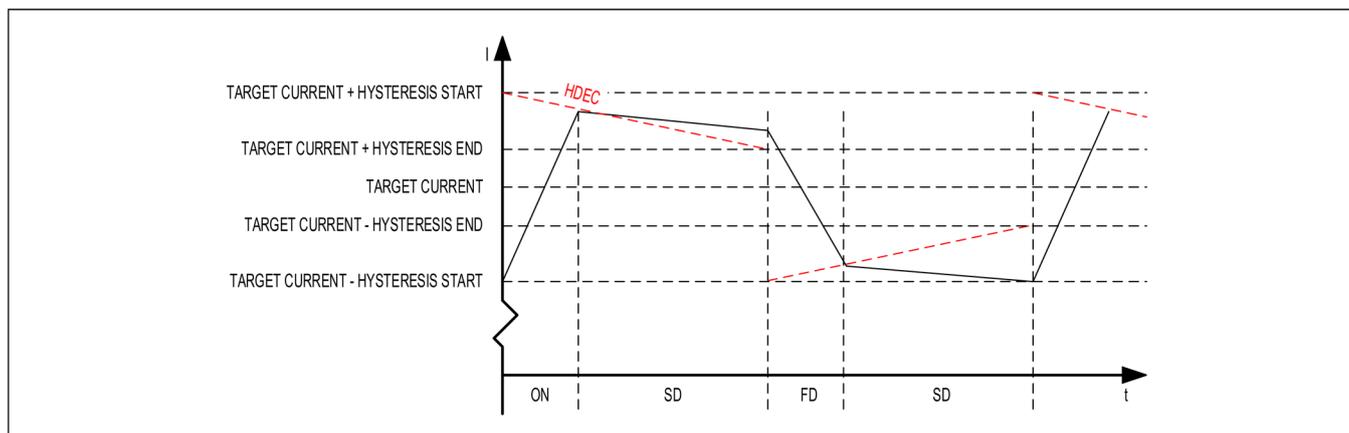


図 15.1 チョッパ・サイクルの間のコイル電流を示す、SpreadCycle チョッパの概略図

表 13. SpreadCycle モードのパラメータ

パラメータ	説明	設定	コメント
HSTRT	ヒステリシス開始設定値。この値は、ヒステリシス終了値 HEND からのオフセットです。 デフォルト=5	0...7	HSTRT = 1...8 この値が HEND に加わります。
HEND	ヒステリシス終了設定値。いくつかのデクリメントを行った後のヒステリシス終了値を設定します。その和 (HSTRT + HEND) は 16 以下にする必要があります。最大値 30 の電流設定の場合 (振幅は 240 に低減)、この和には制限がありません。 デフォルト=2	0...2	-3...-1 : HEND は負
		3	0 : HEND はゼロ
		4...15	1...12 : HEND は正

HSTRT = 0 かつ HEND = 0 の場合でも、TMC5241 はアナログ回路を使用して最小のヒステリシスを設定します。

#### 例：

ヒステリシスとして 4 を選択します。ヒステリシスのデクリメントを使用しないように決めます。この場合、次のように設定します。

HEND = 6 (実効終了値を  $6 - 3 = 3$  に設定)

HSTRT = 0 (最小ヒステリシスを設定。例えば 1 に設定すると、 $3 + 1 = 4$ )

可変ヒステリシスを活用するには、値のほとんどを HSTRT (例えば 4) に設定し、残りの 1 をヒステリシス終了に設定します。その結果、設定レジスタ値は次のようになります。

HEND = 0 (実効終了値を -3 に設定)

HSTRT = 6 (ヒステリシス終了に対する実効開始値を +7 に設定 :  $7 - 3 = 4$ )

## 一般的な定オフ時間チョッパー

一般的な定オフ時間チョッパーを、SpreadCycle の代替として用いることができます。定オフ時間チョッパーでは、各オン・フェーズの後に固定時間の高速減衰を使用します。オン・フェーズの時間はチョッパー・コンパレータによって決定される一方、高速減衰時間はドライバがサイン波の立下がり勾配に追従できるだけの長さであることが必要です。しかし、これは過大なモーター電流リップルおよび消費電力の原因となるため、長すぎはけません。これを調整するには、オシロスコープを用いるか、様々な速度でモーターの平滑さを評価します。良い開始値は、高速減衰時間を低速減衰時間と同じ設定にすることです。

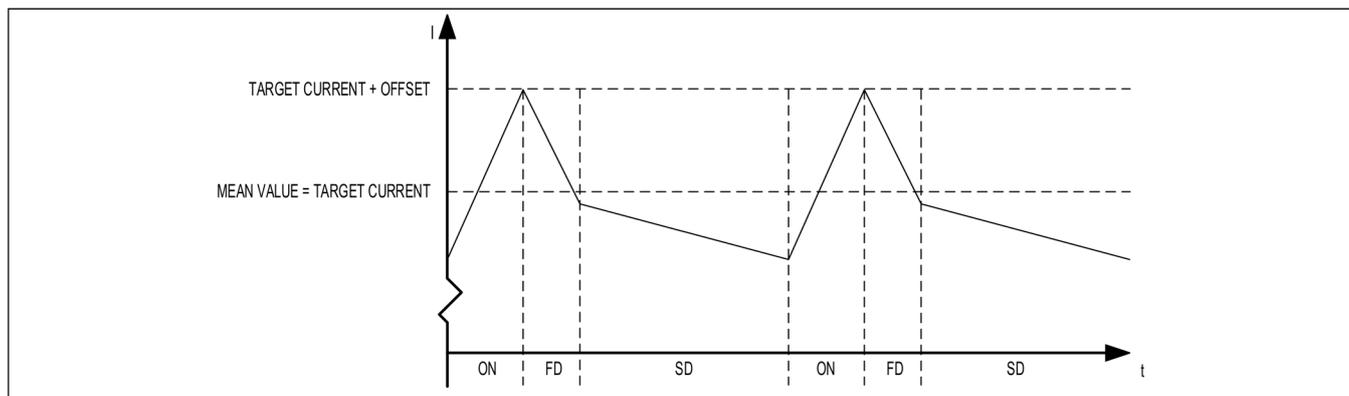


図 16. オフセットを持つ一般的な定オフ時間チョッパーが示すコイル電流

高速減衰時間を調整した後、ゼロ交差が滑らかになるようオフセットを調整する必要があります。これが必要なのは、高速減衰フェーズでは、モーター電流の絶対値が目標電流より小さくなるためです（下図参照）。ゼロ・オフセットが低すぎると、モーターは電流のゼロ交差時に短時間、停止します。ゼロ・オフセットの設定が高すぎると、マイクロステップが増大します。通常、最も滑らかな動作を実現するには、正のオフセット設定が必要です。

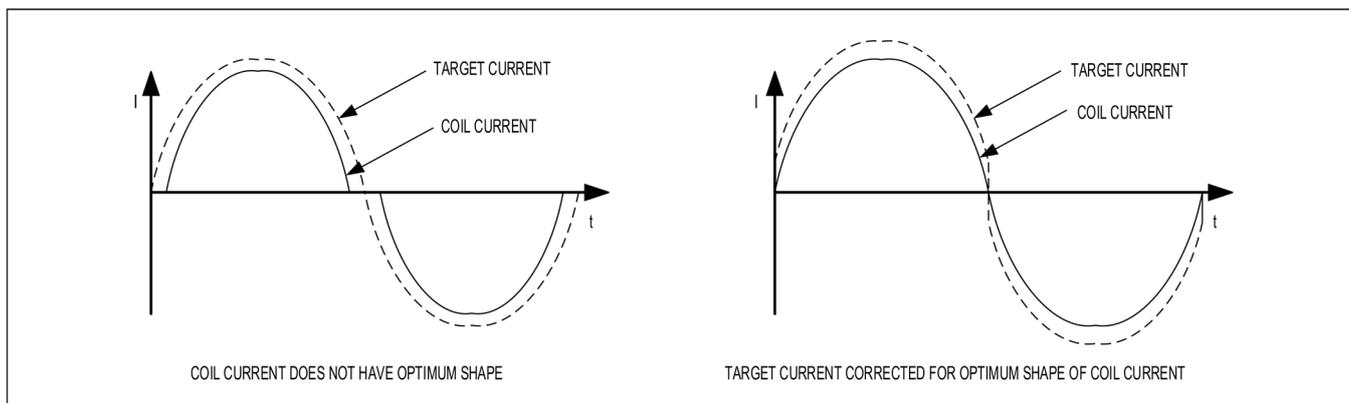


図 17. 一般的なチョッパーのゼロ交差とサイン波オフセットを使用した補正

表 14. 定オフ時間チョッパー・モードを制御するパラメータ

パラメータ	説明	設定	コメント
TFD (fd3 and HSTRT)	高速減衰時間設定。CHM = 1 の場合、これらのビットが各チョッパー・サイクルの高速減衰の部分を制御します。 デフォルト = 5	0	低速減衰のみ
		1...15	高速減衰フェーズの時間
OFFSET (HEND)	サイン波オフセット。CHM = 1 の場合、これらのビットがサイン波のオフセットを制御します。オフセットを正にすると、ゼロ交差エラーを補正します。 デフォルト = 2	0...2	負のオフセット : -3...-1
		3	オフセットなし : 0
		4...15	正のオフセット : 1...12
disfdcc	高速減衰サイクルを終了させるための電流コンパレータの使用を選択します。電流コンパレータがイネーブルされていると、電流が実際の正の値より大きな負の値に達した場合に高速減衰サイクルが終了します。 デフォルト = 0	0	高速減衰サイクルのコンパレータによる終了をイネーブル
		1	時間でのみ終了



IRUNDELAY	開始が検出されてからモーターが起動するまでのクロック・サイクル数を制御します。	0	IRUN への即時パワー・アップ
	動作の開始時に、ホールド電流 (IHOLD) から動作電流 (IRUN) への滑らかな電流インクリメントを可能にします。フル・モーター・トルクを確立するには、短時間でのパワー・アップが重要ですが、わずかな遅延時間を設けることで、可聴ノイズを低減し、また、電源電流の不連続を防止できます。  デフォルト= 4	1...15	電流インクリメント・ステップあたりの遅延は IRUNDELAY × 512 クロックの倍数

## フルスケール電流レンジの設定

フルスケール電流  $I_{FS}$  は、ピーク電流の設定値です。

フルスケール電流は、外部リファレンス抵抗と DRV\_CONF レジスタの 2 ビットを用いて選択できます。

精度 1% の標準的な低電力抵抗で十分です。

様々なモーター・サイズやアプリケーションに適応できるように、3 種類のフルスケール電流レンジが設定可能です。

これは、できる限り最高の電流制御分解能を活用するために必要です。

そのため、I<sub>REF</sub> と GND の間に抵抗を接続して、フルスケール・チョッピング電流  $I_{FS}$  を設定します。

DRV\_CONF レジスタのビット 1 および 0 は、ドライバ段の標準的なオン抵抗を定義し、更に、外付け抵抗に基づきフルスケール・レンジを制御します。

下記の式は、フルスケール電流  $I_{FS}$  を、I<sub>REF</sub> ピンに接続された R<sub>REF</sub> 抵抗と DRV\_CONF レジスタのビット設定の関数として表したものです。

比例定数  $K_{IFS}$  は、選択したフルスケール・レンジ設定 (DRV\_CONF レジスタのビット 1..0) に依存します。外付け抵抗 R<sub>REF</sub> の範囲は 12kΩ ~ 60kΩ です。

$$I_{FS} (RMS) = \frac{K_{IFS}(CURRENT_{RANGE})}{R_{REF}[k\Omega]} \times \frac{GLOBALSCALER}{256} \times \frac{CS + 1}{32} \times \frac{248}{256} / \sqrt{2}$$

この式からコイルあたりのモーター電流の実効値が得られます。

- CS はそれぞれ、IRUN または IHOLD です。IRUN は CoolStep によってスケール・ダウンします。
- 248/256 はデフォルトのマイクロステップ・テーブルの振幅です (StealthChop のみの場合は 255 まで使用できます)。
- GLOBALSCALER の範囲は 1~256 です (256 は GLOBALSCALER = 0 に対応します)。
- 1/SQRT(2) は、サイン波の実効値を計算するための係数です。

表 16.  $I_{FS}$  フルスケール・ピーク・レンジの設定値 (R<sub>REF</sub> = 12kΩ の場合の例)

レジスタ設定 DRV_CONF ビット 1..0	$K_{IFS}$ (A × kΩ)	最大 FS 設定 (ピーク)	R <sub>DS(ON)</sub> 代表値 (HS + LS)	注記
11	36	3A	0.31Ω	効率を最適化して動作範囲を 3A (FS) まで拡大。
10	36	3A	0.31Ω	効率を最適化して動作範囲を 3A (FS) まで拡大。
01	24	2A	0.37Ω	最大動作範囲を 2AFS までに制限。 低電流時に高い精度が必要な場合。
00 (default)	11.75	1A	0.53Ω	最大動作範囲を 1AFS までに制限。 低電流時に高い精度が必要な場合。

次の表に、( $I_{REF}$  ピンでの) 様々なリファレンス抵抗値とフルスケール電流用の様々なピン設定の関係を示します。各セルにはそれぞれの場合の最大実効値電流を示しています。

表 17. DRV\_CONF のビット 1 および 0 の設定と様々な  $R_{REF}$  に基づく、フルスケール実効値電流  $I_{FS}$  (単位: アンペア ( $A_{RMS}$ ))

$R_{REF}$ (k $\Omega$ )	MAX. FULL SCALE CURRENT ( $A_{RMS}$ ) BASED ON CURRENT_SCALER (DRV_CONF BITS 1...0) SETTING AND $K_{IFS}$ (A $\times$ k $\Omega$ )			
	DRV_CONF BITS 1...0 = 11	DRV_CONF BITS 1...0 = 10	DRV_CONF BITS 1...0 = 01	DRV_CONF BITS 1...0 = 00
	$K_{IFS} = 36$	$K_{IFS} = 36$	$K_{IFS} = 24$	$K_{IFS} = 11.75$
12	2,05	2,05	1,37	0,67
15	1,65	1,65	1,09	0,53
18	1,37	1,37	0,91	0,45
22	1,12	1,12	0,75	0,37
27	0,91	0,91	0,61	0,30
33	0,75	0,75	0,49	0,24
39	0,63	0,63	0,43	0,20
47	0,52	0,52	0,35	0,17
56	0,44	0,44	0,29	0,15

**ヒント:** 電流値は係数 248/256 を使用したデフォルトのマイクロステップ・テーブルに対して計算されています。カスタムで作成したテーブルを使用すると、わずかに高い電流値も可能です。

## 速度ベースのモード制御

TMC5241 では、最適なモーター制御を実現するために、様々なチョッパー・モードと動作モードが設定可能です。モーター負荷に応じて、最小ノイズおよび高精度、最高動的性能、あるいは最高速時の最大トルクを実現できるよう、様々なモードを最適化できます。限定された速度範囲では、CoolStep や StallGuard2 などの一部の機能も有用です。いくつかの速度閾値では、広い速度範囲を必要とするアプリケーション内で異なる動作モードを組み合わせることができます。

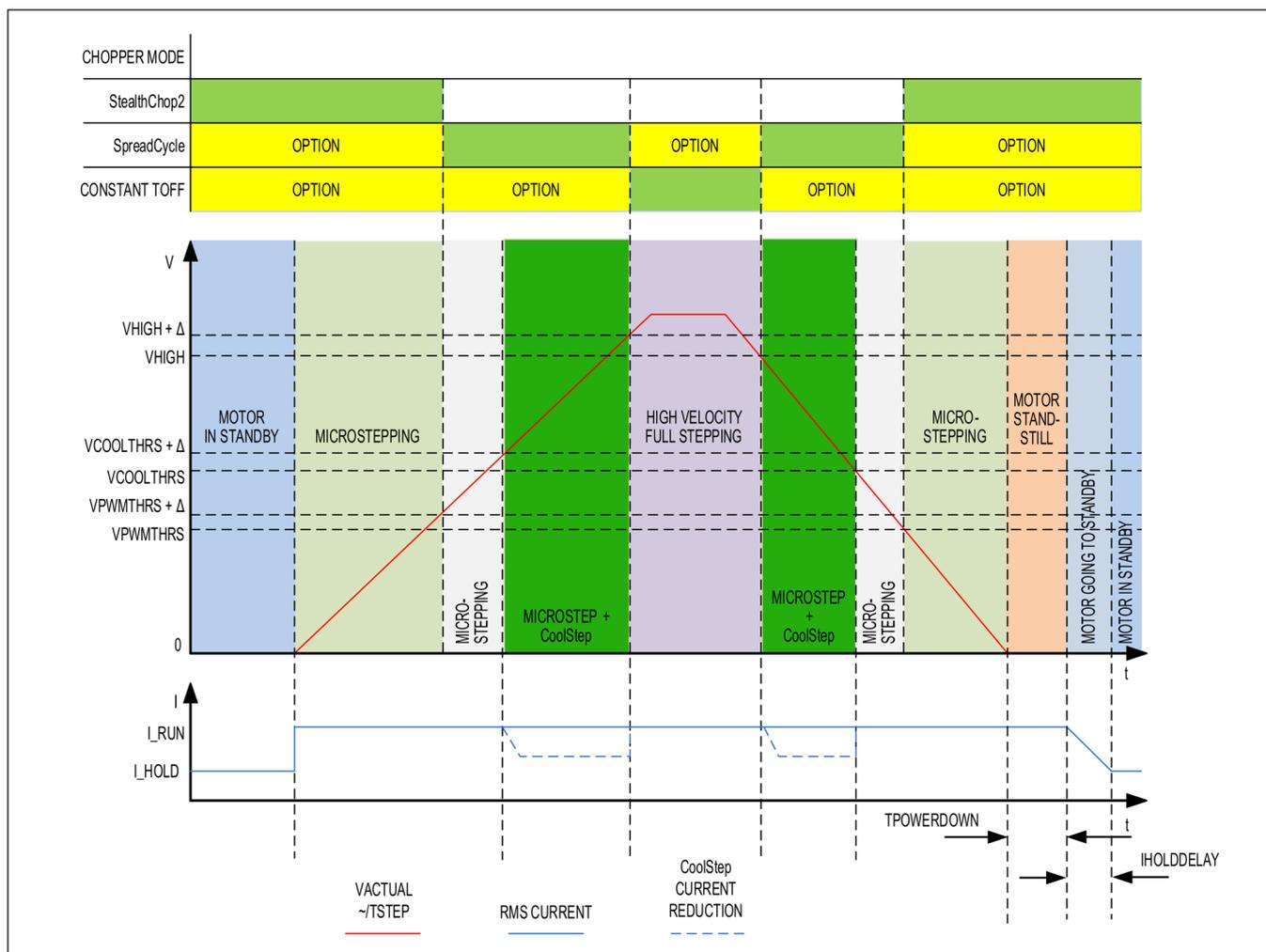


図 18. 速度依存モードの選択

図 18 に、使用可能なすべての閾値と必要な順序を示します。これらの値は、VPWMTHRS、VHIGH、およびVCOOLTHRS の設定値によって決まります。速度は、各 2 ステップ・パルス間の時間間隔 TSTEP で記述されます。それにより、外部ステップ源を用いた場合でも速度を決定できます。TSTEP は、常に 256 マイクロステップに正規化されます。したがって、マイクロステップの分解能が変化した場合でも、閾値をそれに合わせて変化させる必要はありません。閾値は、マイクロステップの設定値とは無関係に、同じモーター速度を表します。TSTEP は、これらの閾値と比較されます。TSTEP の測定でジッタが生じた場合に、比較した結果が連続的に切り替わるのを避けるため、1/16TSTEP または 1/32TSTEP のヒステリシスが適用されます。上限のスイッチング速度は、閾値として設定した 1/16 または 1/32 の値より高くなります (GCONF レジスタの設定ビット `small_hysteresis` で選択できます)。モーター電流は、停止フラグ `stst` に応じて、動作レベルとホールド・レベルにプログラムできます。

自動速度閾値を用いると、アプリケーションを様々な速度レンジに調整できます。CoolStep などの機能は、セットアップに完全かつトランスペアレントに統合できます。したがって、一度パラメータを設定すれば、ソフトウェアで有効化したり無効化したりする必要はありません。

表 18. 速度ベースのモード制御パラメータ

パラメータ	説明	設定	コメント
stst	各動作モードでモーターが停止していることを示します。時間は最後のステップ・パルス後 $2^{20}$ クロックです。  デフォルト/リセット : 0	0/1	ステータス・ビット、読み専用
TPOWERDOWN	これは、モーターの停止 (stst) 後からモーター電流がパワー・ダウンスするまでの遅延時間です。時間範囲は、約 0~4 秒 ( $f_{CLK} = 16\text{MHz}$ の場合) です。0 に設定すると遅延なし、1 に設定すると 1 クロック・サイクルの遅延です。それ以上は、 $2^{18}$ クロック・サイクルの離散的なステップでインクリメントします。  デフォルト : 0xA	0...255	$2^{18} \times t_{CLK}$ の倍数で表した時間
TSTEP	ステップ入力周波数から引き出される 2 つの $1/256$ マイクロステップ間の実際の測定時間 ( $1/f_{CLK}$ 単位)。オーバーフローまたは停止時の測定値は、 $(2^{20}) - 1$ です。  デフォルト/リセット : 0	0...1048575	ステータス・レジスタ、読み専用。 $t_{CLK}$ の倍数で表した実際の測定ステップ時間。
TPWMTHRS	$TSTEP \geq TPWMTHRS$ <ul style="list-style-type: none"> <li>StealthChop2 PWM モードが有効化されず (設定されている場合)。</li> <li>DcStep は無効化されます。</li> </ul> デフォルト : 0	0...1048575	StealthChop2 の動作上限速度閾値を制御するための設定値。
TCOOLTHRS	$TCOOLTHRS \geq TSTEP \geq THIGH$ <ul style="list-style-type: none"> <li>StallGuard2 と CoolStep が有効化されます (設定されている場合)。</li> <li>StealthChop2 電圧 PWM モードは無効化されます。</li> </ul> $TCOOLTHRS \geq TSTEP$ <ul style="list-style-type: none"> <li>StallGuard2 のストール出力信号が外部コントローラで使用可能になります (設定されている場合)。</li> </ul> デフォルト : 0	0...1048575	CoolStep と StallGuard2 の動作下限速度閾値を制御するための設定値。
THIGH	$TSTEP \leq THIGH$ <ul style="list-style-type: none"> <li>CoolStep は無効化されます (モーターは通常電流スケールで動作)。</li> <li>StealthChop2 電圧 PWM モードは無効化されます。</li> <li>vhighchm が設定されていると、チョッパーは <math>TFD = 0</math> (低速減衰のみの定オフ時間) の場合に <math>chm = 1</math> に切り替わりません。</li> <li>チョッパー同期はオフに切り替わりません (<math>SYNC = 0</math>)。</li> <li>vhighfs が設定されていると、モーターはフルステップ・モードで動作し、ストール検出は、DcStep ストール検出に切り替わりません。</li> </ul> デフォルト : 0	0...1048575	CoolStep と StallGuard2、およびオプションの高速ステップ・モードを用いた動作の上限閾値を制御するための設定値。
small_hysteresis	TSTEP (下限速度閾値) と $(TSTEP \times 15/16) - 1$ または $(TSTEP \times 31/32) - 1$ (上限速度閾値) に基づくステップ周波数比較のヒステリシス。  デフォルト : 0	0	ヒステリシスは 1/16
		1	ヒステリシスは 1/32
vhighfs	このビットは、VHIGH を超えた場合にフルステップへの切り替わりをイネーブルします。切り替わりは $45^\circ$ の位置でのみ生じません。フルステップの目標電流は、 $45^\circ$ 位置でのマイクロステップ・テーブルからの電流値を用います。  デフォルト : 0	0	フルステップへの切替えなし
		1	高速時はフルステップ

vhighchm	このビットは、VHIGH を超えた場合に chm = 1 および fd = 0 への切り替わりをイネーブルします。したがって、より高い速度が達成できます。vhighfs = 1 と組み合わせることができます。このビットがセットされた場合、チョッパー周波数が 2 倍になるのを防止するため、TOFF 設定は高速動作時に自動的に 2 倍になります。  デフォルト : 0	0	チョッパー・モードの変化なし
		1	高速時は一般的な定 TOFF チョッパー
en_pwm_mode	StealthChop2 電圧 PWM イネーブル・フラグ (速度閾値に依存)。停止時にのみオフ状態からオン状態に切り替わります。  デフォルト : 0	0	StealthChop2 不使用
		1	StealthChop2 が設定されていて TSTEP > TPWMTHRS の場合に StealthChop2 を有効化

## ランプ・ジェネレータ

ランプ・ジェネレータは、目標ポジションまたは目標速度に基づいた動作を可能にします。ランプ・ジェネレータは、加速度設定と速度設定を考慮して、動作プロファイルを自動で計算します。TMC5241 は、新しいタイプのランプ・ジェネレータを組み込んでおり、一般的な線形加速度ランプに比べ、より高速の機械的動作が可能になります。8 点ランプ・ジェネレータを用いると、加速度のランプをステッピング・モーターのトルク曲線に合わせることができます。これは、加速フェーズと減速フェーズに対してそれぞれ 3 種の加速度設定値を使用することで、ジャークを最小限に抑えたランプを可能にします。

## 現実の単位への変換

TMC5241 は、すべての内部動作に対して、時間基準として内部または外部のクロック信号を使用します。したがって、常に、速度および加速度の設定値は  $f_{CLK}$  を基準にします。安定性と再現性を最高のものにするため、外部水晶発振器をタイム・ベースとして用いるか、マイクロコントローラからクロック信号を供給することを推奨します。

$v$ [TMC5241]および  $a$ [TMC5241]は、TMC5241 の内部単位です。これらは、TMC5241 の速度/加速度レジスタに書き込む値です。

計算ツールは製品のウェブサイトおよび評価用ツールから入手できます。

表 19. ランプ・ジェネレータのパラメータと単位の関係

パラメータ/記号	単位	説明
fCLK	[Hz]	TMC5241 のクロック周波数
s	[s]	秒
US	Microstep	
FS	Fullstep	
USC - microstep count	-	マイクロステップ数で表したマイクロステップ分解能 (2つのフルステップ間のマイクロステップ数 (通常は 256))。
FSC - fullstep count	-	1 回転あたりのモーターのフルステップ、例えば 200
$\mu$ step velocity $v$ [Hz]	Microsteps/s	$v[\text{Hz}] = v[\text{TMC5241}] \times (f_{\text{CLK}}[\text{Hz}]/2^{24})$
$\mu$ step acceleration $a$ [Hz/s]	Microsteps/s <sup>2</sup>	$a[\text{Hz/s}] = a[\text{TMC5241}] \times f_{\text{CLK}}[\text{Hz}]/2^{42}$
Rotations per second $v$ [rps]	Rotations/s	$v[\text{rps}] = v[\text{microsteps/s}]/\text{USC}/\text{FSC}$
RPS acceleration $a$ [rps/s <sup>2</sup> ]	Rotations/s <sup>2</sup>	$a[\text{rps/s}^2] = a[\text{microsteps/s}^2]/\text{USC}/\text{FSC}$
Ramp steps[microsteps] = rs	Microsteps	$rs = (v[\text{TMC5241}])^2/a[\text{TMC5241}]/2^8$ 線形加速ランプ時のマイクロステップ (0 から V への加速を仮定)。
TSTEP, Txxx_THRS	-	$\text{TSTEP} = f_{\text{CLK}}/f_{256\text{STEP}} = f_{\text{CLK}}/(f_{\text{STEP}} \times 256/\text{USC}) = 2^{24}/(\text{VACTUAL} \times 256/\text{USC})$ 速度閾値用の時間基準は、ステップ入力ごとの速度 $v$ [Hz]の実際の $1/256$ マイクロステップ周波数 ( $f_{256\text{STEP}}$ ) を基準にしています。
Ramp generator update rate	[Hz]	$f_{\text{UPDATE}} = f_{\text{CLK}}/512$ VACTUAL はこの周波数に応じて更新されます。

まれに、加速度の上限値がアプリケーションに制限を課すことがあります。クロック周波数を下げて動作させる場合や、モーターのギア比が高く負荷が小さい場合などです。実現可能な実効加速度を増加させるため、シーケンサ入力マイクロステップ分解能は減少できません。CHOPCONF オプションの `intpol` を 1 に、`MRES` を %0001 に設定すると、同じ速度設定に対してモーター速度を 2 倍にできます。そのため、実効的な加速度および減速度も 2 倍になります。この設定ではモーターの滑らかさは同じですが、位置分解能は 1/2 になります。

## 動作プロファイル

### ランプ・モード

ランプ・ジェネレータは、3 フェーズの加速ランプと 3 フェーズの減速ランプに加えて、プログラマブルな始動速度および停止速度が可能です。

加速と減速の 3 種類のセットは自由に組み合わせられます。遷移速度 `V1` と `V2` を用いることで、3 つの加速および減速設定値の間での速度に依存した切り替えが可能です。標準的な高速アプリケーションでは、速度が高くなるほどモーターのトルクが減少するため、加速度値と減速度値は小さくなります。システム内の摩擦を考慮すると、明らかにシステムの減速能力のほうが加速能力より高くなります。そのため、多くのアプリケーションで、減速度値は加速度に比べ高い値に設定できます。時間に制約があるアプリケーションにおいては、この方法によりモーターの動作速度を最大化できます。

目標の位置およびランプ・パラメータは、動作中いつでも変化する可能性があるため、モーション・コントローラは、常に目標に達するための最善の（最も速い）方法を使用しますが、ユーザが設定した加速度の制約は守ります。したがって、動作が自動的に停止し、ゼロを交差し、再度動作することも生じます。例えば、最終的な減速フェーズ時に目標位置が再度変更になり、より近い位置に「プルイン」された場合、設定された減速度値では新しい目標位置に直接達することができません。この場合は、特別なフラグ `second_move` によって通知されます。

また、ランプ・ジェネレータは、加速フェーズから減速フェーズへの遷移、および減速フェーズから加速フェーズへの遷移を滑らかなものにするので、自動ジャーク除去機能にも対応しています。これは、機械的なジャーク応答でも必要とされるように、最短時間の定速セグメント (`TVMAX`) を強制的に入れることで行います。次のグラフに、代表的な（コーナー）ケースにおけるいくつかの例を示します。

**注：** 始動速度は、不要な場合ゼロに設定できます。

停止速度は、不要な場合、低い値（1000~10）に設定できます。

**背景：** `TSTOP=0` とすると、設定されたマイクロステップ目標位置に正確に到達できない可能性があります。

`VSTOP` は常に `VSTART` 以上に設定されるよう、注意してください。これにより、短い動作でも目標ポジションで正しく終了できます。

ジャーク除去を無効化するには `TVMAX` をゼロに設定します。

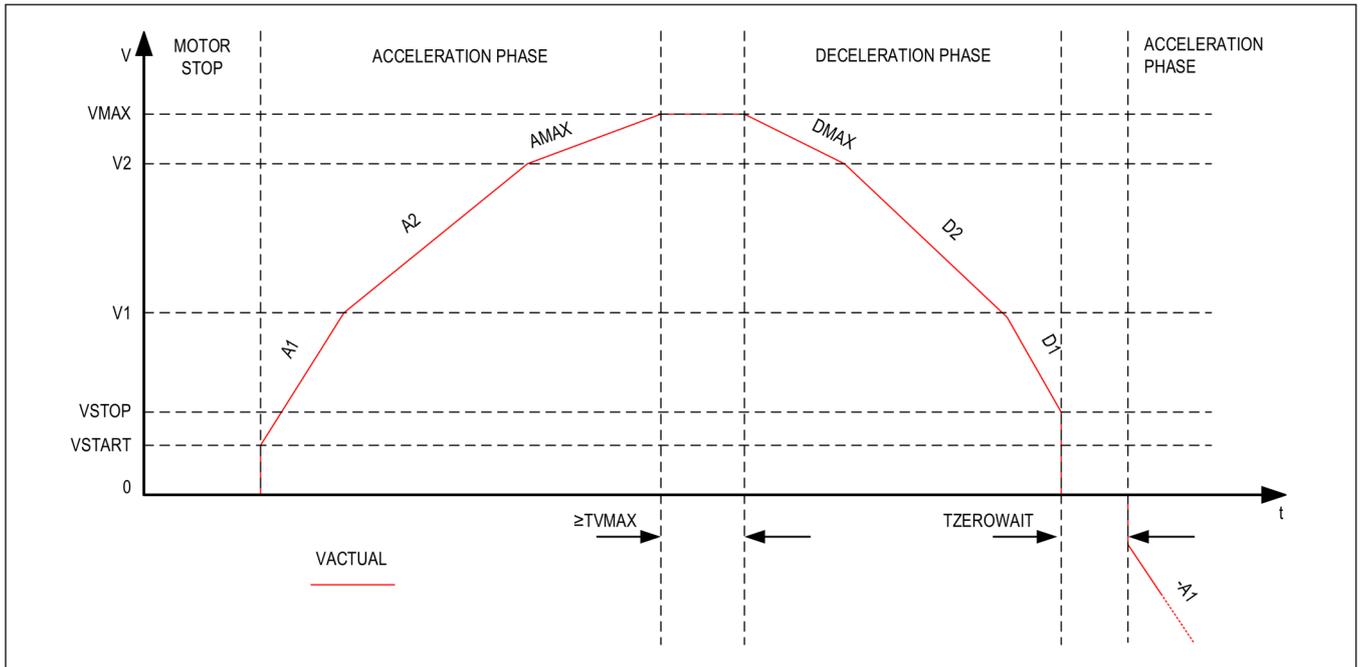


図 19. 2 回目の動作が負の方向に向かうランプ・ジェネレータ速度パターン

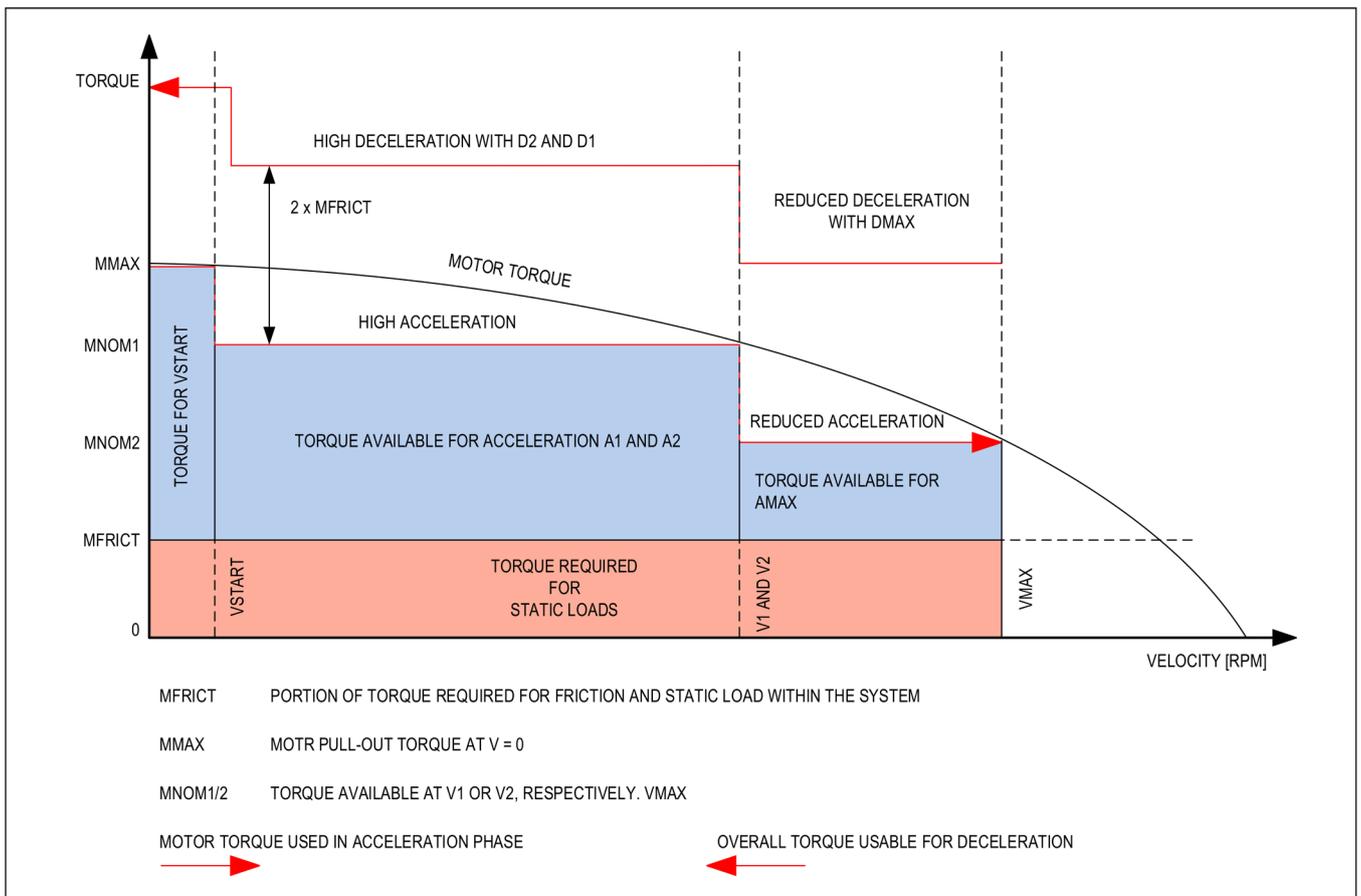


図 20. ランプ・ジェネレータを用いた最適なモーター・トルク使用法を示す図

### 8 点ランプの始動速度および停止速度

レベルの増加した始動速度と停止速度を用いると、その後の反対方向への動作で、VSTART 単独ではなく、VSTART + VSTOP に等しいジャークが生じることが明らかになります。モーターはこれに追従できない可能性が高いため、TZEROWAIT を設定して後続の動作に遅延時間を設定します。有効な遅延時間は、フラグ `t_zerowait_active` で示されます。目標位置に達すると、フラグ `position_reached` がアクティブになります。

3つの加速セグメントおよび減速セグメントからなるセットは2通りの方法で用いることができます。1つは、低速時ほど高い加速度値を用いることによりモーター・トルク曲線に適合させることで、もう1つは、1つの加速セグメントから次のセグメントに移移するときのジャーク（加速度の変化）を低減することです。ジャークを最適化したランプを実現するには、通常、A1、D1、AMAX、DMAXをA2およびD2よりも低い値に設定します。ジャークに関して最も重要な点は、オンザフライの目標位置変更があった場合に、定速セグメントなしに加速から減速に移移すること、および、減速から加速に移移することです。

この両方に対応するため、8点動作プロファイル・ジェネレータは、最小のセグメント時間（TVMAX）に基づく定速セグメントを強制できます。距離が不十分なためこの時間が維持できない場合は、減少したVMAX（VMAX'）が計算され、これが定速セグメントに用いられます。最小VMAXはVSTOPと同じです。

次のいくつかのパターンは、擬似S字設定を実現するための様々な位置間距離に基づいた速度プロファイルを示したものです。

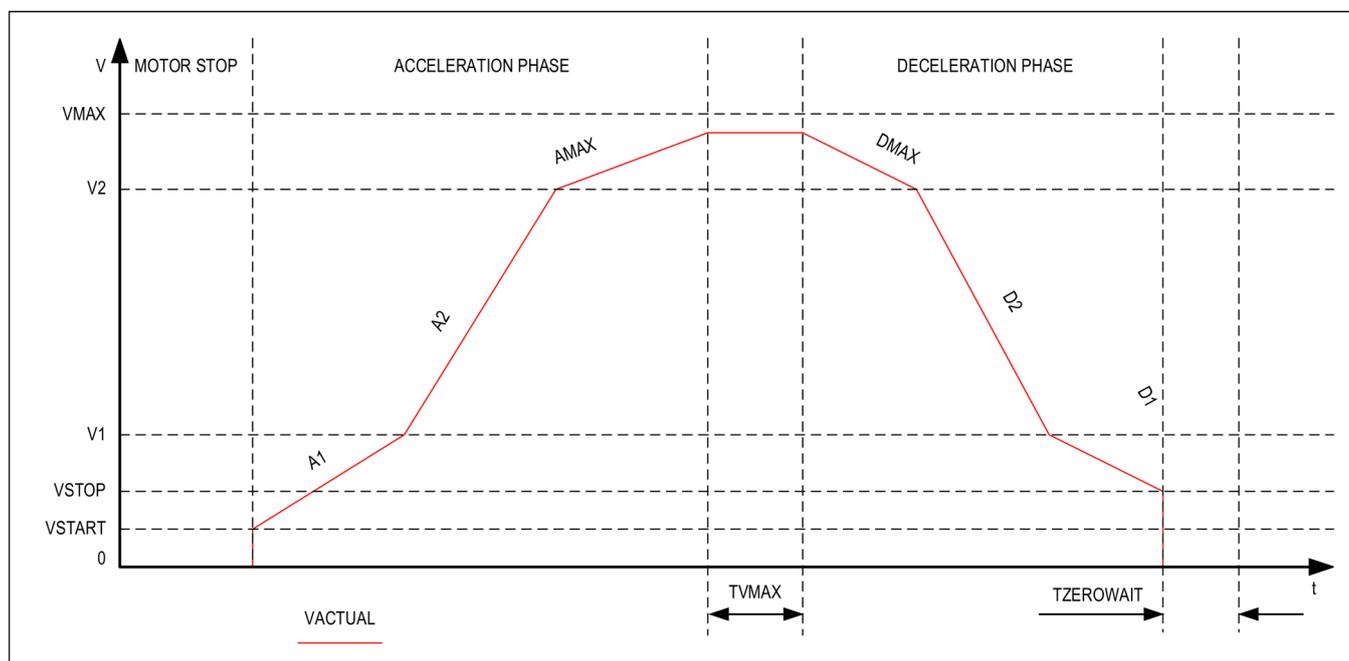


図 21. 距離が短いため VMAX に到達できない 8 点ランプ

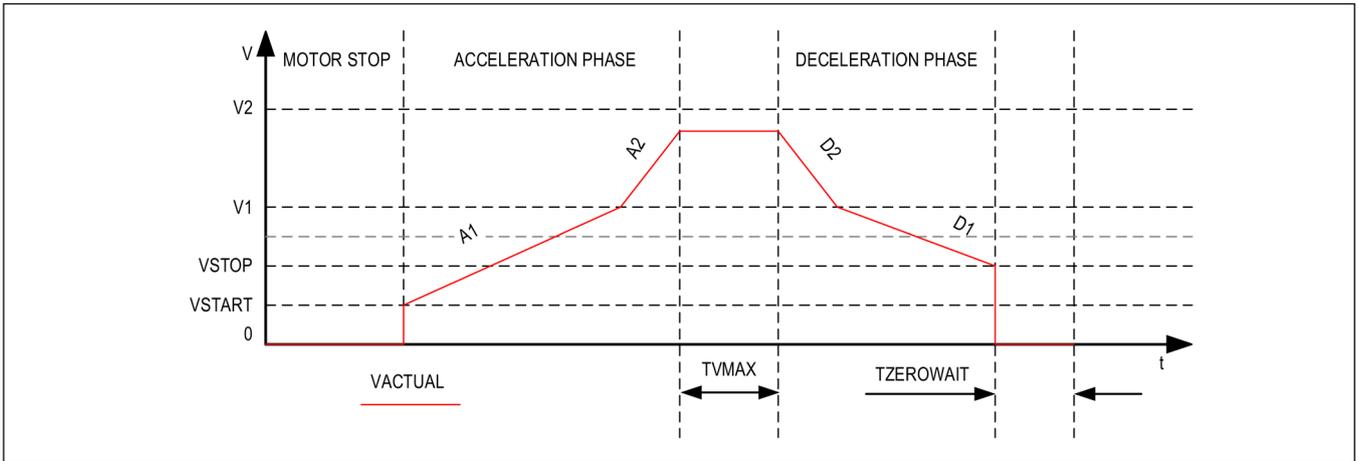


図 22. 距離が短いために V2 に達せず AMAX フェーズと DMAX フェーズがないプロファイル

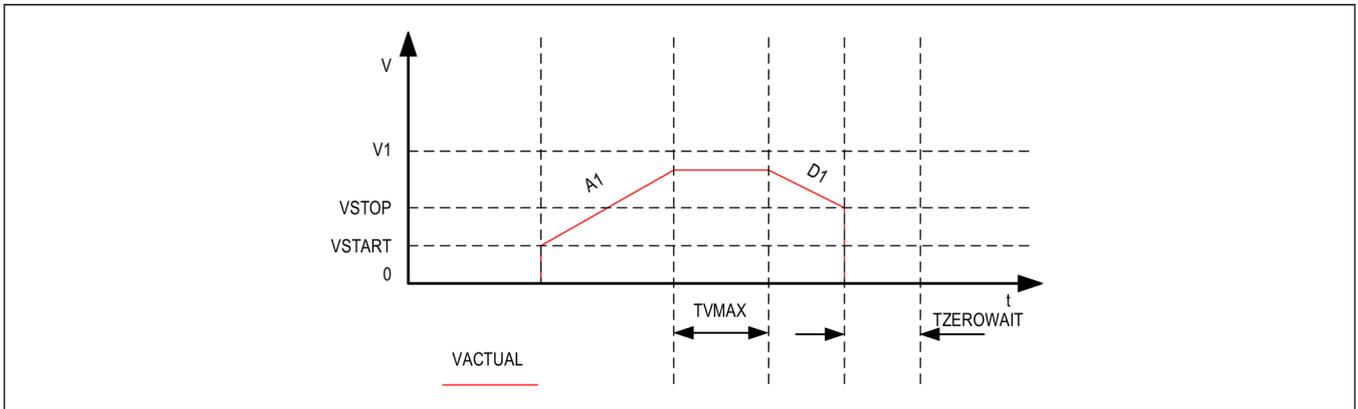


図 23. 距離が短いために V1 に達しないプロファイル

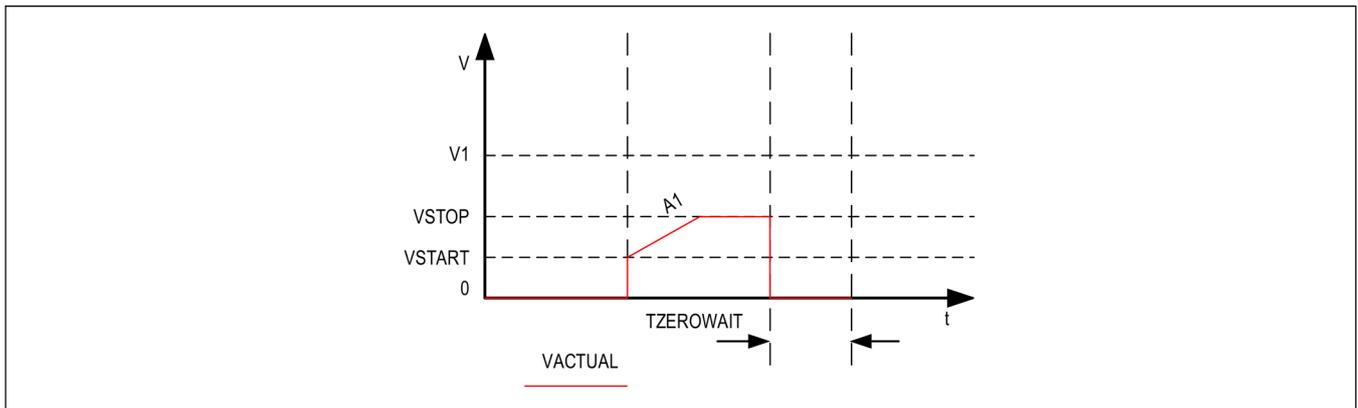


図 24. 距離が短いために TVMAX を維持できないプロファイル

移動距離が短いために VSTOP に達しない場合、A1 を用いた非常に短い線形加速のみが存在し、ランプは XTARGET に達すると直ちに終了します。

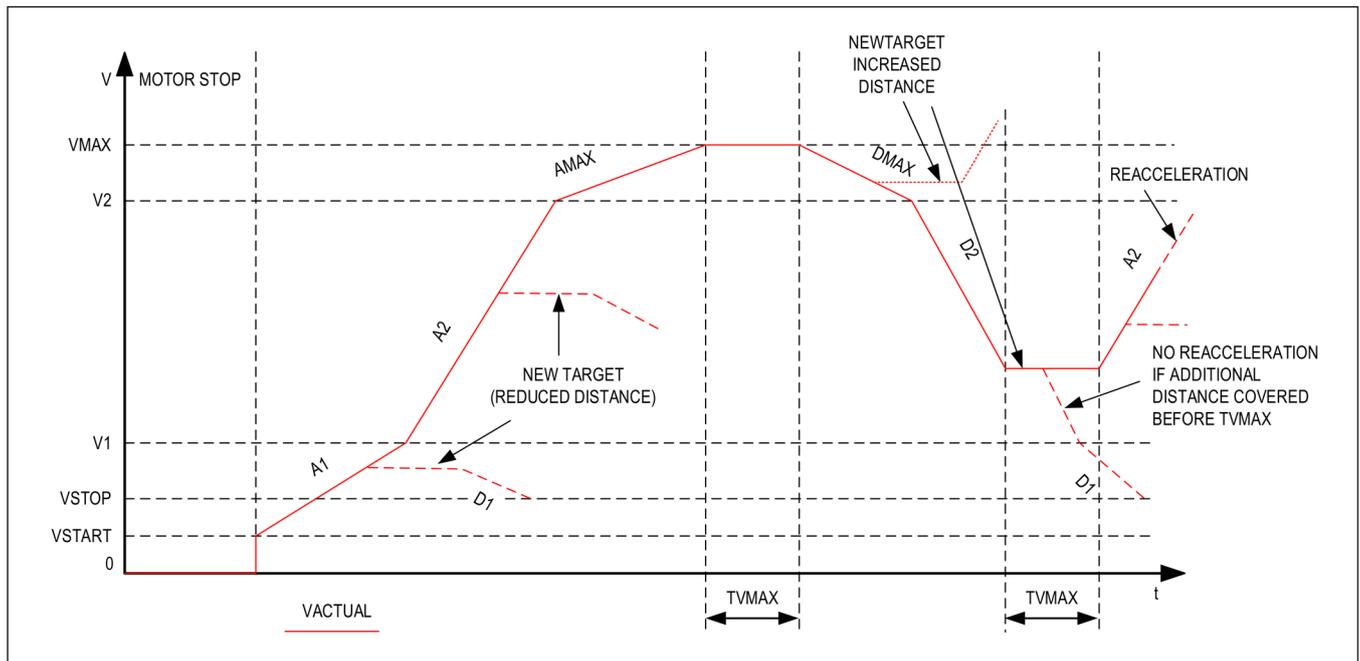


図 25. オンザフライの目標ポジション変更があった場合の 8 点ランプの例

## 速度モード

速度モードの動作では、異なる加速度設定および減速度設定を用いないことで、簡単に使用できるようにしています。速度モードでは、AMAX および VMAX のみに関係します。このモードでは、ランプ・ジェネレータは常に AMAX を用いて加速し、VMAX を用いて減速します。

モーターが停止するまで減速するためには、VMAX をゼロに設定すれば十分です。フラグ `vzero` はモーターの停止を通知します。フラグ `velocity_reached` は、目標速度に達したことを常に通知します。

## ランプの早期終了

システムとインタラクティブであることが可能な場合、アプリケーションによっては、目標位置に達する前に速度ゼロにランプダウンすることで動作を終了させる必要があります。加速設定値を用いて動作を終了させるオプションは、次のとおりです。

- 速度モードに切り替えます。VMAX=0 に設定し、AMAX を必要な減速度値に設定します。これにより、線形ランプを使用してモーターが停止されます。
- ポジショニング・モードでの停止には、VSTART=0、および VMAX=0 に設定します。この場合、VSTOP は使用しません。ドライバは AMAX、A1、A2 (V1 および V2 で定義) を用いて速度ゼロまで減速します。
- DMAX、D1、D2、VSTOP を用いて停止するには、XACTUAL を XTARGET にコピーして減速フェーズをトリガします。TZEROWAIT には、CPU がこの時間中にインタラクティブに動作できるだけの時間を設定します。ドライバは減速を行い、最終的に停止に至ります。a) または b) のオプションを用いて TZEROWAIT 時間中に動作を停止するには、実際の速度をポーリングします。
- 停止スイッチを有効化します。これは、ハードウェア入力（例えば、停止スイッチ入力にワイヤード OR を使用）によって行います。ハードウェア入力を用いず、REFL および REFR を固定レベルに接続している場合は、停止機能 (`stop_l_enable`、`stop_r_enable`) を有効化し、反転機能 (`pol_stop_l`、`pol_stop_r`) を用いることで擬似的にスイッチを有効化します。
- 仮想的な停止スイッチ (`VIRTUAL_STOP_L`、`VIRTUAL_STOP_R`) を利用します。ポジションを比較し (`X_ACTUAL` と `VIRTUAL_STOP_L/R`)、それに応じて停止をトリガします。

## アプリケーション例：ジョイスティック制御

監視カメラなどのアプリケーションは、モーション・コントローラを用いて最適な形で機能強化ができます。ジョイスティック・コマンドはモーターをユーザによる定義の速度で動作させ、また、目標ランプ・ジェネレータは、余すところのない有効な動作範囲を確保します。

### ジョイスティック制御を実現するには：

- ポジショニング・モードを使用して、動作方向の制御および動作制限値の設定を行います。これらの制限値は、仮想停止スイッチとして使用できます（例えば `VIRTUAL_STOP_L` および `VIRTUAL_STOP_R`）。
- ジョイスティックの入力に応じて任意のタイミングで、VMAX を VSTART から最大値までの範囲で変更します。VSTART=0 の場合、IHOLD および IRUN でそれぞれ定義された実際の電流スケーリングを考慮する `CS_ACTUAL` を設定することで、CoolStepMAX=0 で動作を止めることもできます。モーション・コントローラは、V1 および V2 によって決定される A1、A2、AMAX を用いて、ランプアップおよびランプダウンの速度を適応させます。

- 加速設定を変更しない場合、XTARGET は書き換えず、VMAX のみを変更してください。
- DMAX、D1、D2、VSTOP を使用できるのは、ランプ・コントローラが目標位置に達したためにスロー・ダウンする場合、または、目標位置が別の方向の点に変更された場合のみです。

## 速度閾値

ランプ・ジェネレータは、実際の速度 VACTUAL と合わせていくつかの速度閾値を提供します。様々な範囲を用いることで、モーターを最適なステップ・モード、コイル電流、加速度設定値にプログラミングできます。ほとんどのアプリケーションでは、閾値のすべてを必ずしも必要とするわけではありませんが、原理上はすべてのモードを組み合わせることができます。外部ステップ源を用いる場合に速度を決定できるよう、VHIGH および VCOOLTHRS は、設定値 THIGH と TCOOLTHRS で決められます。TSTEP は、これらの閾値と比較されます。TSTEP の測定でジッタが生じた場合に、比較した結果が連続的に切り替わるのを避けるため、 $1/16TSTEP$  または  $1/32TSTEP$  のヒステリシス (GCONF レジスタの small\_hysteresis ビットを参照) が適用されます。スイッチング速度の上限値は、スレッシュホールドの設定値より  $1/16$  または  $1/32$  だけ高くなります。StealthChop の閾値 TPWMTHRS は示されていません。VCOOLTHRS は、StealthChop2 の速度範囲内または SpreadCycle の速度範囲内のいずれかで用いることができます。

様々なチョッパー・モードやセンサーレス動作機能のための速度閾値は、2つのマイクロステップ間の時間 TSTEP と組み合わせられます。

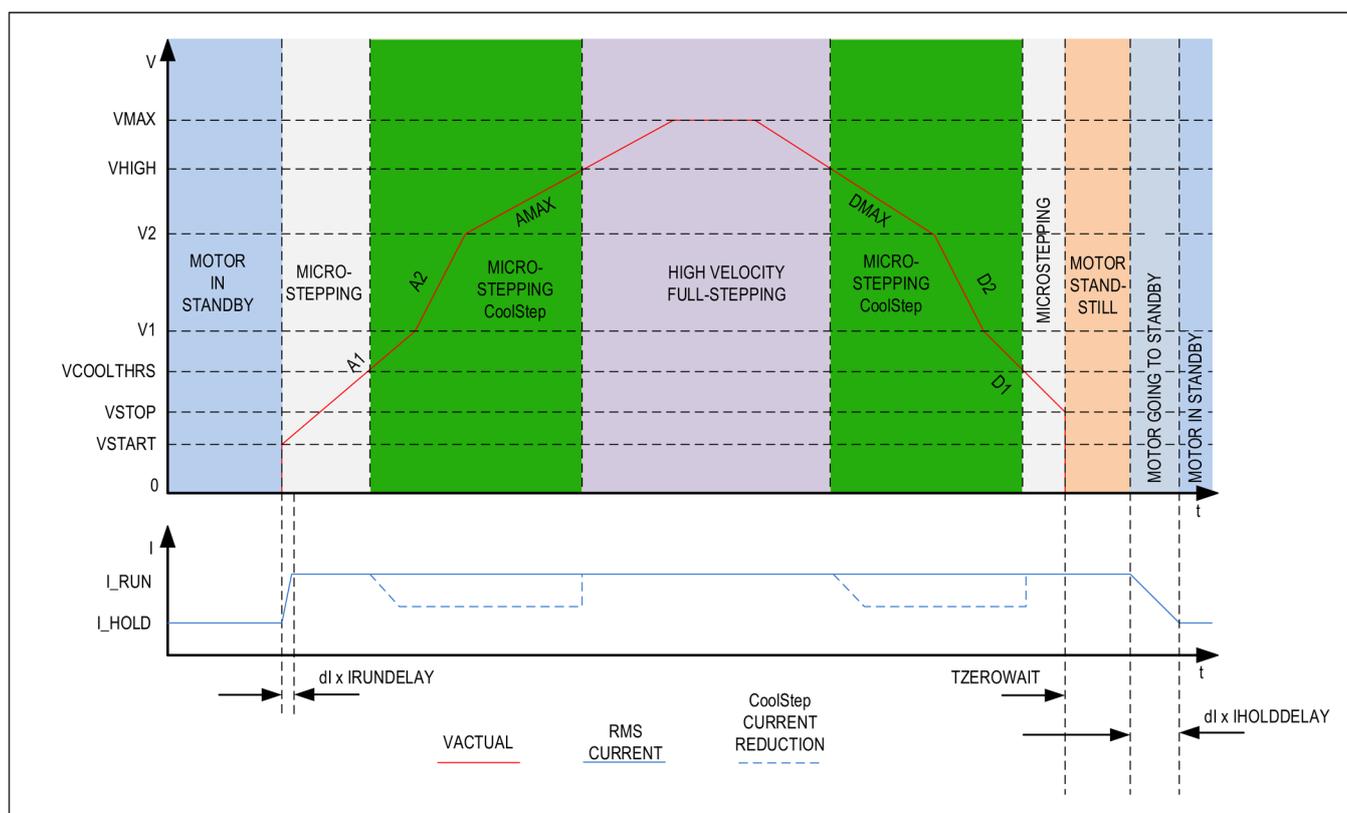


図 26. ランプ・ジェネレータの速度依存モーター制御

## リファレンス・スイッチ

通常の駆動動作の前に、絶対的なリファレンス・ポジションを設定する必要があります。

リファレンス・ポジションは、StallGuard2、StallGuard4、またはリファレンス・スイッチによって機械的な停止を検出することによって見出すことができます。

線形駆動の場合、機械的な動作範囲を超えることはできません。これは、左側リファレンス・スイッチおよび右側リファレンス・スイッチの停止スイッチ機能を有効化することで、異常な状態に対しても確保できます。そのため、ランプ・ジェネレータは、SW\_MODE レジスタで設定されたいくつかの停止イベントに応答します。モーターを停止する方法には次の 2 通りがあります。

- スイッチに達した場合に急激に停止できます。これは緊急時や StallGuard2 に基づくホームイングの場合に有用です。
- ソフト停止機能（ビット `en_softstop = 1`）を用いて、減速設定（`DMAX`、`V2`、`D2`、`V1`、`D1`）によりモーターをゼロまで緩やかに減速できます。

**ヒント：** スイッチ・イベント時にランプ・ポジション `XACTUAL` を保持レジスタ `XLATCH` にラッチすると、リファレンス・スイッチのポジションの正確なスナップショットが得られます。

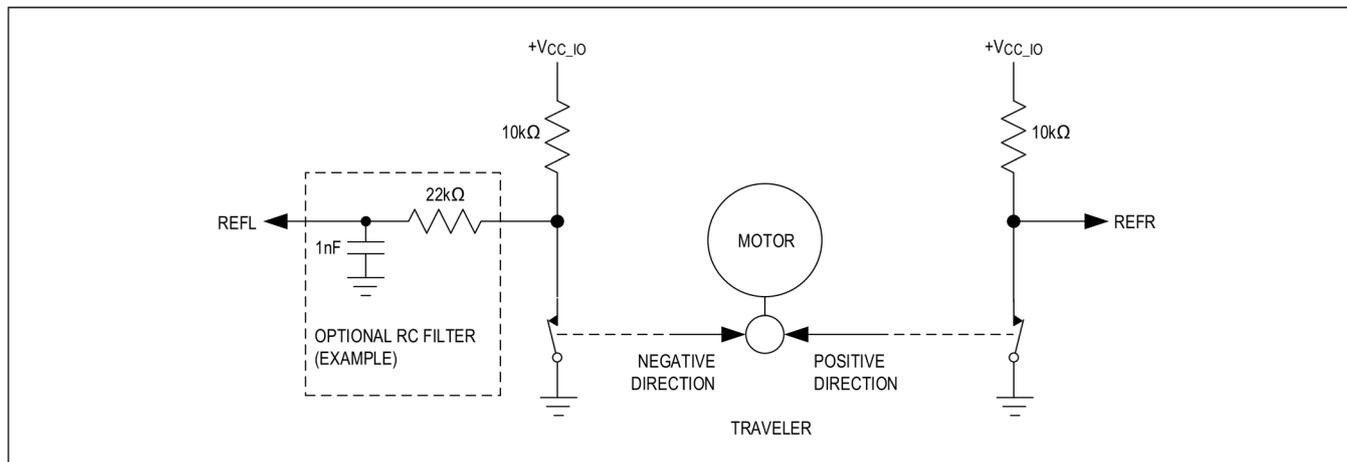


図 27. リファレンス・スイッチの使用（例）

スイッチの極性をプログラミングするか、プルアップまたはプルダウン抵抗の設定を選択することで、ノーマリ・オープンまたはノーマリ・クローズのスイッチを使用できます。ノーマリ・クローズのスイッチは、スイッチ接続の割込みに対してフェールセーフです。使用できるスイッチは次のとおりです。

- 機械式スイッチ
- フォト・インタラプタ
- ホール・センサー

リファレンス・スイッチの抵抗は、スイッチ条件に合ったものを選択するようにしてください。

ケーブルが長い場合、TMC5241 のリファレンス入力付近に RC フィルタを追加する必要が生じることもあります。RC フィルタを追加すると、誤った配線によりロジック・レベル入力を破壊してしまうおそれも低減できますが、アプリケーションに関して考慮すべき一定の遅延が加わります。

### ホームイング・プロシージャの実行

- ホーム・スイッチから距離を置くなどして、ホーム・スイッチが押されることのないようにしてください。
- 目的のスイッチ・イベントでのポジション・ラッチを有効化し、アクティブ・スイッチでのモーターの（ソフト）ストップを有効化します。StallGuard2 ベースのホームイングでは、ハード・ストップ（`en_softstop = 0`）を用いる必要があります。
- 動作ランプをスイッチの方向に起動します（左側スイッチはより負の位置に移動させ、右側スイッチはより正の位置に移動させます）。位置ランピング・コマンドを用いることで、この動作をタイムアウトできます。
- スイッチに達すると直ちに、ポジションがラッチされモーターは停止します。実際の速度 `VACTUAL` をポーリングするか、`vzero` フラグまたは `standstill` フラグをチェックして、モーターが再度停止状態になるまで待ちます。
- ランプ・ジェネレータをホールド・モードに切り替え、ラッチ・ポジションと実際のポジションの差を計算します。StallGuard2 ベースのホームイング時、またはハード・ストップの使用時には、`XACTUAL` は正確にホーム・ポジションで停止します。そのため、差は生じません（0）。
- 計算した差を実際のポジション・レジスタに書き込みます。これでホームイングが終了です。ポジション 0 に移動させると、モーターは正確にスイッチ点に戻ります。StallGuard2 をホームイングに用いた場合は、`RAMP_STAT` に読出しアクセスを行うことで、StallGuard2 の停止イベント `event_stop_sg` がクリアされ、モーターは停止条件から解放されます。

### 3 つ目のスイッチによるホームイング

アプリケーションによっては、機械式のリミット・スイッチとは独立に動作する、追加のホーム・スイッチを用います。TMC5241 のエンコーダ機能は、ポジション・ラッチに対する追加のソースを提供します。この機能では、N チャンネル入力を用いて `XACTUAL` を、立上がりエッジ・イベント時または立下がりエッジ・イベント時あるいはその両方でスナップショットできます。また、この機能は、割込み出力も可能です。

- ラッチ機能を有効化します（`ENCMODE : ignoreAB`、`clr_cont`、`neg_edge` または `pos_edge`、`latch_x_act` をセット）。これによりラッチ機能が割込み出力をトリガできます（割込みが `DIAG0` で示された場合に、`ENC_STATUS` の `n_event` を読み出すことでチェック）。

- N チャンネル・スイッチがあるはずの方向に移動します。ホーム・スイッチが検出される前にモーターが停止スイッチ（REFL または REFR）に達した場合、移動方向を逆にします。
- スwitchがトリガされたら、XLATCH を読み出します。これはスウィッチ・イベントが生じたポジションを示します。
- スwitch・イベントの検出後、モーターを停止させ、実際のポジションから XLATCH を差し引きます（必要なステップの詳細は、上述のホーミング・プロシージャで説明しています）。

### 仮想リファレンス・スイッチ

TMC5241 は、リファレンス・スイッチが 1 つしかない、あるいはまったくないアプリケーションに対応するため、仮想リファレンス・スイッチをサポートし（StallGuard ホーミング）、物理的な移動範囲を安全に制限しています。仮想停止スイッチがアクティブになるのは、実際のモーター・ポジション（XACTUAL）が、正方向への移動時に VIRTUAL\_STOP\_R を超える値になるか、負方向への移動時に VIRTUAL\_STOP\_L より小さい値になる場合です。仮想停止スイッチをイネーブルするには、en\_virtual\_stop\_l または en\_virtual\_stop\_r を設定します。各仮想停止スイッチは、それぞれの方向への動作のみをブロックします。

オプションで、仮想停止を、エンコーダ・ポジション（X\_ENC）のモニタ用に切り替えることもできます。これを選択するには、virtual\_stop\_enc を設定します。仮想停止の値（VIRTUAL\_STOP\_R、VIRTUAL\_STOP\_L）は、符号付き 32 ビット移動範囲のオーバーフロー／アンダーフローに対して十分な距離を保って設定し、ソフト減速を用いた場合にモーターが減速できるようにします。

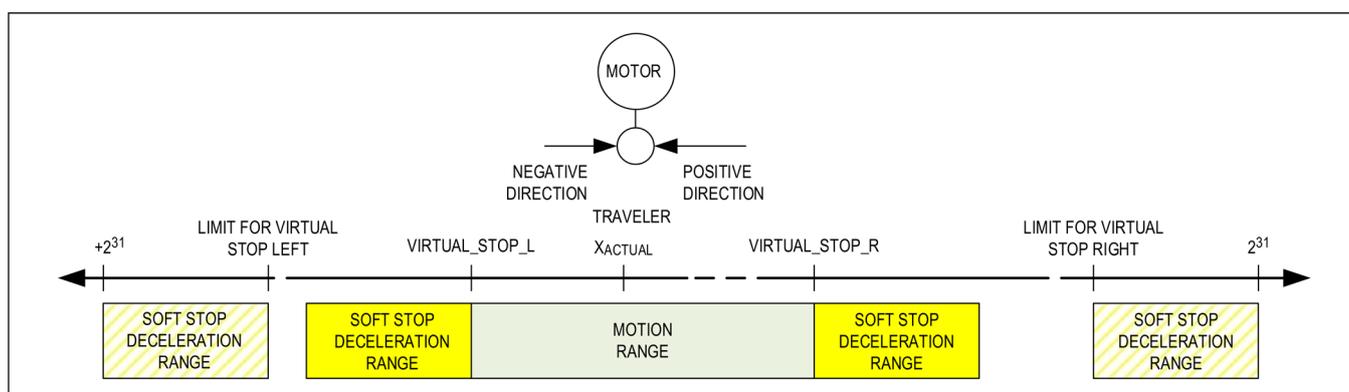


図 28. 仮想停止スイッチと制限の図解

### ランプ・ジェネレータの応答時間

ランプ・ジェネレータはハードウェアで実現され、1 マイクロ秒未満以内にコマンドを実行して目的のモードに切り替わり、目標値が有効になります。

速度アキュムレータは、実際の加速度設定値に基づき 512 クロック・サイクルごとに速度を更新し、滑らかな加速を実現します。

ただし、低移動速度設定時や低加速度設定時、例えば、ランプのポジショニング開始時（VSTART）や終了時（VSTOP）などでは、実際のステップ・パルス・レートは比較的低くなります。

そのため、最初と最後のステップの実行には、選択したマイクロステップ速度に応じて、顕著な遅延が加わる可能性があります。

例えば、10Hz のマイクロステップ速度の場合、ステップとステップの間に 100ms の時間が経過します。ランプの最後のマイクロステップ（の少なくとも一部）が VSTOP に等しい速度で実行されるため、これは目標ポジションに達するまでに大きな遅延が生じる原因となります。ランプを短時間で終了させるには、VSTOP を最小 100~1000 の範囲に設定します（100 の場合は約 10ms 未満、1000 の場合は約 1ms 未満になります）。

### 外部 STEP/DIR ドライバ

TMC5241 は、内部ランプ・ジェネレータを用いることで、ADI-Trinamic の TMC262C、TMC2160、または TMC2241 などの外部 STEP/DIR ドライバを制御し、強力なステッピング・モーター・アプリケーションを実現できます。この構成では、通常内部ドライバは使用ませんが、2 個のモーターを同期動作させる場合などで、外部ドライバに追加して用いることもできます。

DIAG0 出力と DIAG1\_SW 出力は、diag0\_nint\_step および diag1\_nposcomp\_dir の GCONF フラグをセットすることで、STEP と DIR 出力に対してイネーブルされます。DcStep 機能や自動モーター電流制御機能などのその他の内部ドライバ機能は、このモードでは使用できません。外部ドライバから TMC5241 への帰還がないためです。

DIAG1\_SW が低レベルの場合は正方向へのステップ（XACTUAL が増加）に対応し、高レベルの場合は、負方向へのステップ（XACTUAL が減少）に対応します。

外部 STEP 信号には次の 2 つのオプションが選択できます。

- ダブル・エッジ (GCONF.length\_step\_pulse = 0)。STEP 出力の各トグルが 1 ステップに対応します。DIR は少なくとも 1CLK サイクル分先行して有効になります。一致させるには、外部ドライバの dedge 機能を有効化します。
- シングル・エッジ (GCONF.length\_step\_pulse > 0)。STEP 出力の各正パルスが 1 ステップに対応します。CLK サイクルのパルス長は、length\_step\_pulse で制御されます。DIR 信号は、2 つのステップ・パルスの間においてのみ変化し、次のステップ・パルスの前に、STEP パルス長に等しい最小セットアップ・タイムを保持します。

この機能を用いるとステップ同期信号を外部ロジックに供給でき、これにより、例えば測定をトリガできます。

## 位置比較機能

ポジション比較機能を用いると、外部イベントをモーターの動作と同期してトリガできます。この機能は、XACTUAL = X\_COMPARE の場合に常に、アクティブ・ハイのレベルを出力します。そのため、パルスの長さは、XACTUAL が X\_COMPARE に一致している時間、つまり、実際の速度における 1 マイクロステップの長さに対応します。

繰り返し機能を用いれば、周期的な比較パルスをトリガできます。必要な周期 (マイクロステップの距離) をプログラムするには、X\_COMPARE\_REPEAT を用います (最大 224 - 1 ステップ)。

表 20. X\_COMPARE\_REPEAT オプションと周期的なパルス動作

値	説明
0	繰り返しなし
>1	最初の比較ポジションを通過すると連続的なパルス列を生じ、動作方向が変わるまで続きます。 比較パルス間の距離: 2~224 - 1 マイクロステップ

ポジション比較条件 XACTUAL = X\_COMPARE が真から偽の状態になると必ず、X\_COMPARE\_REPEAT にプログラムされた値に応じて、X\_COMPARE が自動的にインクリメントまたはデクリメントされます。X\_COMPARE をインクリメントするかデクリメントするかは、実際の動作方向に基づいて決まります。したがって、動作時の最初の比較イベントは X\_COMPARE の内容で与えられ、後続のイベントの距離は X\_COMPARE\_REPEAT でプログラムされます。動作方向を変更する場合、または、次のパルス・ポジションがソフトウェアによって変わる場合は、最初に X\_COMPARE\_REPEAT = 0 と設定して繰り返しメカニズムを無効化してください。次に、X\_COMPARE を次の目標パルス・ポジションで再度プログラムします。その前に自動生成された次のポジションが依然として以前の動作方向のままとなっており、これに到達しない可能性があるためです。X\_COMPARE への書き込みアクセスを終えたら、繰り返しメカニズムを再度有効化できます。X\_COMPARE への書き込みアクセス時に X\_COMPARE が X\_ACTUAL と同じである場合には、これも繰り返しメカニズムをトリガする原因となるため、最初に繰り返しメカニズムを無効化するステップが必要です。

## StallGuard2 の負荷測定

異なるモーター制御方式を適合させるために、TMC5241 には 2 つの基本チョッパー・モードに対応する、2 種類の StallGuard センサーレス負荷検出機構があります。StallGuard2 は SpreadCycle 動作時に機能し、StallGuard4 は StealthChop2 動作に合わせて最適化されています。

StallGuard2 では、モーターに加わる負荷を正確に測定できます。これは、ストール検出や、モーターをストールさせる下限未満の負荷の場合におけるその他の用途 (CoolStep 負荷適応型電流低減など) で用いることができます。StallGuard2 の測定値は、広範な負荷、速度、電流の各設定値にわたり直線的に変化します。モーターの負荷が増加するにつれ、StallGuard の値 (SG\_RESULT) は減少します。ストールを正しく検出するには調整が必要です。モーターが過負荷/ストール状態となった場合に SG\_RESULT が 0 (あるいはほぼ 0) に達するように、StallGuard の閾値 (SGTHRS) を設定します。

**ヒント:** StallGuard2 と CoolStep を使用するには、最初に SGT 設定を使用して StallGuard2 の感度を調整してください。

**注意:** TMC5240 や TMC2240 と異なり、比較の際に SG4\_THRS の値を 2 倍することで、SG4\_THRS は SG4\_RESULT の 9 ビット全範囲をカバーします。

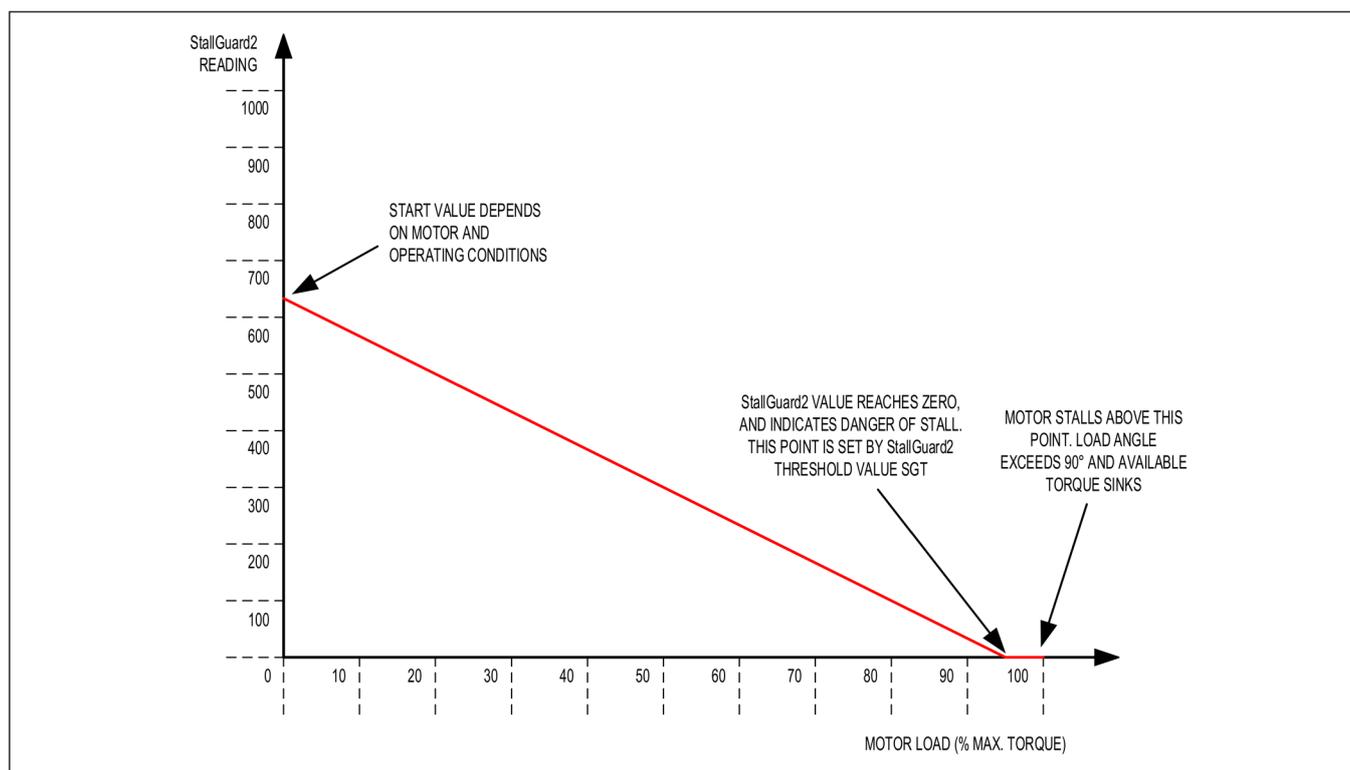


図 29. StallGuard2 の機能原理

表 21. StallGuard2 関連のパラメータ

パラメータ	説明	設定	コメント
SGT	この符号付きの値は、ストール検出用の StallGuard2 閾値レベルを制御し、読出し用に最適な測定範囲を設定します。値を小さくすると感度が高くなります。ゼロはほとんどのモーターで使用できる開始値です。値を大きくすると StallGuard2 の感度は低下し、ストールを示すために必要なトルクが大きくなります。	0	中立値
		+1...+63	低感度
		-1...-64	高感度
sflit	測定精度を向上させるには StallGuard2 のフィルタをイネーブします。これをセットすると、測定頻度がモーターの電氣的周期 (4 フルステップ) あたり 1 測定に減少します。	0	標準モード
		1	フィルタリング・モード
ステータス・ワード	説明	範囲	コメント
SG_RESULT	これは StallGuard2 の結果です。読出し値が大きいほど機械的負荷は小さいことを示します。読出し値が小さいほど負荷が高く、したがって負荷角度も大きいことを示します。モーター・ストール前の最大負荷時に SG_RESULT の読出し値が約 0~100 となるよう、SGT 設定値を調整します。	0...1023	0 : 最大負荷 低値 : 高負荷 高値 : 低負荷

### StallGuard2 の閾値 SGT の調整

StallGuard2 の値 SG\_RESULT は、モーター固有の特性と、負荷、速度、電源電圧、電流レベルに対するアプリケーション固有の要求による影響を受けます。そのため、StallGuard2 の閾値 SGT を特定のモーター・タイプおよび動作条件に合わせて調整するための最も簡単な方法は、実際のアプリケーションでインタラクティブに調整を行うことです。

### StallGuard SGT を調整するための初期手順

- モーターをアプリケーションの通常の動作速度、電源電圧、電流設定で動作させ、SG\_RESULT をモニタします。

- モーターに印加する機械的負荷を徐々に増加します。SG\_RESULT がゼロになる前にモーターがストールした場合は、SGT を小さくします。モーターがストールする前に SG\_RESULT がゼロに達した場合は、SGT を大きくします。SGT の良い開始値はゼロです。SGT は符号付きの値です。そのため、負または正の値が可能です。
- ここで、sg\_stop をイネーブルし、ストール時にいつでもモーターが安全に停止するようにします。ストールが発生する前にモーターが停止する場合は、SGT を増加します。sg\_stop をディスエーブルするか RAMP\_STAT レジスタの event\_stop\_sg (書込み後クリア) をクリアして、モーターを再始動します。
- モーター・ストール直前の負荷増大時に SG\_RESULT が 0 と約 100 の間にあり、負荷がない場合に SG\_RESULT が 100 以上増加すれば、最適設定になっています。ほとんどの場合、SGT は特定の動作速度または動作範囲に合うよう調整できます。設定が特定の範囲 (目的の速度の 80%~120%など)、かつ、極端なモーター条件下 (適用可能な最低および最高温度) でも確実に機能することを確認してください。

### SGT の自動調整を可能にするオプション手順

SGT 設定の背景にある基本的な考え方は、StallGuard 測定をモーター内の抵抗性負荷に対して補償する、というものです。停止時あるいは極めて低い速度時には、抵抗性損失がモーターのエネルギー・バランスの主要因になります。機械的なパワーがゼロまたはほぼゼロであるためです。したがって、SGT は、速度がほぼゼロの場合の最適値に設定できます。このアルゴリズムは、環境条件やモーター浮遊容量などとは無関係に最高の結果を与えるよう、アプリケーション内で SGT を調整する場合に特に有用です。

- 10RPM 未満 (毎秒数フルステップ) の低速度かつ、目標動作電流および電源電圧でモーターを動作させます。この速度範囲では、SG\_RESULT はモーター負荷にはそれほど依存しません。モーターは十分なバック EMF を発生しないためです。そのため、機械的負荷は結果に大きな違いを与えません。
- sflt をスイッチ・オンします。ここで、SGT を 0 から SG\_RESULT が増加を始める値まで増加させます。SGT が大きいと、SG\_RESULT は最大値まで増加します。再び、SG\_RESULT がゼロにとどまる最大値まで減少させます。これで、SGT が可能な限り高感度の値に設定されます。SG\_RESULT は速度が上がるにつれ増加するので、有用なストール検出となります。
- モーター・ストール時には SG\_RESULT がゼロになり、SW\_MODE の sg\_stop をイネーブルすると、ストール時にモーターを停止するようランプ・ジェネレータをプログラムできます。sg\_stop を使用するために、StallGuard が良好な結果をもたらすことのできる下限速度閾値に一致するよう、TCOOLTHRS を設定します。

この設定値を用いてストール検出を行うための上限速度は、モーターのバック EMF が電源電圧に近付き、速度を更に増加させた場合にモーター電流が低下し始める速度で決まります。

システム・クロック周波数は SG\_RESULT に影響します。最高性能が要求されるアプリケーションには、外部水晶安定化クロックを用いる必要があります。電源電圧も SG\_RESULT に影響します。そのため、レギュレーションを厳格に行うほど値の正確さが増します。SG\_RESULT の測定は高分解能であり、以降のセクションで説明するように精度を高めるための方法がいくつかあります。

### TCOOLTHRS および THIGH を制限する可変速度

前述の SGT 調整の結果として選択した SGT 設定は、ある特定の速度範囲で使用できます。この範囲を外れると、ストールが安全に検出されず、CoolStep が最適な結果を与えない場合があります。

多くのアプリケーションでは、ほとんどの時間で単一動作点あるいはそれに近い点での動作が用いられ、1 つの設定で十分です。ドライバは、これに合致する下限および上限速度閾値を用います。決定した動作点の範囲外、例えば、TCOOLTHRS をマッチング値に設定する場合のセンサーレス・ホーミング・プロシージャに先立つ加速フェーズ時では、ストール検出はディスエーブルされます。上限値は THIGH で指定されます。

速度制限値 VHIGH および VCOOLTHRS は、THIGH および TCOOLTHRS の設定値で決まります。

アプリケーションによっては、少数のサポート点と線形補間を用いた SGT 値の速度依存調整が当を得ている場合があります。

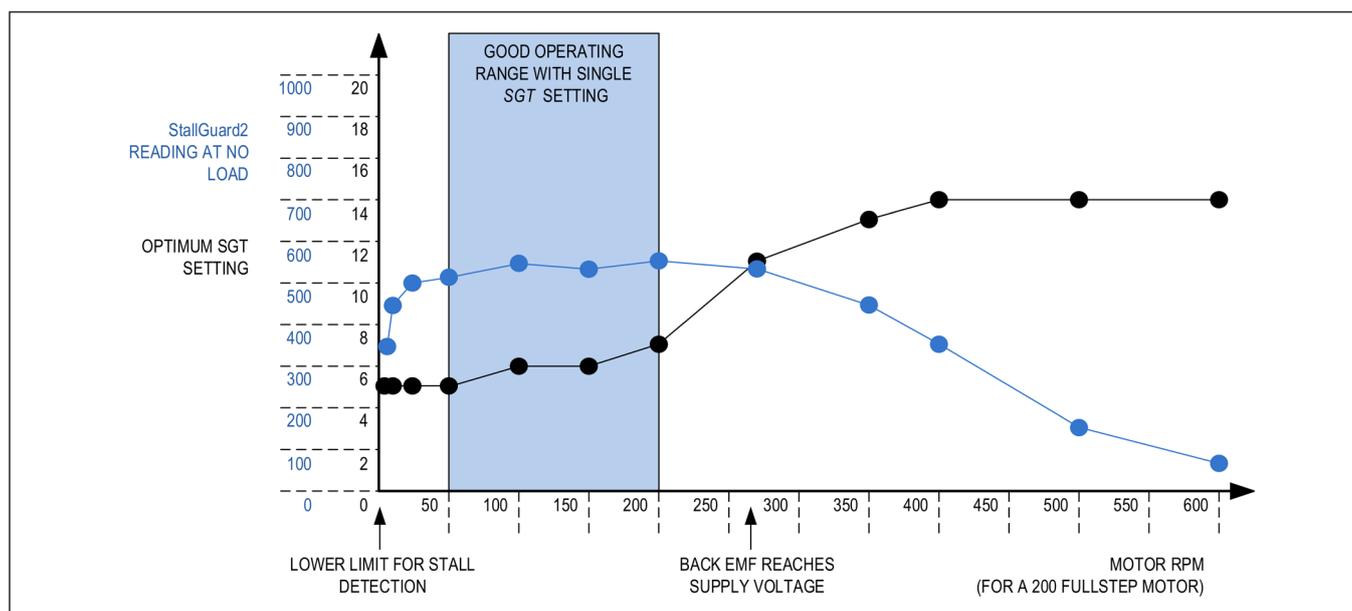


図 30. 例：サンプル・モーターを用いた SGT の最適設定値と StallGuard2 の読出し値

### 高トルク・リップルと共振を伴う小型モーター

保持トルクの高いモーターでは、モーター電流の変化に伴う StallGuard2 の測定値 SG\_RESULT の変動が、特に小電流時に大きくなります。このようなモーターでは、最善の結果が得られるよう、電流依存性を確認する必要があります。

### モーター・コイル抵抗の温度依存性

広い温度範囲で動作するモーターでは、温度上昇に伴いモーターのコイル抵抗が増加するため、温度補正が必要となる場合があります。この補正は、モーター効率の低下と同様に、温度上昇に比例して SG\_RESULT を直線的に減少させることで行うことができます。

### StallGuard2 測定の正確さと再現性

量産環境においては、1 つのモーター・タイプ用のアプリケーション内で、固定 SGT 値を用いることが望ましい場合があります。StallGuard2 測定におけるユニットごとのばらつきのはほとんどは、モーター構築時の製造許容誤差に起因します。StallGuard2 の測定誤差は、他のパラメータはすべて安定していると仮定すると、以下に示す程度まで低くできます。

stallGuard2 の測定誤差 =  $\pm \max(1, |SGT|)$

### StallGuard2 の更新レートおよびフィルタ

StallGuard2 の測定値 SG\_RESULT は、モーターのフルステップごとに更新されます。ストールでは必ず 4 つのフルステップが失われるので、この方法によりストールを十分確実に検出できます。実際のアプリケーション、特に、CoolStep を用いる場合は、より高精度の測定を行うことの方がフルステップごとに更新を行うより重要となる場合があります。機械的な負荷は決してステップごとに瞬時に変化することはないためです。こうしたアプリケーションでは、sfilt ビットによって、4 回の負荷測定にわたりフィルタリング機能を有効化できます。高精度測定が必要な場合は、このフィルタを常に有効化してください。このフィルタは、例えば A 相と B 相の磁石のミスアライメントによるモーター構造のばらつきを補償します。負荷の増加に対して短時間で対応する必要がある場合や、StallGuard を用いたセンサーレス・ホーミングで最善の結果を得ようとする場合は、フィルタを無効化してください。

### モーター・ストールの検出

最適なストール検出を行うには、StallGuard2 のフィルタリング機能を無効化します (sfilt = 0)。モーターのストールを確実に検出するには、特定の SGT 設定値を用いてストール閾値を決定する必要があります。そのため、モーターをストールさせずに駆動できる最大負荷を決定する必要があります。同時に、この負荷での SG\_RESULT の値 (0~100 の範囲のいずれかの値) をモニタします。ストール閾値は、パラメータの浮遊を許容できるよう、動作制限範囲に十分収まる値とします。SGT 設定値が 0 または 0 に近い場合の応答を調べると、信号品質に関する何らかの手がかりが得られます。負荷のない場合と負荷が最大の場合の SG\_RESULT 値をチェックしてください。少なくとも 100 または数 100 の差を示すはずで、これは主としてオフセットと比較されます。最大モーター負荷時に 0 の読出し値が得られるように SGT 値を設定している場合は、ストールは自動的に検出されてモーター停止を指示します。ステップ・ロスの原因となるステップの瞬間には、最小の読出し値が示されます。ステップ・ロスの後、モーターは振動し、SG\_RESULT の読出し値は増加します。

## StallGuard2 を使用したホーミング

線形駆動のホーミングでは、モーターをハード・ストップの方向に移動させることが必要です。StallGuard2 は、動作するには一定の速度が必要なので (TCOOLTHRS で設定)、加速フェーズに必要な距離が得られるよう、開始点がハード・ストップから十分離れているようにしてください。SGT とランプ・ジェネレータ・レジスタを設定した後、ハード・ストップの方向に動作を開始し、ストール時に停止する機能を有効化します (SW\_MODE の sg\_stop をセット)。ストールが検出されると、ランプ・ジェネレータは動作を停止し、VACTUAL をゼロにしてモーターを停止させます。停止条件は、DRV\_STATUS のフラグ StallGuard によっても示されます。モーターが直ちに再始動することのないよう新しい動作パラメータを設定した後、StallGuard2 をディスエーブルできます。あるいは、RAMP\_STAT を読み出すことで、モーターを再度イネーブルできます。動作パラメータが変更されていない場合、RAMP\_STAT の event\_stop\_sg フラグの「読出し後クリア」機能は、TZEROWAIT 時間の経過後、モーターを再始動させます。

## StallGuard2 動作の制限

StallGuard2 は、モーター速度が極端な場合、信頼できる動作ができません。モーター速度が非常に低い (多くのモーターでは 1Rps 未満) 場合、生成されるバック EMF は低く、測定は不安定で環境条件 (温度など) に依存するようになります。前述の自動調整プロセスによってこれを補償できます。その他の条件でも、SGT の設定が極端な値になったり、測定値 SG\_RESULT のモーター負荷に対する応答が低下したりする原因になります。

サイン波電流の必ずしもすべてをモーター・コイルに供給することができないような、非常に高いモーター速度の場合も、低応答の原因となります。これらの速度は、通常、モーターのバック EMF が電源電圧に達することで特徴づけられます。

## StallGuard4 の負荷測定

StallGuard4 は、StealthChop2 を使用した動作に対して最適化されています。これに対して、その先行ツールである StallGuard2 は SpreadCycle を使用して機能します。

いずれにせよ、機能は類似しており、どちらも負荷値を高い値から低い値へと変化させ、高負荷時に低い値となります。

StallGuard2 はストール検出時に「0」の読出し値を示すよう調整されるのに対し、StallGuard4 では、オフセットを加えて測定結果をシフトさせるのではなく、比較値を用いてストール検出をトリガします。

StallGuard4 では、モーターに加わる負荷を正確に測定でき、これを使用して、CoolStep による負荷適応型電流削減の他、ストール検出や負荷の推定が可能です。StallGuard4 の測定値は、図 31 に示すように、広い範囲の負荷、速度、電流の各設定値にわたり直線的に変化します。最大モーター負荷に近付くにつれ、値はモーター固有の低い値まで減少します。これは、ローターのコイルと磁石の磁界が 90° の負荷角になることに対応します。また、これは、モーターにとって最もエネルギー効率の高い動作点でもあります。

StallGuard4 を使用するためには、境目となる条件でモーターの感度をチェックします。

**注意:** TMC2240 や TMC5240 と異なり、比較の際に SG4\_THRS の値を 2 倍することで、SG4\_THRS は SG4\_RESULT の 9 ビット全範囲をカバーします。

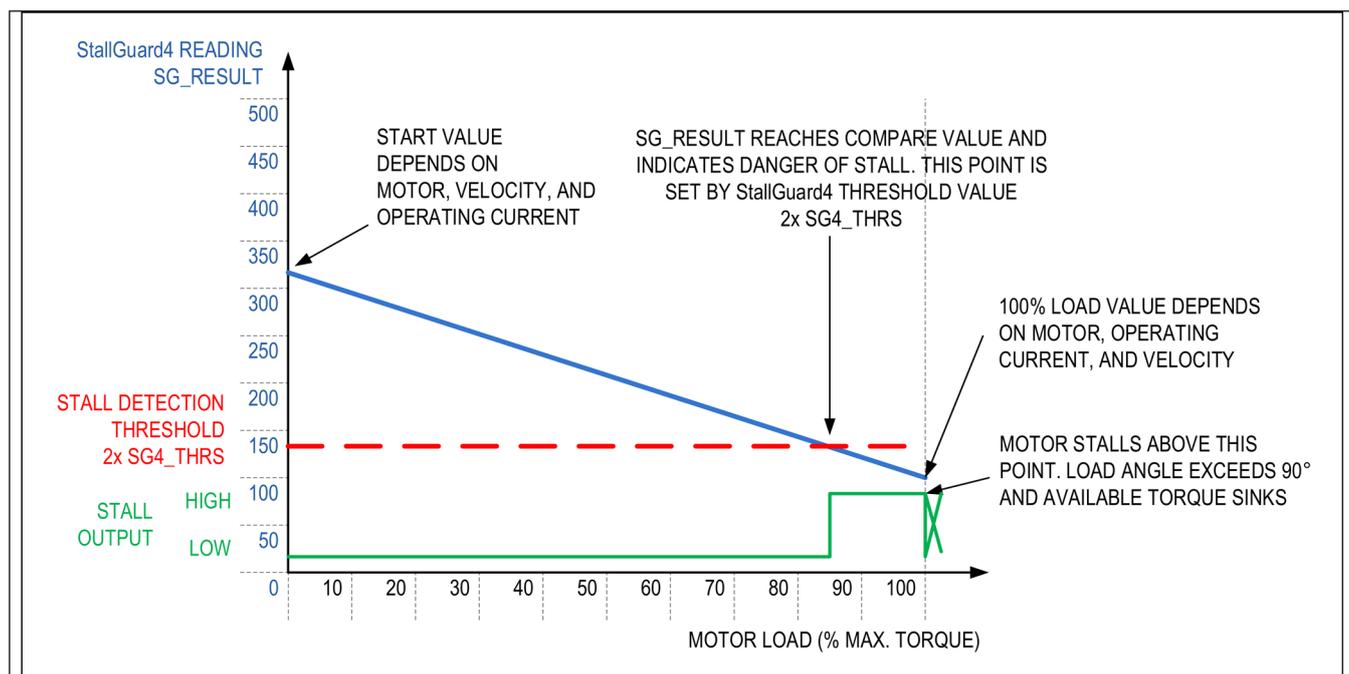


図 31. StallGuard4 の動作モード

表 22. StallGuard4 関連のパラメータ

パラメータ	説明	設定	コメント
SG4_THRS	この値はストール検出のための StallGuard4 閾値レベルを制御します。モーター固有の特性を補償し、また、感度を制御します。値が高いほど感度も高くなります。値を大きくすると StallGuard4 の感度は増加し、ストールを示すために必要なトルクは小さくなります。結果として得られる閾値はこの値の 2 倍になります。	0... 255	この値の 2 倍の値が SG4_RESULT と比較されます。SG4_RESULT が 2 × SG4_THRS より低下した場合、ストール出力がアクティブになります。
ステータス・ワード	説明	範囲	コメント
SG4_RESULT	これは StallGuard4 の結果です。読出し値が大きいほど機械的負荷は小さいことを示します。読出し値が小さいほど負荷が高く、したがって負荷角度も大きいことを示します。この値は、実際のチョップ・モードのようなイネーブル条件や VCOOLTHRS のような速度閾値とは独立に生成されます。結果は SG4_IND_x の計測値から計算され、StallGuard2 より高い精度と同程度の範囲を実現するために 1 ビットが加えられます。	0...510	低値：最大負荷 高値：低/無負荷
SG4_IND_3 SG4_IND_2 SG4_IND_1 SG4_IND_0	モーターの A 相の立下がり (SG4_IND_0) / 立上がり (SG4_IND_1) 遷移または、B 相の立下がり (SG4_IND_2) / 立上がり (SG4_IND_3) 遷移に対する個別の測定値。個別の測定値はフィルタリング・モード (sg4_filt_en = 1) のみ使用できます。SG4_IND_0 は、非フィルタリング・モード (sg4_filt_en = 0) のすべての場合に対応します。	0...255	低値：最大負荷 高値：低/無負荷
sg4_filt_en	0：非フィルタリング動作、SG4_RESULT はフルステップごとに更新します 1：フィルタリング動作、SG4_IND_0...3 を使用可能、SG4_RESULT は最後の 4 つの SG4_IND_x 測定値の平均を示します。	0 1	0：フィルタ・オフ 1：フィルタリング動作、SG4_IND の値を使用可能
sg_angle_offset	このフラグは、StealthChop2 の位相ラグを決定し SpreadCycle の電圧制御動作から電流制御動作への切り替わり時の位相不連続を補償するために SG4_RESULT を用いることで、StealthChop2 と SpreadCycle の間の最適化された切り替えを可能にします。位相オフセットは保存され、StealthChop2 にスイッチ・バックする際に再度減算されます。	0 1	0：角度補正なし 1：StealthChop2 と SpreadCycle のスイッチングを最適化

## StallGuard4 の調整

StallGuard4 の値 SG4\_RESULT は、モーター固有の特性と、負荷、コイル電流、速度に対するアプリケーション固有の要求による影響を受けます。そのため、StallGuard4 の閾値 SG4\_THRS を特定のモーター・タイプおよび動作条件に合わせて調整するための最も簡単な方法は、実際のアプリケーションでインタラクティブに調整を行うことです。

StallGuard の SG4\_THRS を調整するための最初のプロシージャは次のとおりです。

- モーターをアプリケーションの通常の動作速度で動作させ、SG4\_RESULT をモニタします。
- モーターに印加する機械的負荷を徐々に増加します。モーターがストールする前の SG4\_RESULT の最低値をチェックします。この値を SG4\_THRS の開始値として用います（この値の半分を適用）。
- 次に、DIAG 出力を介して StallGuard 出力信号をモニタ（動作の下限速度に合うよう TCOOLTHRS も設定）し、パルスが各出力に表れたらモーターを停止させます。ストール時にはモーターが確実に停止するようにします。ストールが発生する前にモーターが停止する場合は、SG4\_THRS を増加します。
- ストールが確実に検出され、ストールの瞬間に DIAG にパルスが発生したら、最適な設定が実現されています。ほとんどの場合、SG4\_THRS は特定の動作速度または動作範囲に合うよう調整できます。設定が特定の範囲（目的の速度の 80%~120% など）、かつ、極端なモーター条件下（適用可能な最低および最高温度）でも確実に機能することを確認してください。

SG4\_RESULT が  $2 \times \text{SG4\_THRS}$  未満になると、StallGuard は DIAG にパルスを送ります。これが可能となるのは、StealthChop2 モードの場合と、 $\text{TCOOLTHRS} \geq \text{TSTEP} > \text{TPWMTHRS}$  の場合のみです。

外部モーション・コントローラは、必要時にモーターを停止させることで単一パルスに反応することが必要です。StallGuard が良好な結果をもたらすことのできる下限速度閾値に一致するよう、TCOOLTHRS を設定します。

SG4\_RESULT の測定は高分解であり、以降のセクションで説明するように精度を高めるための方法がいくつかあります。

## StallGuard4 の更新レート

StallGuard4 の測定値 SG4\_RESULT は、モーターのフルステップごとに更新されます。ストールでは必ず 4 つのフルステップが失われるので、この方法によりストールを十分確実に検出できます。

StallGuard4 には測定用に次の 2 つのオプションがあります。

- `sg4_filt_en = 0` : 単一測定、各フルステップ後に更新、1 フルステップごとに有効。この測定により、コイル電圧のゼロ伝送ごとに SG4\_RESULT が完全に更新されるため、負荷の変動に対して最速の応答が可能になります。そのため、固い障害物がある場合のストール検出に最適です。
- `sg4_filt_en = 1` : このモードでは、次に示す 4 つの個別の信号が生成されます。コサイン波（コイル A）の立下がり 0 遷移時の SG4\_IND\_0、コサイン波の立上がり 0 遷移時の SG4\_IND\_1、サイン波（コイル B）の立下がり 0 遷移時の SG4\_IND\_2、サイン波の立上がり 0 遷移時の SG4\_IND\_3 です。SG4\_RESULT の実際の値は、4 つのすべての測定値の平均値で、フルステップごとに更新されます。そのため、各フルステップが全体の結果に及ぼす影響は 25% だけです。このモードは、柔らかい障害物がある場合の検出、あるいは、正確さに欠けるモーターでの CoolStep の使用に最適です。フィルタリング・モードでは、負荷の突然的増加（ハード・モーター阻害）に対する感度は減少します。

## モーター・ストールの検出

モーターのストールを確実に検出するには、特定の SG4\_THRS 設定値と特定のモーター速度または速度範囲を用いてストール閾値を決定する必要があります。更に、モーター電流の設定値には一定の影響力があり、一度最適値を決定したら変更はすべきではありません。そのため、所定のアプリケーションに対して、モーターをストールさせずに駆動できる最大負荷を決定する必要があります。同時に、この負荷での SG4\_RESULT をモニタします。ストール閾値は、パラメータの浮遊を許容できるよう、動作制限範囲に十分収まる値とします。評価をより精緻に行っても、固定閾値と比較する場合より、SG4\_RESULT の変化の影響を受ける可能性があります。そのため、絶対値に影響するいくつかの特定の効果を除外できます。

## StallGuard4 動作の制限

StallGuard4 は、モーター速度が極端な場合、信頼できる動作ができません。モーター速度が非常に低い（多くのモーターでは 1Rps 未満）場合、生成されるバック EMF は低く、測定は不安定で環境条件（温度など）に依存するようになります。その他の条件でも、測定値 SG4\_RESULT のモーター負荷に対する応答が低下する原因になります。サイン波電流の必ずしもすべてをモーター・コイルに供給することができないような、非常に高いモーター速度の場合も、低応答の原因となります。これらの速度は、通常、モーターのバック EMF が電源電圧を超えることで特徴づけられます。

## CoolStep の負荷適応型電流スケーリング

CoolStep は、モーターの機械的な負荷に基づきステッピング・モーターに対しスマートなエネルギー最適化を自動的にを行い、これらのモーターを「グリーン（環境にやさしい）」にします。実際のチョッパー・モードに応じて、CoolStep は、StealthChop2 では StallGuard4 の負荷測定結果、SpreadCycle では StallGuard2 の負荷測定結果を自動的に使用します。CoolStep は、使用前に StallGuard2 または StallGuard4 のいずれか（使用するチョッパー・モードに依存）を調整することが必要です。一回の調整ですべての動作点に対応できるわけではありません。

## CoolStep 用の設定

CoolStep はいくつかのパラメータで制御されますが、その作用を理解するのに重要なのは 2 つです。

表 23. CoolStep の重要パラメータ

パラメータ	説明	範囲	コメント
SEMIN	下限閾値を設定する 4 ビットの符号なし整数。SG_RESULT がこの閾値を下回る（負荷が大きいことを示す）場合、CoolStep は両方のコイルへの電流を増加します。4 ビットの SEMIN の値が 32 倍され、10 ビットの SG_RESULT の値の範囲の下半分を占めます。（このパラメータの名前は、CoolStep の以前の名称である smartEnergy に由来するものです）。	0	CoolStep をディスエーブル
		1...15	閾値は SEMIN × 32
SEMAX	上限閾値を制御する 4 ビットの符号なし整数。この閾値以上の値の SG_RESULT が十分な回数サンプリングされる（軽負荷であることを示す）場合、CoolStep は両コイルへの電流を減少します。上限閾値は、(SEMIN + SEMAX + 1) × 32 です。	0...15	閾値は (SEMIN + SEMAX + 1) × 32

図 32 に、CoolStep の動作領域を示します。

- 黒線は SG\_RESULT の測定値を示します。
- 青線はモーターに加えられた機械的負荷を示します。
- 赤線はモーター・コイルに供給される電流を表します。

負荷が増加すると、SG\_RESULT が SEMIN × 32 未満になり、CoolStep により電流が増加します。負荷が減少すると、SG\_RESULT が (SEMIN + SEMAX + 1) × 32 を上回り、電流は減少します。

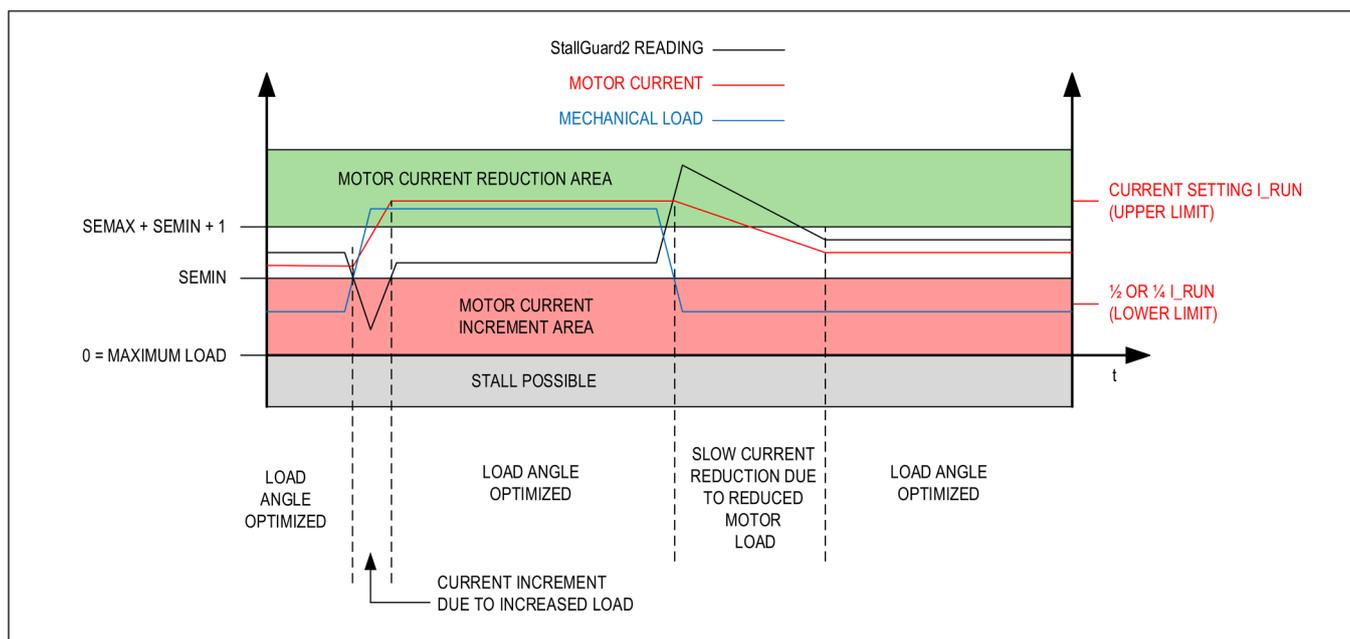


図 32. モーター電流を負荷に適合させる CoolStep

表 24. CoolStep のその他のパラメータとステータス情報

パラメータ	説明	範囲	コメント
SEUP	電流のインクリメント・ステップを設定します。SEMIN で設定された下限閾値未満の新しい StallGuard2 または StallGuard4 の値が測定されると、モーター電流は必ずこの設定値だけインクリメントされます。	0...3	CS 値 CS_ACTUAL のステップ幅は 1、2、4、8 です
SEDN	モーター電流の各電流デクリメントに必要な、上限閾値を上回る StallGuard2/StallGuard4 読み出し値の数を設定します。	0...3	デクリメントあたりの StallGuard2 測定値の数 : 32、8、2、1
SEIMIN	IRUN 電流設定値をスケールして、CoolStep 動作の下限モーター電流制限を設定します。 StealthChop2 を使用する場合、特に、25%まで低減することが必要な場合、StealthChop2 電流レギュレーション用に定められた最小モーター電流を十分上回る電流で動作するようにしてください。	0	0 : IRUN の 1/2 (StealthChop と使用するときは IRUN ≥ 16 とする)
		1	1 : IRUN の 1/4 (StealthChop と使用するときは IRUN ≥ 28 とする)
TCOOLTHRS	CoolStep に切り替えるための下限速度閾値。この速度を下回ると、CoolStep はディスエーブルされます。StallGuard2 が安定した結果を与える速度範囲の下限に合わせてください。  ヒント : VCOOLTHRS を VMAX と同じ値に設定することで、加速フェーズおよび減速フェーズ時に CoolStep をディスエーブルするよう修正できます。	1...2 <sup>20</sup> - 1	閾値の値を TSTEP と比較することで、下限 CoolStep 速度を指定します。
THIGH	CoolStep 用の上限速度閾値。この速度を超えると、CoolStep はディスエーブルされます。StallGuard2/StallGuard4 が安定した結果を与える速度範囲に合わせてください。	1...2 <sup>20</sup> - 1	また、フルステップへの切り替えなど、その他の機能も制御します。
ステータス・ワード	説明	範囲	コメント
CS_ACTUAL	このステータス値は、CoolStep によって制御される実際のモーター電流スケールを示します。値の上限は IRUN の値で、下限は SEIMIN で指定される IRUN の一部です。	0...31	1/32、2/32、... 32/32

## CoolStep の調整

CoolStep を SpreadCycle と併用して調整する前に、まず StallGuard2 の閾値レベル SGT を調整します。これは、負荷測定値 SG\_RESULT の範囲に影響します。CoolStep では、SG\_RESULT を使用してモーターを+90° の最適負荷角度付近で動作させます。CoolStep は、StealthChop2 と共に、SG4\_RESULT を使用します。このモードでは、レベル調整は SEMIN を介して行われます。

電流のインクリメント速度は SEUP で指定され、電流のデクリメント速度は SEDN で指定されます。これらは、異なる応答を必要とする可能性のある異なるイベントでトリガされるため、個別に調整できます。これらのパラメータのエンコーディングでは、コイル電流の増加を減少よりもはるかに速く行うことができます。下限閾値に達することは、高速応答を必要とする可能性のある、より深刻なイベントであるためです。応答が遅すぎる場合、モーターはストールする可能性があります。反対に、上限閾値に達するまでの応答が遅い場合は、電力を節約する機会を失う以外に重大なリスクを生じることはありません。

CoolStep は、電流スケール・パラメータ IRUN と seimin ビットで制御される制限値の間で動作します。

## 応答時間

モーター負荷の増加に対して高速に反応するには、電流インクリメント・ステップ SEUP に大きな値を用います。モーター負荷の変化が緩やかな場合は、電流インクリメント・ステップを下げることでモーターの発振を防止できます。sfilt で制御されたフィルタがイネールされている場合は、測定レートおよびレギュレーション速度は 1/4 になります。

**推奨：**最も一般的で利点の多い使用法は、CoolStep を標準的なシステムの目標動作速度で動作するように調整し、それに応じた速度閾値を設定することです。加速および減速は通常、短時間で行われるものなので、フル・モーター電流が必要ですが、時間が短いため全体の消費電力への影響はわずかなものにすぎません。

## 低速度およびスタンバイ動作

CoolStep は、停止時および非常に低 RPM 時のモーター負荷は測定できないため、ランプ・ジェネレータには下限速度閾値が設けられています。これには、アプリケーション固有のデフォルト値を設定する必要があります。この閾値未満では、IRUN または IHOLD を用いた通常の電流設定が有効です。上限閾値は VHIGH の設定により与えられます。速度制限値 VHIGH および VCOOLTHRS は、THIGH および TCOOLTHRS の設定値で決まります。

どちらの閾値も StallGuard2 と StallGuard4 の調整プロセスの結果として設定できます。

## 診断出力

DIAG 出力はポジションの比較信号を提供し、ソフトウェアをモーター・ランプ内の特定の条件にトリガするために、外部ロジックや割り込み信号を正確にトリガすることを可能にします。オープンドレイン（アクティブ・ロー）の出力信号（デフォルト、GCONF レジスタのビット diag0\_int\_pushpull=0）あるいはアクティブ・ハイのプッシュプル出力信号（GCONF レジスタのビット diag0\_int\_pushpull=1）のいずれかを選択できます。オープンドレイン出力を使用する場合は、複数のドライバ出力信号を OR 接続できます。4.7kΩ~100kΩ の範囲の外部プルアップ抵抗が必要です。DIAG0 は、リセット条件時にもローに駆動されます。この構成では DIAG0 をモニタリングすることでリセット条件の終了を決定することはできません。event\_pos\_reached フラグもリセット時にアクティブになり、リセット条件後もピンはアクティブ・ローであり続けるためです。リセット条件を確実に決定するには、SPI により reset フラグをモニタするか、いずれかのレジスタを読み出してチップがパワー・アップされていることを確認します。

これらの設定オプションにより、TMC5240 でのソフトウェアの互換性を確保しています。更に柔軟性を高めるため、TMC5241 にはレジスタ DIAG\_CONF が追加されています。このレジスタのビットを設定することにより、各診断出力 DIAG0/DIAG1 とは独立した信号を自由に割り当てることができます。

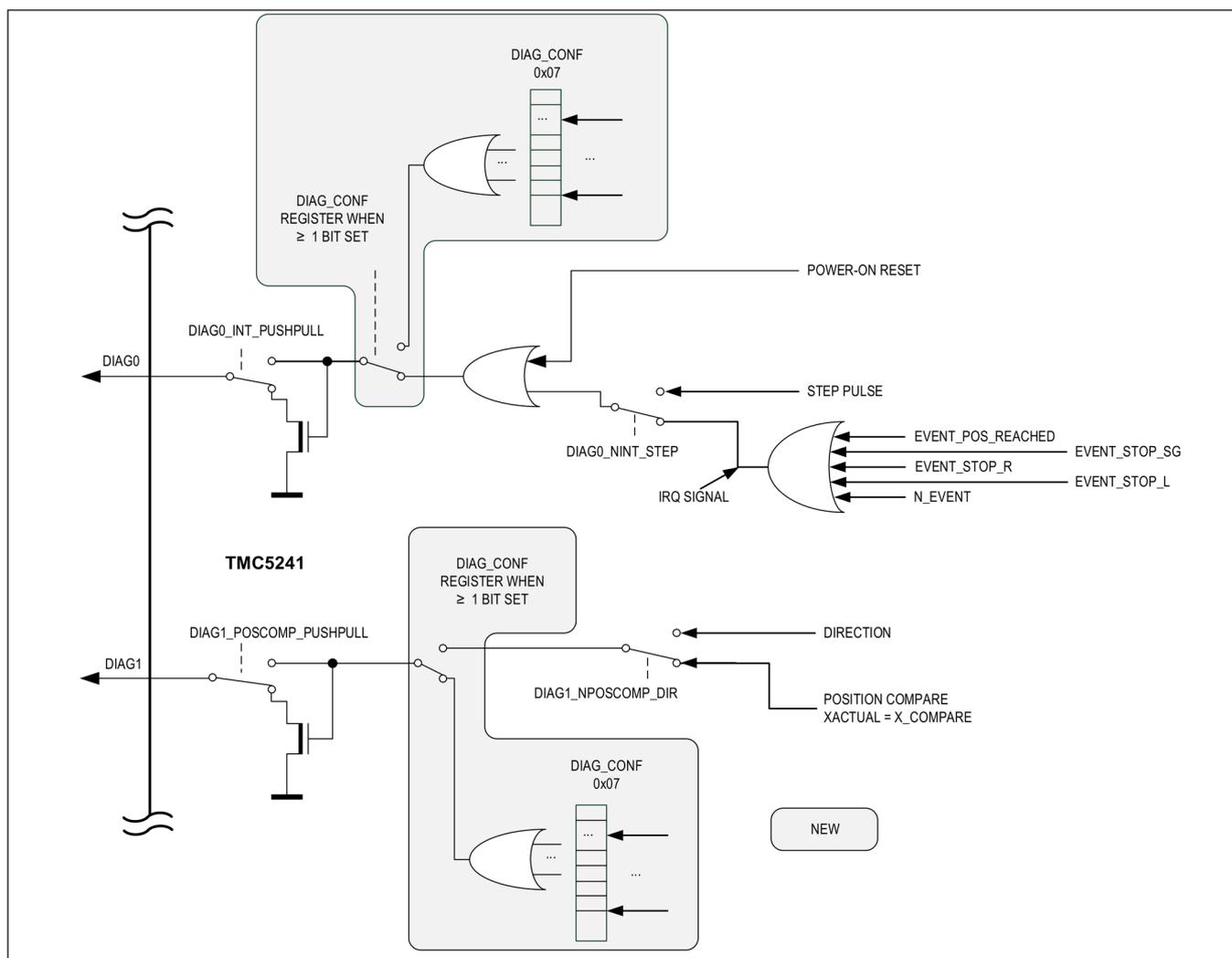


図 33. 診断出力構成のオプション

## DcStep

DcStep はステップング・モーター用の自動整流モードです。これを用いることで、負荷に対応している限り、ステップング・モーターはランプ・ジェネレータから指示された目標速度で動作できます。モーターが過負荷になった場合には、モーターが負荷を駆動できるまで速度を落とします。したがって、ステップング・モーターは決してストールすることなく、重負荷をできる限り高速で駆動できます。低速時に利用可能なより高いトルクと慣性重量による動的トルクにより、機械的なトルク・ピークの補償が可能です。モーターが完全に阻害された場合には、ストール・フラグがセットされます。

## DcStep のデザインイン

従来のアプリケーションでは、動作領域はアプリケーションの最大速度に必要な最大トルクによって制限されます。予期せぬ負荷ピーク、共振によるトルク低下、機械部品の経時劣化などを補償するためには、トルクに対して最大 50%の安全マージンが必要です。DcStep では、利用可能な最大モーター・トルクまですべて使用できます。短時間の動的負荷がかなり高い場合でも、モーターとアプリケーションの慣性重量を用いて、モーター・ストールのおそれなく、これを克服できます。DcStep を用いると、公称アプリケーション負荷を、保持トルク領域付近の安全マージンにのみ制限される、より高いトルク（モーターが実現できる最大トルク）まで拡張できます。更に、最大アプリケーション速度は、実際に実現可能なモーター速度まで増加できます。

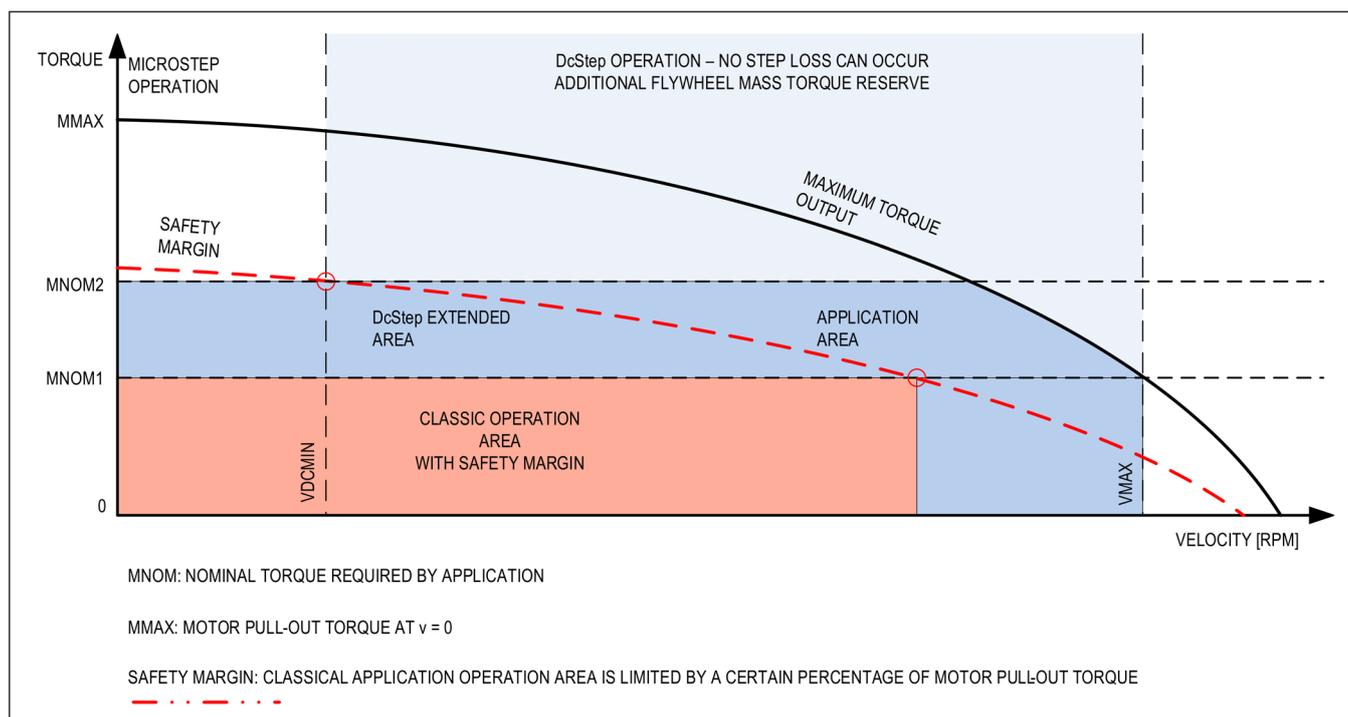


図 34. DcStep により拡大されるアプリケーション動作領域

### DcStep とモーション・コントローラの統合

DcStep に必要な設定はわずかです。DcStep は、モーターの動きをランプ・ジェネレータに直接フィードバックするため、モーターが目標速度に対して過負荷状態になっている場合でも、モーション・ランプに滑らかに統合できます。DcStep は、フルステップ・モードで一定の TOFF チョッパーを用いて、ランプ・ジェネレータの目標速度、VMAX、あるいはモーターが過負荷の場合には減速した速度でモーターを動作させます。これには、最小動作速度 VDCMIN を設定する必要があります。VDCMIN は、DcStep がモーター動作の信頼できる検知を可能にする最小動作速度に設定されます。速度が VDCMIN 未満になるまでブレーキがかけられることのない限り、モーターがストールすることはありません。速度がこの値未満に低下した場合、ストール検出がイネーブル (sg\_stop がセット) されていない限り、モーターは負荷が解放されると再始動します。ストール検出は、速度が VDCMIN 未満になった場合に、StallGuard2 によって処理されません。

**注意：** DcStep では、サイン波の位相極性が、768~255 の MSCNT 範囲内で正、256~767 の範囲内では負となる必要があります。コサイン波の極性は、0~511 では正、512~1023 では負である必要があります。位相が 1 だけシフトすると DcStep の動作を阻害します。そのため、デフォルトの波形を用いることを推奨します。デフォルトのテーブルの初期化については、[サイン波ロックアップ・テーブル](#)のセクションを参照してください。

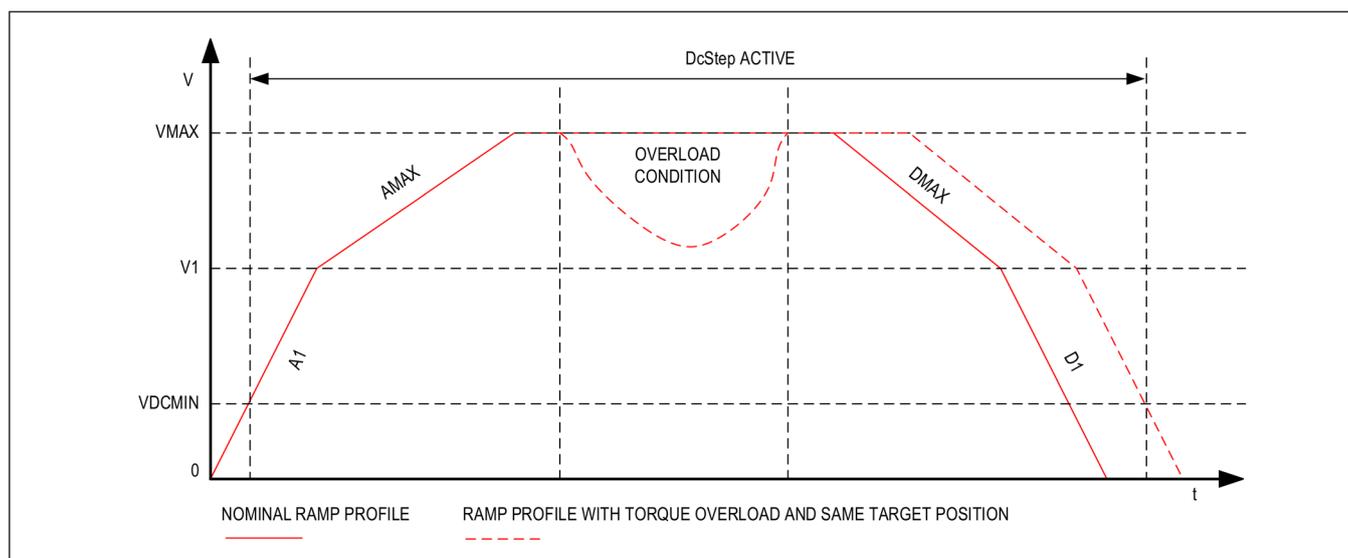


図 35. 過負荷状態時の DcStep の速度プロファイル

### DcStep モードにおけるストール検出

DcStep は過負荷時にモーターを減速できますが、あらゆる動作状況でストールを回避できるわけではありません。モーターがブロックされた場合や、モーター動作がそれ以上安全に検知できないようなモーター依存の最小速度を下回るほど減速した場合は、モーターがストールする、あるいはステップを失う可能性があります。ステップ・ロスを確認し検出しモーターの再始動を回避するために、stop-on-stall (ストール時に停止) をイネーブルできます (フラグ `sg_stop` をセット)。この場合、モーターがストールすると `VACTUAL` はゼロに設定されます。モーターは、`RAMP_STAT` ステータス・フラグを読み出すまで、停止したままになります。フラグ `event_stop_sg` はアクティブな停止条件を示します。モーターは、`event_sg_stop` フラグをリセットするか、`stop-on-stall` (ストール時に停止) をディスエーブルするまで、停止したままになります。StallGuard2 の負荷値は、DcStep 動作時にも使用できます。DcStep では  $0^{\circ} \sim 90^{\circ}$  の負荷角 ( $0 \sim 255$  に対応) のみが生じるので、この値の範囲は、 $0 \sim 255$  に制限されます。特定の状況では、負荷角が一時的に  $90^{\circ} \sim 180^{\circ}$  の範囲に逸脱する場合があります。これは安定的でないとはいえ、読み出し値は最大 511 になります。StallGuard2 をイネーブルするには、`VDCMIN` よりわずかに高い速度または `VMAX` に応じて、`TCOOLTHRS` も設定します。

フライホイール負荷がモーターの軸と緩やかに結合している場合、このモードでのストール検出は共振によって誤ってトリガされる可能性があります。

表 25. DcStep の重要パラメータ

パラメータ	説明	範囲	コメント
<code>vhighfs</code> and <code>vhighchm</code>	CHOPCONF におけるこれらのチョッパー設定フラグは、DcStep 動作に設定されている必要があります。VDCMIN を超過すると直ちに、チョッパーは TOFF が一定のチョッパーに切り替わり、フルステップ動作をするようになります。	0/1	DcStep には 1 を設定。
TOFF	DcStep では、多くの場合、CHOPCONF のオフ時間値が大きいほうがメリットが得られます。2 を超える設定値を推奨します。	2...15	8~15 の設定値は、DcStep の動作に 8 を設定する場合と異なる点はありません。
VDCMIN	これは、内蔵ランプ・ジェネレータを使用する場合の DcStep 動作の下限閾値です。この閾値未満では、モーターは通常のマイクロステップ・モードで動作します。DcStep 動作では、モーターが完全にブロックされている場合でも、最小の VDCMIN で動作します。DC_TIME の設定値と共に調整してください。  StealthChop を有効化しても DcStep はディスエーブルされません。	0...2 <sup>22</sup>	0 : DcStep をディスエーブル DcStep を動作させるためには下限速度を設定します。
DC_TIME	この設定は、DcStep の負荷測定のためのリファレンス・パルス幅を制御します。最大モーター・トルクで安定した動作となるよう最適化する必要があります。値を大きくすると、より大きなトルクとより高い速度が可能となり、値を小さくすると、VDCMIN で設定された下限まで動作速度を下げることができます。  公称動作条件下で最適な設定となることをチェックし、極端な動作条件下 (最低動作電源電圧、最大モーター温度、最大電源電圧、最低モーター温度など) で再チェックします。	0...1023	この設定に対する下限値は、クロック・サイクル単位の <code>tBLANK</code> (TBL で定義) + n で、n は 1~100 の範囲です (標準的なモーターの場合)。

DC_SG	<p>この設定は、DcStep モードでのストール検出を制御します。感度を高くするにはこの値を増加します。</p> <p>ストールは、モーターのハード・ストップを発行することにより、エラー条件として用いることができます。ストール・イベント時にモーターを停止するには sg_stop フラグをイネーブルします。この方法により、ストールが生じるとモーターは停止します。</p>	0...255	DC_TIME/16 よりわずかに大きな値に設定します。
-------	--	---------	------------------------------

### DcStep 動作での実際のモーター速度の測定

DcStep は、モーターが機械的負荷により目標速度よりも低速になってしまう場合に、モーター速度を低減できます。VACTUAL はランプ・ジェネレータの目標速度を示します。これは DcStep の影響を受けません。DcStep の速度の測定は、ポジション・カウンタ XACTUAL に基づいて可能です。

そのため、次式のように、ポジション・カウンタの2つのスナップショットを既知の時間差で取得します。

$$VACTUAL_{DCSTEP} = \frac{XACTUAL(\text{time2}) - XACTUAL(\text{time1})}{\text{time2} - \text{time1}} \times \frac{2^{24}}{f_{CLK}}$$

例：

クロック周波数が 16.0MHz の場合、速度の値に 0.954s の測定遅延が直接生じ、実際の DcStep 速度の 1/100 には 9.54ms の遅延が生じます。時間差をできる限り正確に得るには、IC からの XACTUAL の伝送が開始または終了するごとに、タイマーをスナップショットします。SPI 伝送用の NCS の立上がりエッジが最も正確な時間基準を提供します。

### サイン波ルックアップ・テーブル

TMC5241 は、マイクロステップの電流波形を保存するためにプログラマブルなルックアップ・テーブルを備えています。デフォルトでは、このテーブルはサイン波で事前プログラムされており、ほとんどのステッピング・モーターにとって良い開始点となっています。テーブルをモーター固有の波形に再プログラミングすることで、特に低コスト・モーターでのマイクロステップを大幅に改善できます。ユーザに対する利点は次のとおりです。

- マイクロステッピング：低コスト・モーターで著しく改善。
- モーター：滑らか、かつ静かに動作。
- トルク：機械的な共振が低減することでトルクが向上。
- 低周波数のモーター・ノイズ：実際のモーターの製造公差にサイン波およびコサイン波のシフトを加えることで低減。

### マイクロステップ・テーブル

プログラムするのに必要なメモリとデータ量を最小限に抑えるために、保存される波形は 1/4 のみです。内蔵のマイクロステップ・テーブルはマイクロステップ波形を 0°~90° に割り当てます。これが 360° まで対称的に拡張されます。テーブルの読み出し時には、10 ビットのマイクロステップ・カウンタ MSCNT が、完全に拡張された波形テーブルをアドレス指定します。このテーブルは、エントリごとに 1 ビットを使用して、昇順に保存されています。そのため、波形の 1/4 を保存するのに必要なのは、わずか 256 ビット (ofs00~ofs255) のみです。これらのビットが 8 個の 32 ビット・レジスタに割り当てられます。各 ofs ビットは、テーブルで 1 ステップ進行する際に、勾配 Wx を加算するかどうか、すなわち Wx + 1 とするかどうかを制御します。Wx が 0 の場合、実際のマイクロステップ位置でのテーブルのビットが 1 であれば、これは次のマイクロステップに進む際に「1 を追加」することを意味します。波形は 1 よりも大きな勾配を持つ場合もあるので、1/4 波形内で最大 4 つの柔軟なプログラマブル・セグメントを用いて、基本勾配 Wx を -1、0、1、または 2 にプログラムできます。したがって、負の勾配でも実現できます。4 つの勾配セグメントは、位置レジスタ X1~X3 で制御されます。勾配セグメント 0 は、マイクロステップ位置 0 から X1-1 までの範囲で、その基本勾配は W0 で制御されます。また、セグメント 1 は X1 から X2-1 の範囲で、基本勾配は W1 で制御されます。他のセグメントも同様です。

波形を変更する場合、波形の 1/4 を全波形に拡張するときに滑らかで対称的なゼロ遷移が確保されるよう注意が必要です。実現可能な最大分解能を実現すると同時にヒステリシススペースのチョッパーがオフセットを追加できるヘッドルームを残しておくために、変更の結果得られる波形の最大振幅が -248~248 の範囲となるよう調整する必要があります。

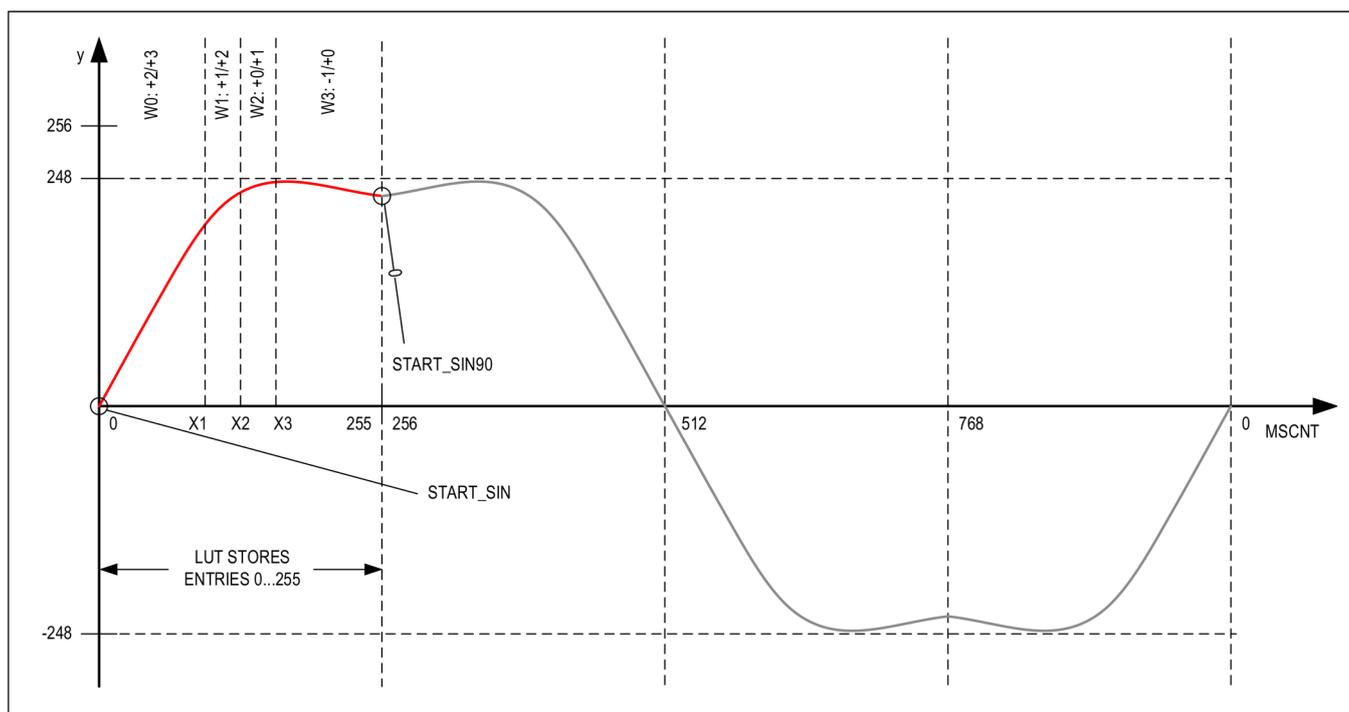


図 36. LUT プログラムの例

マイクロステップ・シーケンサはテーブル内を進行する際に、モーター・コイルの実際の電流値をマイクロステップごとに計算し、それらをレジスタ `CUR_A` および `CUR_B` に保存します。ただし、インクリメント・コーディングは、特にマイクロステップ・テーブルが変更される場合には、絶対的な初期化を必要とします。そのため、`CUR_A` と `CUR_B` は、`MSCNT` がゼロと交差するときは必ず初期化されます。

#### 位相シフトとモーターのマッチング：

2つのレジスタがテーブルの開始値を制御します。

- ゼロでの開始値は必ずしも 0 ではない（1 または 2 の場合がある）ので、これを開始点レジスタ `START_SIN` にプログラムできます。
- 同様に、2つ目のモーター・コイル用の2つ目の波形の開始点は `START_SIN90` に保存する必要があります。このレジスタは、2相モーターの90°の位相シフト用に得られたテーブル・エントリを保存します。モーターの許容誤差に合わせるために、-127~+127の範囲のマイクロステップ・オフセット（レジスタ `OFFSET_SIN90`）を追加することで、位相シフトを90°（256マイクロステップ）から45°~135°の範囲の任意の値に変更できます。モーターの許容誤差に対しては、最大で数10ステップの中程度の調整が必要です。必要な補正オフセットを求めるには、`StallGuard4`の個々の値 `SG4_IND` を使用して両方のコイルが対称の結果を与えるようになるまでオフセットを調整します。

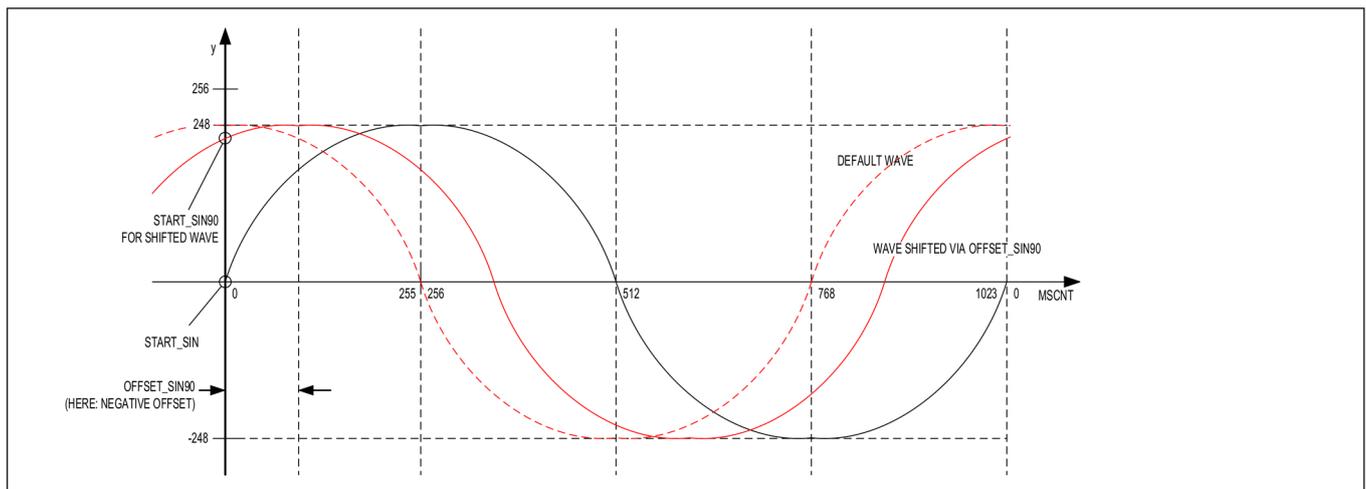


図 37. OFFSET\_SIN90 を使用したコサイン波のシフト

デフォルトのテーブルは、独自テーブルを実現するための良い基礎となります。リセット後のデフォルト・マイクロステップ・テーブルの初期化例を以下に示します。

```
MSLUT[0] = %10101010101010101010101010101010 = 0xAAAAB554
```

```
MSLUT[1] = %01001010100101010101010010101010 = 0x4A9554AA
```

```
MSLUT[2] = %00100100010010010010100100101001 = 0x24492929
```

```
MSLUT[3] = %00010000000100000100001000100010 = 0x10104222
```

```
MSLUT[4] = %11111011111111111111111111111111 = 0xFBFFFFFF
```

```
MSLUT[5] = %101101011011101101110110111011101 = 0xB5BB777D
```

```
MSLUT[6] = %01001001001010010101010101010110 = 0x49295556
```

```
MSLUT[7] = %000000000010000000100001000100010 = 0x00404222
```

```
MSLUTSEL = 0xFFFF8056:
```

```
X1 = 128, X2 = 255, X3 = 255
```

```
W3 = %01, W2 = %01, W1 = %01, W0 = %10
```

```
MSLUTSTART = 0x00F70000:
```

```
START_SIN_0 = 0, START_SIN90 = 247
```

モーターの位相シフトを最適化するには、StealthChop2 で中程度の速度でモーターを動作させ、sg4\_filt\_en = 1 にセットします。位相 A (SG4\_IND\_0+SG4\_IND\_1) と位相 B (SG4\_IND\_2+SG4\_IND\_3) に対する StallGuard4 の結果に一致するよう、位相オフセットを調整します。

位相 A の値が位相 B の値より大きければ、OFFSET\_SIN90 をインクリメントし、小さければデクリメントします。最適な一致が得られるまでこれを繰り返します。START\_SIN90 には必ず正しい値を入力してください。オフセットが-10~+9 の場合は START\_SIN90=247 を用います。-17 または +17 までは START\_SIN90=246 を用います。START\_SIN は常に 0 です。

## ABN インクリメンタル・エンコーダ・インターフェース

TMC5241 は、ABN エンコーダに対応したインクリメンタル・エンコーダ・インターフェースを備えています。エンコーダは、デジタル・インクリメンタル直交信号（通常 A および B と名付けます）およびインデックス信号（通常ヌルには N、ゼロには Z、インデックスには I が用いられます）を通じて位置を示します。

### N 信号

N 信号を使用すると、位置カウンタをクリアするかスナップショットをとることができます。N チャンネルを継続的にモニタし、N チャンネル・イベントが検出されたエンコーダの位置をクリア、あるいはラッチするトリガを発するには、フラグ clr\_cont をセットします。あるいは、エンコーダの次の N チャンネル・イベントにのみ応答し、最初の N 信号イベント後のエンコーダ位置のクリアまたはラッチを自動的にディスエーブルすることもできます（フラグ clr\_once）。エンコーダがこの信号を供給するのは 1 回転ごとに一度であるため、この機能が必要となる可能性があります。

エンコーダがラッチされたイベントをチェックするには、次の手順に従います。

- **オプション 1** : ENC\_LATCH に変更がないか確認します。これは 0 から始まり、動作開始後最初に N イベントが生じたエンコーダ・カウンタを示します。連続的な回転のため、この値は増加/減少し、常に変化します。
- **オプション 2** : 割込み出力がアクティブであるかどうか確認し、アクティブな割込み出力に続くフラグのみを読み出します。GCONF レジスタのビット diag0\_nint\_step を用いて、割込みラインの DIAG0 ピンを設定する必要があります。

エンコーダによっては、A および B の極性の特定の設定によって N 信号を検証することが必要となります。これは、ENCMODE レジスタの pol\_A と pol\_B のフラグで制御できます。例えば、pol\_A と pol\_B の両方がセットされている場合、アクティブな N イベントが受け入れられるのは、A と B の両方のチャンネルがハイ極性になっている間のみです。

エンコーダ位置 ENC\_POS を次のアクティブな N イベントでクリアするには、clr\_enc\_x = 1 および clr\_once = 1、または clr\_cont = 1 に設定します。

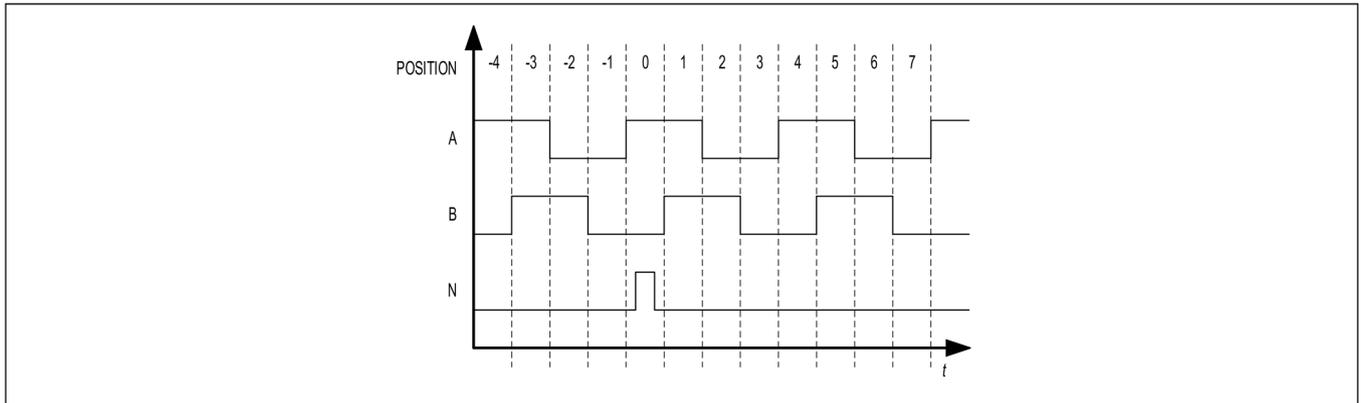


図 38. インクリメンタル・エンコーダの ABN 信号の概要

### エンコーダ・カウンタ X\_ENC

エンコーダ・カウンタ X\_ENC は、電流エンコーダを読み出し可能な位置に保持します。信号 A、B、N の処理に関する様々なモードは、様々なタイプのエンコーダに見られるアクティブ・ローおよびアクティブ・ハイの信号を考慮します。

### レジスタ ENC\_STATUS

レジスタ ENC\_STATUS は、N チャンネル信号でのエンコーダ・クリアのイベントに関するステータスを保持します。レジスタ ENC\_LATCH は、N 信号イベント時の実際のエンコーダ位置を常時保存します。

### エンコーダ定数 ENC\_CONST

エンコーダ定数 (またはエンコーダ係数) ENC\_CONST は、インクリメンタル・エンコーダの直交信号 AB の極性が変化すると、エンコーダ・カウンタに加算、あるいはエンコーダ・カウンタから減算されます。エンコーダ定数 ENC\_CONST は、符号付きの固定小数点数 (16.16) を表し、モーターとエンコーダ間の一般的な調整を容易なものにします。10 進数モードでは、下位 16 ビットは 0~9999 の数を表します。インクリメンタル・エンコーダを備えたステップング・モーターでは、固定小数点表示により、パラメータ化が非常に容易になります。また、機械的なギアのかみ合わせを容易に考慮できます。ENC\_CONST の符号を反転すると、モーターとエンコーダの方向に一致するようカウンタ方向を逆転できます。

例 :

- 1.0 のエンコーダ係数 :  $ENC\_CONST = 0x0001.0x0000 = FACTOR.FRACTION$
- -1.0 のエンコーダ係数 :  $ENC\_CONST = 0xFFFF.0x0000$ 。これは  $0x00010000$  の 2 の補数です。これは、 $(2^{16} - (FACTOR + 1)) \times (2^{16} - FRACTION)$  に等しくなります。
- 10 進数モードのエンコーダ係数 25.6 :  $00025.6000 = 0x0019.0x1770 = FACTOR.DECIMALS$  (DECIMALS = 小数点以下 4 桁)
- 10 進数モードのエンコーダ係数 -25.6 :  $(2^{16} - (25 + 1)) \times (10000 - 6000) = (2^{16} - 26) \times (4000) = 0xFFE6.0x0FA0$
- 負のエンコーダ定数は、次式を用いて計算できます。  $(2^{16} - (FACTOR + 1)) \times (10000 - DECIMALS)$

## モーターの分解能に合うエンコーダ設定

モーター・パラメータに対するエンコーダの設定例：

- USC = 256 マイクロステップ
- FSC = 200 フルステップ・モーター
- ファクタ = FSC × USC / エンコーダ分解能

表 26. 200 フルステップのモーターに対して 256 マイクロステップを使用するエンコーダの設定例

ENCODER RESOLUTION	REQUIRED ENCODER FACTOR	COMMENT
200	256	
360	142.2222 = 9320675.5555/216 = 1422222.2222/10000	No exact match possible!
500	102.4 = 6710886.4/216 = 1024000/10000	Exact match with decimal setting.
1000	51.2	Exact match with decimal setting.
1024	50	
4000	12.8	Exact match with decimal setting.
4096	12.5	
16384	3.125	

例：

エンコーダ定数レジスタは、10 進数モードで 51.2 にプログラムされます。そのため次のように設定します。

$$\text{ENC\_CONST} = 51 \times 2^{16} + 0.2 \times 10000$$

## リセット、ディスエーブル/停止、パワー・ダウン

### 緊急停止

ドライバは、負のアクティブ・イネーブル・ピン DRV\_ENN を備えており、すべてのパワー-MOSFET を確実にスイッチ・オフします。これにより、モーターをフリーホイール状態にすることができます。更に、これは、ソフトウェアと連携しない緊急停止が必要な場合には常に、安全なハードウェア機能となります。アプリケーションによっては、ドライバがアクティブな保持電流の状態または受動ブレーキング・モードの状態になることが要求されます。これを行うには、ピン ENCA がステップ・ディスエーブル機能として作用するようプログラミングします。GCONF のフラグ stop\_enable をセットするとこのオプションを有効化できます。ENCA がハイにプルアップされ、それがハイにとどまる限り、IHOLD、IHOLD\_DELAY、および StealthChop2 の停止オプション (StealthChop2 を使用している場合) を通じた設定に従い、モーターは突然に停止し、パワー・ダウン状態になります。

### 外部リセットおよびスリープ・モード

リセットおよびスリープ・モードは SLEEPN ピンで制御されます。

持続時間が 30µs を超える短パルスで SLEEPN に印加すると、チップ・リセットが生じます (診断出力でも認識可能)。

30µs 未満の非常に短いパルスはフィルタ除去され、動作には影響しません。

SLEEPN が GND に維持されている場合は、IC は低消費電力のスタンバイ状態 (スリープ・モード) になります。内部電源はすべてスイッチ・オフされます。

リセットとスタンバイのどちらの場合も、すべての内部レジスタの値と設定値はクリアされてそのデフォルト値に設定され、電源ブリッジはオフになります。

パワー・アップ後、あるいはスリープ・モードやリセット状態からの解放後、レジスタは再設定する必要があります。

IC の再設定の間、DRV\_ENN を用いてブリッジ・ドライバをディスエーブルしたままにすることを推奨します。

モーターからのエネルギー帰還によりチップが損傷する可能性があるため、モーター速度が高い場合には使用しないでください。

使用しない場合は、Vs または V<sub>CC10</sub> (これは高電圧ピンです) に接続します。

## 保護およびドライバ診断

TMC5241 のドライバでは、GND への短絡保護や低電圧検出など、フルセットの診断機能および保護機能が利用できます。開放負荷状態の検出では、モーター・コイルの接続が切断されているかどうかをテストできます。詳細については [DRV\\_STATUS レジスタ](#) の表を参照してください。

ステータス・フラグの他、TMC5241 では、チップ温度の測定および読出しや、モーター位相用の巻線温度へのフィードバックが可能です。システムの信頼性と回路全体の保護能力を向上するため、TMC5241 には過電圧コンパレータとトリガ出力 OV が内蔵され、電源電圧の過剰な上昇があった場合に外部スイッチを制御します。

### 過電流保護

過電流保護 (OCP) は、レール (電源電圧とグラウンド) への短絡と出力 (OUT1A、OUT2A、OUT1B、OUT、2B) 間の短絡からデバイスを保護します。

OCP 閾値は、選択したフルスケール電流範囲に依存します。また、それぞれの閾値については [電气的特性](#) の表を参照してください。

フルスケール・レンジは、DRV\_CONF レジスタの CURRENT\_RANGE パラメータで選択できます。

デグリッチ時間 (ブランキング時間) より長い間にわたって出力電流が OCP 閾値を上回る場合、OCP イベントが検出されます。

OCP イベントが検出されると、Hブリッジが直ちにディスエーブルされます。

短絡保護が3回行われると、フォルト・フラグ (DRV\_STATUS レジスタの s2ga、s2gb、ss 2vsa、s2vsb) がセットされブリッジが連続的にディスエーブルされます。

デバイスは依然としてアクティブで、設定やステータスの読出しが可能です。

電源ブリッジを再度イネーブルするには、DRV\_ENN ピンをローにして再度ハイにする必要があります。別のオプションは、CHOPCONF の TOFF を 0 にして電源ブリッジをディスエーブルし、TOFF > 0 にしてブリッジを再度イネーブルすることです。

### 過熱保護およびサーマル・シャットダウン

TMC5241 は過熱保護機能を内蔵しています。

ダイ温度が 165°C (代表値) を超過した場合、フォルト指示であるフォルト・フラグ (DRV\_STATUS の ot) がセットされ、ドライバはジャンクション温度が約 145°C (代表値) 未満になるまでスリーステートとなります。その後、ドライバは再度イネーブルされます。

更に、TMC5241 は、ADC ベースで設定可能な過熱事前警告レベルをサポートしています。これは、パラメータ OVERTEMPPREWARNING\_VTH を用いてレジスタ OTW\_OV\_VTH で設定できます。ADC がチップの平均温度を検出するのに対し、ドライバ段はこれよりはるかに高い温度となっている可能性があります。このことは、TMC5241 がサーマル・シャットダウン状態になる可能性があり、事前警告は低温に設定されている場合でもアサートされないことがある、ということを示すにすぎません。

熱は主としてモーター・ドライバ段で生成され、電圧が増加すると内部電圧レギュレータによって生成されます。ドライバ MOSFET が過熱するような重大な状況のほとんどは、GND への短絡保護機能を有効化していれば回避できます。多くのアプリケーションでは、過熱事前警告は異常な動作状況にあることを示すもので、これを用いて、ユーザ警告を発したり、モーター電流低減などの電力低減措置を開始したりできます。サーマル・シャットダウンは単なる緊急措置であり、シャットダウン・レベルまで温度が上昇することは、設計によって回避しなくてはなりません。

### 温度測定

TMC5241 は内部チップ温度およびモーター温度を測定する機能を備えています。

これらの診断機能は、アプリケーションにおいて、チップまたは PCB の温度およびモーター温度の時間変化をモニタし、システムの堅牢性を向上したり、予防的メンテナンスのための追加情報を取得したりするのに有用です。

### チップ温度の測定

過熱の事前警告や過熱フラグの他、チップ温度自体が、ADC\_TEMP レジスタの ADC\_TEMP パラメータを用いて決定できます。

°C 単位の最終温度は次の式で計算できます。

$$TEMP[°C] = \frac{ADC\_TEMP - 2038}{7.7}$$

## モーター温度の測定

PWM\_SCALE レジスタは、StealthChop2 動作での実際のデューティサイクルを示します。所定のモーター電流に対して、デューティサイクルはモーターの位相抵抗に依存します。

位相抵抗には温度依存性があるため、PWM\_SCALE を用いて実際のモーター温度を推定し、モーター温度の時間変化をモニタできます。この測定は、モーターの停止時または低速動作時に行うことを推奨します。

通常、モーター温度は短時間では変化しません。

## 過電圧保護と OV ピン

ステップング・モーター・アプリケーションでは、特にモーターが高速状態から短時間で減速する場合やモーターがストールする場合に、大きな過電圧が発生します。

この電圧は、ドライバ出力段で電源レールに帰還されます。

代表的な NEMA17 やそれより大型のモーター、および十分な慣性重量を持つ小型のモーターでは、帰還されるエネルギーは相当な量となるため、電源コンデンサや回路でのエネルギー消費では、電源をその制限値以内に保持するには不十分です。

ドライバや接続回路を保護するために、TMC5241 では、過電圧の検出および保護のための機構を備えています。

OV 出力を用いることで、NPN または MOSFET に電力抵抗（ブレーキ抵抗）を取り付けて過剰なエネルギーを抵抗に送り出すことができます。

トランジスタは約 3kHz~4kHz（クロック周波数に依存）でチョッピングされ、電源を制限範囲内に維持します。

電源電圧は内蔵 ADC で常にモニタされています。

所定のアプリケーションに対する電源電圧の上限レベルは、レジスタ OTW\_OV\_VTH のパラメータ OVERVOLTAGE\_VTH を用いて設定できます。

電源電圧の実際の ADC 値は、レジスタ ADC\_VSUPPLY\_AIN のパラメータ ADC\_VSUPPLY で読み出すことができます。

ADC の値を  $V_S$  に変換したり、あるいはその逆を行ったりするには、次の式を用います。

$$V_S = \text{ADC\_VSUPPLY} \times 17.6\text{mV}$$

OV 出力ピンは、過電圧モニタの実際の状態を示します。

ADC\_VSUPPLY が OVERVOLTAGE\_VTH 以上になるとすぐに、またその状態である限り、OV 出力ピンはスリーステートの高インピーダンス状態に変化します。

OV 出力ピンはオープンドレイン・ピンです。ブレーキ・チョッパー回路の例を次の図に示します。

デバイスをスリープ・モード（SLEEPN=LOW）にする場合には、特に注意を払う必要があります。この場合、OV はフロート状態です。

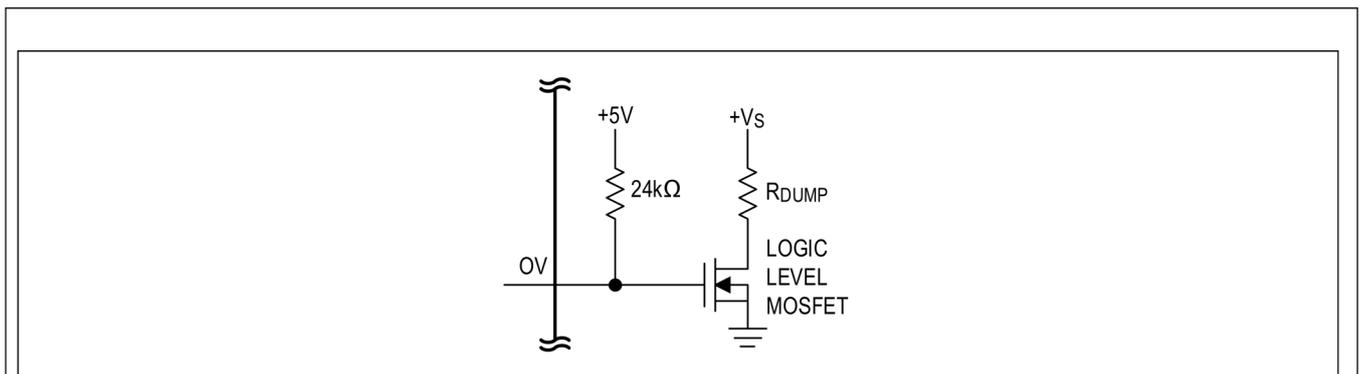


図 39. ブレーキ・チョッパー回路の例

## 短絡保護（GND への短絡および $V_S$ への短絡）

TMC5241 の電力段は、ハイサイド MOSFET を流れる電流を追加測定することで、回路の短絡状態に対し保護されます。大半の短絡状態は、システムのグラウンドに導電素子が接触した場合など、モーター・ケーブルの絶縁不良が原因です。短絡検出は、3 度再試行してからモーターをスイッチ・オフすることで、静電放電（ESD）などによる誤トリガから保護されます。

短絡状態が確実に検出されると、対応するドライバ・ブリッジがスイッチ・オフされ、s2ga フラグまたは s2gb フラグがセットされます。モーターを再始動するには、ドライバをディスエーブルした後再度イネーブルするというユーザの介入が必要です。短絡イベントは様々であり、また、外部コンポーネントの複合的なネットワークが影響を受ける可能性があるため、GND 短絡保護では、必ずしも起こり得るすべての短絡イベントからシステムと電力段を保護できるとは限らない点に注意が必要です。そのため、基本的に短絡を回避する必要があります。

フルスケール電流設定に応じて、様々な過電流保護閾値でローサイドの短絡保護がトリガします。

表 27. フルスケール電流設定に基づく過電流保護閾値

FULL-SCALE CURRENT SETTING (BITS)	OVERCURRENT PROTECTION THRESHOLD [A]
10 (and 11)	5.0
01	3.33
00	1.67

### 開放負荷診断

断線したケーブルは、コネクタが確実に差し込まれていない場合などのように、よく見られるシステム故障の原因です。TMC5241 は、目的のモーター・コイル電流に達したかどうかをチェックすることで、開放負荷条件を検出します。したがって、低電圧条件、高モーター速度設定、あるいは短絡および過熱条件が、開放負荷フラグをトリガする可能性があります。モーター停止時には、結果的にコイルの電流がゼロであるため、開放負荷を測定できません。

断線したコイル接続を確実に検出するには、SpreadCycle で動作させ、低速あるいは通常のモーター速度動作のみを用い、選択したマイクロステップ分解能の最低 4 倍 (4 フルステップ) の単一方向動作の後に開放負荷フラグをチェックします。ただし、ola フラグと olb フラグは、単に情報を伝えるだけの性質しかなく、ドライバの何らかの動作の原因になるわけではありません。

### 低電圧ロックアウト保護

TMC5241 は、 $V_S$ 、 $V_{CC10}$ 、チャージ・ポンプに対する UVLO 保護機能を備えています。

$V_S$  の UVLO 条件は 4.05V (最大値) 未満でトリガされます。

$V_{CC10}$  の UVLO 条件は 1.95V (最大値) 未満でトリガされます。

チャージ・ポンプの UVLO 条件は、コンデンサ値が不適切である場合など、チャージ・ポンプのエラー条件が生じた場合にトリガされません。

$V_S$  の UVLO 条件は、レジスタ GSTAT のフラグ `vm_uvlo` で読み出せます。このフラグは「書き込んでクリア」フラグです。これをクリアするにはアクティブに 1 に設定する必要があります。

$V_{CC10}$  の UVLO 時には、IC との通信は不可能で、ドライバはディスエーブルされます。DIAG0 ピンはアクティブ・ロー (オープンドレイン) です。

### ESD 保護

チップには、ピンごとに ESD 保護機能が内蔵されています。

TMC5241 のモーター位相出力ピンは、正電圧電源 ( $V_S$  ピン) に  $1\mu\text{F}$  以上のバイパス・コンデンサを用いたアプリケーションで、最大 8kV の人体モデル (HBM) に対して保護されています。

これはモーターのホット・プラグに対する保護ではありません。

### 外部アナログ入力 AIN のモニタリング

TMC5241 は、外部アナログ入力 AIN を備えており、内部 ADC で連続的にサンプリングされます。

ADC のサンプリング値は、レジスタ `ADC_VSUPPLY_AIN` のパラメータ `ADC_AIN` で読み出せます。

ADC の値を  $V_{AIN}$  に変換したり、あるいはその逆を行ったりするには、次の式を用います。

$$V_{AIN} = \text{ADC\_AIN} \times 305.2\mu\text{V}$$

AIN 入力を用いることで、システム・レベルの状態を表し、システムの状態に更に帰還を提供することのできる、外部アナログ変数やパラメータをモニタできます。

## クロック発振器とクロック入力

### 内部クロックの使用

内部クロック発振器を使用するには、CLK 入力ピンを IC 付近で直接 GND に接続します。内部クロックは、12.5MHz（代表値）の周波数で動作しています。

### 外部クロックの使用

外部クロックを使用できる場合、最適な性能を得るために推奨する周波数は 8MHz～20MHz です。

クロック信号の必要な最小および最大のデューティサイクルは、[電気的特性](#)のセクションで定義されています。

特に、20MHz に近いクロック周波数では、クロックのデューティサイクル条件を満たす必要があります。

高いクロック周波数を用いる場合、クロック・ソースにはノイズのない安定した CMOS 出力ロジック・レベルと急峻なスロープを供給するものを使用してください。

外部クロック入力は、外部クロックが CLK ピンに供給されると直ちに有効化されます。

レジスタ IOIN の読出しビット `ext_clk` は、現在どのクロック源を使用しているかをフィードバックします（1=外部クロック）。

外部クロックが使用できなくなった場合、あるいはスイッチ・オフされた場合、内部クロックに継ぎ目なく切り替わり、ドライバの損傷を自動的に防止します。

## クイック設定ガイド

このガイドは、最初のレジスタ設定を決定してドライバ調整のための測定と決定をできるだけ少なくすることのできる、実用的なツールを提供することを目的としています。ここではすべての先進的機能とオプションをカバーしているわけではなく、モーターをスムーズに動作させるための基本的な機能セットに重点を置いています。モーターが動作するようになった後、その他の機能やより高度な機能をより詳細に調べることができます。1 つのモーター・コイルに電流プローブを配置することは、最適な設定を見出す上で良い助けとなります。

## 電流設定

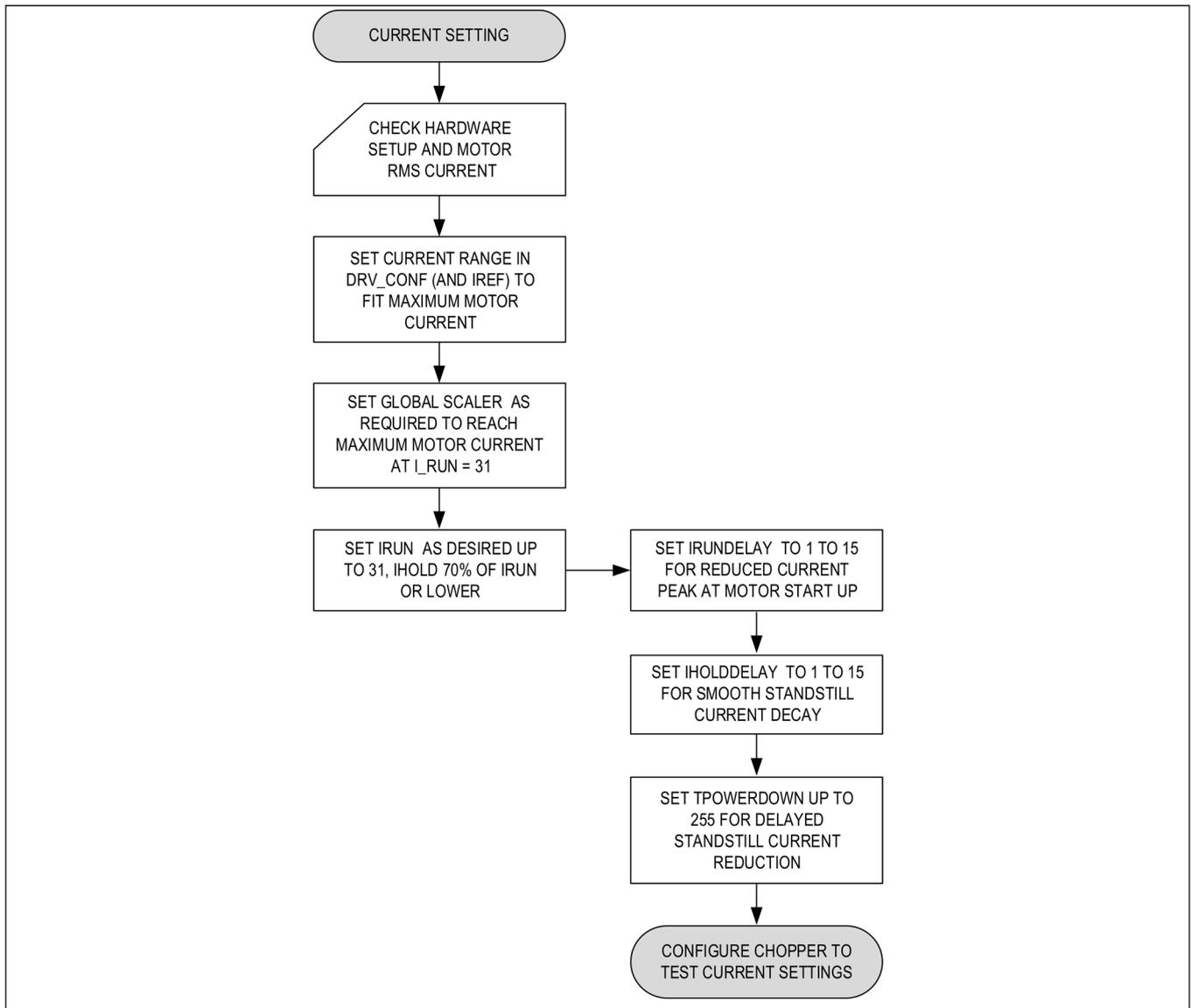


図 40. 電流設定のためのクイック設定ガイド

StealthChop2 の設定

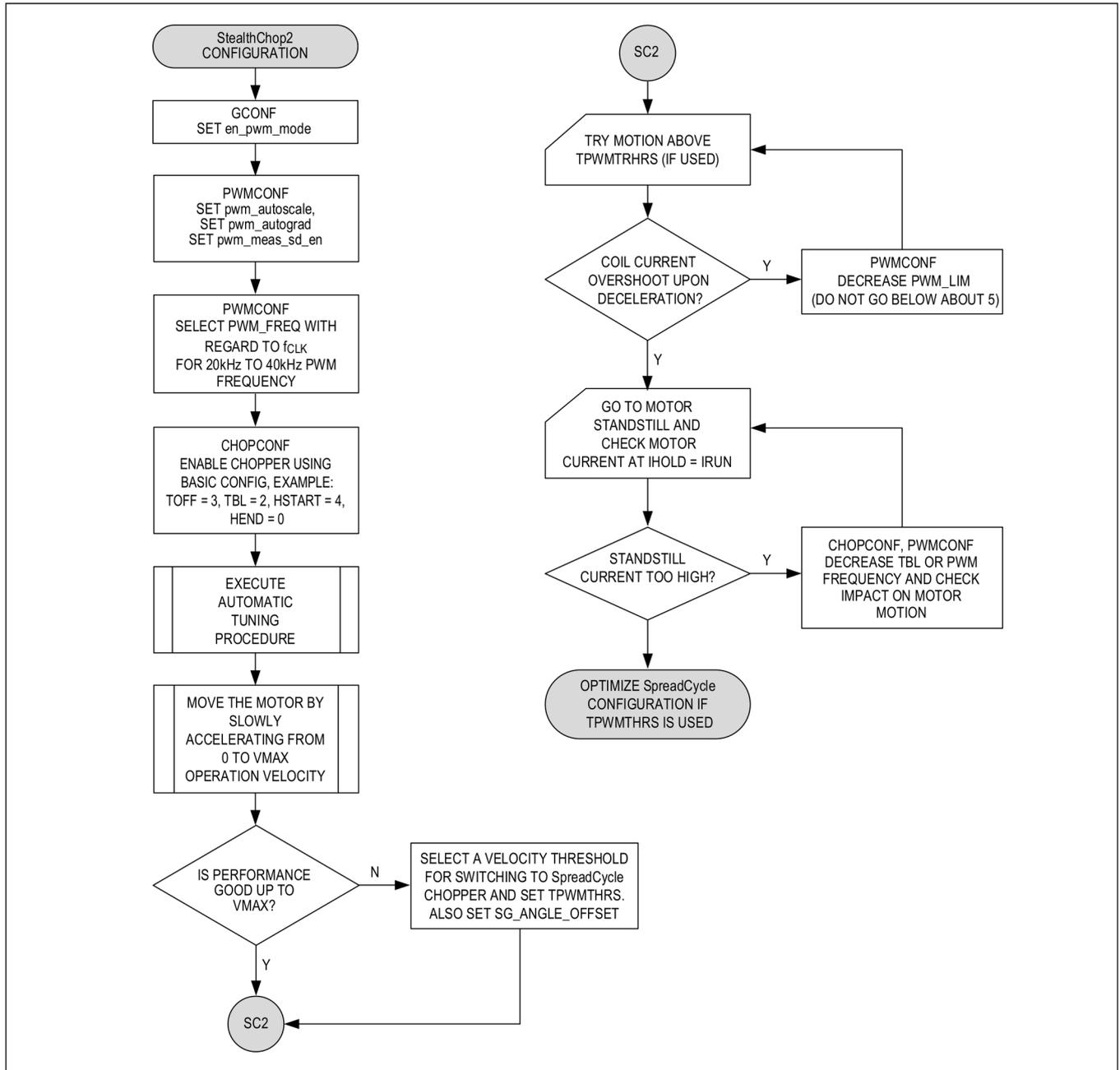


図 41. StealthChop2 のクイック設定ガイド

## SpreadCycle 設定

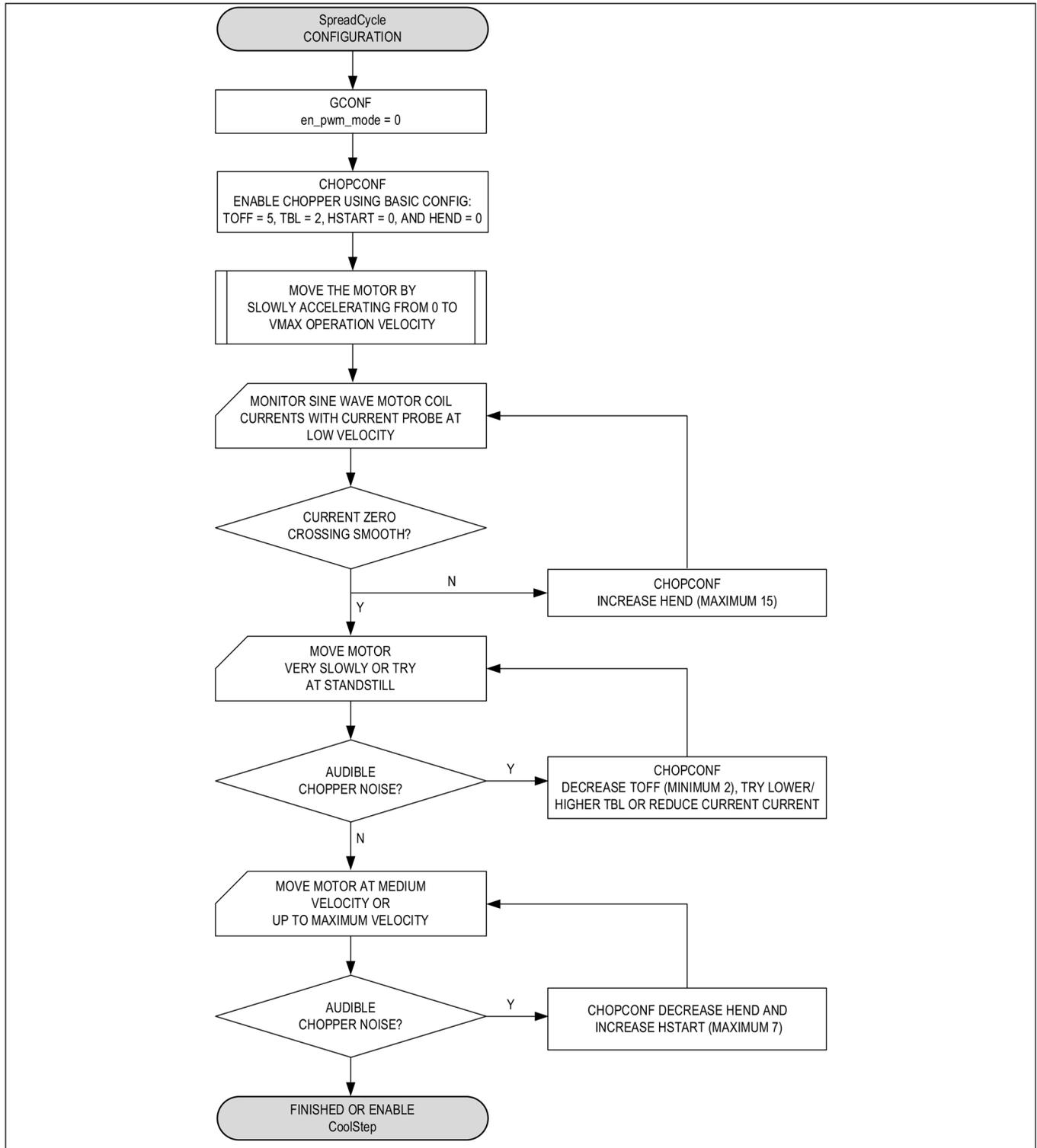


図 42. SpreadCycle のクイック設定ガイド

CoolStep を StealthChop2 と組み合わせてイネーブル

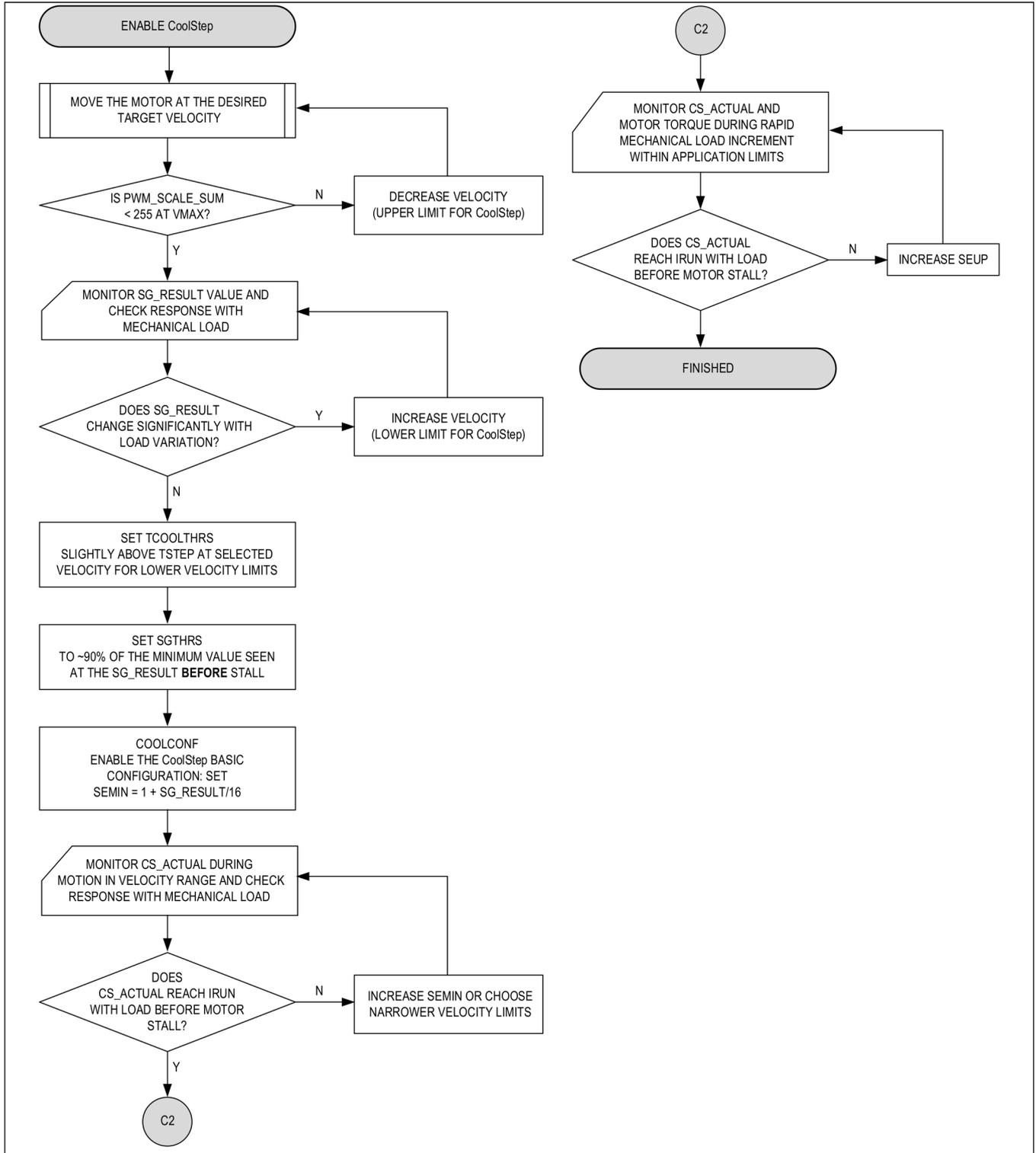


図 43. StealthChop2 との組み合わせによる CoolStep のクイック設定ガイド

CoolStep を SpreadCycle と組み合わせてイネーブル

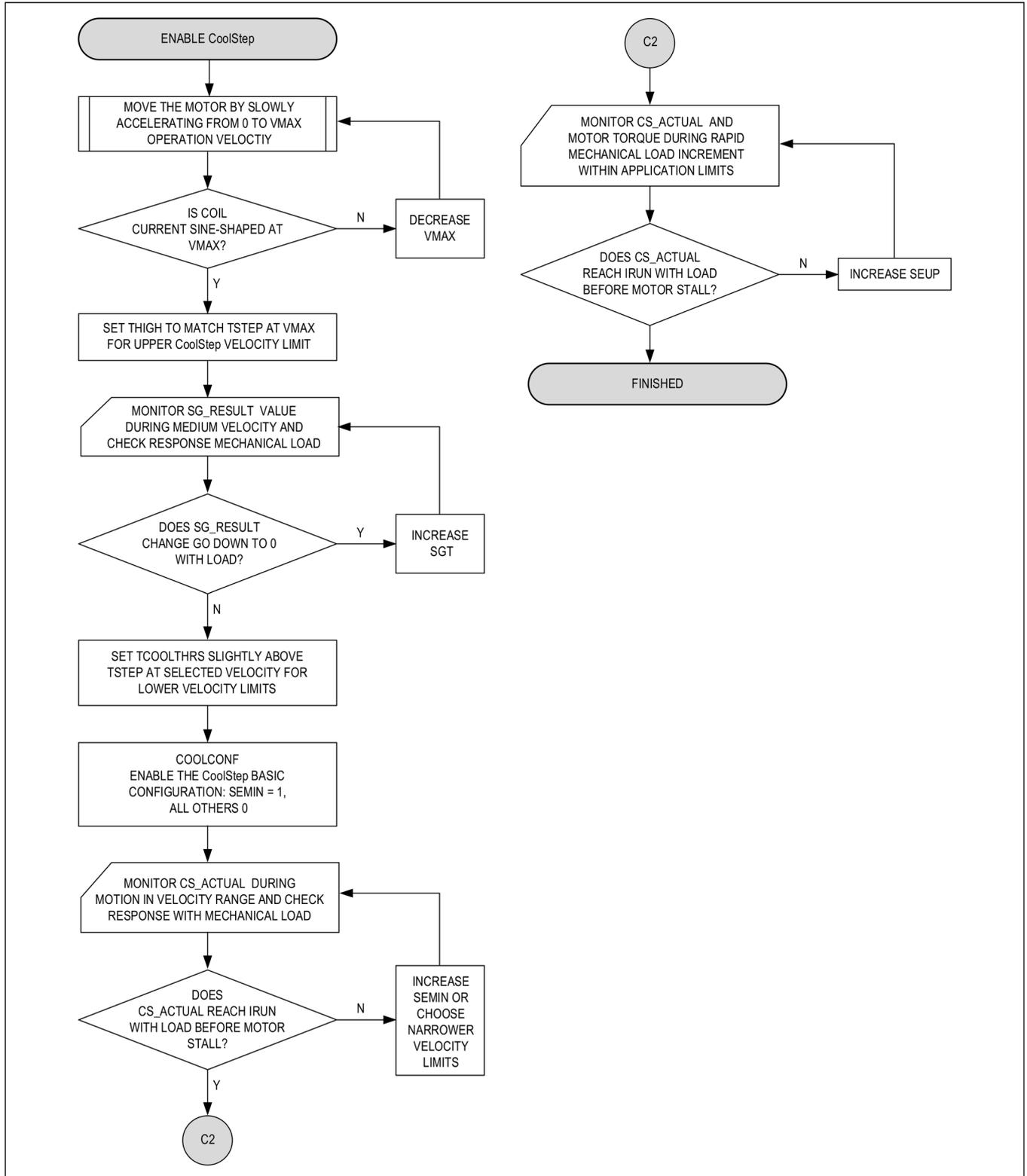


図 44. SpreadCycle との組み合わせによる CoolStep のクイック設定ガイド

モーション・コントローラを用いたモーターの作動

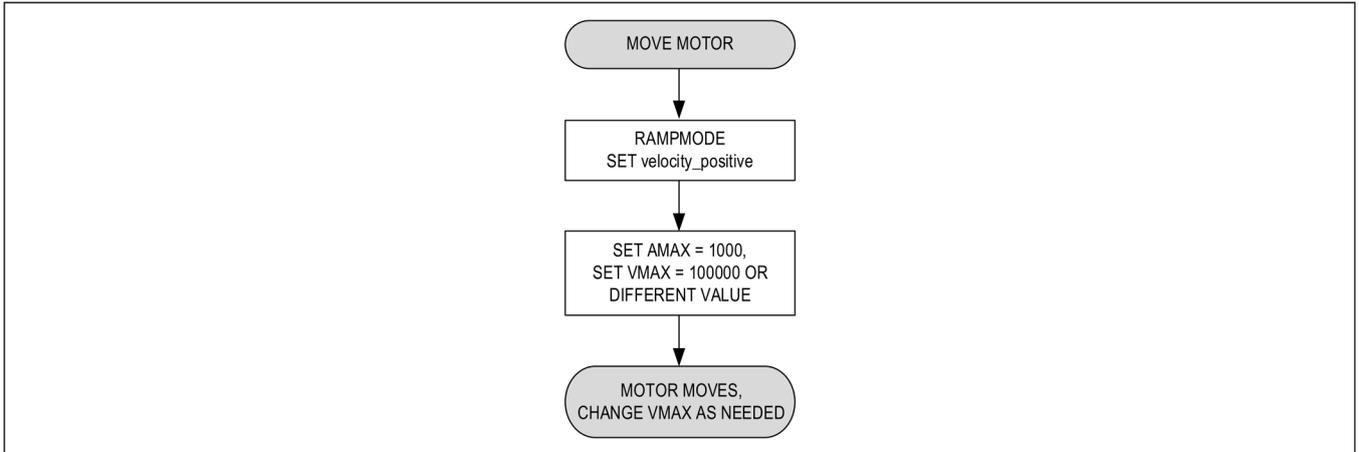


図 45. 速度モードでモーターを作動させるためのクイック設定ガイド

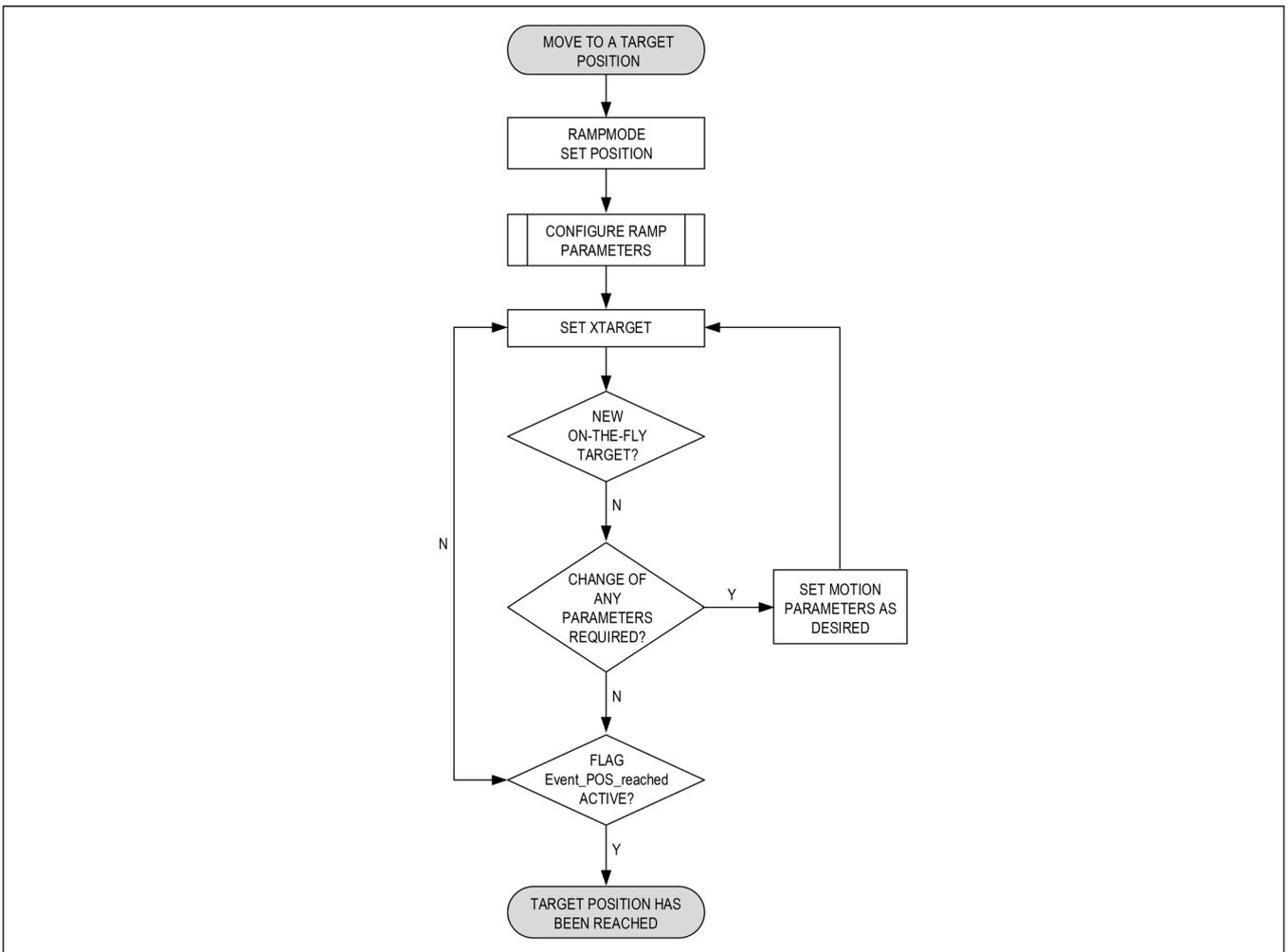


図 46. 目標ポジションへモーターを作動させるためのクイック設定ガイド

## モーション・ランプ・パラメータの設定

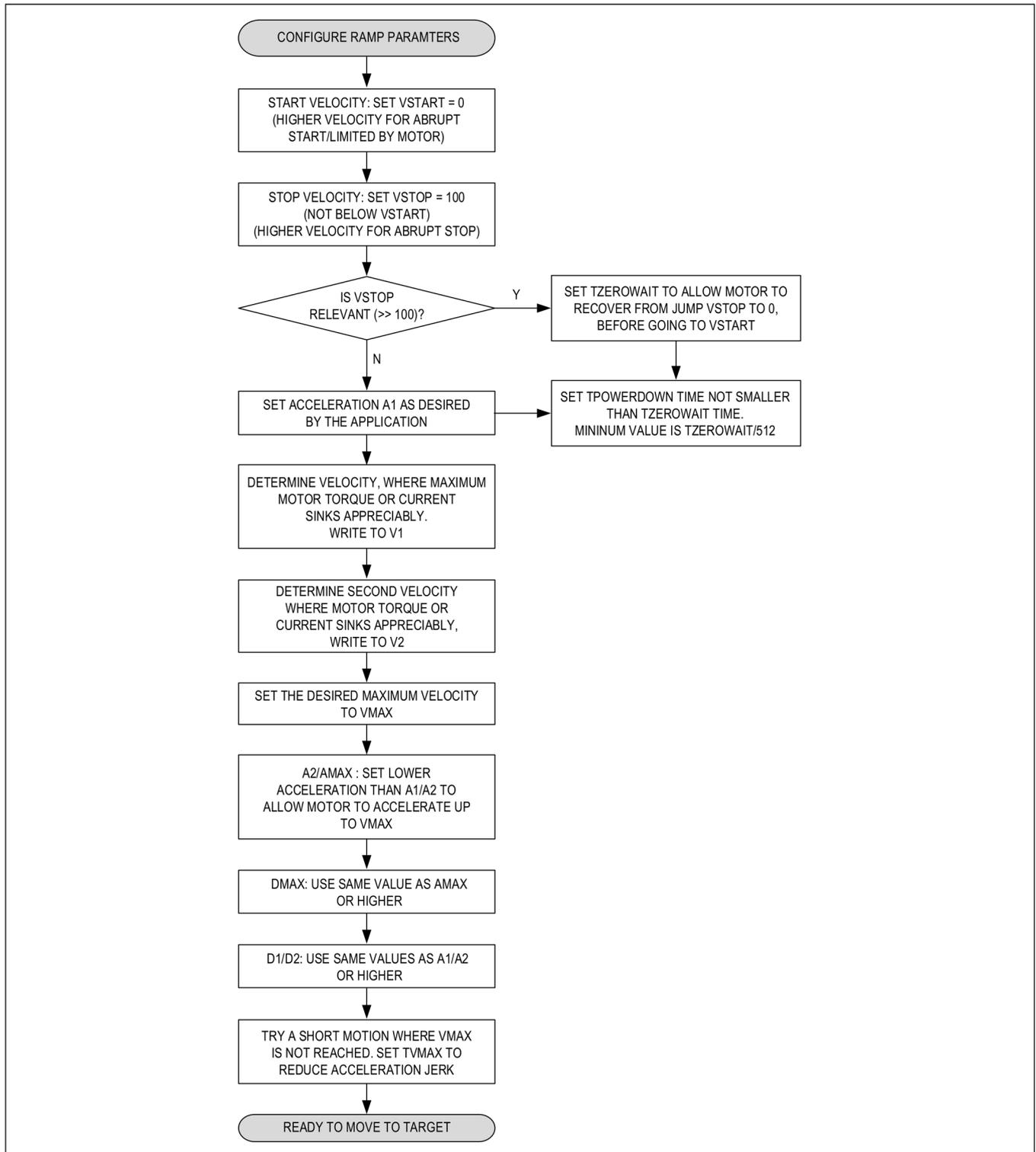


図 47. モーション・ランプ・パラメータを設定するためのクイック設定ガイド

## DcStep 動作のイネーブル

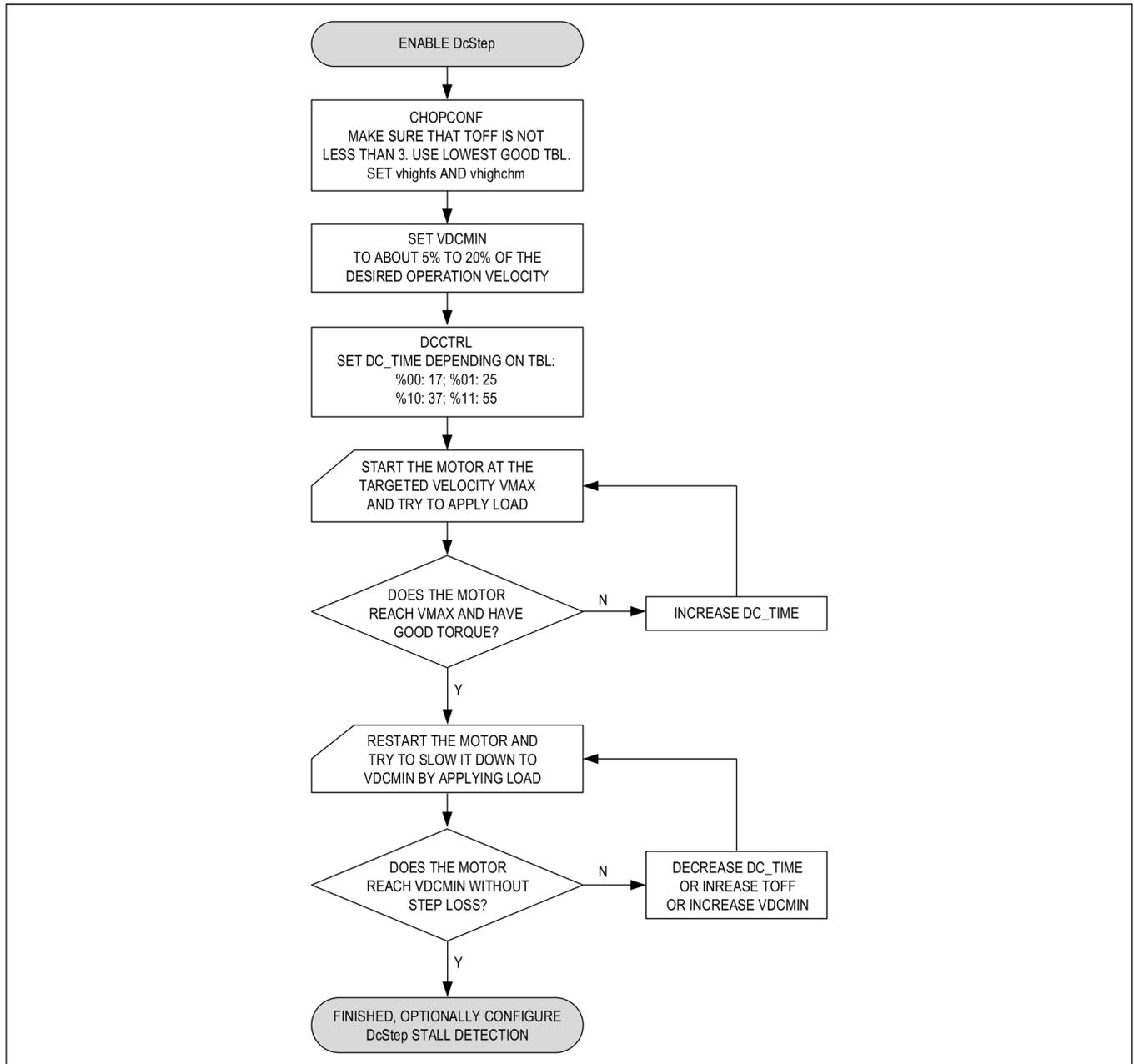


図 48. DcStep のクイック設定ガイド

## DcStep でストール検出を使用

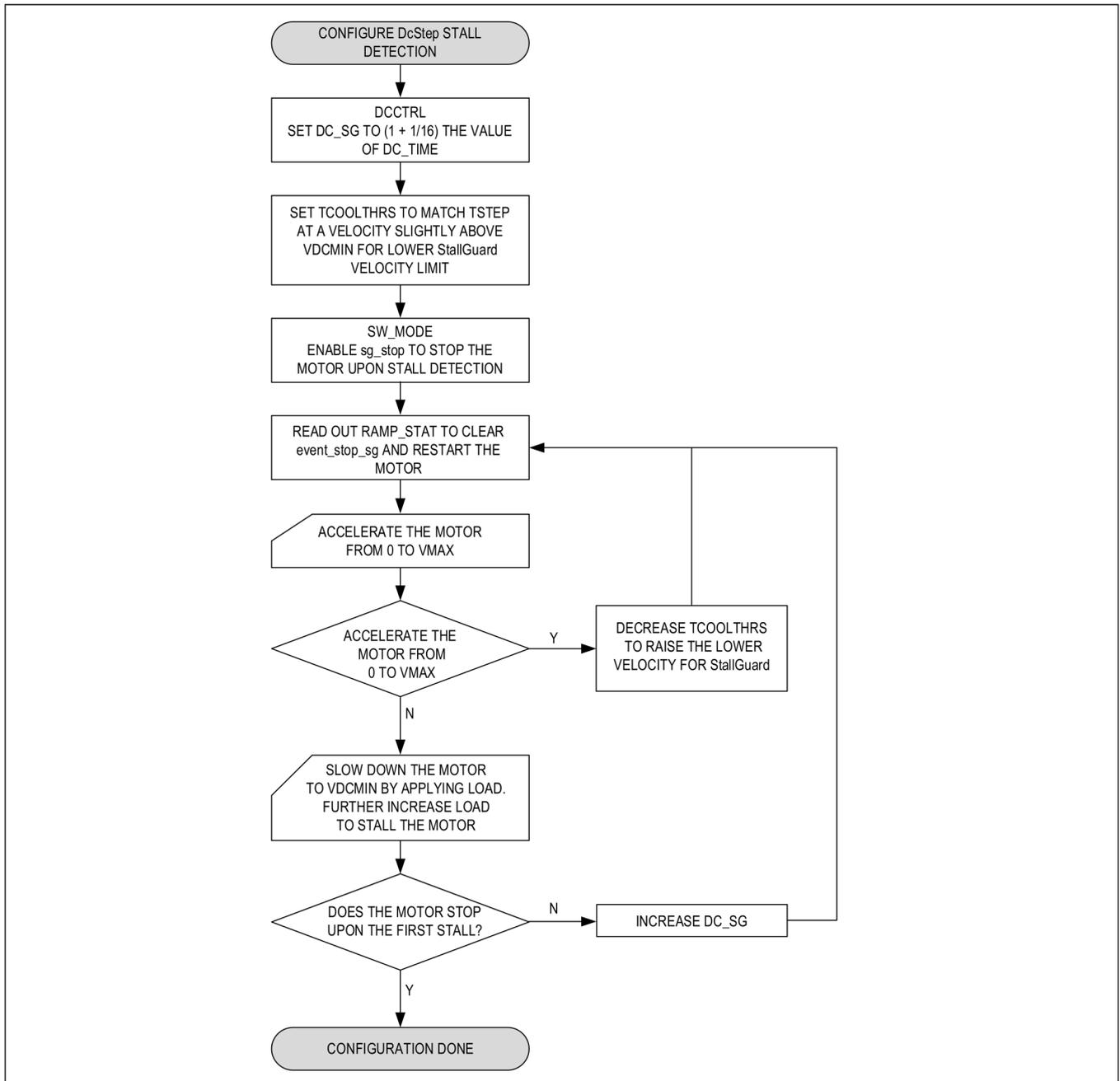


図 49. DcStep でストール検出を使用するためのクイック設定ガイド

## 全般的なレジスタ・マッピングおよびレジスタ情報

このセクションでは、レジスタ・マップに関する全般的な情報を示します。

すべてのレジスタおよびその内容に関する詳細は、[レジスタ・マップ](#)のセクションに示します。

- すべてのレジスタは、特に指定のない限り、パワー・アップ時に 0 にリセットされます。
- 書き込みアクセス時にはアドレス Addr に 0x80 を追加してください。

表 28. レジスタ・マップの概要

レジスタ	説明
全般的な設定レジスタ	これらのレジスタは以下の内容を含みます： <ul style="list-style-type: none"> <li>• グローバル設定</li> <li>• グローバル・ステータス・フラグ</li> <li>• インターフェース設定</li> <li>• I/O 信号設定</li> </ul>
ランプ・ジェネレータのモーション・コントロール・レジスタ・セット	このレジスタ・セットは以下を目的とするレジスタで構成されます： <ul style="list-style-type: none"> <li>• ランプ・モードの選択</li> <li>• 速度の選択</li> <li>• ホーミング</li> <li>• 加減速</li> <li>• 目標への位置制御</li> <li>• リファレンス・スイッチおよび StallGuard2 イベントの設定</li> <li>• ランプおよびリファレンス・スイッチのステータス</li> </ul>
速度依存ドライバ機能制御レジスタ・セット	このレジスタ・セットは以下を目的とするレジスタで構成されます： <ul style="list-style-type: none"> <li>• ドライバ電流コントロール</li> <li>• CoolStep 動作閾値の設定</li> <li>• 様々なチョッパー・モード用閾値の設定</li> <li>• DcStep 動作閾値の設定</li> </ul>
直接モード・レジスタ	このレジスタ・グループは、直接コイル電流制御モードで用いるレジスタで構成されます。
エンコーダ・レジスタ・セット	エンコーダ・レジスタ・セットは、適切な ABN エンコーダ動作に必要なすべてのレジスタで構成されます。
ADC レジスタ	このレジスタ・グループは、内部 ADC の制御と読出しを行うレジスタで構成されます。
モーター・ドライバ・レジスタ・セット	このレジスタ・セットは以下を目的とするレジスタで構成されます： <ul style="list-style-type: none"> <li>• マイクロステップ・テーブルおよびカウンタの設定／読出し</li> <li>• チョッパーおよびドライバの設定</li> <li>• CoolStep および StallGuard の設定</li> <li>• DcStep の設定</li> <li>• StallGuard の値およびドライバ・エラー・フラグの読出し</li> </ul>

## 標準アプリケーション回路

## 標準的なアプリケーション回路

標準的なアプリケーション回路では、最低限のセットの追加コンポーネントを使用します。電源をフィルタにより除去するには、低 ESR の電解コンデンサを使用します。コンデンサは、チョッパー動作によって生じる電流リップルを処理する必要があります。最適な性能を得るために、 $V_S$  に最小  $100\mu\text{F}$  のコンデンサを用いることを推奨します。電源コンデンサの電流リップルは、電源の内部抵抗とケーブル長にも依存します。 $V_{CC\_IO}$  は、低ドロップアウト  $3.3\text{V}$  レギュレータなどの外部電源から供給する必要があります。

すべてのフィルタ・コンデンサは、関連する IC ピンのできるだけ近くに配置してください。すべての GND 接続には、べた膜の共通グラウンド・プレーンを用います。 $V_{DDIV8}$  フィルタリング・コンデンサは、 $V_{DDIV8}$  ピンに直接接続します。

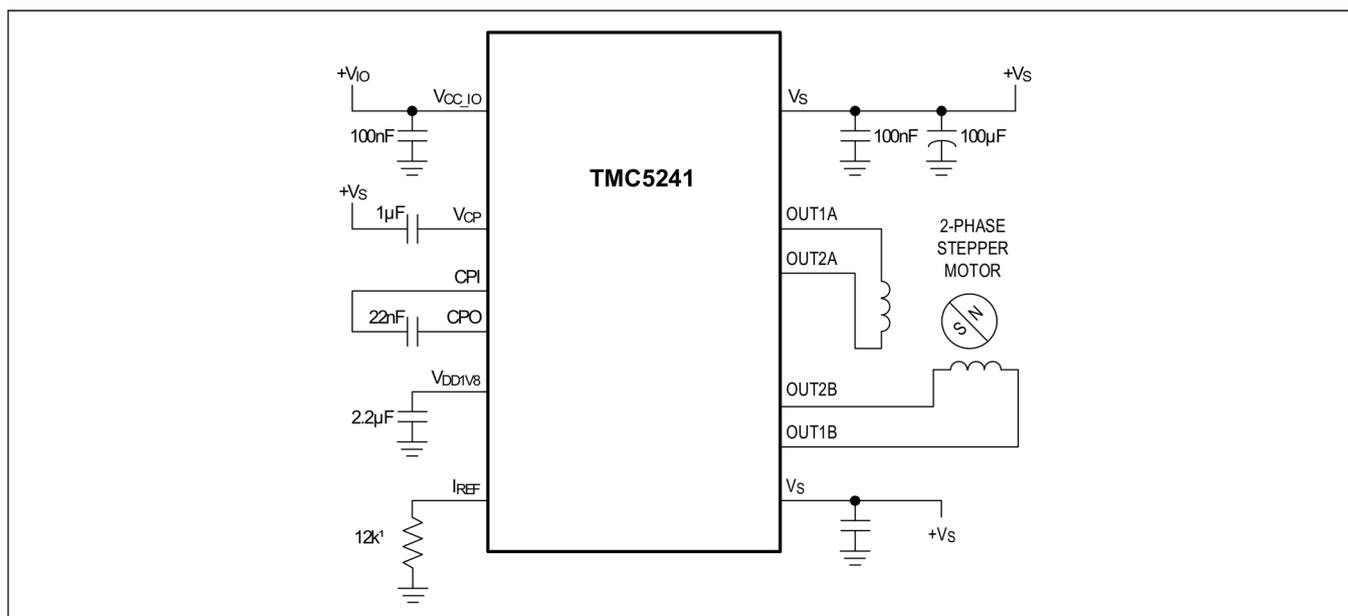


図 50. 標準的なアプリケーション回路

## 大モーター電流

大モーター電流での動作時、MOSFET のスイッチオン抵抗によるドライバの消費電力が、ドライバの温度を著しく上昇させます。大きなデューティサイクルで動作している場合、この消費電力により、PCB の冷却機構も過熱します。これがドライバの更なる温度上昇につながります。温度が約  $100^\circ\text{C}$  上昇すると、MOSFET の抵抗が約 50% 増加します。これは MOSFET スイッチの代表的な挙動です。そのため、デューティサイクルが大きく高負荷条件の場合は、特に、周囲温度の上昇が見込まれる際には、熱特性に特別な注意を払う必要があります。熱特性については [パッケージ情報](#) のセクションを、レイアウト例についてはオンライン評価用キットの情報を参照してください。

おおよその目安として、PCB 設計の熱的性質は、 $24\text{V}$  を超える電源電圧の場合、またはモーター電流が長時間にわたり  $1.5\text{A}_{\text{RMS}}$  以上となった場合に重要になる可能性があります。抵抗による消費電力はモーター電流の 2 乗に比例して増加することに注意してください。一方、このことは、モーター電流をわずかに減らすだけでも、大幅に放熱を減らしエネルギーを節約できることを意味します。

## 勾配コントロール

ドライバは、 $100\text{V}/\mu\text{s}$  から  $800\text{V}/\mu\text{s}$  の範囲で 4 段階の勾配コントロールを選択できます。高速の勾配はドライバの消費電力を低減し、低速の勾配は電磁放射を低減します。一般的には、消費電力を低く抑えてデバイスの発熱を低減することが重要なため、最も高速の勾配設定が多くアプリケーションに適しています。特に、モーター電流が  $1\text{A}$  を超え、電源電圧が  $24\text{V}$  を超えるアプリケーションでは、最高速の勾配設定のメリットが得られます。電磁放射が重要となる場合は、勾配を低くすることを検討します。ただし、デバイスの温度を注意深くモニタしてください。 $100\text{V}/\mu\text{s}$  の勾配設定は、 $14\text{V}$  以下の低電源電圧アプリケーションでのみ使用してください。

## ドライバ保護および EME 回路

アプリケーションによっては、モーター動作または外部の影響による ESD イベントに対応する必要があります。ドライバ・チップは ESD 回路を内蔵していますが、動作時に発生する ESD イベントは、エネルギーによっては、リセットの原因となったり、モーター・ドライバの破壊を引き起こしたりする可能性があります。特に、プラスチックのハウジングやベルト・ドライブ・システムは、数 kV にも及ぶ重大な ESD イベントを引き起こす傾向にあります。モーター自体を PCB グラウンドに取り付けたり、導電性プラスチック部品を用いたりするなど、取り付け部品にはすべて導電体で構成されたものを用いることで、ESD イベントを防止するのがベスト・プラクティスです。更に、ドライバは、ESD イベントまたは、モーター・コネクタ端子に高電圧大電流が供給される原因にもなるモーターの活線挿抜に対して、ある程度までは保護可能です。

簡単な方法は、ドライバ出力にコンデンサを用いて ESD イベントによる  $dV/dt$  を低減することです。コンデンサの値が大きいほど、ESD 抑制に関する利点は大きくなりますが、各チョッパー・サイクルに流れる電流増加の原因となり、それによって特に高電源電圧時にドライバの消費電力が大きくなります。以下に示す値は例です（100pF~1nF の範囲で変動する場合があります）。また、コンデンサは、アプリケーションの PCB 回路のデジタル部分から注入される高周波ノイズを緩和し、それによって電磁放射を減少させます。

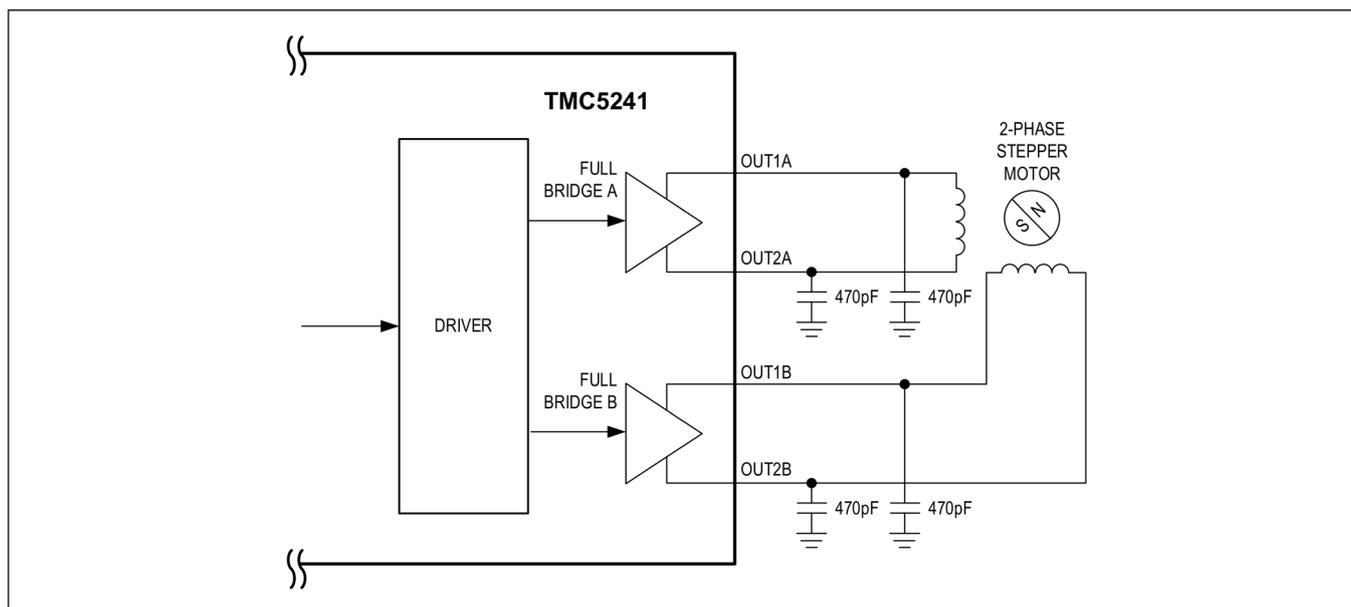


図 51. 簡単な ESD 対策

より精巧な方法は、LC フィルタを用いてドライバ出力をモーター・コネクタとデカップリングすることです。コイル端子間のバリスタ V1 および V2 は、活線接続に起因するコイルの過電圧をなくします。オプションで、バリスタ (V1A、V1B、V2A、および V2B) により、すべての出力を ESD 電圧から保護します。バリスタは電源電圧の定格に合わせてください。SMD (表面実装) インダクタはモーターの全コイル電流を導通するため、それに応じたものを選択する必要があります。

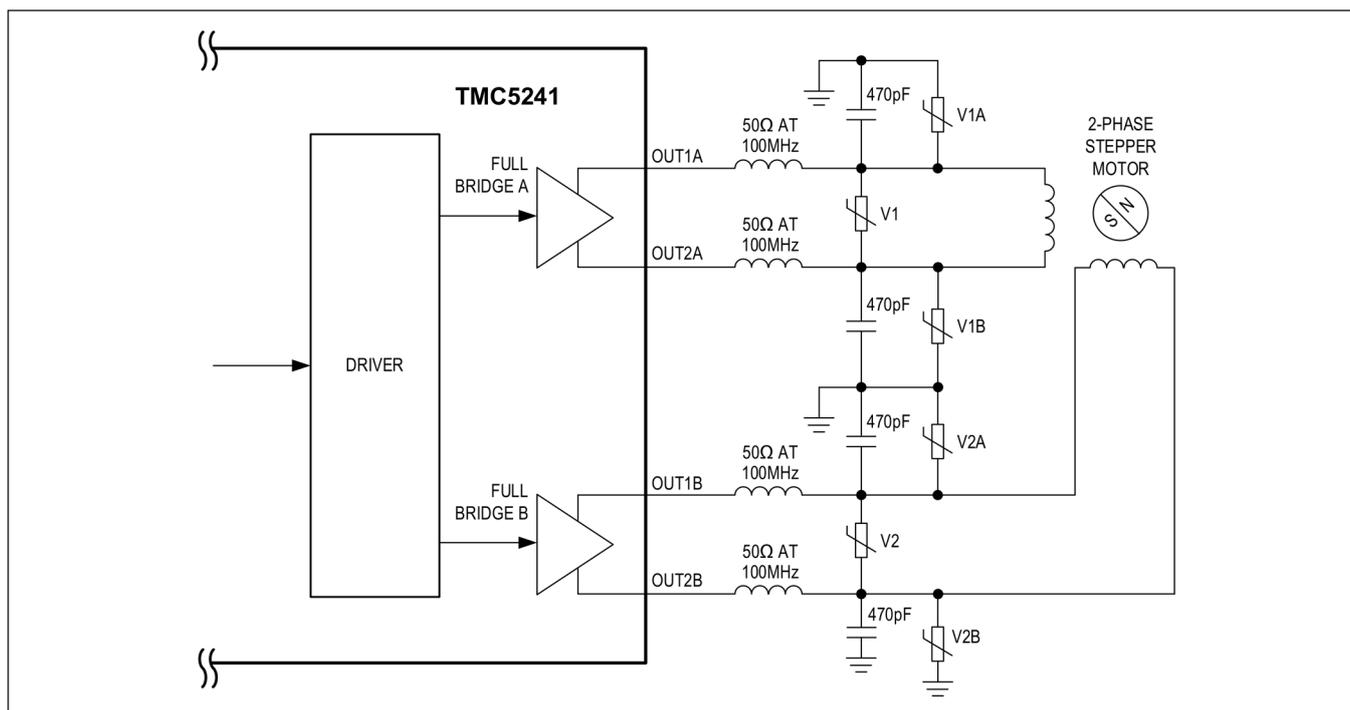


図 52. モーター出力保護の強化

## レジスタ・マップ

## GCR レジスタの概要

GCR ブロックに関連するすべてのレジスタを示します。

ADDRESS AND NAME	FIELDS								MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)
0x00 <b>GCONF</b>									
					length_step_pulse				direct_mode
	stop_enable	small_hysteresis	diag1_poscomp_pushpull	diag0_int_pushpull					diag1_nposcomp_dir
	diag0_nint_step			shaft		en_pwm_mode	fast_standstill		
0x01 <b>GSTAT</b>									
				vm_uvlo	register_reset	uv_cp	drv_err	reset	
0x02 <b>IFCNT</b>									
	IFCNT								
0x03 <b>NODECONF</b>									
						SENDDelay			
	NODEADDR								
0x04 <b>IOIN</b>									
		EXT_CLK	EXT_RES_DET	OUTPUT					
		UART_EN	ENCN	DRV_ENN	ENCA	ENCB	REFR	REFL	
0x05 <b>X_COMPARE</b>	X_COMPARE[31:24]								
	X_COMPARE[23:16]								
	X_COMPARE[15:8]								
	X_COMPARE[7:0]								
0x06 <b>X_COMPARE_REPEAT</b>									
	X_COMPARE_REPEAT[23:16]								
	X_COMPARE_REPEAT[15:8]								
	X_COMPARE_REPEAT[7:0]								

ADDRESS AND NAME	FIELDS				MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)			
0x07 <b>DIAG_CONF</b>					diag1_ overvoltage	diag1_ev_n_ deviation	diag1_ev_ pos_ reached	diag1_ev_ stop_sg
	diag1_ev_ stop_ref	diag1_ xcomp	diag1_dir	diag1_step	diag1_index	diag1_stall	diag1_otpw	diag1_error
					diag0_ overvoltage	diag0_ev_n_ deviation	diag0_ev_ pos_ reached	diag0_ev_ stop_sg
	diag0_ev_ stop_ref	diag0_ xcomp	diag0_dir	diag0_step	diag0_index	diag0_stall	diag0_otpw	diag0_error
0x0A <b>DRV_CONF</b>								
			SLOPE_CONTROL				CURRENT_RANGE	
0x0B <b>GLOBAL_SCALER</b>								
	GLOBALSCALER							
0x10 <b>IHOLD_IRUN</b>					IRUNDELAY			
					IHOLDDELAY			
					IRUN			
					IHOLD			
0x11 <b>TPOWERDOWN</b>								
	TPOWERDOWN							
0x12 <b>TSTEP</b>					TSTEP[19:16]			
	TSTEP[15:8]							
	TSTEP[7:0]							
0x13 <b>TPWMTHRS</b>					TPWMTHRS[19:16]			
	TPWMTHRS[15:8]							
	TPWMTHRS[7:0]							
0x14 <b>TCOOLTHRS</b>					TCOOLTHRS[19:16]			
	TCOOLTHRS[15:8]							
	TCOOLTHRS[7:0]							

ADDRESS AND NAME	FIELDS								MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)							
0x15 <b>THIGH</b>																
									THIGH[19:16]							
									THIGH[15:8]							
									THIGH[7:0]							
0x20 <b>RAMPMODE</b>																
									RAMPMODE							
0x21 <b>XACTUAL</b>									XACTUAL[31:24]							
									XACTUAL[23:16]							
									XACTUAL[15:8]							
									XACTUAL[7:0]							
0x22 <b>VACTUAL</b>																
									VACTUAL[23:16]							
									VACTUAL[15:8]							
									VACTUAL[7:0]							
0x23 <b>VSTART</b>																
									VSTART[17:16]							
									VSTART[15:8]							
									VSTART[7:0]							
0x24 <b>A1</b>																
									A1[17:16]							
									A1[15:8]							
									A1[7:0]							
0x25 <b>V1</b>																
									V1[19:16]							
									V1[15:8]							
									V1[7:0]							
0x26 <b>AMAX</b>																
									AMAX[17:16]							
									AMAX[15:8]							
									AMAX[7:0]							
0x27 <b>VMAX</b>																
									VMAX[22:16]							
									VMAX[15:8]							
									VMAX[7:0]							

ADDRESS AND NAME	FIELDS								MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)																				
0x28 <b>DMAX</b>																													
																	DMAX[17:16]												
																	DMAX[15:8]												
																	DMAX[7:0]												
0x29 <b>TVMAX</b>																													
																	TVMAX[15:8]												
																	TVMAX[7:0]												
0x2A <b>D1</b>																													
																	D1[17:16]												
																	D1[15:8]												
																	D1[7:0]												
0x2B <b>VSTOP</b>																													
																	VSTOP[17:16]												
																	VSTOP[15:8]												
																	VSTOP[7:0]												
0x2C <b>TZEROWAIT</b>																													
																	TZEROWAIT[15:8]												
																	TZEROWAIT[7:0]												
0x2D <b>XTARGET</b>																													
																	XTARGET[31:24]												
																	XTARGET[23:16]												
																	XTARGET[15:8]												
0x2E <b>V2</b>																													
																	V2[19:16]												
																	V2[15:8]												
																	V2[7:0]												
0x2F <b>A2</b>																													
																	A2[17:16]												
																	A2[15:8]												
																	A2[7:0]												
0x30 <b>D2</b>																													
																	D2[17:16]												
																	D2[15:8]												
																	D2[7:0]												

ADDRESS AND NAME	FIELDS								MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)								
0x33 <b>VDCMIN</b>																	
									VDCMIN[14:8]								
									VDCMIN[7:0]								
									reserved								
0x34 <b>SW_MODE</b>																	
			virtual_stop_enc	en_virtual_stop_r	en_virtual_stop_l	en_softstop	sg_stop	en_latch_encoder	latch_r_inactive								
	latch_r_active	latch_l_inactive	latch_l_active	swap_lr	pol_stop_r	pol_stop_l	stop_r_enable	stop_l_enable									
0x35 <b>RAMP_STAT</b>																	
	status_virtual_stop_r	status_virtual_stop_l	status_sg	second_move	t_zerowait_active	vzero	position_reached	velocity_reached									
	event_pos_reached	event_stop_sg	event_stop_r	event_stop_l	status_latch_r	status_latch_l	status_stop_r	status_stop_l									
0x36 <b>XLATCH</b>									XLATCH[31:24]								
									XLATCH[23:16]								
									XLATCH[15:8]								
									XLATCH[7:0]								
0x38 <b>ENCMODE</b>																	
					pos_neg_edge	clr_once	clr_cont	ignore_AB	enc_sel_decimal	latch_x_act	clr_enc_x						
			pos_neg_edge	clr_once	clr_cont	ignore_AB	pol_N	pol_B	pol_A								
0x39 <b>X_ENC</b>									X_ENC[31:24]								
									X_ENC[23:16]								
									X_ENC[15:8]								
									X_ENC[7:0]								
0x3A <b>ENC_CONST</b>									ENC_CONST[31:24]								
									ENC_CONST[23:16]								
									ENC_CONST[15:8]								
									ENC_CONST[7:0]								
0x3B <b>ENC_STATUS</b>																	
									deviation_warn								
								n_event									

ADDRESS AND NAME	FIELDS	MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)
0x3C <b>ENC_LATCH</b>		ENC_LATCH[31:24]
		ENC_LATCH[23:16]
		ENC_LATCH[15:8]
		ENC_LATCH[7:0]
0x3D <b>ENC_DEVIATION</b>		
		ENC_DEVIATION[19:16]
		ENC_DEVIATION[15:8]
		ENC_DEVIATION[7:0]
0x3E <b>VIRTUAL_STOP_L</b>		VIRTUAL_STOP_L[31:24]
		VIRTUAL_STOP_L[23:16]
		VIRTUAL_STOP_L[15:8]
		VIRTUAL_STOP_L[7:0]
0x3F <b>VIRTUAL_STOP_R</b>		VIRTUAL_STOP_R[31:24]
		VIRTUAL_STOP_R[23:16]
		VIRTUAL_STOP_R[15:8]
		VIRTUAL_STOP_R[7:0]
0x50 <b>ADC_VSUPPLY_AIN</b>		ADC_AIN[12:8]
		ADC_AIN[7:0]
		ADC_VSUPPLY[12:8]
		ADC_VSUPPLY[7:0]
0x51 <b>ADC_TEMP</b>		
		ADC_TEMP[12:8]
		ADC_TEMP[7:0]
0x52 <b>OTW_OV_VTH</b>		OVERTEMPPREWARNING_VTH[12:8]
		OVERTEMPPREWARNING_VTH[7:0]
		OVERVOLTAGE_VTH[12:8]
		OVERVOLTAGE_VTH[7:0]
0x60 <b>MSLUT_0</b>		MSLUT_0[31:24]
		MSLUT_0[23:16]
		MSLUT_0[15:8]
		MSLUT_0[7:0]
0x61 <b>MSLUT_1</b>		MSLUT_1[31:24]
		MSLUT_1[23:16]
		MSLUT_1[15:8]
		MSLUT_1[7:0]

ADDRESS AND NAME	FIELDS				MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)			
0x62 <b>MSLUT_2</b>					MSLUT_2[31:24]			
					MSLUT_2[23:16]			
					MSLUT_2[15:8]			
					MSLUT_2[7:0]			
0x63 <b>MSLUT_3</b>					MSLUT_3[31:24]			
					MSLUT_3[23:16]			
					MSLUT_3[15:8]			
					MSLUT_3[7:0]			
0x64 <b>MSLUT_4</b>					MSLUT_4[31:24]			
					MSLUT_4[23:16]			
					MSLUT_4[15:8]			
					MSLUT_4[7:0]			
0x65 <b>MSLUT_5</b>					MSLUT_5[31:24]			
					MSLUT_5[23:16]			
					MSLUT_5[15:8]			
					MSLUT_5[7:0]			
0x66 <b>MSLUT_6</b>					MSLUT_6[31:24]			
					MSLUT_6[23:16]			
					MSLUT_6[15:8]			
					MSLUT_6[7:0]			
0x67 <b>MSLUT_7</b>					MSLUT_7[31:24]			
					MSLUT_7[23:16]			
					MSLUT_7[15:8]			
					MSLUT_7[7:0]			
0x68 <b>MSLUTSEL</b>					X3			
					X2			
					X1			
		W3	W2	W1	W0			
0x69 <b>MSLUTSTART</b>					OFFSET_SIN90			
					START_SIN90			
					START_SIN			
0x6A <b>MSCNT</b>								
								MSCNT[9:8]
					MSCNT[7:0]			

ADDRESS AND NAME	FIELDS								MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)							
0x6B <b>MSCURACT</b>									CUR_A[8]							
									CUR_A[7:0]							
									CUR_B[8]							
									CUR_B[7:0]							
0x6C <b>CHOPCONF</b>			reserved		intpol		MRES									
	TPFD				vhighchm		vhighfs				TBL[1]					
	TBL[0:0]		chm		disfdcc		fd3		HEND_OFFSET[3:1]							
	HEND_OFFSET[0:0]		HSTRT_TFD210				TOFF									
0x6D <b>COOLCONF</b>									sflt							
									sgt							
	seimin		sedn				semax									
	seup				semin											
0x6E <b>DCCTRL</b>									DC_SG							
									DC_TIME[9:8]							
									DC_TIME[7:0]							
0x6F <b>DRV_STATUS</b>	stst	olb	ola	s2gb		s2ga		otpw	ot	stallguard						
									CS_ACTUAL							
	fsactive	stealth		s2vsb	s2vsa				SG_RESULT[9:8]							
									SG_RESULT[7:0]							
0x70 <b>PWMCONF</b>	PWM_LIM				PWM_REG											
	pwm_dis_reg_stst	pwm_meas_sd_enable	FREEWHEEL			pwm_autograd	pwm_autoscale	PWM_FREQ								
									PWM_GRAD							
									PWM_OFS							
0x71 <b>PWM_SCALE</b>									PWM_SCALE_AUTO[8]							
									PWM_SCALE_AUTO[7:0]							
									PWM_SCALE_SUM[9:8]							
0x72 <b>PWM_AUTO</b>									PWM_SCALE_SUM[7:0]							
									PWM_GRAD_AUTO							
									PWM_OFS_AUTO							

ADDRESS AND NAME	FIELDS								MSB (TOP LEFT) TO LSB (BOTTOM RIGHT)								
0x74 SG4_CONF																	
																sg_angle_offset	sg4_filt_en
	SG4_THRS																
0x75 SG4_RESULT																	
																	SG4_RESULT[9:8]
	SG4_RESULT[7:0]																
0x76 SG4_IND	SG4_IND_3																
	SG4_IND_2																
	SG4_IND_1																
	SG4_IND_0																

**0x00 : GCONF**

グローバル設定フラグ

ビットおよび名称	タイプおよびリセット	説明
[20:17] length_step_pulse	RW 0x0	DIAG0またはDIAG1で選択されている場合、ステップ・パルス長を設定します。 length_step_pulse = 0 : STEP出力はステップごとに切り替わり、その結果、ステップ周波数の半分になります。 length_step_pulse = 1...15 : クロック・サイクル数で表したSTEPピンのハイ時間。
	0: HALF	出力はステップごとに切り替わる
	1: CLK_1	各ステップ・パルスの長さは1クロック・サイクル
	2: CLK_2	各ステップ・パルスの長さは2クロック・サイクル
	3: CLK_3	各ステップ・パルスの長さは3クロック・サイクル
	4: CLK_4	各ステップ・パルスの長さは4クロック・サイクル
	5: CLK_5	各ステップ・パルスの長さは5クロック・サイクル
	6: CLK_6	各ステップ・パルスの長さは6クロック・サイクル
	7: CLK_7	各ステップ・パルスの長さは7クロック・サイクル
	8: CLK_8	各ステップ・パルスの長さは8クロック・サイクル
	9: CLK_9	各ステップ・パルスの長さは9クロック・サイクル
	10: CLK_10	各ステップ・パルスの長さは10クロック・サイクル
	11: CLK_11	各ステップ・パルスの長さは11クロック・サイクル
	12: CLK_12	各ステップ・パルスの長さは12クロック・サイクル
	13: CLK_13	各ステップ・パルスの長さは13クロック・サイクル
	14: CLK_14	各ステップ・パルスの長さは14クロック・サイクル
	15: CLK_15	各ステップ・パルスの長さは15クロック・サイクル

ビットおよび名称	タイプおよびリセット	説明
[16] direct_mode	RW 0x0	シリアル・インターフェースを通じて直接モーター位相電流制御をイネーブル。
	0: NORMAL_MODE 1: DIRECT_MODE	通常動作 モーター・コイル電流および極性をシリアル・インターフェースから直接プログラム。レジスタXTARGET (0x2D) は符号付きコイルA電流 (ビット8..0) とコイルB電流 (ビット24..16) を指定します。このモードでは、電流はIHOLDの設定値によって変化します。StealthChop2での速度ベースによる電流レギュレーションは、このモードでは使用できません。StealthChop2での自動電流レギュレーションはステッピング・モーターが低速の場合にのみ作動します。
[15] stop_enable	RW 0x0	モーター・ハード・ストップ機能を有効化。
	0: NORMAL 1: EM_STOP	通常動作 緊急停止：ハイに接続された場合、ENCAはシーケンサを停止します (シーケンサによってステップは一切実行されず、モーターは停止状態になります)。
[14] small_hysteresis	RW 0x0	
	0: HYST_1_16 1: HYST_1_32	ステップ周波数比較のためのヒステリシスは1/16 ステップ周波数比較のためのヒステリシスは1/32
[13] diag1_poscomp_pushpull	RW 0x0	DIAG1の出力タイプ設定。
	0: OPEN_DRAIN 1: PUSH_PULL	DIAG1はオープン・コレクタ出力 (アクティブ・ロー) DIAG1のプッシュプル出力をイネーブル (アクティブ・ハイ)
[12] diag0_int_pushpull	RW 0x0	DIAG0の出力タイプ設定。
	0: OPEN_DRAIN 1: PUSH_PULL	DIAG0はオープン・コレクタ出力 (アクティブ・ロー) DIAG0のプッシュプル出力をイネーブル (アクティブ・ハイ)
[8] diag1_nposcomp_dir	RW 0x0	UARTを使用しない場合のDIAG1の出力設定。
	0: XCOMP 1: DIR	DIAG1はポジションの比較信号を出力 DIAG1を外部STEP/DIRドライバのDIR出力としてイネーブル。
[7] diag0_nint_step	RW 0x0	DIAG0の出力設定。
	0: INTERRUPT 1: STEP	DIAG0は割り込み信号を出力 DIAG0を外部STEP/DIRドライバのSTEP出力としてイネーブル (周波数は半分、デュアル・エッジでトリガ、またはlength_step_pulse != 0の場合はその周波数)。
[4] shaft	RW 0x0	モーターの方向/方向符号の変更
	0: DIR 1: DIR_INV	モーターはデフォルトの方向 モーターは反転方向
[2] en_pwm_mode	RW 0x0	StealthChop2モードを有効化
	0: SPREADCYCLE 1: STEALTHCHOP	StealthChop2不使用 StealthChop2電圧PWMモードを有効化 (速度閾値に依存)。停止状態およびIHOLD = 公称IRUN電流の場合にのみ、オフからオンに切り替わりません。
[1] fast_standstill	RW 0x0	ステップ実行から停止検出までのタイムアウト
	0: STST_2_20 1: STST_2_18	通常的时间：2 <sup>20</sup> クロック 短い時間：2 <sup>18</sup> クロック

**0x01 : GSTAT**

グローバル・ステータス・フラグ

(各フラグをクリアするには「1」のビットを再書き込みします)

ビットおよび名称	タイプおよびリセット	説明
[4] vm_uvlo	RW, W1C 0x1	1: 最後のリセット後にVMの低電圧が発生。(ヒント: 最初のブートアップ後にアクティブになります。デバイスが動作しているときにフォルトを検出するにはブートアップ後にフラグをクリアします)。
	0: OPERATIONAL 1: VM_UVLO_DET	通常動作 最後のリセット後にVMの低電圧を検出。
[3] register_reset	RW, W1C 0x1	ヒント: 最初のブートアップ後にアクティブになります。デバイスが動作しているときにレジスタ・マップのリセットを検出するにはブートアップ後にフラグをクリアします。
	0: OPERATIONAL 1: REG_RES_DET	通常動作 レジスタ・マップがリセットされたことを示します。すべてのレジスタはリセット値にクリアされます。
[2] uv_cp	RW, W1C 0x1	チャージ・ポンプ低電圧条件フラグ。(ヒント: 最初のブートアップ後にアクティブになります。デバイスが動作しているときにフォルトを検出するにはブートアップ後にフラグをクリアします)。#type=COW
	0: OPERATIONAL 1: UV_CP_DET	通常動作 チャージ・ポンプが低電圧状態であることを示します。低電圧の間、ドライバは無効化されます。このフラグは通知のためにラッチされます。
[1] drv_err	RW, W1C 0x0	ドライバ・エラー・フラグ、#type = COW
	0: OPERATIONAL 1: DRV_DIS_DET	通常動作 過熱または短絡検出によりドライバがシャットダウンしていることを示します。詳細については、 <a href="#">DRV_STATUS</a> を参照してください。このフラグは、温度が制限値未満に戻った場合にのみクリアされます。
[0] reset	RW, W1C 0x1	リセット・フラグ (ヒント: 最初のブートアップ後にアクティブになります。動作中にデバイスがリセットされたことを検出するにはブートアップ後にフラグをクリアします)。#type=COW
	0: NO_RESET 1: RESET_DET	通常動作 ICがリセットされたことを示します。

**0x02 : IFCNT**

インターフェース伝送カウンタ。

ビットおよび名称	タイプおよびリセット	説明
[7:0] IFCNT	R, unsigned 0x00	インターフェース伝送カウンタ。このレジスタは、UARTインターフェースの書き込みアクセスが正常に行われるたびにインクリメントします。これを読み出すことでシリアル伝送にデータ喪失がないことを確認できます。読出しアクセスでは内容は変化しません。SPI動作ではディスエーブルされます。カウンタは255の後は0に戻ります。

## 0x03 : NODECONF

ビットおよび名称	タイプおよびリセット	説明
[11:8] SENDDelay	RW 0x0	SWUARTで設定可能な、読み出し要求後の応答遅延。
	0: BIT_T_8	8ビット分の時間（複数のノードは使用できません）
	2: BIT_T_24	3 × 8ビット分の時間
	4: BIT_T_40	5 × 8ビット分の時間
	6: BIT_T_56	7 × 8ビット分の時間
	8: BIT_T_72	9 × 8ビット分の時間
	10: BIT_T_88	11 × 8ビット分の時間
12: BIT_T_104	13 × 8ビット分の時間	
14: BIT_T_120	15 × 8ビット分の時間	
[7:0] NODEADDR	RW, unsigned 0x00	NODEADDR : これらの8ビットは、UARTインターフェース用のデバイス・アドレスを設定します。アドレスはSDI、SCK、CSNIによる定義に従って1ずつ7までインクリメントします。 CSN, SCK, SDI 000: +0 001: +1 010: +2 011: +3 100: +4 101: +5 110: +6 111: +7 範囲: 0~254 (254より先へはインクリメントしません)

## 0x04 : IOIN

利用可能なすべての入力ピンの状態を読み出し、最上位バイトでICリビジョンを返します

ビットおよび名称	タイプおよびリセット	説明
[18:16] SILICON_RV	R, unsigned 0x0	シリコン・リビジョン番号
[14] EXT_CLK	R 0x0	0: クロック信号（12.5MHz）を生成するために内部発振器が用いられます。 1: クロック信号を生成するために外部発振器が用いられます。
	0: INT_OSC 1: EXT_OSC	12.5MHzの内部発振器を使用。 外部クロックを検出して使用。
[13] EXT_RES_DET	R 0x0	外部リファレンス抵抗の検出。外部抵抗を通じてI <sub>REF</sub> 電流が供給され、デバイスが動作可能かどうかをチェックします。
	0: REF_RES_FAULT 1: REF_RES_DET	抵抗は未検出。 通常動作
[12] OUTPUT	RW 0x1	UART_ENピンを通じてUARTがイネーブルされている場合のSDOピンの出力極性。主な目的は、NAO（次のアドレス出力）信号としてSDOを用い、複数のICのチェーン・アドレス指定を行うことです。 <b>注意</b> ：単線チェーンで次のICへのNAOとして用いる場合、リセット値は1です。
	0: NAO_LOW 1: NAO_HI	SDO/NAOはロー・ロジック・レベルにセット。 SDO/NAOはハイ・ロジック・レベルにセット。

ビットおよび名称	タイプおよびリセット	説明
[6] UART_EN	R 0x0	1 = UARTインターフェースがイネーブル
	0: LOW 1: HIGH	UART_ENはロジック・レベル・ロー UART_ENはロジック・レベル・ハイ
[5] ENCN	R 0x0	Nチャンネルの状態
	0: LOW 1: HIGH	ENC_Nはロジック・レベル・ロー ENC_Nはロジック・レベル・ハイ
[4] DRV_ENN	R 0x0	ドライバのディスエーブル/イネーブル状態。
	0: LOW 1: HIGH	DRV_ENNはロジック・レベル・ロー DRV_ENNはロジック・レベル・ハイ
[3] ENCA	R 0x0	Aチャンネルの状態
	0: LOW 1: HIGH	ENCAはロジック・レベル・ロー ENCAはロジック・レベル・ハイ
[2] ENCB	R 0x0	Bチャンネルの状態
	0: LOW 1: HIGH	ENCBはロジック・レベル・ロー ENCBはロジック・レベル・ハイ
[1] REFR	R 0x0	ピンREFRの状態。
	0: LOW 1: HIGH	REFRピンはロー・ロジック・レベル REFRピンはハイ・ロジック・レベル
[0] REFL	R 0x0	ピンREFLの状態。
	0: LOW 1: HIGH	REFLピンはロー・ロジック・レベル REFLピンはハイ・ロジック・レベル

## 0x05 : X\_COMPARE

ビットおよび名称	タイプおよびリセット	説明
[31:0] X_COMPARE	RW, signed 0xFFFFFFFF	<p>モーション・コントローラ・ポジション・ストローブ用のポジション比較レジスタ。X_COMPAREは絶対ポジションです。 ポジション・パルスは出力SWP_DIAG1で使用できます。</p> <p>XACTUAL = X_COMPARE : 出力信号PP (ポジション・パルス) がハイになります。 ポジションが不一致の場合はロー状態に戻ります。</p> <p>X_COMPARE_REPEATが1より大きい場合、X_COMPAREは、periodicポジション・ストローブ・トリガ出力のポジション・リファレンスです。</p>

## 0x06 : X\_COMPARE\_REPEAT

ビットおよび名称	タイプおよびリセット	説明
[23:0] X_COMPARE_REPEAT	RW, unsigned 0x000000	<p>このレジスタは、相対距離を（MRESの設定に基づき）マイクロステップ単位で定義します。</p> <p>これを1より大きな値に設定した場合、複数のX_COMPARE_REPEATマイクロステップが刻まれるたびに、ポジション比較パルスが発生します。</p> <p>それによって、X_COMPAREレジスタは、正または負の方向に向かって行われたX_COMPARE_REPEATステップのモジュール計算を行うためのベース位置を定義します。</p> <p>X_COMPAREポジションに達すると、X_COMPAREはX_COMPARE_REPEATだけインクリメント/デクリメントされます。</p>

## 0x07 : DIAG\_CONF

ビットおよび名称	タイプおよびリセット	説明
[27] diag1_overvoltage	RW 0x0	OVピンをDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でOVを有効化
[26] diag1_ev_n_deviation	RW 0x0	Nイベントおよび偏差警告をDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でエンコーダ・イベントを有効化
[25] diag1_ev_pos_reached	RW 0x0	RAMPSTAT_EVENT_POS_REACHEDをDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でEVENT_POS_REACHEDを有効化
[24] diag1_ev_stop_sg	RW 0x0	RAMPSTAT_EVENT_SG_STOPをDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でSG_STOPを有効化
[23] diag1_ev_stop_ref	RW 0x0	RAMPSTAT_EVENT_STOPL    RAMPSTAT_EVENT_STOPRをDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でSTOP_EVENTを有効化
[22] diag1_xcomp	RW 0x0	ポジション・コンパレータをDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でX_COMPAREを有効化
[21] diag1_dir	RW 0x0	DIR出力信号をDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でDIRを有効化

ビットおよび名称	タイプおよびリセット	説明
[20] diag1_step	RW 0x0	STEP出力信号をDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でSTEPを有効化
[19] diag1_index	RW 0x0	インデックス信号をDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 インデックス・パルスをDIAG1に割当て
[18] diag1_stall	RW 0x0	StallGuard信号SGをDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 ストール情報をDIAG1に割当て
[17] diag1_otpw	RW 0x0	OTPW（過熱事前警告）をDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でOTPWを有効化
[16] diag1_error	RW 0x0	ドライバ・エラー条件をDIAG1に割り当てます
	0: DISABLE 1: ENABLE	無効化 DIAG1でエラー条件を有効化
[11] diag0_overvoltage	RW 0x0	OVピンをDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でOVを有効化
[10] diag0_ev_n_deviation	RW 0x0	Nイベントおよび偏差警告をDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でエンコーダ・イベントを有効化
[9] diag0_ev_pos_reached	RW 0x0	RAMPSTAT_EVENT_POS_REACHEDをDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でEVENT_POS_REACHEDを有効化
[8] diag0_ev_stop_sg	RW 0x0	RAMPSTAT_EVENT_SG_STOPをDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でSG_STOPを有効化
[7] diag0_ev_stop_ref	RW 0x0	RAMPSTAT_EVENT_STOPL    RAMPSTAT_EVENT_STOPRをDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でSTOP_EVENTを有効化
[6] diag0_xcomp	RW 0x0	ポジション・コンパレータをDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でX_COMPAREを有効化
[5] diag0_dir	RW 0x0	8ポイント・ランプ・ジェネレータのDIR出力をDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でDIRを有効化

ビットおよび名称	タイプおよびリセット	説明
[4] diag0_step	RW 0x0	8ポイント・ランプ・ジェネレータのSTEP出力をDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でSTEPを有効化
[3] diag0_index	RW 0x0	インデックス信号をDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 インデックス・パルスにDIAG0に割当て
[2] diag0_stall	RW 0x0	StallGuard信号SGをDIAG0に割り当てます。TCOOLTHRSに達した後に有効化されます。
	0: DISABLED 1: ENABLED	無効化 ストール情報をDIAG0に割当て
[1] diag0_otpw	RW 0x0	OTPW（過熱事前警告）をDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でOTPWを有効化
[0] diag0_error	RW 0x0	ドライバ・エラー条件をDIAG0に割り当てます
	0: DISABLED 1: ENABLED	無効化 DIAG0でエラー条件を有効化

**0x0A : DRV\_CONF**

ビットおよび名称	タイプおよびリセット	説明
[5:4] SLOPE_CONTROL	RW 0x0	勾配制御設定
	0: SC_100V_US	100V/μs
	1: SC_200V_US	200V/μs
	2: SC_400V_US	400V/μs
	3: SC_800V_US	800V/μs
[1:0] CURRENT_RANGE	RW 0x0	この設定により、モーター電流範囲に合わせたドライバのRDSon電流検出機能の基本的な調整ができます。最高の電流精度を実現するには、最小のフィッティング範囲を選択してください。値はピーク電流の設定値です。
	0: CR_1A	1A
	1: CR_2A	2A
	2: CR_3A	3A
	3: CR_3A_RED	3A

## 0x0B : GLOBAL\_SCALER

ビットおよび名称	タイプおよびリセット	説明
[7:0] GLOBALSCALER	RW, unsigned 0x00	<p>モーター電流のグローバル・スケーリング。ドライブを特定のモーター・タイプに合わせるために、この値が電流スケーリングに乗じられます。この値を選択してから、他の設定値を調整してください。この値はチョッパのヒステリシスにも影響するためです。この値は、モーター電流を微調整することだけを目的としています。</p> <p>0: フルスケール（または256を書込み） 1 ... 31: 動作には許容されません 32 ... 255: 最大電流の32/256 ... 255/256</p> <p>ヒント：最高の結果を得るには、128より大きな値にすることを推奨します。</p>

## 0x10 : IHOLD\_IRUN

ビットおよび名称	タイプおよびリセット	説明
[27:24] IRUNDELAY	RW 0x4	開始が検出されてからモーターが起動するまでのクロック・サイクル数を制御します。0：即時パワー・アップ、1...15：電流インクリメント・ステップあたりの遅延はIRUNDELAY × 512クロックの倍数
	0: INSTANT	動作開始時にIRUNへ瞬時にジャンプ。
	1: DELAY_512	電流インクリメントあたり512 × t_clkの遅延。
	2: DELAY_1024	電流インクリメントあたり1024 × t_clkの遅延。
	3: DELAY_1536	電流インクリメントあたり1536 × t_clkの遅延。
	4: DELAY_2048	電流インクリメントあたり2048 × t_clkの遅延。
	5: DELAY_2560	電流インクリメントあたり2560 × t_clkの遅延。
	6: DELAY_3072	電流インクリメントあたり3072 × t_clkの遅延。
	7: DELAY_3584	電流インクリメントあたり3584 × t_clkの遅延。
	8: DELAY_4096	電流インクリメントあたり4096 × t_clkの遅延。
	9: DELAY_4608	電流インクリメントあたり4608 × t_clkの遅延。
	10: DELAY_5120	電流インクリメントあたり5120 × t_clkの遅延。
	11: DELAY_5632	電流インクリメントあたり5632 × t_clkの遅延。
	12: DELAY_6144	電流インクリメントあたり6144 × t_clkの遅延。
	13: DELAY_6656	電流インクリメントあたり6656 × t_clkの遅延。
	14: DELAY_7168	電流インクリメントあたり7168 × t_clkの遅延。
	15: DELAY_7680	電流インクリメントあたり7680 × t_clkの遅延。

ビットおよび名称	タイプおよびリセット	説明
[19:16] IHOLDDELAY	RW 0x1	停止が検出され (stst = 1) TPOWERDOWNの時間が経過すると直ちに、動作後のモーター・パワー・ダウンを行うためのクロック・サイクル数を制御します。滑らかな遷移にすることで、パワー・ダウン時のモーター・ジャークを回避できます。  0: 即時パワー・ダウン 1..15: 電流低減ステップあたりの遅延は $2^{18}$ クロックの倍数
	0: INSTANT 1: DELAY_1_218 2: DELAY_2_218 3: DELAY_3_218 4: DELAY_4_218 5: DELAY_5_218 6: DELAY_6_218 7: DELAY_7_218 8: DELAY_8_218 9: DELAY_9_218 10: DELAY_10_218 11: DELAY_11_218 12: DELAY_12_218 13: DELAY_13_218 14: DELAY_14_218 15: DELAY_15_218	TPOWERDOWN時間が経過すると直ちにIHOLD電流に減少。 $2^{18} \times t_{clk}$ ごとに1ずつデクリメント $2^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $3^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $4^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $5^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $6^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $7^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $8^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $9^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $10^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $11^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $12^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $13^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $14^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント $15^2 \times 2^{18} \times t_{clk}$ ごとに1ずつデクリメント
[12:8] IRUN	RW, unsigned 0x1F	モーター動作電流 (0 = 1/32、...、31=32/32)  ヒント: 最高のマイクロステップ性能を得るには、設定値の範囲を16~31にします。
[4:0] IHOLD	RW, unsigned 0x08	停止時電流 (0 = 1/32、...、31=32/32) StealthChop2モードと組み合わせた場合、IHOLD = 0と設定することで、フリーホイールまたはコイル短絡回路を選択してモーターを停止できます。

## 0x11 : TPOWERDOWN

ビットおよび名称	タイプおよびリセット	説明
[7:0] TPOWERDOWN	RW, unsigned 0x0A	TPOWERDOWNは、モーターの停止 (stst) 後からモーター電流がパワー・ダウンするまでの遅延時間を設定します。時間範囲は、約0~4秒です。  注意: StealthChop2のPWM_OFFS_AUTOを自動調整するには、少なくとも2に設定する必要があります。 リセット後のデフォルト= 10 $0 \dots ((2^8) - 1) \times 2^{18} t_{CLK}$

## 0x12 : TSTEP

ビットおよび名称	タイプおよびリセット	説明
[19:0] TSTEP	R, unsigned 0x00000	<p>ステップ入力周波数から引き出される2つの1/256マイクロステップ間の実際の測定時間（<math>1/f_{CLK}</math>単位）。オーバーフローまたは停止時の測定値は、<math>(2^{20}) - 1</math>です。</p> <p>TSTEPに関連する閾値はすべて、比較値の1/16のヒステリシスを使用してクロックまたはステップ周波数のジッタを補償します。フラグsmall_hysteresisは、このヒステリシスをより小さな値である1/32に変更します。</p> <p><math>(T_{xxx} \times 15/16) - 1</math>または <math>(T_{xxx} \times 31/32) - 1</math>が、各比較値の2番目の比較値として使用されます。つまり、下限スイッチング速度は計算された設定値に等しくなりますが、上限スイッチング速度はヒステリシス設定で定義された値よりも高くなります。</p> <p>モーション・コントローラを使用して動作する場合、所定の速度Vに対して測定されるTSTEPは次の範囲になります。</p> $(2^{24}/V) \leq TSTEP \leq 2^{24}/V - 1$ <p>DcStepモードでは、TSTEPはモーターの平均速度を示すのではなく、各マイクロステップの速度を示します。これは、安定していない可能性があるため、モーターが目標速度よりも低い速度で動作する場合、実際のモーター速度を表すわけではありません。</p>

## 0x13 : TPWMTHRS

ビットおよび名称	タイプおよびリセット	説明
[19:0] TPWMTHRS	RW, unsigned 0x00000	<p>これは、StealthChop2の電圧PWMモードでの上限速度です。</p> <p>StealthChop2 PWMモードが有効化されます（設定されている場合）。DcStepは無効化されます。</p>

## 0x14 : TCOOLTHRS

ビットおよび名称	タイプおよびリセット	説明
[19:0] TCOOLTHRS	RW, unsigned 0x00000	<p>これは、スマート・エネルギーCoolStepおよびStallGuard機能をスイッチ・オンするための下限閾値速度です。（符号なし）</p> <p>このパラメータを設定すると、信頼できる動作を行うことのできない低速時にCoolStepがディスエーブルされます。ストール時に停止する機能（内部モーション・コントローラを用いた場合にsg_stopでイネーブル）およびストール出力信号は、この速度を超えると有効化されます。DcStep以外のモードでは、速度がこの閾値未満になると再び無効化されます。</p> <p>TCOOLTHRS <math>\geq</math> TSTEP <math>\geq</math> THIGH :</p> <p>CoolStepがイネーブルされます（設定されている場合）。</p> <p>TCOOLTHRS <math>\geq</math> TSTEP :</p> <p>ストール時に停止する機能が有効化されます（設定されている場合）。 ストール出力信号（DIAG0/1）が有効化されます（設定されている場合）。</p>

## 0x15 : THIGH

ビットおよび名称	タイプおよびリセット	説明
[19:0] THIGH	RW, unsigned 0x00000	<p>この速度設定は、トルクを最大にするために、異なるチョッパ・モードおよびフルステップ動作への速度に依存したスイッチングを可能にします。（符号なし）</p> <p>ストール検出機能は、THIGH閾値を通過すると必ず2~3電氣的周期分スイッチ・オフされ、スイッチング・モードの影響を補償します。</p> <p>TSTEP <math>\leq</math> THIGH :</p> <p>CoolStepは無効化されます（モーターは通常電流スケールで動作）。 StealthChop2電圧PWMモードは無効化されます。 vhighchmが設定されていると、チョッパはTFD = 0（低速減速のみの定オフ時間）の場合にchm = 1に切り替わります。 vhighfsが設定されていると、モーターはフルステップ・モードで動作し、ストール検出は、DcStepストール検出に切り替わります。</p>

## 0x20 : RAMPMODE

ビットおよび名称	タイプおよびリセット	説明
[1:0] RAMPMODE	RW 0x0	モーション・コントローラのランピング・モード
	0: POS_MODE 1: VEL_POS 2: VEL_NEG 3: HOLD	ポジショニング・モード（すべてのA、D、Vパラメータを使用）。 正のVMAX側の速度モード（加速度AMAXを使用）。 負のVMAX側の速度モード（加速度AMAXを使用）。 ホールド・モード（停止イベントが発生しない限り、速度は変更されません）。

## 0x21 : XACTUAL

ビットおよび名称	タイプおよびリセット	説明
[31:0] XACTUAL	RW, signed 0x00000000	実際のモーター・ポジション（符号付き）  <b>ヒント：</b> 駆動のホーミングを行う場合、通常必要なのはこの値が変更されることだけです。ポジショニング・モードでは、このレジスタの内容を変更すると、動作が開始されます。

## 0x22 : VACTUAL

ビットおよび名称	タイプおよびリセット	説明
[23:0] VACTUAL	R, signed 0x000000	ランプ・ジェネレータによる実際のモーター速度（符号付き）。  符号は動作方向に一致します。負の符号は、XACTUALが減少する方向への移動を意味します。  $\pm(2^{23} - 1)$ [μsteps/t]

## 0x23 : VSTART

ビットおよび名称	タイプおよびリセット	説明
[17:0] VSTART	RW, unsigned 0x00000	モーターの始動速度（符号なし）  一般的な使用では、VSTOP ≥ VSTARTとなるように設定します。VSTARTからVSTOPに確実に減速できるだけの移動距離がある場合は、この条件は不要です。  $0 \dots (2^{18}) - 1$ [μsteps/t]

## 0x24 : A1

ビットおよび名称	タイプおよびリセット	説明
[17:0] A1	RW, unsigned 0x00000	VSTARTからV1への最初の加速（符号なし）  $0 \dots (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ]

## 0x25 : V1

ビットおよび名称	タイプおよびリセット	説明
[19:0] V1	RW, unsigned 0x00000	最初の加速／減速フェーズの閾値速度（符号なし）。  0 : A1およびD1フェーズをディスエーブルし、AMAX、DMAXのみを使用。  $0 \dots (2^{20}) - 1$ [μsteps/t]

**0x26 : AMAX**

ビットおよび名称	タイプおよびリセット	説明
[17:0] <b>AMAX</b>	RW, unsigned 0x00000	V1からVMAXへの2回目の加速（符号なし）。  これは、速度モードの加速度値および減速度値です。  $0 \dots (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ]

**0x27 : VMAX**

ビットおよび名称	タイプおよびリセット	説明
[22:0] <b>VMAX</b>	RW, unsigned 0x000000	モーション・ランプ目標速度（ポジショニングを行うには確実にVMAX ≥ VSTARTとなるようにしてください）（符号なし）  これは速度モードでの目標速度です。動作中いつでも変更できます。  $0 \dots (2^{23}) - 512$ [μsteps/t]

**0x28 : DMAX**

ビットおよび名称	タイプおよびリセット	説明
[17:0] <b>DMAX</b>	RW, unsigned 0x00000	VMAXからV1への減速（符号なし）。  $0 \dots (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ]

## 0x29 : TVMAX

ビットおよび名称	タイプおよびリセット	説明
[15:0] TVMAX	RW, unsigned 0x0000	512クロックの倍数で表した一定速度セグメントの最小時間。  0 : 一定速度フェーズでの最小時間設定を無効化します。 >0 : ジャークを減らすため、加速から減速への変化または減速から加速への変化の間に最小時間の一定速度が挿入されます。  $(0 \dots (2^{16}) - 1) \times 512 t_{CLK}$ <b>注 :</b> ポジション・モード時と停止時に、VMAXの設定後、このレジスタを設定します。ランプ・モードに戻す場合、TVMAXの遅延がトリガされないよう、速度モード時にはTVMAX = 0に設定します。

## 0x2A : D1

ビットおよび名称	タイプおよびリセット	説明
[17:0] D1	RW, unsigned 0x0000A	V1からVSTOPへの減速（符号なし）。  <b>注意 :</b> ポジショニング・モードでは、V1=0であっても、この値を0には設定しないでください。  $1 \dots (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ] リセット後のデフォルト= 10

## 0x2B : VSTOP

ビットおよび名称	タイプおよびリセット	説明
[17:0] VSTOP	RW, unsigned 0x0000A	モーター停止速度（符号なし）  <b>ヒント :</b> 短距離でのポジショニングを可能にするにはVSTOP ≥ VSTARTとなるよう設定します。  <b>注意 :</b> ポジショニング・モードではこの値を0には設定しないでください。最小でも10とすることを推奨します。  $1 \dots (2^{18}) - 1$ [μsteps/t] リセット後のデフォルト= 10

## 0x2C : TZEROWAIT

ビットおよび名称	タイプおよびリセット	説明
[15:0] TZEROWAIT	RW, unsigned 0x0000	速度ゼロへのランプダウン後から次の動作または方向反転を開始できるまでの待機時間を定義します。時間範囲は、約0~2秒です。  この設定により、例えばVSTOPからVSTARTへといった、過剰な加速を避けることができます。  $0 \dots (2^{16}) - 1 \times 512 t_{CLK}$

## 0x2D : XTARGET

ビットおよび名称	タイプおよびリセット	説明
[31:0] XTARGET	RW, signed 0x00000000	ランプ・モード用の目標ポジション（符号付き）。RAMPMODE = 0でランプ・ジェネレータのポジショニングを有効化するために、このレジスタに新しい目標ポジションを書き込みます。それ以前の速度、加速度、減速度のパラメータはすべて初期化します。  <b>ヒント</b> ：ポジションはラップ・アラウンド可能です。そのため、XTARGETの値は、符号なしの数値として処理することもできます。  <b>ヒント</b> ：可能な最大変位は $\pm((2^{31}) - 1)$ です。  <b>ヒント</b> ：動作中にV1、D1、またはDMAXを増加させる場合、必要に応じて、その後XTARGETに再書き込みして2回目の加速フェーズをトリガします。  $-2^{31} \dots + (2^{31}) - 1$

## 0x2E : V2

ビットおよび名称	タイプおよびリセット	説明
[19:0] V2	RW, unsigned 0x00000	AMAX/2およびDMAX/2で加速セグメントを有効化する場合のVMAXとの速度差 0 : AMAX/2およびDMAX/2フェーズをディスエーブルし、AMAX、DMAXのみを使用。  $0 \dots (2^{20}) - 1$ [μsteps/t]

## 0x2F : A2

ビットおよび名称	タイプおよびリセット	説明
[17:0] A2	RW, unsigned 0x00000	V1からV2への加速（符号なし）。  $0 \dots (2^{18}) - 1$ [ $\mu\text{steps}/\text{ta}^2$ ]

## 0x30 : D2

ビットおよび名称	タイプおよびリセット	説明
[17:0] D2	RW, unsigned 0x0000A	V2からV1への減速（符号なし）。  <b>注意：</b> ポジショニング・モードでは、 $V2 = 0$ であっても、この値を0には設定しないでください。  $1 \dots (2^{18}) - 1$ [ $\mu\text{steps}/\text{ta}^2$ ] リセット後のデフォルト= 10

## 0x33 : VDCMIN

dcStep の開始速度

ビットおよび名称	タイプおよびリセット	説明
[22:8] VDCMIN	RW, unsigned 0x0000	自動整流DcStepは、VDCMINを超える速度でイネーブルされます（符号なし）。  このモードでは、実際の位置はセンサーレス・モーター整流によって決定され、XACTUALにフィードバックされます。モーターが重負荷になった場合、VDCMINも最小ステップ速度として用いられます。ステップ・ロスを検出するには、ストール時に停止する機能（sg_stop）を有効化します。  0 : ディスエーブル、DcStepがオフ  VACT  $\geq$ VDCMIN $\geq$ 256 :  THIGHの設定値を超えた場合と同じ動作をトリガします。 自動整流DcStepをスイッチ・オンします。  <b>ヒント：</b> DcStepを動作させるには、DCCTRLパラメータも設定します。  (値および比較のためにはビット22~8のみが用いられます)。

ビットおよび名称	タイプおよびリセット	説明
[7:0] reserved	R, unsigned 0x00	常に0を読み出します

**0x34 : SW\_MODE**

スイッチ・モード設定

ビットおよび名称	タイプおよびリセット	説明
[14] virtual_stop_enc	RW 0x0	仮想停止比較 (VIRTUAL_STOP_LおよびVIRTUAL_STOP_R) のためのソースとしてX_ENCかXACTUALかを選択します。
	0: SRC_XACTUAL 1: SRC_X_ENC	XACTUALをVIRTUAL_STOP_L/Rと比較 X_ENCをVIRTUAL_STOP_L/Rと比較
[13] en_virtual_stop_r	RW 0x0	右側の仮想停止条件がアクティブとなっている場合に自動モーター停止を有効化。
	0: DISABLED 1: ENABLED	無効化 有効化
[12] en_virtual_stop_l	RW 0x0	左側の仮想停止条件がアクティブとなっている場合に自動モーター停止を有効化。
	0: DISABLED 1: ENABLED	無効化 有効化
[11] en_softstop	RW 0x0	ソフトストップ・モードでは、常にDMAx、V1、D1、V2、D2、VSTOP、TZEROWAITの減速ランプ設定を用いてモーターを停止させます。停止が発生するのは、速度の符号がリファレンス・スイッチのポジション (負の速度の場合はREFL および VIRTUAL_STOP_L、正の速度の場合はREFR および VIRTUAL_STOP_R) に合致し、かつ、それぞれのスイッチ停止機能が有効化されている場合です。  ハード・ストップも、モーターが解放される前にTZEROWAITを用います。  <b>注意:</b> ソフト・ストップは、StallGuardと組み合わせて用いないでください。高速度でのStealthChop動作にはソフト・ストップを用いてください。この場合、ハード・ストップは重度の過電流の原因となるため、避ける必要があります。
	0: HARD_STOP 1: SOFT_STOP	ハード・ストップ ソフト・ストップ
[10] sg_stop	RW 0x0	StallGuardによる停止をイネーブル (DcStepモードでの使用可)。ディスエーブルすると停止イベント後にモーターを解放します。速度閾値にはTCOOLTHRSを設定します。  <b>ヒント:</b> モーターの回転上昇中にはイネーブルしないでください。モーター速度が、StallGuardによって安定した結果が得られる値を超えるまで待ってください。この速度閾値は、TCOOLTHRSを用いて設定する必要があります。
	0: DISABLED 1: ENABLED	無効化 ストール時に停止する機能が有効化されます
[9] en_latch_encoder	RW 0x0	ポジションXACTUALに加えて、ラッチ・イベント時のリファレンス切り替えイベントにおいて、エンコーダ・ポジションX_ENCをENC_LATCHIにラッチします。(latch_r_active、_inactive、latch_l_active、_inactive)
	0: DISABLED 1: ENABLED	無効化 X_ENCをラッチ

ビットおよび名称	タイプおよびリセット	説明
[8] latch_r_inactive	RW 0x0	右側リファレンス・スイッチ入力REFRにおいて非有効化へのエッジが発生したときに、ポジションXACTUALをXLATCHにラッチします。有効化レベルは pol_stop_r で定義されます。
	0: DISABLED 1: ENABLED	無効化 status_stop_r の立下がりエッジでのラッチを有効化
[7] latch_r_active	RW 0x0	右側リファレンス・スイッチ入力REFRにおいて有効化へのエッジが発生したときに、ポジションXACTUALをXLATCHにラッチします。 ヒント： status_latch_r の読出しによって誤停止イベントを検出するには、latch_r_active を有効化します。
	0: DISABLED 1: ENABLED	無効化 status_stop_r の立上がりエッジでのラッチを有効化
[6] latch_l_inactive	RW 0x0	左側リファレンス・スイッチ入力REFLにおいて非有効化へのエッジが発生したときに、ポジションXACTUALをXLATCHにラッチします。有効化レベルは pol_stop_l で定義されます。
	0: DISABLED 1: ENABLED	無効化 status_stop_l の立下がりエッジでのラッチを有効化
[5] latch_l_active	RW 0x0	左側リファレンス・スイッチ入力REFLにおいて有効化へのエッジが発生したときに、ポジションXACTUALをXLATCHにラッチします。 ヒント： status_latch_l の読出しによって誤停止イベントを検出するには、latch_l_active を有効化します。
	0: DISABLED 1: ENABLED	無効化 status_stop_l の立上がりエッジでのラッチを有効化
[4] swap_lr	RW 0x0	停止イベントにおいて、左側リファレンス・スイッチ入力REFLと右側リファレンス・スイッチ入力REFRを入れ替えます。
	0: DEFAULT 1: SWAPPED	デフォルトの割当て REFLとREFRを切り替え
[3] pol_stop_r	RW 0x0	右側リファレンス・スイッチ入力でアクティブな極性を設定します。
	0: NON_INV 1: INV	非反転、ハイでアクティブ：REFRのレベルが高くなるとモーターが停止します。 反転、ローでアクティブ：REFRのレベルがローになるとモーターが停止します。
[2] pol_stop_l	RW 0x0	左側リファレンス・スイッチ入力でアクティブな極性を設定します。
	0: NON_INV 1: INV	非反転、ハイでアクティブ：REFLのレベルが高くなるとモーターが停止します。 反転、ローでアクティブ：REFLのレベルがローになるとモーターが停止します。
[1] stop_r_enable	RW 0x0	1：右側リファレンス・スイッチ入力が入力されている間の自動モーター停止を有効化します ヒント：停止スイッチが解放されるとモーターは再始動します。
	0: DISABLED 1: ENABLED	無効化 有効化
[0] stop_l_enable	RW 0x0	1：左側リファレンス・スイッチ入力が入力されている間の自動モーター停止を有効化します ヒント：停止スイッチが解放されるとモーターは再始動します。
	0: DISABLED 1: ENABLED	無効化 有効化

## 0x35 : RAMP\_STAT

ランプのステータスおよびスイッチ・イベントのステータス

ビットおよび名称	タイプおよびリセット	説明
[15] status_virtual_stop_r	R 0x1	右側仮想リファレンス・スイッチのステータス (1=アクティブ)
	0: INACTIVE 1: ACTIVE	右側仮想停止は非アクティブ 右側仮想停止を検出
[14] status_virtual_stop_l	R 0x1	左側仮想リファレンス・スイッチのステータス (1=アクティブ)
	0: INACTIVE 1: ACTIVE	左側仮想停止は非アクティブ 左側仮想停止はアクティブ
[13] status_sg	R 0x0	1: StallGuardが有効な場合、アクティブなStallGuard入力を示します。  ヒント: このフラグをポーリングする場合、ストール・イベントが検出されない可能性があります。ストール・イベントを確実に検出できるようにsg_stopを有効化してください。
	0: NO_STALL 1: STALL	ストール不検出 ストール
[12] second_move	RW, W1C 0x0	1: オンザフライのパラメータ変更などにより自動ランプの動作を反対方向に戻すことが必要であることを指示します。#type = COW
	0: INACTIVE 1: ACTIVE	目標に到達するために2回目の動作は不要。 目標ポジションに到達するために方向転換が必要。
[11] t_zerowait_active	R 0x0	1: モーターの停止後TZEROWAITがアクティブであることを指示します。この時間、モーターは停止状態です。
	0: INACTIVE 1: ACTIVE	非アクティブ 動作間でのTZEROWAITの中断がアクティブ
[10] vzero	R 0x0	1: 実際の速度が0であることを示します。
	0: VEL_NOT_0 1: VEL_0	VACTUAL = 0 VACTUAL = 0
[9] position_reached	R 0x0	1: 目標ポジションに達していることを示します。 このフラグは、XACTUALがXTARGETと一致している場合にセットされます。
	0: POS_NOT_REACHED 1: POS_REACHED	目標ポジションにまだ達していない。 目標ポジションへの移動が完了。
[8] velocity_reached	R 0x0	1: 目標速度に達していることを示します。 このフラグは、VACTUALがVMAXと一致している場合にセットされます。
	0: VMAX_NOT_REACHED 1: VMAX_REACHED	VMAXに達していない。 VMAXに到達。
[7] event_pos_reached	RW, W1C 0x0	1: 目標ポジションに達したことを示します (position_reachedがアクティブになります)。 このビットは、DIAG0の割込み出力信号とOR接続されます。#type = COW
	0: INACTIVE 1: ACTIVE	目標ポジションに達していない。 このフラグが最後にクリアされた後に、目標ポジションに到達。
[6] event_stop_sg	RW, W1C 0x0	1: StallGuard停止イベントがアクティブであることを示します。 レジスタをリセットするとストール条件がクリアされ、モーターは、モーション・コントローラが停止していない限り、動作を再開します。 このビットは、DIAG0の割込み出力信号とOR接続されます。#type = COW
	0: INACTIVE 1: ACTIVE	ストール・イベントは非アクティブ。 ストールを検出したためモーターは停止。

ビットおよび名称	タイプおよびリセット	説明
[5] event_stop_r	R 0x0	1: 停止スイッチまたは仮想停止により右側停止条件を有効化。 停止条件と割込み条件は、RAMP_MODEをホールド・モードに設定するか反対方向への動作を指示することで、解除できます。soft_stopモードの場合、この条件はモーターが停止スイッチ方向への動作を停止するまでアクティブのままとなります。停止スイッチまたは停止機能を無効化してもこのフラグをクリアできませんが、モーターは動作を続けます。 このビットは、DIAG0の割込み出力信号とOR接続されます。
	0: INACTIVE 1: ACTIVE	停止イベントなし 右方向の停止イベントを有効化
[4] event_stop_l	R 0x0	1: 停止スイッチまたは仮想停止により左側停止条件を有効化。 停止条件と割込み条件は、RAMP_MODEをホールド・モードに設定するか反対方向への動作を指示することで、解除できます。soft_stopモードの場合、この条件はモーターが停止スイッチ方向への動作を停止するまでアクティブのままとなります。停止スイッチまたは停止機能を無効化してもこのフラグをクリアできませんが、モーターは動作を続けます。 このビットは、DIAG0の割込み出力信号とOR接続されます。
	0: INACTIVE 1: ACTIVE	停止イベントなし 左方向の停止イベントを有効化
[3] status_latch_r	RW, W1C 0x0	1: 停止イベント後に右側ラッチの準備が完了します。 (SW_MODEのlatch_r_activeまたはlatch_r_inactiveの設定を用いてポジションのラッチをイネーブルします)。#type = COW
	0: INACTIVE 1: ACTIVE	ポジションをラッチしない ポジションをラッチ
[2] status_latch_l	RW, W1C 0x0	1: 停止イベント後に左側ラッチの準備が完了します。 (SW_MODEのlatch_l_activeまたはlatch_l_inactiveの設定を用いてポジションのラッチをイネーブルします)。#type = COW
	0: INACTIVE 1: ACTIVE	ポジションをラッチしない ポジションをラッチ
[1] status_stop_r	R 0x0	右側リファレンス・スイッチのステータス (1 = アクティブ)。極性およびREFL/REFR入力の入れ替え (SW_MODEのswap_lrを参照) を考慮した後の状態を示します。このビットはリファレンス・スイッチのステータスのみを示します。停止イベントが起きたかどうかを確認するには、event_stop_rをチェックしてください。
	0: INACTIVE 1: ACTIVE	非アクティブ リファレンス・スイッチはアクティブ
[0] status_stop_l	R 0x0	左側リファレンス・スイッチのステータス (1 = アクティブ)。極性およびREFL/REFR入力の入れ替え (SW_MODEのswap_lrを参照) を考慮した後の状態を示します。このビットはリファレンス・スイッチのステータスのみを示します。停止イベントが起きたかどうかを確認するには、event_stop_lをチェックしてください。
	0: INACTIVE 1: ACTIVE	非アクティブ リファレンス・スイッチはアクティブ

**0x36 : XLATCH**

ランプ・ジェネレータのラッチ位置

ビットおよび名称	タイプおよびリセット	説明
[31:0] XLATCH	R, signed 0x00000000	ランプ・ジェネレータのラッチ・ポジション。プログラマブルなスイッチ・イベント時にXACTUALをラッチします（ <a href="#">SW_MODE</a> を参照）。  ヒント：エンコーダ・ポジションは、整合性チェックができるよう、XLATCHと共にENC_LATCHにラッチされる場合もあります。

**0x38 : ENCMODE**

ビットおよび名称	タイプおよびリセット	説明
[10] enc_sel_decimal	RW 0x0	エンコーダ・プリスケアラのモード選択
	0: BINARY 1: DECIMAL	エンコーダ・プリスケアラの除算2進数モード： ENC_CONST（小数部）/65536をカウント  エンコーダ・プリスケアラの除算10進法モード： ENC_CONST（小数部）/10000をカウント
[9] latch_x_act	RW 0x0	ポジション・ラッチの設定
	0: DISABLED 1: LATCH_XACTUAL	無効化 X_ENCと一緒にXACTUALポジションをラッチします。 pos_edgeおよびneg_edgeによって選択したNチャンネル・イベントでランプ・ジェネレータ・ポジションをラッチできます。
[8] clr_enc_x	RW 0x0	エンコーダ・ラッチの設定
	0: KEEP 1: CLEAR	Nイベントにおいて、X_ENCはENC_LATCHにラッチされるのみです。 Nイベントにおいて、ラッチされた後エンコーダ・カウンタX_ENCをクリアします。
[7:6] pos_neg_edge	RW 0x0	Nチャンネル・イベントの感度
	0: HIGH_ACTIVE	Nチャンネル・イベントは、Nイベント・レベルがアクティブな間アクティブになります。
	1: RISING_EDGE	Nチャンネルは、Nイベントがアクティブになる立上がり時に有効になります。
	2: FALLING_EDGE	Nチャンネルは、Nイベントが非アクティブになる立下がり時に有効になります。
3: BOTH_EDGES	Nチャンネルは、Nイベントがアクティブになるとき、および非アクティブになるときに有効になります。	
[5] clr_once	RW 0x0	ポジション・ラッチの設定
	0: OFF 1: LATCH_ONCE	無効化 書込みアクセスに続く次のNイベントで、X_ENCをラッチ、またはラッチしてクリアします。
[4] clr_cont	RW 0x0	ポジション・ラッチの設定
	0: OFF 1: LATCH_CONT	無効化 Nイベント時に常にX_ENCをラッチ、またはラッチしてクリアします（1回転あたり1回、この設定はエッジ応答側のNイベントと組み合わせることを推奨します）。

ビットおよび名称	タイプおよびリセット	説明
[3] ignore_AB	RW 0x0	Nイベントの設定
	0: N_AB_MATCH 1: N_AB_IGNORED	Nイベントは、pol_N、pol_A、pol_Bで与えられた極性が一致した場合にのみ発生します。 Nチャンネル・イベントにおいてAおよびBの極性を無視します。
[2] pol_N	RW 0x0	Nのアクティブな極性を定義
	0: LOW_ACTIVE 1: HIGH_ACTIVE	ロー・アクティブ ハイ・アクティブ
[1] pol_B	RW 0x0	Nチャンネル・イベントに必要なBの極性
	0: NEG 1: POS	負の極性 正の極性
[0] pol_A	RW 0x0	Nチャンネル・イベントに必要なAの極性
	0: NEG 1: POS	負の極性 正の極性

**0x39 : X\_ENC**

ビットおよび名称	タイプおよびリセット	説明
[31:0] X_ENC	RW, signed 0x00000000	実際のエンコーダ・ポジション (符号付き)

## 0x3A : ENC\_CONST

ビットおよび名称	タイプおよびリセット	説明
[31:0] ENC_CONST	RW, signed 0x00010000	<p>加算定数（符号付き） 16ビットの整数部分、16ビットの小数部分</p> <p>X_ENCで加算される量は以下のとおりです +/- ENC_CONST / (2<sup>16</sup> × X_ENC) (2進数) または +/- ENC_CONST / (10<sup>4</sup> × X_ENC) (10進数)</p> <p>ENCMODEのビットenc_sel_decimalで10進数設定と2進数設定の間の切り替えが行われます。 回転方向を合わせるには符号を用いてください。</p> <p>2進数： ± [μsteps/2<sup>16</sup>] ±(0 ... 32767.999847) 10進数： ±(0.0 ... 32767.9999) リセット後のデフォルト = 1.0 (= 65536)</p>

## 0x3B : ENC\_STATUS

エンコーダのステータス情報

ビットおよび名称	タイプおよびリセット	説明
[1] deviation_warn	RW, W1C 0x0	このビットが最後にクリアされた後に、X_ACTUALとX_ENCの間の絶対差がENC_DEVIATIONに達したことを示します。#type = COW
	0: NO_DEVIATION 1: DEVIATION	警告なし 警告が続いている間はdeviation_warnはクリアできません。無効にするにはENC_DEVIATIONを0にセットします。
[0] n_event	RW, W1C 0x0	このビットが最後にクリアされた後に、Nイベントを検出。#type = COW
	0: NO_EVENT 1: EVENT_DETECTED	イベントなし イベントを検出。このステータス・ビットをクリアするには、対応するポジションにおいて1ビットを書き込みます。

## 0x3C : ENC\_LATCH

ビットおよび名称	タイプおよびリセット	説明
[31:0] ENC_LATCH	R, unsigned 0x00000000	Nイベント時にラッチされたエンコーダ・ポジションX_ENC。

## 0x3D : ENC\_DEVIATION

ビットおよび名称	タイプおよびリセット	説明
[19:0] ENC_DEVIATION	RW, unsigned 0x00000	偏差警告の原因となる、エンコーダ・カウンタX_ENCとXACTUALのステップ差の最大値。 これにより、ENC_STATUSのdeviation_warnフラグがセットされます。 0 = この機能をオフ。

## 0x3E : VIRTUAL\_STOP\_L

ビットおよび名称	タイプおよびリセット	説明
[31:0] VIRTUAL_STOP_L	RW, signed 0x00000000	エンコーダまたはランプの位置に基づく仮想停止スイッチ。停止は、符号を含めた比較に基づき行われます virtual_stop_enc = 1 : X_ENC <= VIRTUAL_STOP_L virtual_stop_enc = 0 : XACTUAL <= VIRTUAL_STOP_L  -2 <sup>31</sup> ... + (2 <sup>31</sup> ) - 1

## 0x3F : VIRTUAL\_STOP\_R

ビットおよび名称	タイプおよびリセット	説明
[31:0] VIRTUAL_STOP_R	RW, signed 0x00000000	エンコーダに基づく仮想停止スイッチ。停止は、符号を含めた比較に基づき行われます Virtual_stop_enc = 1 : X_ENC >= VIRTUAL_STOP_R Virtual_stop_enc = 0 : XACTUAL >= VIRTUAL_STOP_R  -2 <sup>31</sup> ... + (2 <sup>31</sup> ) - 1

## 0x50 : ADC\_VSUPPLY\_AIN

ビットおよび名称	タイプおよびリセット	説明
[28:16] ADC_AIN	R, signed 0x0000	AINピンの電圧値（整数）。 更新レート= 2048クロックごと
[12:0] ADC_VSUPPLY	R, signed 0x0000	V <sub>S</sub> 電圧の実際の値（ローパス・フィルタを使用）。 更新レート：2048クロックごと V <sub>S</sub> = ADC_VSUPPLY x 17.64 mV

## 0x51 : ADC\_TEMP

ビットおよび名称	タイプおよびリセット	説明
[12:0] ADC_TEMP	R, signed 0x0000	実際の温度（ローパス・フィルタを使用） 更新レート：2048クロックごと。

## 0x52 : OTW\_OV\_VTH

ビットおよび名称	タイプおよびリセット	説明
[28:16] OVERTEMPPREWARNING_VTH	RW, unsigned 0x0B92	過熱警告閾値レジスタ： ADC_TEMP >= OVERTEMPPREWARNING_VTH 過熱事前警告がトリガされます。 （リセット：0xB92が120°Cに相当します）
[12:0] OVERVOLTAGE_VTH	RW, unsigned 0x0F25	出力OV用の過電圧閾値。デフォルト：68.4V

**0x60 : MSLUT\_0**

マイクロステップ・テーブルの項目 0~31

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_0	RW, unsigned 0xAAAAB554	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

**0x61 : MSLUT\_1**

マイクロステップ・テーブルの項目 32~63

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_1	RW, unsigned 0x4A9554AA	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

**0x62 : MSLUT\_2**

マイクロステップ・テーブルの項目 64~95

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_2	RW, unsigned 0x24492929	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

**0x63 : MSLUT\_3**

マイクロステップ・テーブルの項目 96~127

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_3	RW, unsigned 0x10104222	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

**0x64 : MSLUT\_4**

マイクロステップ・テーブルの項目 128~159

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_4	RW, unsigned 0xFBFFFFFF	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

**0x65 : MSLUT\_5**

マイクロステップ・テーブルの項目 160~191

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_5	RW, unsigned 0xB5BB777D	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

**0x66 : MSLUT\_6**

マイクロステップ・テーブルの項目 192~223

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_6	RW, unsigned 0x49295556	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

**0x67 : MSLUT\_7**

マイクロステップ・テーブルの項目 224~255

ビットおよび名称	タイプおよびリセット	説明
[31:0] MSLUT_7	RW, unsigned 0x00404222	<p>各ビットは、対応するMSLUTSELのWビットと組み合わせた場合のエントリxとエントリx + 1の差を示します。</p> <p>0 : W = %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1 : W = %00: +0 %01: +1 %10: +2 %11: +3</p> <p>これは、波形の最初の1/4について差分をコード化したものです。CUR_AとCUR_Bの開始値は、START_SINとSTART_SIN90のMSCNTのポジション0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00 ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト=サイン波テーブル</p>

## 0x68 : MSLUTSEL

ビットおよび名称	タイプおよびリセット	説明
[31:24] X3	RW, unsigned 0xFF	<p>LUTのセグメント1の開始</p> <p>サイン波のルックアップ・テーブルは、個別のステップ幅制御エントリWxを用いて、最大4つのセグメントに分割できます。セグメントの境界は、X1、X2、X3で選択できます。</p> <p>セグメント0は0～X1 - 1の範囲です。 セグメント1はX1～X2 - 1の範囲です。 セグメント2はX2～X3 - 1の範囲です。 セグメント3はX3～255の範囲です。</p> <p>定義された応答を得るには、各値が以下を満たす必要があります。 <math>0 &lt; X1 &lt; X2 &lt; X3</math></p>
[23:16] X2	RW, unsigned 0xFF	<p>LUTのセグメント1の開始</p> <p>サイン波のルックアップ・テーブルは、個別のステップ幅制御エントリWxを用いて、最大4つのセグメントに分割できます。セグメントの境界は、X1、X2、X3で選択できます。</p> <p>セグメント0は0～X1 - 1の範囲です。 セグメント1はX1～X2 - 1の範囲です。 セグメント2はX2～X3 - 1の範囲です。 セグメント3はX3～255の範囲です。</p> <p>定義された応答を得るには、各値が以下を満たす必要があります。 <math>0 &lt; X1 &lt; X2 &lt; X3</math></p>
[15:8] X1	RW, unsigned 0x80	<p>LUTのセグメント1の開始</p> <p>サイン波のルックアップ・テーブルは、個別のステップ幅制御エントリWxを用いて、最大4つのセグメントに分割できます。セグメントの境界は、X1、X2、X3で選択できます。</p> <p>セグメント0は0～X1 - 1の範囲です。 セグメント1はX1～X2 - 1の範囲です。 セグメント2はX2～X3 - 1の範囲です。 セグメント3はX3～255の範囲です。</p> <p>定義された応答を得るには、各値が以下を満たす必要があります。 <math>0 &lt; X1 &lt; X2 &lt; X3</math></p>

ビットおよび名称	タイプおよびリセット	説明
[7:6] W3	RW 0x1	ofs(X3)~ofs255の範囲でのLUT幅の選択  幅制御ビットのコーディングW0~W3: %00: MSLUTのエントリ0、1に対する選択 : -1、+0 %01: MSLUTのエントリ0、1に対する選択 : +0、+1 %10: MSLUTのエントリ0、1に対する選択 : +1、+2 %11: MSLUTのエントリ0、1に対する選択 : +2、+3
	0: W3_SUB1_ADD0 1: W3_ADD0_ADD1 2: W3_ADD1_ADD2 3: W3_ADD2_ADD3	現在のMSLUTのエントリ0 (1) : -1 (+0) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +0 (+1) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +1 (+2) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +2 (+3) をサイン波に選択
[5:4] W2	RW 0x1	ofs(X2)~ofs(X3-1)でのLUT幅の選択  幅制御ビットのコーディングW0~W3: %00: MSLUTのエントリ0、1に対する選択 : -1、+0 %01: MSLUTのエントリ0、1に対する選択 : +0、+1 %10: MSLUTのエントリ0、1に対する選択 : +1、+2 %11: MSLUTのエントリ0、1に対する選択 : +2、+3
	0: W2_SUB1_ADD0 1: W2_ADD0_ADD1 2: W2_ADD1_ADD2 3: W2_ADD2_ADD3	現在のMSLUTのエントリ0 (1) : -1 (+0) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +0 (+1) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +1 (+2) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +2 (+3) をサイン波に選択
[3:2] W1	RW 0x1	ofs (X1) ~ofs (X2-1) でのLUT幅の選択  幅制御ビットのコーディングW0~W3: %00: MSLUTのエントリ0、1に対する選択 : -1、+0 %01: MSLUTのエントリ0、1に対する選択 : +0、+1 %10: MSLUTのエントリ0、1に対する選択 : +1、+2 %11: MSLUTのエントリ0、1に対する選択 : +2、+3
	0: W1_SUB1_ADD0 1: W1_ADD0_ADD1 2: W1_ADD1_ADD2 3: W1_ADD2_ADD3	現在のMSLUTのエントリ0 (1) : -1 (+0) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +0 (+1) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +1 (+2) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +2 (+3) をサイン波に選択
[1:0] W0	RW 0x2	ofs00~ofs(X1-1)の範囲でのLUT幅の選択  幅制御ビットのコーディングW0~W3: %00: MSLUTのエントリ0、1に対する選択 : -1、+0 %01: MSLUTのエントリ0、1に対する選択 : +0、+1 %10: MSLUTのエントリ0、1に対する選択 : +1、+2 %11: MSLUTのエントリ0、1に対する選択 : +2、+3
	0: W0_SUB1_ADD0 1: W0_ADD0_ADD1 2: W0_ADD1_ADD2 3: W0_ADD2_ADD3	現在のMSLUTのエントリ0 (1) : -1 (+0) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +0 (+1) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +1 (+2) をサイン波に選択 現在のMSLUTのエントリ0 (1) : +2 (+3) をサイン波に選択

**0x69 : MSLUTSTART**

リファレンス位置 MSCNT=0 を通過した場合、常に開始値がマイクロステップ・レジスタ CUR\_A および CUR\_B に転送されます。

ビットおよび名称	タイプおよびリセット	説明
[31:24] OFFSET_SIN90	RW, unsigned 0x00	コサイン波に対する符号付きオフセット (+/-127マイクロステップ)。MSCNT = 0の位置でマイクロステップ波形テーブルに適合するよう、START_SIN90を修正します。
[23:16] START_SIN90	RW, unsigned 0xF7	START_SIN90は、MSCNT = 0 (テーブル位置256 + OFFSET_SIN90) の位置でのコサイン波マイクロステップ・テーブル項目に絶対値を与えます。
[7:0] START_SIN	RW, unsigned 0x00	START_SINは、マイクロステップ・テーブル項目0に絶対値を与えます。

**0x6A : MSCNT**

ビットおよび名称	タイプおよびリセット	説明
[9:0] MSCNT	R, unsigned 0x000	マイクロステップ・カウンタ。実際の位置をCUR_Aのマイクロステップ・テーブルに示します。CUR_Bは256のオフセットを使用します (2相モーター)。 ヒント: MSLUTSTARTまたはMSLUTおよびMSLUTSELを再初期化する前に、MSCNTがゼロとなるポジションに移動させてください。

**0x6B : MSCURACT**

ビットおよび名称	タイプおよびリセット	説明
[24:16] CUR_A	R, signed 0x0F7	MSLUTから読み出した、モーター位相A (コサイン波) の実際のマイクロステップ電流 (電流によるスケールなし)。
[8:0] CUR_B	R, signed 0x000	MSLUTから読み出した、モーター位相B (サイン波) の実際のマイクロステップ電流 (電流によるスケールなし)。

**0x6C : CHOPCONF**

ビットおよび名称	タイプおよびリセット	説明
[29] reserved	RW 0x0	予約済み、使用不可

ビットおよび名称	タイプおよびリセット	説明
[28] intpol	RW 0x1  0: DISABLED 1: ENABLED	最も滑らかなモーター動作を実現するため、実際のマイクロステップ分解能 (MRES) を 256 マイクロステップに外挿します。  補間なし 256 マイクロステップへ補間
[27:24] MRES	RW 0x0  0: RES_256 1: RES_128 2: RES_64 3: RES_32 4: RES_16 5: RES_8 6: RES_4 7: RES_HS 8: RES_FS	マイクロステップの分解能の選択。  %0000 : 元来の 256 マイクロステップ設定。通常内部モーション・コントローラではこの設定を使用します。  %0001~%1000 : 128、64、32、16、8、4、2、FULLSTEP 低下したマイクロステップ分解能。 この分解能は、1/4 サイン波あたりのマイクロステップ・エントリの数を示します。 低いマイクロステップ分解能を用いる場合、ドライバは自動的に、波形が対称的となるマイクロステップ位置を用います。 ステップ幅 = $2^{\wedge}(\text{MRES})$ [microsteps]
[23:20] TPFD	RW, unsigned 0x4	受動高速減衰時間  TPFD を用いることで、モーターの中間領域での共振の緩和が可能です。 受動高速減衰時間の設定により、ブリッジの極性変更後に挿入される高速減衰フェーズの時間が制御されます。 $N_{\text{CLK}} = 128 \times \text{TPFD}$ %0000 : ディスエーブル %0001~%1111 : 1~15
[19] vhighchm	RW 0x0  0: CTOFF_THIGH_DIS 1: CTOFF_THIGH_EN	高速チョッパー・モード。  このビットは、VHIGH を超えた場合に chm = 1 および fd = 0 への切り替わりをイネーブルします。したがって、より高い速度が達成できます。vhighfs = 1 と組み合わせることができます。このビットがセットされた場合、チョッパー周波数が 2 倍になるのを防止するため、TOFF 設定は高速動作時に自動的に 2 倍になります。  無効化 THIGH に達したときに定 tOFF チョッパーに切り替え。
[18] vhighfs	RW 0x0  0: FS_THIGH_DIS 1: FS_THIGH_EN	高速フルステップ選択。  このビットは、VHIGH を超えた場合にフルステップへの切り替わりをイネーブルします。切り替わりは 45° の位置でのみ生じます。フルステップの目標電流は、45° 位置でのマイクロステップ・テーブルからの電流値を用います。  無効化 THIGH に達したときにマイクロステップからフルステップに切り替え。

ビットおよび名称	タイプおよびリセット	説明
[16:15] TBL	RW 0x2	TBLブランキング時間設定。 コンパレータのブランキング時間をクロック・サイクル数単位で設定します。 ヒント：ほとんどのアプリケーションには24クロックまたは36クロックを推奨します。  TBL = 0x0の場合の制限：8MHz以下の外部クロック発振器を組み合わせる場合のみ使用できます。 TBL = 0x1の場合の制限：内部クロックまたは13MHz以下の外部クロックを適用する場合に使用できます。
	0: TBL_16 1: TBL_24 2: TBL_36 3: TBL_48	16クロック 24クロック 36クロック 48クロック
[14] chm	RW 0x0	チョッパー・モードの選択。 これが有効になるのは、en_pwm_modeが0に設定されている場合かTSTEP < TPWMTHRSの場合です。
	0: SPREADCYCLE 1: CLASSIC_CHOP	標準モード (SpreadCycle) 定オフ時間で高速減衰時間を使用。高速減衰時間は負の公称電流に達した場合にも終了します。オン時間後の高速減衰。
[12] disfdcc	RW 0x0	chm = 1の場合の高速減衰モード
	0: DISABLED 1: ENABLED	高速減衰サイクルの終了に、電流コンパレータの使用を有効化 高速減衰サイクルの終了に、電流コンパレータの使用を無効化
[11] fd3	RW 0x0	TFD[3]  chm = 1の場合： 高速減衰時間設定TFDのMSB
	0: TFD3_0 1: TFD3_1	TFD設定値のMSB : 0 TFD設定値のMSB : 1
[10:7] HEND_OFFSET	RW, unsigned 0x2	chm = 0の場合： HEND = ヒステリシスの低い値 %0000~%1111： ヒステリシスは-3、-2、-1、0、1、…、12 (この設定値の1/512が電流設定値に加算されます) これは、ヒステリシス・チョッパーで用いられるヒステリシス値です。  chm = 1の場合： OFFSET = サイン波のオフセット %0000~%1111： オフセットは-3、-2、-1、0、1、…、12 これは、サイン波のオフセットで、この値の1/512が各サイン波項目の絶対値に加算されます。

ビットおよび名称	タイプおよびリセット	説明
[6:4] HSTRT_TFD210	RW, unsigned 0x5	<p>chm = 0の場合： HENDに加算されるヒステリシス開始値HSTRT。</p> <p>%000~%111： ヒステリシスの低い値HENDに1、2、…、8が加算されます (この設定値の1/512が電流設定値に加算されます)</p> <p><b>注意：</b>実効HEND + HSTRT ≤ 16。 <b>ヒント：</b>ヒステリシスのデクリメントは16クロックごとに行われます。</p> <p>chm = 1の場合： TFD[2..0]高速減衰時間設定 高速減衰時間設定 (MSB : fd3) : %0000~%1111： N<sub>CLK</sub> = 32 × TFDの高速減衰時間設定値TFD (%0000 : 低速減衰のみ)</p>
[3:0] TOFF	RW 0x0	<p>オフ時間TOFFおよびドライバのイネーブル。</p> <p>オフ時間設定値は低速減衰フェーズの時間を制御します。 N<sub>CLK</sub> = 24 + 32 × TOFF %0000 : ドライバ・ディスエーブル、全ブリッジ・オフ %0001 : 1 – TBL ≥ 2の場合にのみ使用 %0010~%1111 : 2~15</p>
	0: DRIVER_OFF 1: TOFF_56 2: TOFF_88 3: TOFF_120 4: TOFF_152 5: TOFF_184 6: TOFF_216 7: TOFF_248 8: TOFF_280 9: TOFF_312 10: TOFF_344 11: TOFF_376 12: TOFF_408 13: TOFF_440 14: TOFF_472 15: TOFF_504	<p>ドライバ・ディスエーブル、全ブリッジ・オフ。</p> <p>低速減衰フェーズの時間 : 56 × tclk。 TBL ≥ 2の場合に使用。</p> <p>低速減衰フェーズの時間 : 88 × tclk。</p> <p>低速減衰フェーズの時間 : 120 × tclk。</p> <p>低速減衰フェーズの時間 : 152 × tclk。</p> <p>低速減衰フェーズの時間 : 184 × tclk。</p> <p>低速減衰フェーズの時間 : 216 × tclk。</p> <p>低速減衰フェーズの時間 : 248 × tclk。</p> <p>低速減衰フェーズの時間 : 280 × tclk。</p> <p>低速減衰フェーズの時間 : 312 × tclk。</p> <p>低速減衰フェーズの時間 : 344 × tclk。</p> <p>低速減衰フェーズの時間 : 376 × tclk。</p> <p>低速減衰フェーズの時間 : 408 × tclk。</p> <p>低速減衰フェーズの時間 : 440 × tclk。</p> <p>低速減衰フェーズの時間 : 472 × tclk。</p> <p>低速減衰フェーズの時間 : 504 × tclk。</p>

## 0x6D : COOLCONF

ビットおよび名称	タイプおよびリセット	説明
[24] sfilt	RW 0x0	StallGuard2フィルタをイネーブル。
	0: FILT_DISABLED 1: FILT_ENABLED	<p>標準モード、StallGuard用高時間分解能。</p> <p>フィルタリング・モード、StallGuard信号は、モーターの極数の公差を補償するためにのみ、4フルステップごとに更新されます。</p>

ビットおよび名称	タイプおよびリセット	説明
[22:16] sgt	RW, unsigned 0x00	StallGuard2の閾値  この符号付きの値は、ストール出力用のStallGuard2レベルを制御し、読み出し用に最適な測定範囲を設定します。値を小さくすると感度が高くなります。ゼロはほとんどのモーターで使用できる開始値です。 -64~+63: 値を大きくするとStallGuard2の感度は低下し、ストールを示すために必要なトルクが大きくなります。
[15] seimin	RW 0x0	スマート電流制御用の最小電流。
	0: IRUN_DIV2  1: IRUN_DIV4	電流設定 (IRUN) の1/2 (StealthChopと共に使用する場合はIRUN ≥ 16であることが必要)。  電流設定 (IRUN) の1/4 (StealthChopと共に使用する場合はIRUN ≥ 28であることが必要)。
[14:13] sedn	RW 0x0	電流のダウン・ステップ速度  %00: 32個のStallGuard2ごとに1ステップずつ値は減少します。 %01: 8個のStallGuard2ごとに1ステップずつ値は減少します。 %10: 2個のStallGuard2ごとに1ステップずつ値は減少します。 %11: 1個のStallGuard2ごとに1ステップずつ値は減少します。
	0: STEP_DOWN_EACH_32 1: STEP_DOWN_EACH_8 2: STEP_DOWN_EACH_2 3: STEP_DOWN_EACH_1	32個のStallGuard2/StallGuard4ごとに1ステップずつ値は減少します。 8個のStallGuard2/StallGuard4ごとに1ステップずつ値は減少します。 2個のStallGuard2/StallGuard4ごとに1ステップずつ値は減少します。 1個のStallGuard2/StallGuard4ごとに1ステップずつ値は減少します。
[11:8] semax	RW, unsigned 0x0	スマート電流制御用のStallGuard2ヒステリシス値  StallGuard2の結果が前記の(SEMIN + SEMAX + 1) × 32以上の場合、エネルギー節約のためモーター電流が減少します。 %0000 ... %1111: 0 ... 15
[6:5] seup	RW 0x0	電流アップ・ステップ幅  StallGuard2測定値ごとの電流インクリメント・ステップ。
	0: STEP_UP_1 1: STEP_UP_2 2: STEP_UP_4 3: STEP_UP_8	Stallguard2/4値ごとに1ずつインクリメント Stallguard2/4値ごとに2ずつインクリメント Stallguard2/4値ごとに4ずつインクリメント Stallguard2/4値ごとに8ずつインクリメント
[3:0] semin	RW, unsigned 0x0	スマート電流制御用および電流イネーブルのための最小StallGuard2値。  StallGuard2の結果がSEMIN × 32未満になった場合、モーター負荷角を減少するためにモーター電流が増加します。 %0000: スマート電流制御CoolStepをオフ %0001 ... %1111: 1 ... 15

**0x6E : DCCTRL**

DcStep (DC) 自動整流設定レジスタ (VDCMIN を使用してイネーブル)。

**ヒント:** 高マイクロステップ分解能または補間操作を用いると、DcStep の供給する StallGuard 信号が向上します。

vhighfs が有効化されている場合、VHIGH を超えている場合でも DC\_SG を使用できます。最高の結果を得るには、vhighchm をセットしてください。

ビットおよび名称	タイプおよびリセット	説明
[23:16] DC_SG	RW, unsigned 0x00	DcStepモードでのDcStep StallGuard2を用いたステップ・ロス検出のための最大PWMオン時間 (DC_SG × 16/f <sub>CLK</sub> )。  DC_TIME/16よりわずかに大きな値に設定します。  0 = ディスエーブル
[9:0] DC_TIME	RW, unsigned 0x000	整流のためのPWMオン時間上限値 (DC_TIME × 1/f <sub>CLK</sub> )。実効ブランキング時間TBLよりわずかに大きい値に設定します。

**0x6F : DRV\_STATUS**

ビットおよび名称	タイプおよびリセット	説明
[31] stst	R 0x0	停止インジケータ  このフラグは、各動作モードでモーターが停止していることを示します。これは最後のステップ・パルスから2 <sup>20</sup> (20)クロック後に生じます。
	0: INACTIVE 1: ACTIVE	モーターは動作中 モーターは停止中
[30] olb	R 0x0	位相Bの開放負荷インジケータ。 <b>ヒント:</b> これは単なる情報フラグです。ドライバは、このフラグに基づく処理は行いません。高速動作時および停止時に誤検出が生じる可能性があります。低速動作時にのみチェックしてください。
	0: OPERATIONAL 1: OPEN_LOAD	通常動作 位相Bで開放負荷を検出
[29] ola	R 0x0	位相Aの開放負荷インジケータ
	0: OPERATIONAL 1: OPEN_LOAD	通常動作 位相Aで開放負荷を検出。 <b>ヒント:</b> これは単なる情報フラグです。ドライバは、このフラグに基づく処理は行いません。高速動作時および停止時に誤検出が生じる可能性があります。低速動作時にのみチェックしてください。
[28] s2gb	R 0x0	位相Bの地絡インジケータ。ドライバはディスエーブルされます。このフラグは、ソフトウェアによる (TOFF = 0) か DRV_ENN 入力によってドライバがディスエーブルされるまで有効なままです。
	0: 0 1: 1	通常動作 位相Bで地絡を検出。ドライバはディスエーブルされます。

ビットおよび名称	タイプおよびリセット	説明
[27] s2ga	R 0x0	位相Aの地絡インジケータ。ドライバはディスエーブルされます。このフラグは、ソフトウェアによる (TOFF = 0) か DRV_ENN 入力によってドライバがディスエーブルされるまで有効なままです。
	0: 0 1: 1	通常動作 位相Aで地絡を検出。ドライバはディスエーブルされます。
[26] otpw	R 0x0	過熱事前警告フラグ
	0: INACTIVE 1: ACTIVE	通常動作 過熱事前警告の閾値を上回っている。過熱事前警告フラグは両ブリッジで共通です。
[25] ot	R 0x0	過熱フラグ
	0: OPERATIONAL 1: ERROR	通常動作 過熱制限値に到達。ICが冷えるまでドライバはディスエーブルされます。過熱フラグは両ブリッジで共通です。
[24] stallguard	R 0x0	StallGuard2/StallGuard4のステータス
	0: INACTIVE 1: ACTIVE	通常動作 StallGuard2 (SpreadCycleで動作中)、StallGuard4 (StealthChop2で動作中)、またはDcStepストール (DcStepモードで動作中) によってモーター・ストールを検出。
[20:16] CS_ACTUAL	R, unsigned 0x00	実際のモーター電流/スマート・エネルギー電流 実際の電流制御スケールリング、COOLCONFレジスタの設定により制御されるスマート・エネルギー電流スケールリングのモニタリング用、または自動電流スケールリング機能のモニタリング用。
[15] fsactive	R 0x0	フルステップ・アクティブ・インジケータ
	0: USTEP 1: FSTEP	マイクロステップがアクティブ チョッパー・モード設定および速度閾値による定義に従って、ドライバがフルステップに切り替わったことを表示。
[14] stealth	R 0x0	StealthChop2インジケータ
	0: SPREADCYCLE_CTOFF 1: STEALTHCHOP	StealthChop2は非アクティブ ドライバはStealthChop2モードで動作
[13] s2vsb	R 0x0	位相Bの電源への短絡インジケータ。ドライバはディスエーブルされます。このフラグは、ソフトウェアによる (TOFF = 0) か DRV_ENN 入力によってドライバがディスエーブルされるまで有効なままです。
	0: OPERATIONAL 1: ERROR	エラーなし 位相Bで電源への短絡を検出。ドライバはディスエーブルされます。
[12] s2vsa	R 0x0	位相Aの電源への短絡インジケータ。ドライバはディスエーブルされます。このフラグは、ソフトウェアによる (TOFF = 0) か DRV_ENN 入力によってドライバがディスエーブルされるまで有効なままです。
	0: OPERATIONAL 1: ERROR	エラーなし 位相Aで電源への短絡を検出。ドライバはディスエーブルされます。

ビットおよび名称	タイプおよびリセット	説明
[9:0] SG_RESULT	R, unsigned 0x000	<p>StallGuard2の結果、StallGuard4の結果（実際のチョッパー・モードに依存）、またはSpreadCycleを使用しモーター温度検出のために停止しているコイルAのPWMオン時間。</p> <p><b>機械的負荷測定：</b> StallGuard2/StallGuard4の結果は、機械的なモーター負荷を測定する手段を与えます。値が大きいと機械的負荷が小さいことを意味します。StallGuard2では、値が0の場合負荷が最大であることを示します。最適なSGT設定値を用いた場合、これはモーター・ストールのインジケータとなります。ストール検出では、SG_RESULTと0を比較してストールの検出を行います。SG_RESULTは、これをプログラマブルな上限値および下限値と比較することで、CoolStep動作の基盤として使用されます。StealthChop2モードでは使用できません。</p> <p>StallGuard2は、マイクロステップ動作またはDcStepで最もよく機能します。</p> <p>SpreadCycleモード時の温度測定： 停止時には、StallGuard2の結果は取得できません。代わりに、SG_RESULTは、モーター・コイルAのチョッパー・オン時間を示します。モーターを定められたマイクロステップ位置に特定の電流設定値で動作させると、チョッパー・オン時間を読み出すことでモーター温度を大まかに推定できます。モーター温度が上昇するにつれ、コイル抵抗が増し、チョッパー・オン時間も増加します。StallGuard4の仕様については、SG4_RESULTを参照してください。</p>

## 0x70: PWMCONF

ビットおよび名称	タイプおよびリセット	説明
[31:28] PWM_LIM	RW, unsigned 0xC	<p>スイッチ・オン時のPWM自動スケール振幅制限。</p> <p>SpreadCycleからStealthChop2にスイッチ・バックする場合のPWM_SCALE_AUTOの制限値。この値は、スイッチ・バック時の自動電流制御のビット7~4に対する上限値を定義します。StealthChop2へのモード切り替え時の電流ジャークを低減するよう設定できます。</p> <p>PWM_GRADまたはPWM_GRAD_AUTOオフセットは制限しません。 (デフォルト= 12)</p>
[27:24] PWM_REG	RW, unsigned 0x4	<p>レギュレーション・ループの勾配</p> <p>pwm_autoscale = 1を使用した場合の、ユーザ定義による半波あたりの最大PWM振幅変化。(1~15)：</p> <p>1 : 0.5インクリメント (最遅レギュレーション) 2 : 1インクリメント 3 : 1.5インクリメント 4 : 2インクリメント (リセット後のデフォルト) ... 8 : 4インクリメント ... 15 : 7.5インクリメント (最速レギュレーション)</p>

ビットおよび名称	タイプおよびリセット	説明
[23] pwm_dis_reg_stst	RW 0x0	1 = モーターの停止時および電流低下 (IRUN未満) 時に電流レギュレーションをディスエーブル。このオプションは、停止時のレギュレーション・ノイズを抑制します。
	0: CTRL_ACTIVE 1: CTRL_INACTIVE	電流レギュレーションはアクティブ モーターの停止時および電流低下 (IRUN未満) 時に電流レギュレーションをディスエーブル。このオプションは、停止時のレギュレーション・ノイズを抑制します。
[22] pwm_meas_sd_enable	RW 0x0	低速減衰フェーズのローサイド電流測定制御。
	0: DISABLED 1: ENABLED	低速減衰時のローサイド電流測定をディスエーブル。 ローサイドの低速減衰フェーズを使用してモーターを測定。電流制限の下限値より低い電流が可能になります。
[21:20] FREEWHEEL	RW 0x0	様々な停止モードを可能にします モーター電流設定値がゼロ (I_HOLD = 0) の場合の停止オプション。
	0: NORMAL 1: FREEWHEEL 2: LS_SHORT 3: HS_SHORT	通常動作 フリーホイールリング LSドライバを用いたコイル短絡 HSドライバを用いたコイル短絡
[19] pwm_autograd	RW 0x1	PWM自動勾配調整
	0: FIXED 1: AUTO	PWM_GRADに固定値を使用 (PWM_GRAD_AUTO = PWM_GRAD) 自動調整 (pwm_autoscale = 1の場合のみ) (リセット後のデフォルト)。pwm_autograd = 0の間、PWM_GRAD_AUTOはPWM_GRADで初期化され、動作中は自動的に最適化されます。前処理としてPWM_OFS_AUTOは自動的に初期化されます。これには、IRUNで130msを超える時間停止している必要があり、a) 停止の検出、b) IRUNで128チョッパー・サイクルを超える時間待機、c) PWM_OFS_AUTOのレギュレーションが行われます。これは、+/-1の変化につき8フルステップでPWM_GRAD_AUTOを調整するために、-1 PWM_SCALE_AUTOのモーター動作と、 $1.5 \times \text{PWM\_OFS\_AUTO} \times (\text{IRUN} + 1) / 32$ PWM_SCALE_SUM PWM_OFS_AUTO $\times (\text{IRUN} + 1) / 32$ および PWM_SCALE_SUMの時間を必要とするためです。また、PWM_OFS_AUTOを調整するためにチョッパー周波数を低減して使用することもできます。
[18] pwm_autoscale	RW 0x1	PWM自動振幅スケールリング
	0: USER 1: AUTO	ユーザによる定義によるフィードフォワードPWM振幅。電流設定IRUNおよびIHOLDは影響を及ぼしません。得られるPWM振幅 (0...255に制限) は次式で表されます。 $\text{PWM\_OFS} \times ((\text{CS\_ACTUAL} + 1) / 32) + \text{PWM\_GRAD} \times 256 / \text{TSTEP}$ 自動電流制御を有効化 (リセット後のデフォルト)。
[17:16] PWM_FREQ	RW 0x0	PWM周波数の選択。
	0: FCLK_2DIV1024 1: FCLK_2DIV683 2: FCLK_2DIV512 3: FCLK_2DIV410	$f_{\text{PWM}} = 2/1024 f_{\text{CLK}}$ $f_{\text{PWM}} = 2/683 f_{\text{CLK}}$ $f_{\text{PWM}} = 2/512 f_{\text{CLK}}$ $f_{\text{PWM}} = 2/410 f_{\text{CLK}}$

ビットおよび名称	タイプおよびリセット	説明
[15:8] <b>PWM_GRAD</b>	RW, unsigned 0x00	<p>PWM振幅の速度依存勾配：  <math>PWM\_GRAD \times 256/TSTEP</math>            この値は、速度に依存したモーター・バックEMFを補償するためにPWM_OFSに            加算されます。</p> <p>自動スケーリングの初期値としてPWM_GRADを用いると、自動調整プロセスを            加速できます。これを行うには、pwm_autoscale = 0とし、PWM_GRADを定め            られたアプリケーション固有の値に設定します。その後のみ、pwm_autoscale            = 1に設定します。終了後はStealthChop2をイネーブルします。</p> <p><b>ヒント：</b>初期調整の後、必要な初期値はPWM_GRAD_AUTOで読み出せます。</p>
[7:0] <b>PWM_OFS</b>	RW, unsigned 0x1D	<p>停止時のモーターのフル電流（CS_ACTUAL = 31）に関連したユーザ定義による            PWM振幅オフセット（0~255）。            （リセット後のデフォルト= 30）</p> <p>自動スケーリングの初期値としてPWM_OFSを用いると、自動調整プロセスを加            速できます。これを行うには、pwm_autoscale = 0とし、PWM_OFSを定められ            たアプリケーション固有の値に設定します。その後のみ、pwm_autoscale = 1            に設定します。終了後はStealthChop2をイネーブルします。</p> <p>PWM_OFS = 0とすると、モーター電流をモーター固有の下限測定閾値未満にス            ケーリング・ダウンできなくなります。この設定は特定の条件下でのみ、すなわ            ち電源電圧が2倍もしくはそれ以上変動する可能性がある場合にのみ、用いてく            ださい。それにより、モーターがレギュレーションから外れるのを防止できま            すが、同時に、レギュレーション制限値未満にパワー・ダウンすることができな            くなります。</p> <p>PWM_OFS &gt; 0とすると、PWMデューティサイクルを自動的にスケール・ダウン            できます。この場合、下限レギュレーション閾値未満にすることも可能です。そ            れにより、実際の（ホールド）電流スケール（IHOLD_IRUNレジスタ）に基づい            て低（停止）電流設定値にすることができます。</p>

**0x71 : PWM\_SCALE**

StealthChop2 振幅レギュレータの結果。これらの値は、自動PWM振幅スケーリングをモニタするために使用できます（255=最大電圧）。

ビットおよび名称	タイプおよびリセット	説明
[24:16] <b>PWM_SCALE_AUTO</b>	R, unsigned 0x000	
[9:0] <b>PWM_SCALE_SUM</b>	R, unsigned 0x000	<p>ビット：9~0：[0~1023]PWM_SCALE_SUM：実際のPWMデューティサイク            ル。この値は、サイン波テーブルから読み出されるCUR_AとCUR_Bの値をス            ケーリングするために用いられます。1023：最大デューティサイクル。この値            は、より高精度のデューティサイクル読出しのために、2ビット[1,0]に拡張でき            ます。ビット9...2は、他のPWMデューティサイクル関連レジスタの8ビット値に            対応します。</p>

**0x72 : PWM\_AUTO**

これらの自動生成値は、PWM\_GRAD と PWM\_OFS のデフォルト/パワー・アップ設定を決定するために読み出すことができます。

ビットおよび名称	タイプおよびリセット	説明
[23:16] PWM_GRAD_AUTO	R, unsigned 0x00	自動的に決定される勾配値
[7:0] PWM_OFS_AUTO	R, unsigned 0x00	自動的に決定されるオフセット値

**0x74 : SG4\_CONF**

ビットおよび名称	タイプおよびリセット	説明
[9] sg_angle_offset	RW 0x1	1 : TPWMTHRSを介した制御によりStealthChop2からSpreadCycleにスイッチングする際の、StallGuard4に基づく自動位相シフト補償。
	0: DISABLE 1: ENABLE	位相シフトのための補償なし 位相シフトを補償（推奨）
[8] sg4_filt_en	RW 0x0	SG4フィルタをイネーブル。 1 : SG4_RESULTはSG4_IND_0、_1、_2、_3の平均値
	0: FILT_DISABLE 1: FILT_ENABLE	SG4フィルタをディスエーブルします SG4フィルタをイネーブルします。
[7:0] SG4_THRS	RW, unsigned 0x00	ストールの検出閾値。StallGuard4値SG4_RESULTがこの閾値と比較されます。ストールが示されるのは以下の場合です。 SG4_RESULT ≤ SG4_THRS SG4_THRSは可能なSG4_RESULTの範囲の半分を占めます

**0x75 : SG4\_RESULT**

ビットおよび名称	タイプおよびリセット	説明
[9:0] SG4_RESULT	R, unsigned 0x000	StallGuard4専用のStallGuard結果。  SG4_RESULTは、TCOOLTHRSおよびSG4THRSとは無関係に、フルステップごとに更新されます。値が高いと、モーター負荷が低くトルク・ヘッドルームが大きいことを意味します。 StealthChop2モードのみを対象としたものです。ビット9およびビット0は常に0を示します。10ビットにスケールリングされているのはStallGuard2との互換性を確保するためです。

## 0x76 : SG4\_IND

ビットおよび名称	タイプおよびリセット	説明
[31:24] SG4_IND_3	R, unsigned 0x00	SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定3の値を示します。
[23:16] SG4_IND_2	R, unsigned 0x00	SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定2の値を示します。
[15:8] SG4_IND_1	R, unsigned 0x00	SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定1の値を示します。
[7:0] SG4_IND_0	R, unsigned 0x00	SG4測定の値を示します SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定0の値を示します。

## 型番

PART NUMBER	TEMPERATURE RANGE	PIN-PACKAGE
TMC5241ATU+	-40°C to +125°C	38 TQFN - 5mm x 7mm
TMC5241ATU+T	-40°C to +125°C	38 TQFN - 5mm x 7mm

+は鉛 (Pb) フリー/ROHS 適合のパッケージであることを示します。

Tはテープのリール巻きを示します。

## 改訂履歴

版数	改訂日	説明	改訂ページ
0	08/25	初版発行	-

この製品のデータシートに間違いがありましたので、お詫びして訂正いたします。  
この正誤表は、2026年1月7日現在、アナログ・デバイセズ株式会社で確認した誤りを記したものです。

なお、英語のデータシート改版時に、これらの誤りが訂正される場合があります。

正誤表作成年月日： 2026年1月7日

製品名： TMC5241

対象となるデータシートのリビジョン(Rev)： Rev.0

訂正箇所： 103頁、上の表、ENCAの欄内の説明

## 【誤】

	I. HIGH	DRV_ENIN is logic level high
[3] ENCA	R 0x0	A-channel state
	0: LOW 1: HIGH	ENCA is logic level low ENCB is logic level high

## 【正】

ENCB is logic level high ⇒ ENCA is logic level high