



## TMC5271

2.1V~20V、1.6A<sub>RMS</sub> の  
ステッピング・モータ・ドライバおよびコントローラ

## 概要

TMC5271 は、小型でスマートな高性能 1 軸ステッピング・モータ・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 および UART) および豊富な診断機能を備えています。最高の集積度、高いエネルギー効率、小さなフォーム・ファクタという特長により、コスト効率の高いソリューション向けの小型でスケラブルなシステムを実現します。

StealthChop2™ が、最大の効率と最適なモータ・トルク制御に加え、まったくノイズのない動作を確実に実現します。

TMC5271 は、256 マイクロステップの内蔵インデクサに基づくスマート・ステッピング・モータ・ドライバにより自動ポジショニングを実現するジャークの少ない柔軟な 8 点ランプ・ジェネレータに、20V、2.24A<sub>MAX</sub> の全機能内蔵型 H ブリッジ 2 個、および大きな外部センス抵抗が不要なためスペースと消費電力の節約ができるエネルギー散逸のない内蔵電流検出 (ICS) 機能を組み合わせたものです。

内蔵パワー-MOSFET は低インピーダンスであるため、高い駆動効率を実現すると共に発熱を最小限に抑えます。

合計の R<sub>DS(ON)</sub> (ハイサイド+ローサイド) の代表値は 0.155Ω で、H ブリッジあたりの最大実効値電流 I<sub>RMS</sub> は 1.6A<sub>RMS</sub>、H ブリッジあたりの最大出力電流 I<sub>MAX</sub> は 3.0A<sub>MAX</sub> (過電流保護により制限) です。

TMC5271 は豊富な診断機能と、短絡保護/過電流保護、サーマル・シャットダウン、低電圧ロックアウトなどの保護機能を備えています。

TMC5271 は、2.97mm × 3.13mm の小型 36 ピン WLCSP パッケージを採用しています。

## アプリケーション

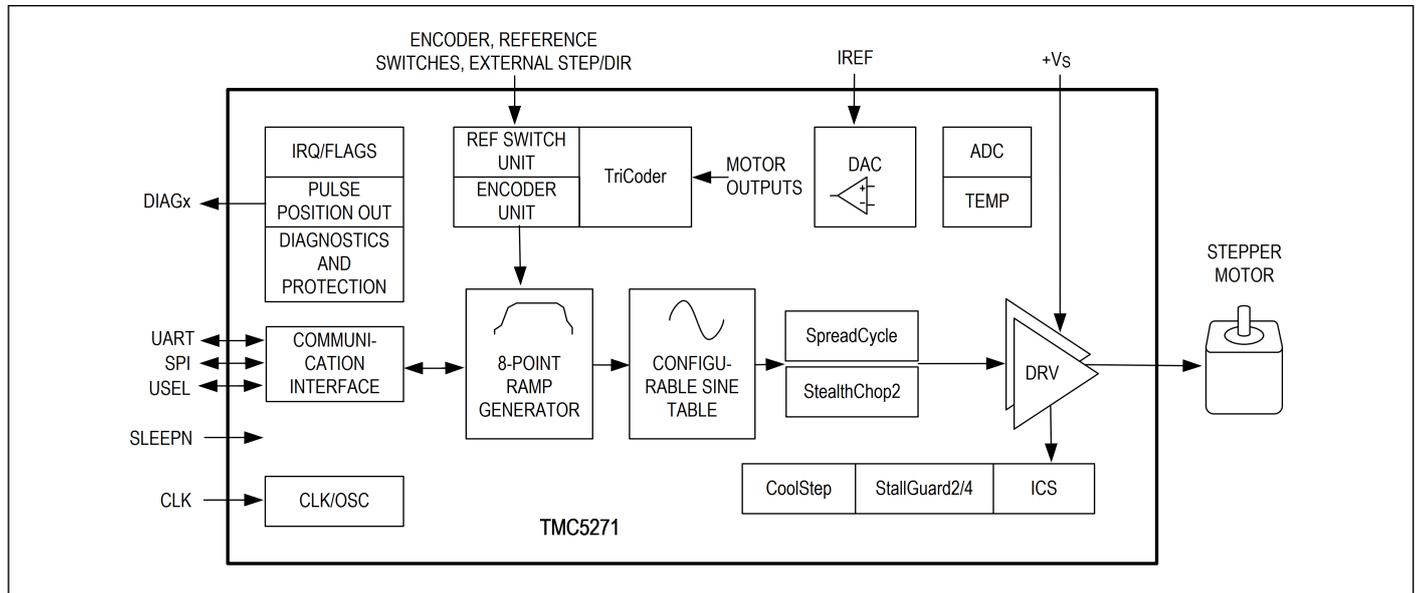
- ウェアラブル・デバイス
- 個人向けポータブル機器
- 光学システム、レンズ制御
- CCTV、監視、会議システムのフォーカス/ズームおよびパン/チルト制御
- パッチおよびインスリン・ポンプ
- ピペットおよびリキッド・ハンドリング
- 小型の印刷およびスキャン機器
- ラボおよびオフィス・オートメーション
- スペースに制約のあるステッピング・モータ・アプリケーション

## 利点と特長

- DC 電圧範囲 : 2.1V~20V
- 2 個の 20V H ブリッジ
- 低 R<sub>DS(ON)</sub> (ハイサイド (HS) + ローサイド (LS)) : 0.155Ω (代表値、T<sub>A</sub> = +25°C)
- H ブリッジあたりの電流定格 (+25°C で代表値) :
  - ブリッジ・ピーク電流 I<sub>MAX</sub> = 3.0A<sub>MAX</sub>
  - フルスケール電流 IFS = 1.6A<sub>RMS</sub> (2.24A のサイン波ピーク)
- 全機能内蔵無損失電流検出 (ICS)
- 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
- MicroPlyer™ ステップ補間機能を備えたステップ/方向インターフェース
- SPI または単線式 UART インターフェース
- 直交エンコーダ・インターフェース
- リファレンス・スイッチ入力
- TriCoder センサーレス停止ステップロス検出およびフル・ステップ・エンコーダ
- フル・ステップあたり 256 マイクロステップの最大分解能
- StealthChop2 によるサイレント・モータ動作
- SpreadCycle™ による高ダイナミック・チョッパー・モード
- StallGuard2™ および StallGuard4™ によるセンサー不要のモータ負荷検出
- 最大 75% のエネルギー削減を実現する CoolStep® 電流制御
- 受動ブレーキおよびフリーホイール・モード
- ブリッジのディスエーブル、リセット、低消費電力モード
- モータ位相とチップ温度の測定
- すべてを網羅する保護および診断機能
- 2.97mm × 3.13mm、36 ピン WLCSP

型番はデータシート末尾に記載されています。

簡略化したブロック図



## 目次

概要.....	1
アプリケーション.....	1
利点と特長.....	1
簡略化したブロック図.....	2
絶対最大定格.....	10
パッケージ情報.....	10
36 WLCSP .....	10
電気的特性.....	10
ピン配置.....	15
36 WLCSP .....	15
端子説明.....	15
機能図.....	18
TMC5271 .....	18
詳細.....	19
動作原理.....	19
一軸モーション・コントローラおよびドライバ.....	19
主要コンセプト.....	20
制御インターフェース.....	21
モータの運動と制御.....	21
内蔵 8 点モーション・コントローラ .....	21
ステップおよび方向モード.....	22
自動停止パワーダウンおよびパワーアップ .....	22
StealthChop2 および SpreadCycle ドライバ.....	22
StallGuard2 および StallGuard4—機械的負荷のセンシング .....	23
CoolStep—負荷適応型電流制御 .....	23
エンコーダ・インターフェース .....	23
停止ステップロス検出.....	23
SPI.....	23
SPI データグラム構造 .....	23
読出し/書込みの選択 (WRITE_notREAD) .....	24
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.....	25
データ・アライメント.....	25
SPI 信号.....	25
SPI のタイミング.....	25
UART 単線式インターフェース.....	26
UART データグラム構造.....	26
UART 書込みアクセス .....	26
UART 読出しアクセス .....	27
UART 信号.....	28

## 目次 (続き)

CRC の計算.....	28
CRC 計算の C コード例.....	28
複数ノードのアドレス指定.....	29
UART デイジーチェーン接続.....	29
UART リング・モード.....	30
ステップ/方向インターフェース.....	31
タイミング.....	31
分解能の変更.....	32
MicroPlyer ステップ・インターポレータおよび停止検出.....	33
StealthChop2.....	34
自動調整.....	34
StealthChop2 のオプション.....	36
StealthChop2 電流レギュレータ.....	37
下限電流値.....	40
速度ベースのスケールリング.....	40
モータの逆起電力 (BEMF) 定数の解釈.....	42
StealthChop2 と SpreadCycle の結合.....	42
StealthChop2 でのフラグ.....	44
開放負荷フラグ.....	44
PWM_SCALE_SUM からのモータ状態に関する情報.....	44
フリーホイールリングおよび受動ブレーキング.....	44
StealthChop2 を制御するパラメータ.....	45
SpreadCycle および一般的なチョッパ.....	47
SpreadCycle チョッパ.....	48
一般的な定オフ時間チョッパ.....	50
内蔵電流検出機能 (ICS).....	51
モータ電流の設定.....	51
フルスケール電流の設定.....	53
速度ベースのモード制御.....	55
ランプ・ジェネレータ.....	58
現実の単位への変換.....	58
動作プロファイル.....	59
ランプ・モード.....	59
8点ランプ.....	61
速度モード.....	64
ランプの早期終了.....	64
アプリケーション例：ジョイスティック制御.....	65
速度スレッシュホールド.....	65
リファレンス・スイッチ.....	66

## 目次 (続き)

仮想リファレンス・スイッチ .....	68
ランプ・ジェネレータの応答時間 .....	68
ポジション比較機能 .....	69
エンコーダを用いたクローズド・ループ・ポジション制御 .....	69
StallGuard2 の負荷測定 .....	70
StallGuard2 のスレッシュホールド SGT の調整 .....	72
TCOOLTHRS および THIGH を制限する可変速度 .....	73
高トルク・リップルと共振を伴う小型モータ .....	73
モータ・コイル抵抗の温度依存性 .....	73
StallGuard2 測定の正確さと再現性 .....	74
StallGuard2 の更新レートおよびフィルタ .....	74
モータ・ストールの検出 .....	74
StallGuard2 動作の制限 .....	74
StallGuard4 の負荷測定 .....	74
StallGuard4 の調整 .....	76
StallGuard4 の更新レート .....	76
モータ・ストールの検出 .....	77
StallGuard4 動作の制限 .....	77
StallGuard2 および StallGuard4 を使用したホーミング .....	77
CoolStep の負荷適応型電流スケーリング .....	77
CoolStep 用の設定 .....	77
CoolStep の調整 .....	79
応答時間 .....	79
低速度およびスタンバイ動作 .....	79
診断出力およびステータス出力 .....	80
DcStep .....	81
DcStep のデザインイン .....	81
DcStep とモーション・コントローラの統合 .....	82
DcStep モードにおけるストール検出 .....	83
DcStep 動作での実際のモータ速度の測定 .....	84
サイン波ルックアップ・テーブル .....	84
マイクロステップ・テーブル .....	85
TriCoder—逆起電力センサーレス停止ステップロス検出 .....	87
TriCoder BEMF デコーダの動作原理 .....	87
TriCoder 動作に対するモータおよび速度の条件 .....	88
イネーブルまでの時間 .....	89
関連設定 .....	89
ABN インクリメンタル・エンコーダ・インターフェース .....	89
モータの分解能に合うエンコーダ設定 .....	91

## 目次 (続き)

チップの緊急停止およびパワーダウン・モード.....	91
SPI または UART を使用した緊急停止 .....	92
停止オプションを用いた緊急停止 .....	93
ハード緊急停止.....	93
低消費電力自己消費モード.....	93
外部リセットおよびスリープ・モード.....	93
クロック発振器とクロック入力.....	94
内部クロックの使用.....	94
外部クロックの使用.....	94
保護およびドライバ診断.....	94
過熱保護およびサーマルシャットダウン .....	94
モータ温度の測定.....	94
短絡保護.....	95
開放負荷診断.....	95
低電圧ロックアウト保護.....	95
ESD 保護.....	95
クイック設定ガイド.....	95
電流設定.....	96
StealthChop2 設定 .....	97
SpreadCycle 設定 .....	98
CoolStep を StealthChop2 と組み合わせてイネーブル .....	99
CoolStep を SpreadCycle と組み合わせてイネーブル.....	100
モーション・コントローラを用いたモータの作動.....	101
DcStep 動作のイネーブル .....	104
モータの適合化.....	105
レジスタ・マップ.....	107
TMC5271 .....	107
レジスタの詳細.....	115
標準アプリケーション回路.....	182
標準的なアプリケーション回路 .....	182
ドライバ保護および EME 回路 .....	182
型番.....	184
改訂履歴.....	185

## 図一覧

図 1. 一軸モーション・コントローラおよびドライバ .....	20
図 2. 停止時およびランプアップ時の自動モータ電流制御 .....	22
図 3. SPI データグラム構造 .....	24
図 4. SPI タイミング図 .....	26
図 5. UART 書込みアクセスのデータグラム構造 .....	26
図 6. UART 読出しアクセス要求データグラムの構造 .....	27
図 7. UART 読出しアクセス返答データグラムの構造 .....	28
図 8. UART のデジタイズチェーン接続例 .....	30
図 9. UART リング・モードの配線例 .....	31
図 10. STEP/DIR 信号のタイミング .....	32
図 11. STEP/DIR 信号の入力フィルタ構造 .....	32
図 12. ステップ信号周波数が増加する場合の MicroPlyer マイクロステップ補間 (例: 16~256) .....	34
図 13. StealthChop2 の自動調整手順 .....	36
図 14. StealthChop2 : PWM_REG の設定は良好 .....	38
図 15. StealthChop2 : AT#2 の間の PWM_REG の設定が過小 .....	38
図 16. 正しく決定された PWM_GRAD(AUTO)および PWM_OFS(AUTO) .....	39
図 17. PWM_GRAD の過小設定の例 .....	39
図 18. 速度ベースの PWM スケーリング (pwm_autoscale = 0) .....	42
図 19. SpreadCycle への切り替えオプションのための TPWMTHRS .....	43
図 20. 代表的なチョッパー減衰フェーズ .....	47
図 21. 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 .....	49
図 22. コイル電流を表すオフセットを持つ一般的な定オフ時間チョッパー .....	50
図 23. 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 .....	51
図 24. 速度依存モードの選択 .....	56
図 25. 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 .....	60
図 26. ランプ・ジェネレータを用いた最適なモータ・トルク使用法を示す図 .....	61
図 27. 距離が短すぎるために VMAX に到達できない 8 点ランプ .....	62
図 28. 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 .....	62
図 29. 距離が短いために V1 に達しないプロファイル .....	63
図 30. 距離が短いために TVMAX を維持できないプロファイル .....	63
図 31. 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 .....	64
図 32. ランプ・ジェネレータの速度依存モータ制御 .....	66
図 33. リファレンス・スイッチの使用 (例) .....	67
図 34. 仮想停止スイッチと制限の図解 .....	68
図 35. StallGuard2 の機能原理 .....	71
図 36. サンプル・モータを用いた SGT の最適設定値と StallGuard2 の読出し値 .....	73
図 37. StallGuard4 の動作モード .....	75
図 38. モータ電流を負荷に適合させる CoolStep .....	78
図 39. DIAG0 の出力回路 .....	80

## 図一覧 (続き)

図 40. DIAG1 の出力回路 .....	81
図 41. DcStep により拡大されるアプリケーション動作領域 .....	82
図 42. 過負荷状態時の DcStep の速度プロファイル .....	83
図 43. LUT プログラミングの例 .....	85
図 44. OFFSET_SIN90 を使用したコサイン波のシフト .....	86
図 45. モータ動作の検知 .....	88
図 46. インクリメンタル・エンコーダの ABN 信号の概要 .....	90
図 47. クイック設定ガイド-電流設定 .....	96
図 48. クイック設定ガイド-StealthChop2 設定 .....	97
図 49. クイック設定ガイド-SpreadCycle .....	98
図 50. クイック設定ガイド-StealthChop2 を組み合わせた CoolStep .....	99
図 51. クイック設定ガイド-SpreadCycle を組み合わせた CoolStep .....	100
図 52. クイック設定ガイド-速度モードでのモータの作動 .....	101
図 53. クイック設定ガイド-目標ポジションへのモータの作動 .....	102
図 54. 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 .....	103
図 55. クイック設定ガイド-DcStep .....	104
図 56. クイック設定ガイド-DcStep でストール検出を使用 .....	105
図 57. TMC5271 の標準的なアプリケーション回路 .....	182
図 58. 簡単な ESD 対策 .....	183
図 59. モータ出力保護の強化 .....	184

## 表一覧

表 1. TMC5271 の主要コンセプト .....	20
表 2. SPI 読出し/書込みフローの例 .....	24
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 .....	25
表 4. TMC5271 UART のインターフェース信号 .....	28
表 5. 最大 255 ノードをアドレス指定する UART 例 .....	30
表 6. 位相 A/B のコイル電流用フル・ステップ/ハーフ・ステップのルックアップ・テーブル値 .....	33
表 7. StealthChop2 自動調整 AT#1 および AT#2 の制約事項と条件 .....	35
表 8. StealthChop2 の PWM 周波数の選択 .....	37
表 9. StealthChop2 を制御するパラメータ .....	45
表 10. 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 .....	48
表 11. SpreadCycle モードのヒステリシス・パラメータ .....	50
表 12. 定オフ時間チョッパ・モードを制御するパラメータ .....	51
表 13. モータ電流を制御するパラメータ .....	52
表 14. FSR_IREF 設定に基づく KIFS_IREF スケーリング係数 .....	53
表 15. I <sub>FS</sub> フルスケール・レンジ設定 (標準的な R <sub>REF</sub> = 10kΩ) .....	54
表 16. I <sub>FS</sub> フルスケール・レンジ設定 (R <sub>REF</sub> = 36kΩ の場合の例) .....	54
表 17. 速度ベースのモード制御パラメータ .....	57
表 18. ランプ・ジェネレータのパラメータと単位の関係 .....	58
表 19. X_COMPARE_REPEAT オプションと周期的なパルス動作 .....	69
表 20. クローズド・ループ・ポジション制御に関連するパラメータ .....	69
表 21. StallGuard2 関連のパラメータ .....	71
表 22. StallGuard4 関連のパラメータ .....	75
表 23. CoolStep の重要パラメータ .....	77
表 24. その他の CoolStep パラメータとステータス情報 .....	78
表 25. DcStep モードにおけるストール検出のパラメータ .....	83
表 26. 200 フル・ステップのモータに対して 256 マイクロステップを使用するエンコーダの設定例 .....	91
表 27. チップのモード比較 .....	91
表 28. フルスケール電流設定に基づく過電流保護スレッシュホールド .....	95

## 絶対最大定格

V <sub>S</sub> ~GND .....	-0.3V~22V	V <sub>CC10</sub> ~GND .....	-0.3V~5.5V
V <sub>DD1V8</sub> ~GND .....	-0.3V~min(2.2, V <sub>S</sub> + 0.3)V	ロジック入力電圧~GND .....	-0.3V~V <sub>CC10</sub> + 0.5V
PGND~GND .....	-0.3V~+0.3V	動作温度範囲.....	-40°C~+125°C
OA1、OA2、OB1、OB2 .....	-0.3V~V <sub>S</sub> + 0.3V	ジャンクション温度.....	+160°C
SLEEPN .....	-0.3V~V <sub>S</sub> + 0.3V	保存温度範囲 .....	-65°C~+150°C
IREF~GND .....	-0.3V~V <sub>DD</sub> + 0.3V	はんだ付け処理温度 (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。

## パッケージ情報

## 36 WLCSP

Package Code	W362A3+1
Outline Number	21-100643
Land Pattern Number	Refer to Application Note 1891
<b>Thermal Resistance, Four-Layer Board: JEDEC PCB, 2 x 2 Via Array, M1 Traces, M2/M3 90% Cu, M4 25% Cu</b>	
Junction to Ambient ( $\theta_{JA}$ )	33.22°C/W
Junction to Board ( $\theta_{JB}$ )	4.13°C/W
Junction to Case ( $\theta_{JC}$ )	0.19°C/W
Junction-to-Board Characterization Parameter ( $\psi_{JB}$ )	3.89°C/W
Junction-to-Top (of Package) Characterization Parameter ( $\psi_{JT}$ )	0.59°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、[www.maximintegrated.com/packages](http://www.maximintegrated.com/packages) 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、[www.maximintegrated.com/thermal-tutorial/](http://www.maximintegrated.com/thermal-tutorial/)を参照してください。

## 電気的特性

(V<sub>S</sub> = 2.1V~20V、R<sub>REF</sub> = 10k $\Omega$ ~60k $\Omega$ 、代表値はT<sub>A</sub> = +25°C、V<sub>S</sub> = 16V、SLEEPN = V<sub>S</sub>、V<sub>CC10</sub> = 3.3Vを前提としています。制限値はT<sub>A</sub> = +25°Cで100%テストされています。動作温度範囲および関連する電源電圧範囲における制限値は、設計および特性評価により裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>POWER SUPPLY</b>						
Supply Voltage Range	V <sub>S</sub>		2.1		20	V
Sleep Mode Current Consumption	I <sub>VS</sub>	V(SLEEPN) = 0, HV outputs pulled up		28n	2u	A
Quiescent Current Consumption	I <sub>VS</sub>	V(SLEEPN) = 1, qsc_sts_ena = 1, adc_en = 0, qsc_enc_en = 0		480	800	$\mu$ A
Driver Off Current Consumption	I <sub>VS</sub>	V(SLEEPN) = 1, qsc_sts_ena = 0		3.4	5	mA
1.8V Regulator Output Voltage	V <sub>VDD</sub>	Internally short-circuit protected, V <sub>S</sub> = 2.1V, I <sub>LOAD</sub> = 20mA		1.8		V
V <sub>DD</sub> Current Limit	I <sub>V18LIM</sub>	Internally short-circuit protected	20			mA

## 電气的特性（続き）

( $V_S = 2.1V \sim 20V$ 、 $R_{REF} = 10k\Omega \sim 60k\Omega$ 、代表値は  $T_A = +25^\circ C$ 、 $V_S = 16V$ 、 $SLEEPN = V_S$ 、 $V_{CC\_IO} = 3.3V$  を前提としています。制限値は  $T_A = +25^\circ C$  で 100%テストされています。動作温度範囲および関連する電源電圧範囲における制限値は、設計および特性評価により裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Logic I/O Supply	$V_{CC\_IO}$	$V_{CC\_IO}$ can be either 1.8V, 3.3V, or 5V level and is defined by the user application circuit	1.6		5.5	V
Sleep Mode Current Consumption	$I_{VCC}$	$V(SLEEPN) = 0$			1	$\mu A$
Quiescent Current Consumption	$I_{VCC}$	$V(SLEEPN) = 1$		36	70	$\mu A$
	$I_{SLEEPN}$			32	60	
<b>LOGIC LEVEL INPUTS/OUTPUTS</b>						
Input Voltage Level—High	$V_{IH}$		1.2			V
Input Voltage Level—Low	$V_{IL}$				0.65	V
Input Hysteresis	$V_{HYS}$			85		mV
Pull-Down Current	$I_{PD}$	To GND	6	13	22	$\mu A$
Pull-Up Current	$I_{PU}$	To $V_{DD}$	8	15	23	$\mu A$
Open-Drain Output Logic-Low Voltage	$V_{OL}$	$I_{LOAD} = 5mA$			0.4	V
Push-Pull Output Logic-High Voltage	$V_{OH}$	$I_{LOAD} = 5mA$			$V_{CC\_IO} - 400mV$	V
Open-Drain Output Logic-High Leakage Current	$I_{OH}$	$V(PIN) = 5.5V$ , $V_{CCIO} = 5.5V$	-1		+1	$\mu A$
SLEEPN Voltage Level High	$V_{IHSLEEPN}$		1.3			V
SLEEPN Voltage Level Low	$V_{ILSLEEPN}$				0.4	V
<b>OUTPUT SPECIFICATIONS</b>						
Output On-Resistance Low Side	$R_{DS(ON),LS}$	Full-scale bits = 00		0.19	0.375	$\Omega$
		Full-scale bits = 01		0.1	0.195	
		Full-scale bits = 10		0.07	0.135	
		Full-scale bits = 11		0.055	0.11	
Output On-Resistance High Side	$R_{DS(ON),HS}$			0.1	0.19	$\Omega$
Output Leakage	$I_{LEAK}$		-10		+10	$\mu A$
Output Slew Rate	SR			200		V/ $\mu s$
<b>PROTECTION CIRCUITS</b>						
Overcurrent Protection Threshold	OCP	Full-scale bits = 00	0.75			A
		Full-scale bits = 01	1.5			
		Full-scale bits = 10	2.25			
		Full-scale bits = 11	3			
Overcurrent Protection Blanking Time	TOCP		0.9	1.6	2.6	$\mu s$

## 電气的特性（続き）

( $V_S = 2.1V \sim 20V$ 、 $R_{REF} = 10k\Omega \sim 60k\Omega$ 、代表値は  $T_A = +25^\circ C$ 、 $V_S = 16V$ 、 $SLEEPN = V_S$ 、 $V_{CC\_IO} = 3.3V$  を前提としています。制限値は  $T_A = +25^\circ C$  で 100% テストされています。動作温度範囲および関連する電源電圧範囲における制限値は、設計および特性評価により裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
UVLO Threshold on $V_S$	UVLO	$V_S$ rising	1.7	1.8	1.9	V
UVLO Threshold on $V_S$ Hysteresis	UVLOHYS			0.11		V
UVLO Threshold on $V_{CC\_IO}$	UVLOVCC	$V_{CC\_IO}$ rising			1.3	V
UVLO Threshold on $V_S$ Hysteresis	UVLOVCCHYS			65		mV
Thermal-Protection Threshold Temperature	TSD			+165		$^\circ C$
Thermal-Protection Temperature Hysteresis				+20		$^\circ C$
<b>CURRENT REGULATION</b>						
IREF Pin Resistor Range	$R_{REF}$		10		60	k $\Omega$
IREF Output Voltage	$V_{REF}$		0.882	0.9	0.918	V
Full-Scale Current Constant	$K_{IFS}$	Value is in RMS	FSR_M = 0, FSR_IREF_M = 3	4.1		A x k $\Omega$
			FSR_M = 1, FSR_IREF_M = 3	8.16		
			FSR_M = 2, FSR_IREF_M = 3	12.06		
			FSR_M = 3, FSR_IREF_M = 3	16		
Current Trip Regulation Accuracy	DITRIP1	ITRIG from 15% to 100% FSR, $R_{REF} = 10k\Omega$ , FSR_IREF_M = 3	-4		+4	%
<b>FUNCTIONAL TIMING</b>						
Sleep Time	$t_{SLEEP}$	SLEEPN = 1 to OUT_x three-state			50	$\mu s$
Wake-Up Time from Sleep	$t_{WAKE}$	SLEEPN = 0 to normal operation		300	500	$\mu s$
Enable Time	$t_{EN}$	Time from CSN/AD2 pin rising edge to driver on			1	$\mu s$
Disable Time	$t_{DIS}$	Time from CSN/AD2 pin rising edge to driver off			1	$\mu s$
<b>CLOCK OSCILLATOR AND INPUT</b>						
Internal Clock Frequency	$f_{CLKOSC}$		11.9	12.5	13.2	MHz
External Clock Frequency	$f_{CLK}$		8	16	20	MHz
External Clock Duty Cycle	$t_{CLKL}$		40		60	%
External Clock Detection in Cycles			4		8	ns

## 電气的特性（続き）

( $V_S = 2.1V \sim 20V$ 、 $R_{REF} = 10k\Omega \sim 60k\Omega$ 、代表値は  $T_A = +25^\circ C$ 、 $V_S = 16V$ 、 $SLEEPN = V_S$ 、 $V_{CC_{IO}} = 3.3V$  を前提としています。制限値は  $T_A = +25^\circ C$  で 100%テストされています。動作温度範囲および関連する電源電圧範囲における制限値は、設計および特性評価により裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
External Clock Timeout Detection in Cycles of Internal $f_{CLKOSC}$			12		16	Cycles
External Clock Detection Lower Frequency Threshold			4			MHz
<b>SPI TIMING</b>						
SCK Valid Before or After Change of CSN	$t_{CC}$		$t_{CLK}$			ns
CSN High Time	$t_{CSH}$		$4 \times t_{CLK}$			ns
SCK Low Time	$t_{CL}$		20			ns
SCK High Time	$t_{CH}$		20			ns
SCK Frequency	$f_{SCK}$				10	MHz
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	$t_{DO}$	$V_{CC_{IO}} = 1.8V$ , $SPI\_FLT\_SEL = 00$		27	40	ns
SPI Input Filter	$t_{FILT}$	Rising and falling edge, $SPI\_FLT\_SEL = 10$		10		ns
<b>STEP/DIRECTION TIMING</b>						
STEP Frequency	$f_{STEP}$	Maximum microstep resolution	Dedge = 1		$f_{CLK}/4$	MHz
			Dedge = 0		$f_{CLK}/2$	
Full-Step Frequency	$f_{PS}$				$f_{CLK}/512$	MHz
STEP High Time	$t_{SH}$		$t_{CLK} + 20$			ns
STEP Low Time	$t_{SL}$		$t_{CLK} + 20$			ns
DIR to STEP Setup Time	$t_{DSU}$		20			ns
DIR to STEP Hold Time	$t_{DSH}$		20			ns
DIR/STEP to CLK Setup Time	$t_{SU}$		10			ns
DIR/STEP to CLK Hold Time	$t_{SH}$		10			ns
<b>ENCODER TIMING</b>						
Encoder Counting Frequency	$f_{CNT}$			$< 2/3 f_{CLK}$	$f_{CLK}$	MHz
ABN Input Low Time	$t_{ABNL}$		$3 \times t_{CLK} + 20$			ns

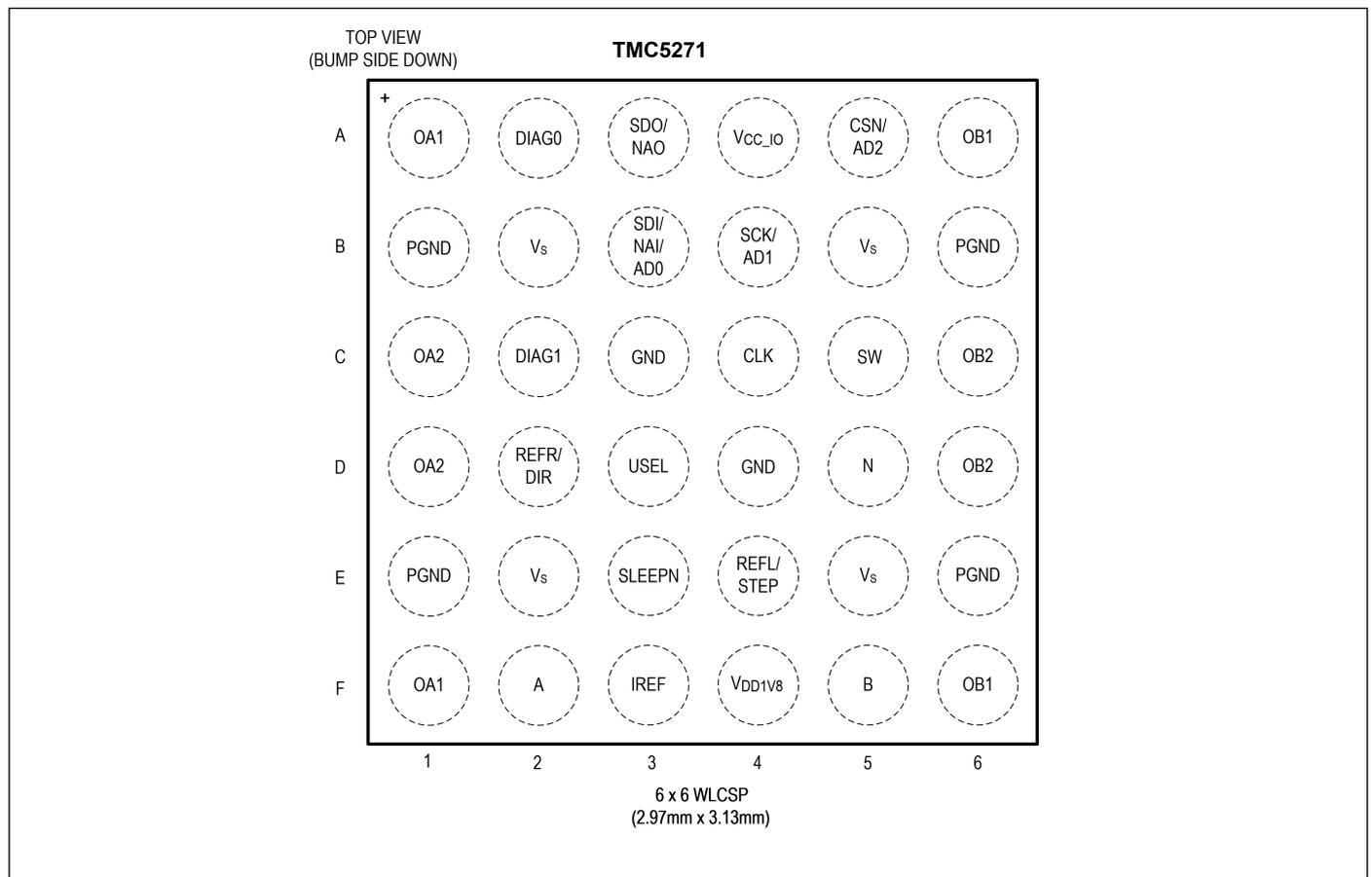
## 電气的特性（続き）

( $V_S = 2.1V \sim 20V$ 、 $R_{REF} = 10k\Omega \sim 60k\Omega$ 、代表値は  $T_A = +25^\circ C$ 、 $V_S = 16V$ 、 $SLEEPN = V_S$ 、 $V_{CC_{IO}} = 3.3V$  を前提としています。制限値は  $T_A = +25^\circ C$  で 100%テストされています。動作温度範囲および関連する電源電圧範囲における制限値は、設計および特性評価により裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
ABN Input High Time	$t_{ABNH}$		$3 \times t_{CLK} + 20$			ns
ABN Spike Filtering Time	$t_{FILTABN}$	Rising and falling edge	$3 \times t_{CLK}$			ns
<b>ADC/TEMPERATURE MEASUREMENT</b>						
ADC Resolution			8			Bit
Temperature Measurement Resolution	$T_{MEAS}$		2.5			$^\circ C$
Temperature Measurement Accuracy	$T_{SIGMA}$		1.5			$^\circ C$
ADC Sample Rate	$f_{SAMPLE,ADC}$	$f_{CLK,EXT} = 20MHz$ , $f_{CLK,INT} = 12.5MHz$	$f_{CLK,INT}/396$			MHz
<b>BEMF ENCODER</b>						
Pull-Down Resistance on OA1/OB1	$R_{PDBEMF}$	BEMF encoder activated	345 600			$\Omega$
BEMF Encoder Hysteresis	BEMFHYS	bemf_hyst = 0	8	10	12	mV
		bemf_hyst = 1	23	25	27	
		bemf_hyst = 2	47	50	53	
		bemf_hyst = 3	71	75	79	
		bemf_hyst = 4	95	100	105	
		bemf_hyst = 5	142	150	158	
		bemf_hyst = 6	190	200	210	
		bemf_hyst = 7	236	250	264	

## ピン配置

## 36 WLCSP



## 端子説明

ピン	名称	説明	リファレンス電源	タイプ
E2, E5, B2, B5	V <sub>s</sub>	モータの電源電圧。最も近い PGND ピン（または GND プレーンを使用）との間のピン近くに、短いループでフィルタリング・コンデンサを配置します。	-	Supply
F4	V <sub>DD1V8</sub>	内蔵 1.8V レギュレータの出力。最高性能を実現するには、GND 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
F3	IREF	電流スケール用のアナログ・リファレンス電流。GND との間に外部抵抗 R <sub>REF</sub> を配置します。	V <sub>DD1V8</sub>	Analog Input
A4	V <sub>CC_IO</sub>	回路の IO レベルを定めるために外部電源から供給されるデジタル IO 電源電圧。	-	Supply
C4	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。	-	Digital Input (Pull-Down)
A5	CSN/AD2	SPI チップ・セレクト入力（アクティブ・ロー）（USEL = 0）または UART モードのアドレス入力 2（+4）（USEL = 1）	-	Digital Input (Pull-Up)

## 端子説明 (続き)

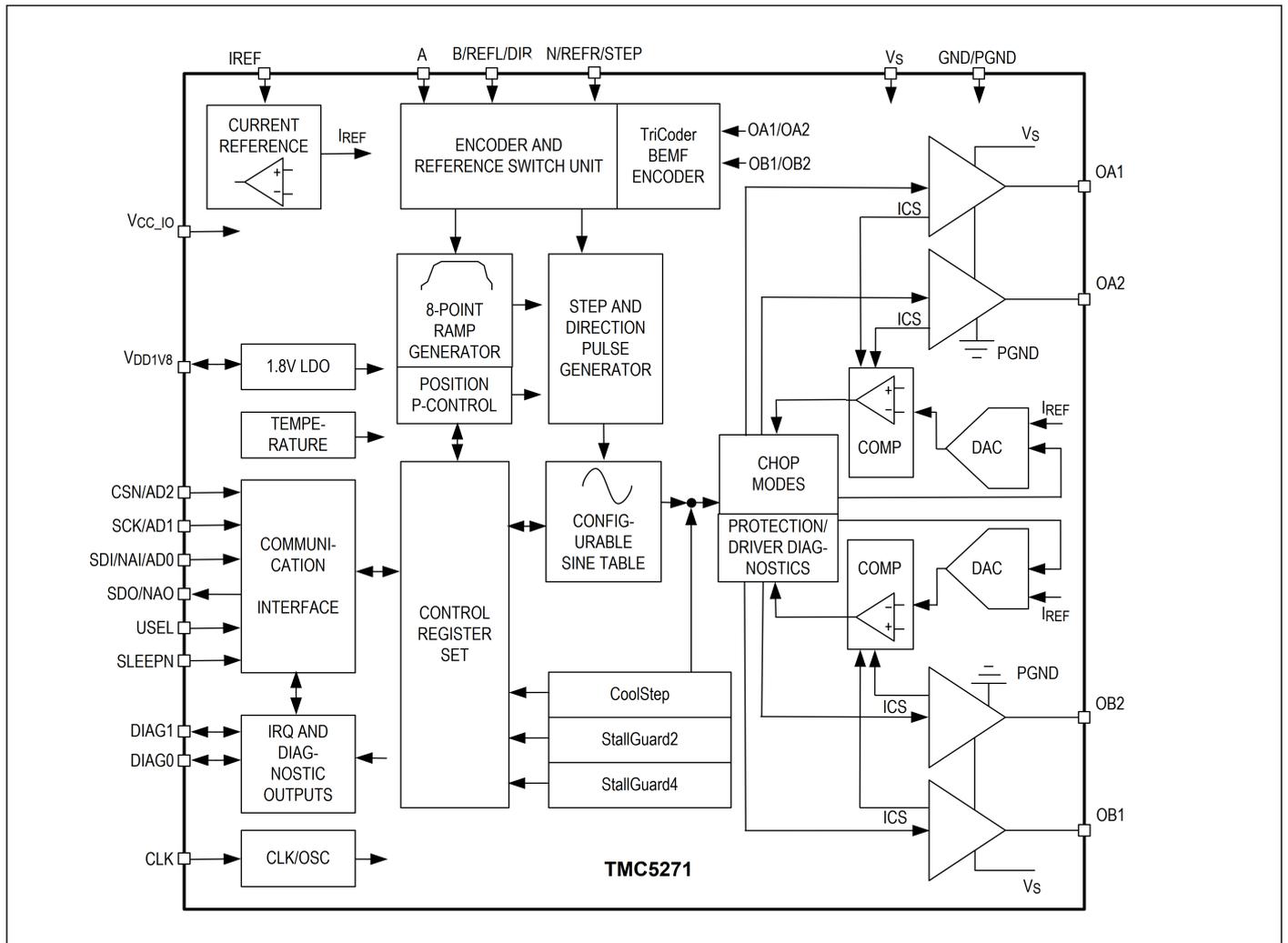
ピン	名称	説明	リファレンス電源	タイプ
B4	SCK/AD1	SPI シリアル・クロック入力 (USEL = 0) または UART モードのアドレス入力 1 (+2) (USEL = 1)。	-	Digital Input (Pull-Up)
B3	SDI/NAI/AD0	SPI データ入力 (USEL = 0) またはデジチェーン UART モードの次のアドレス入力 または デフォルト UART モードのアドレス入力 0 (+1) (USEL = 1)	-	Digital Input (Pull-Up)
A3	SDO/NAO	SPIモードのSPIデータ出力 (スリーステート) (USEL = 0)。 UART モードの次のアドレス出力 (NAO) (USEL = 1)。 UART リング・モードの UART データ出力 (AD0 = AD1 = AD2 = 1)。	V <sub>CC_IO</sub>	Digital Output
D3	USEL	インターフェース選択ピン。 ローに接続された場合、SPI 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。	-	Digital Input (Pull-Down)
A2	DIAG0	設定可能な診断出力 DIAG0。 オープン・ドレイン・モードでは 4.7kΩ 以下の外部プルアップ抵抗をします。	V <sub>CC_IO</sub>	Digital Output
E3	SLEEPN	アクティブ・ローのパワーダウン入力/リセット入力。  SLEEPN ピンと GND の間に短パルスを押加するとデバイスがリセットされ、電源ドライバがディスエーブルされます。デバイスをスリープ・モードにするには、ロー・レベルを押加します。  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・モードに入る前の最後のレジスタ内容は保存されません。IC の再設定時には、GCONF レジスタの drv_enn ビットを用いて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。  使用しない場合は、V <sub>CC_IO</sub> または V <sub>S</sub> に接続します。(このピンは高電圧ピンです。)  注：ハード・ストップや逆起電力スパイクから保護するために、モータ速度が高い場合には使用しないでください。	V <sub>S</sub>	Analog input (Pull-Down)
A1, F1	OA1	モータ・コイル A の出力 1、モータ 1	V <sub>S</sub>	Analog Output
C1, D1	OA2	モータ・コイル A の出力 2、モータ 1	V <sub>S</sub>	Analog Output
A6, F6	OB1	モータ・コイル B の出力 1、モータ 1	V <sub>S</sub>	Analog Output
C6, D6	OB2	モータ・コイル B の出力 2、モータ 1	V <sub>S</sub>	Analog Output
B1, B6, C3, D4, E1, E6	GND/PGND	電源およびシステムのグランド。GND プレーンに接続します。熱伝導を最大限にするために、GND プレーンへ堅固に接続します。	-	GND
C2	DIAG1	設定可能な診断出力 DIAG1。オープン・ドレイン・モードでは 4.7kΩ 以下の外部プルアップ抵抗をします。	V <sub>CC_IO</sub>	Digital Output

## 端子説明 (続き)

ピン	名称	説明	リファレンス 電源	タイプ
F2	A	エンコーダの A チャンネル入力 または 制御された緊急停止を行うための外部入力トリガ		Digital Input (Pull-Down)
F5	B	エンコーダの B チャンネル入力		Digital Input (Pull-Down)
D5	N	エンコーダの N チャンネル入力		Digital Input (Pull-Down)
C5	SW	UART 通信モードでの単線式 UART データ・バスの入出力	V <sub>CC_IO</sub>	Digital Input/ Output
D2	REFR/DIR	右リファレンス・スイッチ入力 REFR または S/D モードでの方向入力 DIR または ハード緊急停止を行うための外部入力トリガ (REFL も使用)		Digital Input (Pull-Down)
E4	REFL/STEP	左リファレンス・スイッチ入力 REFL または S/D モードでの STEP 入力 または ハード緊急停止を行うための外部入力トリガ (REFR も使用)		Digital Input (Pull-Down)

機能図

TMC5271



## 詳細

### 動作原理

TMC5271 モーション・コントローラおよびステッピング・モータ・ドライバ・チップは、スマートな電力変換部品です。MCU と接続でき、モータを制御するためにソフトウェアを追加する必要がありません。基本的なチップ設定の他には、いずれかの使用可能なインターフェースを通じて、加速度パラメータ、速度パラメータ、目標ポジションを伝達する必要があるだけです。TMC5271 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、ユニークな機能を数多く備えています。

### 一軸モーション・コントローラおよびドライバ

TMC5271 は、ステッピング・モータの制御と駆動に必要なあらゆる機能を備えています。モータを制御する外部のモーション・コントローラやステップ・パルス・ジェネレータは不要で、シリアル・インターフェースを通じて目標ポジションなどのランプ・パラメータを提供するだけで済みます。更に、リファレンス・スイッチまたはインクリメンタル・エンコーダを接続すれば、ホーミングやモニタリングなどを行うこともできます。モーション・コントローラ・モードがデフォルトの動作モードです。図 1 にブロック図の概略を示します。

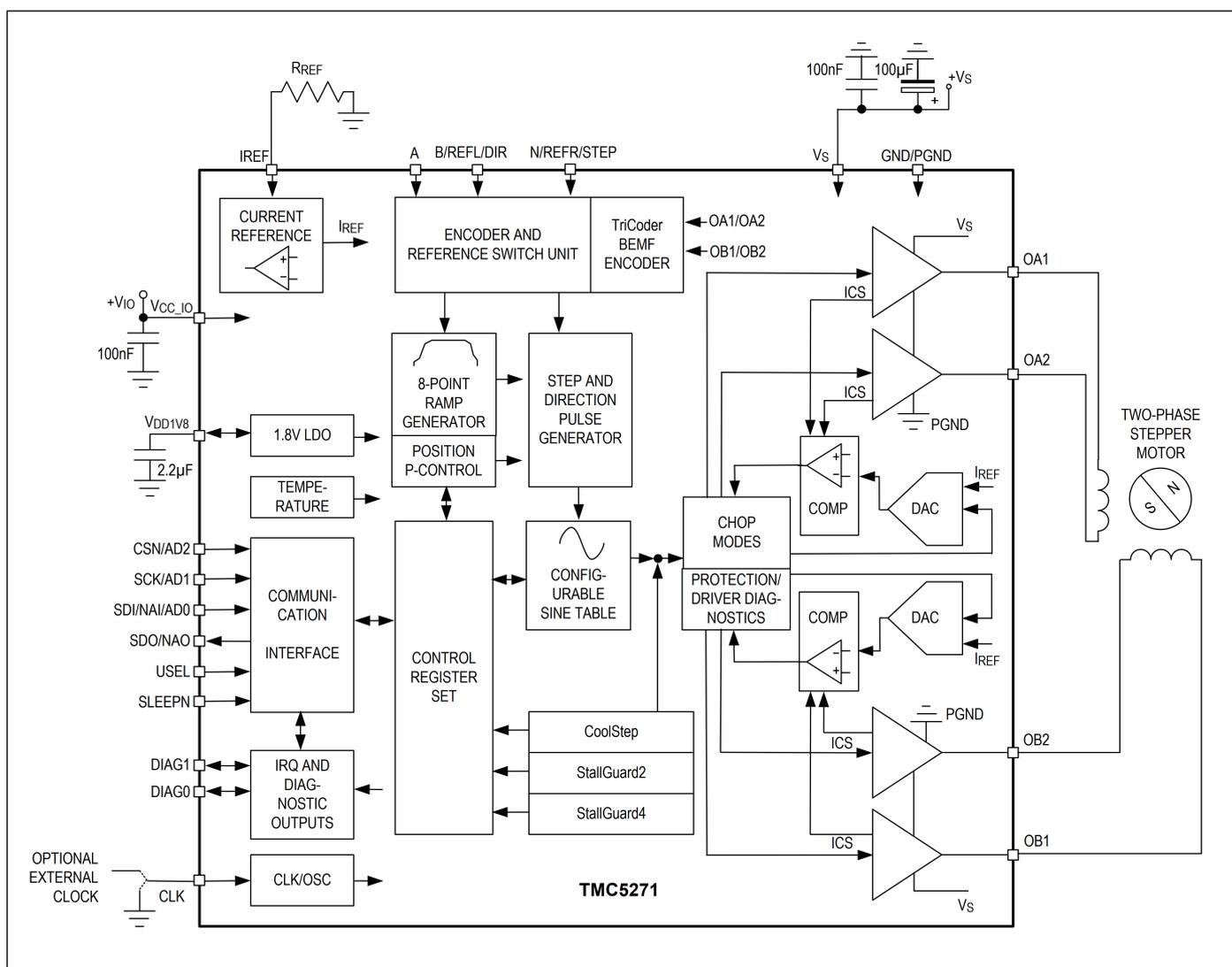


図 1. 一軸モーション・コントローラおよびドライバ

### 主要コンセプト

TMC5271 には、アナログ・デバイセズ製品独自の先進機能が備わっています。その主要機能を表 1 に示します。これらの機能は、ステッピング・モータ・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。

表 1. TMC5271 の主要コンセプト

主要コンセプト	説明	詳細
StealthChop2	モータの無音動作および無音停止を実現する、無騒音高精度の電流制御アルゴリズム。StealthChop®よりも急速なモータ加減速を可能にし、StealthChop を小さい停止モータ電流まで対応できるようにします。	詳細については、 <a href="#">StealthChop2</a> のセクションを参照してください。

表 1. TMC5271 の主要コンセプト (続き)

主要コンセプト	説明	詳細
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。	詳細については <a href="#">Spreadcycle</a> および <a href="#">一般的なチョッパ</a> のセクションを参照してください。
StallGuard2/4	センサーレス・ストール検出および機械的負荷測定。センサーレス・ホーミング機能によりエンド・スイッチを省略でき、また、モータ過負荷時に警告を発することができます。	詳細については、 <a href="#">StallGuard2 の負荷測定</a> および <a href="#">StallGuard4 の負荷測定</a> のセクションを参照してください。
CoolStep	StallGuard2/4 測定を使用して、モータやドライバの効率を最大にし温度上昇を最小限に抑えるよう、モータ電流を適応させます。	詳細については、 <a href="#">CoolStep の負荷適応型電流スケーリング</a> のセクションを参照してください。
MicroPlyer	フル・ステップより低い分解能のステップ入力でフル 256 マイクロステップの滑らかさを実現するマイクロステップ・インターポレータ。	詳細については、 <a href="#">ステップ/方向インターフェース</a> のセクションを参照してください。
TriCoder	モータがディスエーブルされ保持電流を印加できないアプリケーションでモータ動作を検出するために、モータの逆起電力 (BEMF) を利用する、センサーレス停止ステップロス検出機能。	詳細については、 <a href="#">TriCoder-逆起電力センサーレス停止ステップロス検出</a> のセクションを参照してください。

これらの性能向上機能に加え、アナログ・デバイゼズのモータ・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ています。

### 制御インターフェース

TMC5271 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の選択は、USEL ピンを通じて行われます。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 V<sub>CC10</sub> にハードワイヤ接続できます。

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 TMC5271 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。

この単線式インターフェースにより、ホストと TMC5271 を直接結ぶ双方向単線式インターフェースが可能となります。これは標準的な UART で駆動できます。ポー・レートの設定は不要です。

更に、1 つのホストを用いて複数の軸を制御するために、リング・モードとデイジーチェーン接続にも対応しています。

### モータの運動と制御

#### 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、モータを自動で目標ポジションに駆動したり、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、速度スレッシュホールドによって選択される 2 番目と 3 番目の加減速設定と併せてサポートされ、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・プロファイルをモータ・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抑制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 に対して、即時対応が可能です。

内蔵の 8 点モーション・コントローラの利点は以下のとおりです。

- 柔軟なランプ成形
- 加減速に対してモータ・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
- ジャーク抑制のための擬似 S 字形状ランプ
- 停止状態およびストール状態への即時反応

## ステップおよび方向モード

S 字のモーション・プロファイルが必要な場合、内蔵のモーション・コントローラをディスエーブルし、TMC4361A (TMC4361A のデータシート参照) などの外部モーション・コントローラまたは CPU から直接、ステップおよび方向の信号を供給できます。この場合、TMC5271 は電流制御をつかさどり、ドライバおよびモータの状態についてフィードバックを行います。MicroPlyer は運動を自動的に平滑化します。このモードは、SPI または UART を通じた設定によりイネーブルできます。STEP 入力のアクティブ・エッジは、立上がりまたは立下がりとすることができます。両方のエッジを用いることで STEP 信号のトグル・レートを半分に短縮できます。これは、光学的に絶縁を行うカプラなど動作の遅いインターフェースで制御する場合に便利です。DIR 入力は、前方にステップするか後方にステップするかを決定します。

## 自動停止パワーダウンおよびパワーアップ

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。図 2 に、電流レベルの変化と時間の関係を示します。

動作電流を半分に低減するだけで、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。

自動フリーホイールリングおよび受動モータ・ブレーキは、停止用オプションとして提供されています。

受動ブレーキにより、効果的な減衰およびブレーキを維持したまま、モータ停止の消費電力をゼロに減少できます。

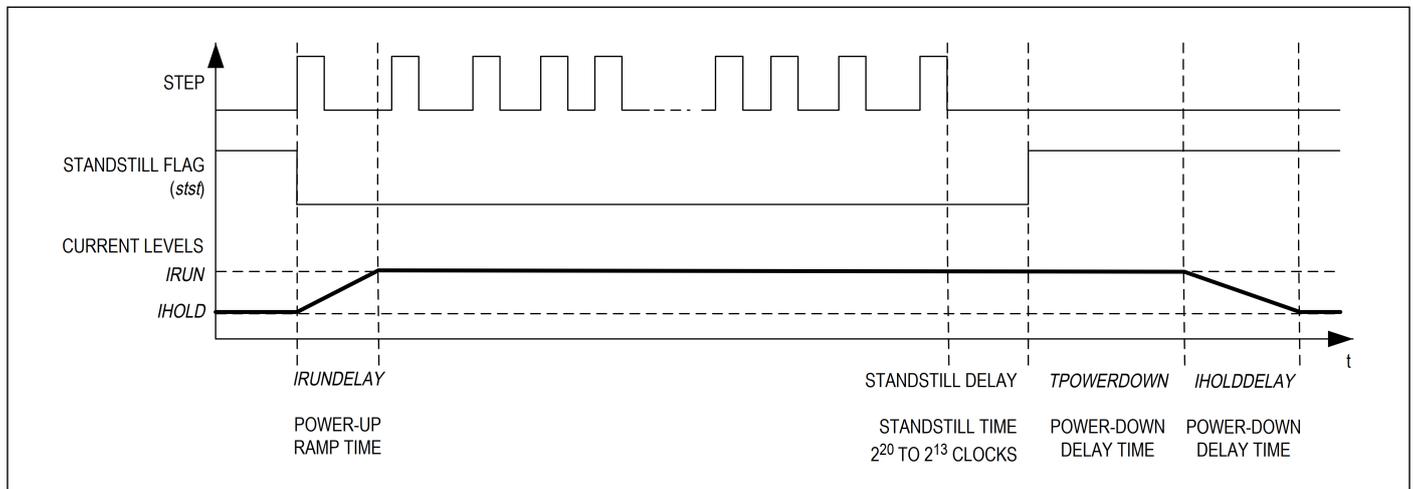


図 2. 停止時およびランプアップ時の自動モータ電流制御

## StealthChop2 および SpreadCycle ドライバ

StealthChop2 は電圧チョッパー・ベースの電流コントローラです。これにより、特に停止動作時および低速動作時には、ボール・ベアリングで発生する雑音を除き、まったく静寂なモータが可能となります。

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、またモータ電流を制御することもできます。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ムिंग・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、モータ速度の変化に対して一度の応答で高いモータ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 は両方の利点を活用するために組み合わせて使用するよう意図されています。StealthChop2 は無騒音の停止や静音かつ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。

StealthChop2 および SpreadCycle ドライバの利点は次のとおりです。

- 低コスト・モータでのマイクロステッピング性能が大幅に向上
- モータ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 および StallGuard4—機械的負荷のセンシング

StallGuard2 および StallGuard4 では、モータ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。

StallGuard2 および StallGuard4 の利点は次のとおりです。

- 追加センサーが不要なリファレンス検索
- 機械的な状態インジケータとして使用
- 定められた負荷条件に反応

### CoolStep—負荷適応型電流制御

CoolStep は、最適な電流でモータを駆動します。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、モータが最適な電流で駆動されるため、トルク・リザーブが約 50%の標準的な動作に比べ、モータ効率が増加します。

CoolStep の利点は以下のとおりです。

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

### エンコーダ・インターフェース

TMC5271 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。また、エンコーダ・インターフェースは、クローズド・ループ・ポジション・コントロール用の特別な P レギュレータ・モードで用いることもできます。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。

### 停止ステップロス検出

センサーレス停止ステップロス検出機能は BEMF デコーダを基盤としています。BEMF デコーダは、モータがディスエーブル時（モータ・コイルに有効な電流が加えられていないとき）のモータ動作を検出できます。この機能は、消費電力を節約する要求があるために保持電流を印加できないながらも、通常一定量のコギング・トルクまたは摩擦がモータを定位置に保持するようなデバイスに対して、特に有用です。モータが外部の力によって回転するような場合に、その動きが探知されます。その結果を利用して、モータが動いた場合の新しいホーミング・シーケンスをトリガしたり、ステップ数の追跡を行い続けその後にそれを補正したりできます。

### SPI

#### SPI データグラム構造

TMC5271 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を行うことができます。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・ビットでアクセスされます。データグラム構造を図 3 に示します。

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。
- 書き込みアクセスでは、アドレス・バイトの最上位ビットが 1 です。

すべてのレジスタが読出し可能です。それらのほとんどは読出し／書き込みの両方が可能です。一部は読出し専用レジスタで、また、1 を書き込んでクリアできるレジスタもあります (GSTAT レジスタなど)。

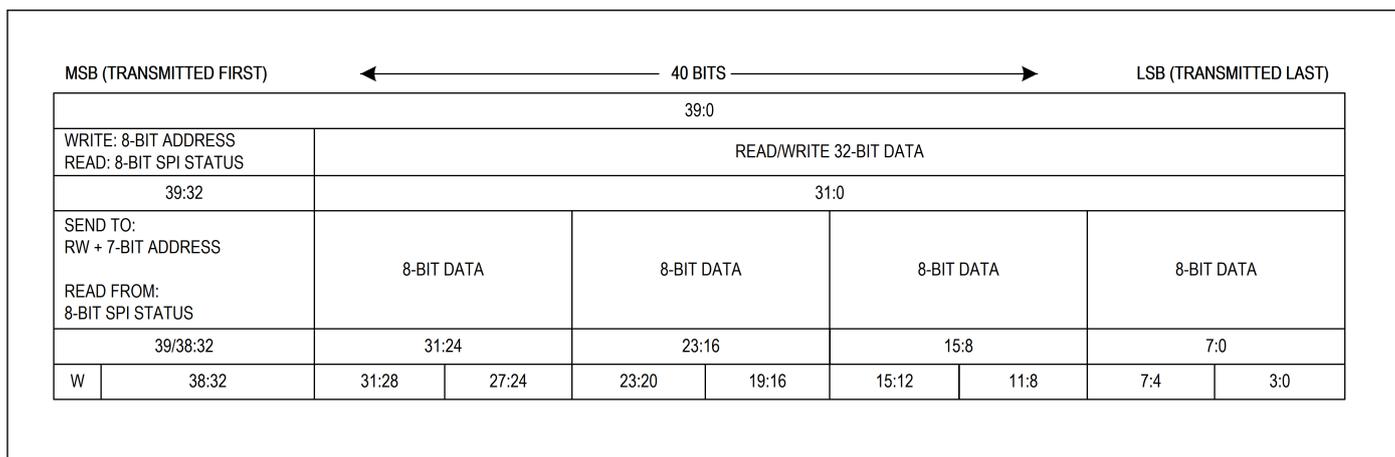


図 3. SPI データグラム構造

### 読出し／書き込みの選択 (WRITE\_notREAD)

読出しと書き込みの選択は、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、書き込みアクセスの場合は 1 です。W と記されたビットは WRITE\_notREAD 制御ビットです。アクティブ・ハイの書き込みビットはアドレス・バイトの MSB です。したがって、書き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き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き込みデータを反映したものになります。そのため、読出しアクセスではアドレス指定されたレジスタにデータが転送されない点が、読出しアクセスと書き込みアクセスの違いとなります。読出しアクセスではアドレスを伝送するのみで、その 32 データ・ビットはダミーです。その次の読出しまたは書き込みアクセスで、その前の読出しサイクルで伝送されたアドレスからの読出しデータが返されます。

読出しアクセス要求のデータグラムはダミーの書き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き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。

TMC5271 との間でデータの読出しや書き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。

SPI の読出し／書き込みフローの例は表 2 を参照してください。

アドレス 0x18 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 0x18 に設定する必要があります。レジスタ (VACTUAL) への書き込みアクセスの場合、アドレス・バイトは 0x80 + 0x19 = 0x99 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、これらは 0 に設定できます。

表 2. SPI 読出し／書き込みフローの例

ACTION	DATA SENT TO TMC5271	DATA RECEIVED FROM TMC5271
Read XACTUAL	0x1800000000	0xSS and unused data*

表 2. SPI 読み出し/書き込みフローの例 (続き)

ACTION	DATA SENT TO TMC5271	DATA RECEIVED FROM TMC5271
Read XACTUAL	0x1800000000	0xSS and XACTUAL
Write VMAX = 0x00ABCDEF	0xA100ABCDEF	0xSS and XACTUAL
Write VMAX = 0x00123456	0xA100123456	0xSS00ABCDEF

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

### 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 にこれらのステータス・フラグの一覧を示します。

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	注
7	Not used	常時 0
6	status_stop	1: モーション・コントローラが停止していることを指示 (stop_l、stop_r、stop_sg、virtual_stop_l、virtual_stop_r)
5	Not used	常時 0
4	position_reached	RAMP_STAT[9] = 1: モータが目標ポジションに到達したことを指示 (モーション・コントローラ専用)
3	Not used	常時 0
2	velocity_reached	RAMP_STAT[8] = 1: モータが目標速度に到達したことを指示 (モーション・コントローラ専用)
1	driver_error	GSTAT[1] = 1: ドライバにエラーが生じたことを指示 (GSTAT レジスタのビットをクリアする (そのビット・ポジションに 1 を書き込む) ことでクリア)
0	reset_flag	GSTAT[0] - 1: リセットが発生したことを指示 (GSTAT の読み出しでクリア) (GSTAT レジスタのビットをクリアする (そのビット・ポジションに 1 を書き込む) 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 信号

TMC5271 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。TMC5271 でバス・トランザクションを行うには、最低でも 40SCK 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

図 4 に SPI のタイミング図を示します。SPI は SPI モード 3 を使用します。SPI の最大周波数は 10MHz 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。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。タイミング値は電気的特性のセクションに記載されています。

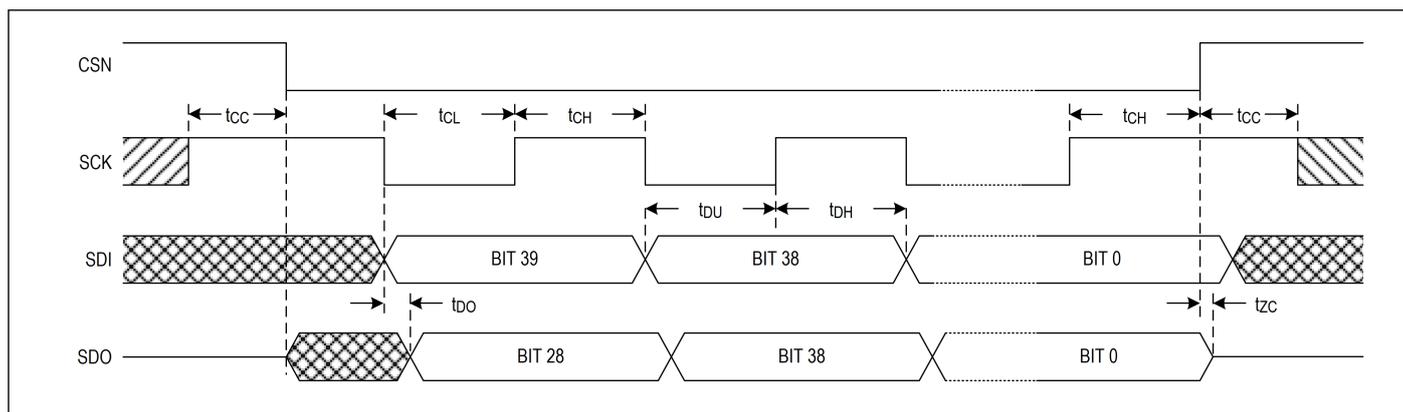


図 4. SPI タイミング図

## 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 TMC5271 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。自動ポー・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。様々な配線方式およびアドレス指定方式が可能です。

## UART データグラム構造

### UART 書込みアクセス

EACH BYTE IS LSB TO MSB, HIGHEST BYTE TRANSMITTED FIRST											
0:63											
SYNC + RESERVED				8 BIT NODE ADDR		RW + 7 BIT REGISTER ADDR.		32 BIT DATA		CRC	
1	0	1	0	RESERVED / DON'T CARES, BUT INCLUDED IN CRC		REGISTER ADDRESS		1	DATA BYTES 3, 2, 1, 0 (HIGH TO LOW BYTE)	CHECKSUM	
0	1	2	3	4	5	6	7	8:15	16:23	24:55	56:63

図 5. UART 書込みアクセスのデータグラム構造

図 5 に UART 書込みアクセスのデータグラム構造を示します。同期ニブルが TMC5271 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、内部ポー・レート分周器と UART ホスト・クロックの同期が可能です。実際のポー・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、ポー・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 のロジック 0、ロー・レベル) で始まり、停止ビット (DIAG1 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 ボーの最小ポー・レートが可能です (低ポー・レートの最も厳しい場合)。最大ポー・レートは、ポー・クロックの必要な安定性を確保するため、 $f_{CLK}/16$  です。

低消費電力モードに切り替えると、クロック・ソースは内部クロックに切り替わります。これにより、最大ポー・レートは内部発振器の  $1/16$  に減少します。

NODEADDR は 8 ビットのアドレスで、NODECONF レジスタ (0x3) で設定できます。

パワーアップ後またはリセット後、0~7 の範囲の 3 つのアドレス信号 CSN\_AD2、SCK\_AD1、SDI\_AD0 によって最初のペリフェラル・アドレスが選択され、これらが NODEADDR の値に追加されます。

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。

例：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：レジスタ・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、ポー・レートが直前の値の 15%未満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リッジされます。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。

## UART 読出しアクセス

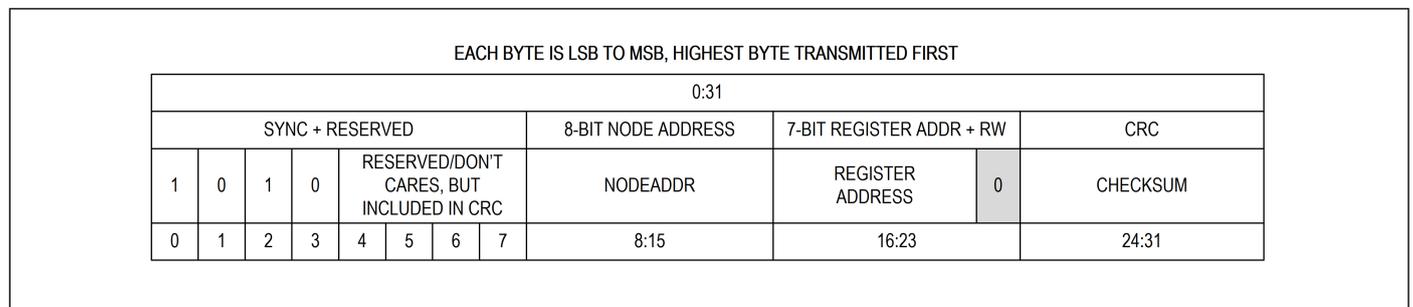


図 6. UART 読出しアクセス要求データグラムの構造

読出しアクセス要求データグラムの構造は、図 6 に示すように、書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。TMC5271 は、UART ホストが読出し要求のために用いるポー・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、TMC5271 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。この遅延時間は、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。

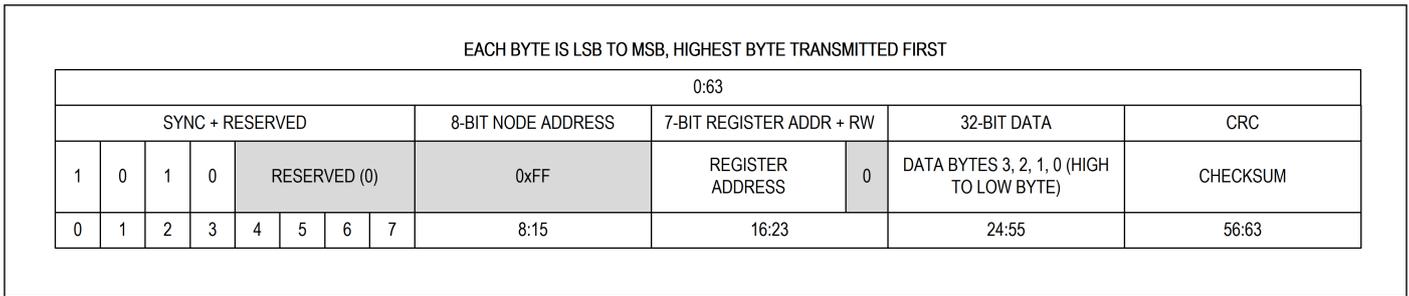


図 7. UART 読出しアクセス返答データグラムの構造

読出し応答は、図 7 に示すように、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。ノードにはこのアドレスを使用することはできません。

## UART 信号

TMC5271 の UART インターフェースは、表 4 に示す 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 を絶えずチェックします。ピンはこの期間、入力に切り替わります。UART 書込みアクセスのセクションで説明したように、このピンは同期ニブルに基づくボー・レートに対応します。読出しアクセスの場合、このピンは、DIAG1 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 4. TMC5271 UART のインターフェース信号

信号	説明
DIAG1	デフォルトの UART モードではデータ入出力。 UART リング・モードではデータ入力。
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/NAI/AD0	UART アドレスのビット 0 をインクリメント (+1) シーケンシャルなアドレス指定方法の場合はチェーン内の前の IC の NAO に接続します。
SDO/NAO	チェーン化されたシーケンシャル・アドレス指定方法の場合は NAO (次のアドレスの出力) ピン (リセット後のデフォルト = ハイ)。 UART リング・モードではデータ出力。

## CRC の計算

8 ビット CRC 多項式を用いて、読出しおよび書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。TMC5271 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)
{
    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
}

```

### 複数ノードのアドレス指定

複数のノードを接続するには、次の3通りの選択肢があります。

- 1~7個のみのノードを接続する場合は直接ノード指定
- ノード数が7を超える場合はデジチェーン接続 ([UART デジチェーン接続](#)のセクションを参照)
- ノード数が7を超えノード間のパターン長が短い場合はリング・モード ([UART リング・モード](#)のセクションを参照)

1~7個のみの TMC5271 デバイスを単一の 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は、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>CC10</sub>および GND に接続することで行います。

注：3つのアドレス・ピン (AD0、AD1、AD2) がすべて 1 に設定されている場合、または V<sub>CC10</sub>に接続されている場合、UART の動作はリング・モード状態になります ([UART リング・モード](#)のセクションを参照)。

### UART デジチェーン接続

8個以上のノードを同じ 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。[図 8](#) および [表 5](#) に示すように、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 TMC5271 の SDI/NAI/AD0 を GND に接続します。
- 最初の TMC5271 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 TMC5271 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。

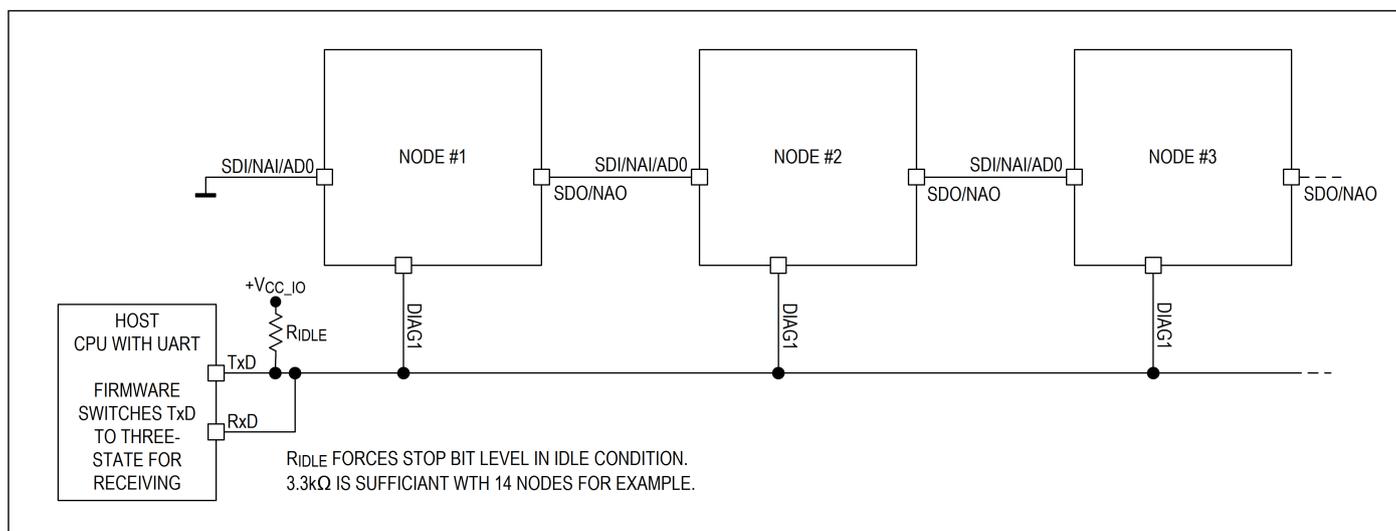


図 8. UART のデジチェーン接続例

表 5. 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	—	—

## UART リング・モード

UART リング・モードでは、すべてのノードがリング構造にカスケード接続されます。リング・モードを有効化するには、3 つの UART アドレス指定ピン SDI/NAI/AD0、SCK/AD1、CSN/AD2 をすべて 1 に設定 ( $V_{CC\_IO}$  に接続、またはプルアップ抵抗を内蔵している場合は開放状態を維持) します。DIAG1 ピンは UART データ入力として使用します。SDO/NAO ピンは UART データ出力として使用します。SDO/NAO はハイ状態 (アイドル状態) を維持し、NODEADDR がプログラムされるまで SW ピンに入力されるデータを転送しません。すべてのデータは、NODEADDR が 0 以外のアドレス値にプログラムされた後、DIAG1 から SDO/NAO に転送されます。構造を図 9 に示します。

目的は、アドレス指定ピン SDI/NAI/AD0、SCK/AD1、CSN/AD2 を用いずに、あるいは、チェーン接続されたシーケンシャルなアドレス指定方式を用いずに、単純で柔軟なアドレス指定方法を可能にすることです。同時に、チェーン構造により各 2 つのノード間の距離は短いままに保たれ、各ラインの負荷が 1 つの入力だけとなります。そのため、配線コストや PCBA レベルの配線コストが重要となる場合に論理リングを最適化できます。ボード上の物理構造がリングではなく直線に近い場合は、各 2 番目のノードが左から右方向に接続され次には右から左に接続されるような、インターリーブ方式でノードをカスケード接続できます。このようにすることで、最後のノードのデータ出力からホストへの距離が 3 個のノード間の距離より長くなることはありません。

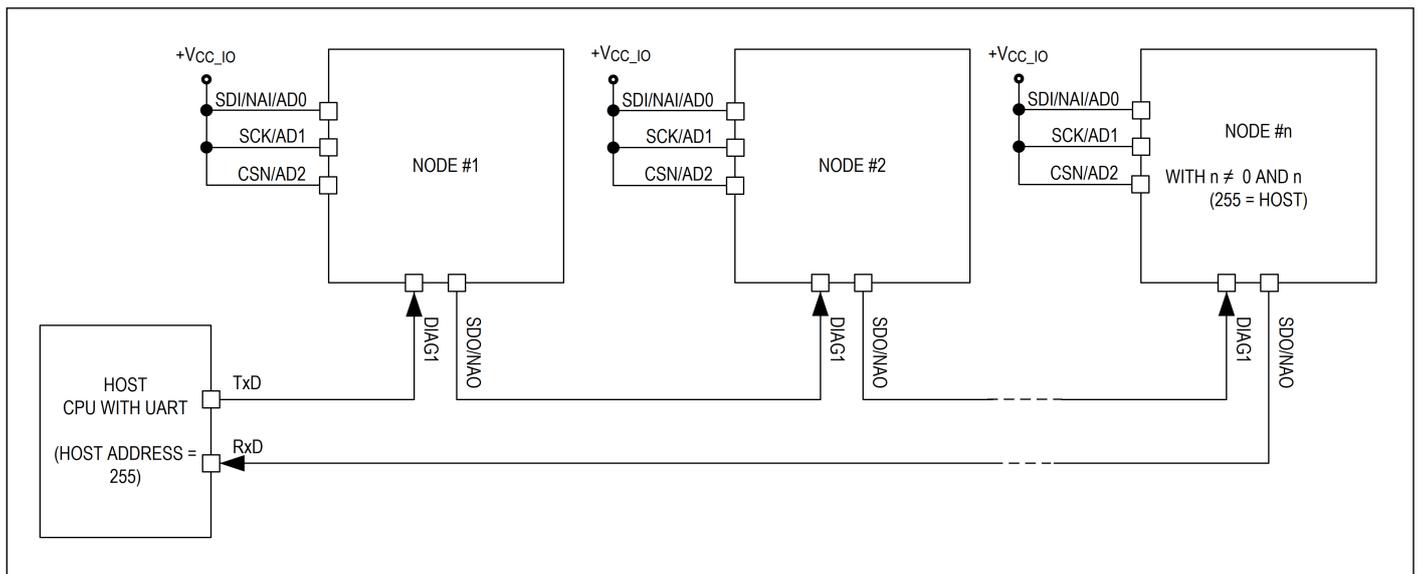


図 9. UART リング・モードの配線例

すべてのノードに 0 以外の固有のアドレスが指定された後にのみ、リング構造を通じてデータをノードからホストに伝達できる点に注意してください。2 個の同じアドレスを指定することのないようにしてください。

## ステップ/方向インターフェース

STEP 入力および DIR 入力は、既存の多くのモーション・コントローラに対応できる、単純で標準的なインターフェースを備えています。MicroPlyer ステップ・パルス・インターポレータは、元来は粗いステッピング用に設計されたアプリケーションに対して、高分解能マイクロステッピングの滑らかなモータ動作を可能にします。

ステップ/方向モードを有効にするには、GCONF レジスタの SD ビットを 1 に設定する必要があります。その後内蔵モーション・コントローラをオフにします。この場合、アクティブなままの唯一のモーション・コントローラ・レジスタは、IHOLD\_IRUN レジスタの電流レベル設定値です。ステップ/方向モードでは、インクリメンタル・エンコーダ・オプションとリファレンス・スイッチ入力も使用できません。

**注：**ステップ/方向モードではモーション・コントローラがオフになるだけでなく、完全にディスエーブルされクロック供給もされないため、全体の消費電力を削減できます。

## タイミング

図 10 および図 11 に、STEP 信号および DIR 信号のタイミング・パラメータと入力フィルタ構造を示します。CHOPCONF レジスタの dedge モード・ビットが設定されている場合、STEP のどちらのエッジもアクティブになります。dedge がクリアされると立上がりエッジのみがアクティブになります。STEP および DIR は、システム・クロックに対応してサンプリングと同期を行います。約 10ns の内蔵アナログ・フィルタが、長い PCB パターンによるグリッチなどの信号に存在するグリッチを除去します。信号源がチップから離れた場所にある場合、特に信号がケーブルを通じて伝送される場合は、信号を差動でフィルタリングまたは伝送する必要があります。

仕様規定されたタイミング・パラメータについては電氣的特性のセクションを参照してください。

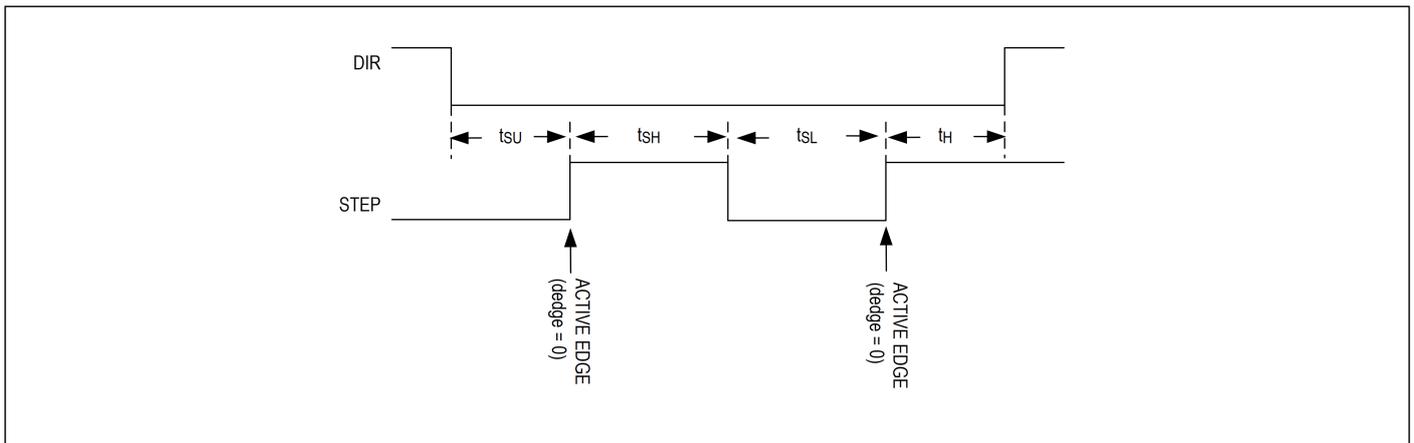


図 10. STEP/DIR 信号のタイミング

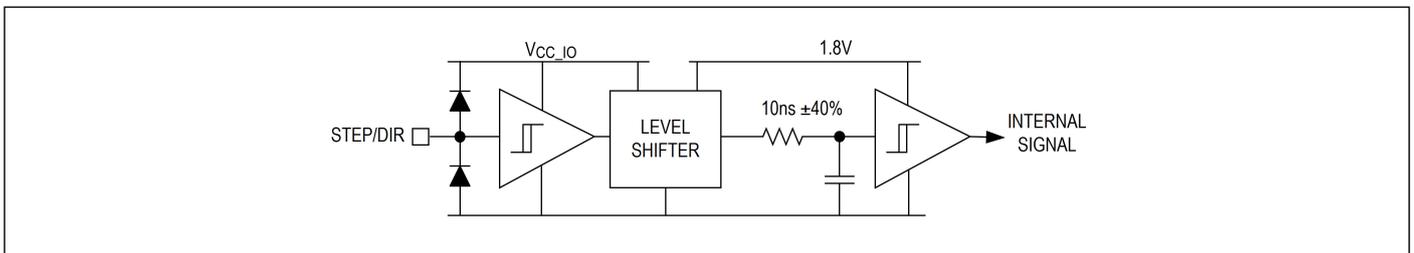


図 11. STEP/DIR 信号の入力フィルタ構造

## 分解能の変更

マイクロステップ分解能を低下させることで、ステップおよび方向インターフェースのステップ周波数を制限したり、旧式で性能が劣るドライバに対応したりすることができます。1024 個のサイン波エントリを持つ内蔵マイクロステップ・テーブルにより、サイン波モータ・コイル電流が生成されます。これらの 1024 個のエントリは、1 つの電氣的回転または 4 つのフル・ステップに対応します。マイクロステップ分解能の設定は、テーブル内で用いられるステップ幅を決定します。DIR の入力に応じて、ステップ・パルスごとにマイクロステップ・カウンタがステップ幅分だけ増加 (DIR=0) または減少 (DIR=1) します。マイクロステップ分解能がインクリメント量またはデクリメント量を決定します。最大分解能の場合、シーケンサはステップ・パルスごとに 1 ステップ前進します。分解能が半分の場合、2 ステップ進みます。インクリメントはフル・ステップで最大 256 ステップです。シーケンサは、いつでも異なるマイクロステップ・レートにシームレスに切り替えることのできる、特別な機能を備えています。低マイクロステップ分解能に切り替わると、シーケンサは、目標分解能に最も近いステップを計算し、そのポジションでの電流ベクトルを読み出します。この動作は、フル・ステップやハーフ・ステップなどの低分解能の場合に特に重要です。ステップ・シーケンスに何らかの不具合があると、時計方向および反時計方向に動作するモータと比較して、非対称な動作となるためです。表 6 に、ハーフ・ステップ動作およびフル・ステップ動作のコイル電流値を示します。

例：

**フル・ステップ：** テーブル・ポジション全体でのサイクル：128、384、640、896 (45°、135°、225°、315° の電氣的ポジション、両コイルが同一電流でオン)。各ポジションのコイル電流は、実効値 (振幅の 0.71 倍) に対応します。ステップ・サイズは 256 (90° の電氣的ポジション) です。

**ハーフ・ステップ：** 最初のテーブル・ポジションは 64 (22.5° の電氣的ポジション) です。ステップ・サイズは 128 (45° のステップ) です。

**クォータ・ステップ：** 最初のテーブル・ポジションは 32 (90°/8 = 11.25° の電氣的ポジション) です。ステップ・サイズは 64 (22.5° のステップ) です。

このようにして等距離のステップが生じ、これらはどちらの回転方向に対しても同じです。一部の旧式のドライバは、ステップ・テーブル内でゼロ電流 (テーブル・エントリ 0、0°) とフル電流 (90°) も使用します。この種のステップ動作は、トルクが少なく、また、ドライバおよびモータの消費電力も増加するため、回避します。

表 6. 位相 A/B のコイル電流用フル・ステップ/ハーフ・ステップのルックアップ・テーブル値

STEP POSITION	TABLE POSITION	CURRENT COIL A	CURRENT COIL B
Half step 0	64	38.3%	92.4%
Full step 0	128	70.7%	70.7%
Half step 1	192	92.4%	38.3%
Half step 2	320	92.4%	-38.3%
Full step 1	384	70.7%	-70.7%
Half step 3	448	38.3%	-92.4%
Half step 4	576	-38.3%	-92.4%
Full step 2	640	-70.7%	-70.7%
Half step 5	704	-92.4%	-38.3%
Half step 6	832	-92.4%	38.3%
Full step 3	896	-70.7%	70.7%
Half step 7	960	-38.3%	92.4%

### MicroPlyer ステップ・インターポレータおよび停止検出

STEP 入力の各アクティブ・エッジに対して、MicroPlyer は 256 倍の分解能でマイクロステップを生成します。これは、最後のステップ間隔に基づき、ステップ入力での 2 つのステップ・インパルス間の時間を補間します。したがって、2 マイクロステップ (256 マイクロステップに対して 128 マイクロステップの補間) から 256 マイクロステップ (256 マイクロステップに対するフル・ステップ入力) が単一ステップ・パルスに対して駆動されます。

MicroPlyer 機能は、CHOPCONF レジスタの intpol ビットを設定することで有効になります。

補間された 2 マイクロステップ~256 マイクロステップに対するステップ・レートは、直前のステップ期間の時間間隔を測定しそれを最大 256 等分することで求められます。2 つのマイクロステップ間の最大時間は、256 マイクロステップに等分した場合、 $2^{20}$  (約 100 万システム・クロック・サイクル) に相当します。その結果、システム・クロック周波数が 16MHz の場合、MicroPlyer 動作の最小ステップ入力周波数は 16Hz になります。ステップ・レートがこれより低いと、STST ビットが設定され、停止が生じたことを指示します。その周波数では、(システム・クロック周波数)/ $2^{16}$  (約 256Hz) のレートのマイクロステップが生じます。停止が検出されると、ドライバはモータ電流を自動的に保持電流 IHOLD に切り替えます。

**注:** MicroPlyer は、ステップ周波数が安定している場合のみ完全に機能します。STEP ピンのステップ信号のデューティサイクルが 50% でない場合は、dedge オプションは使用しないでください。

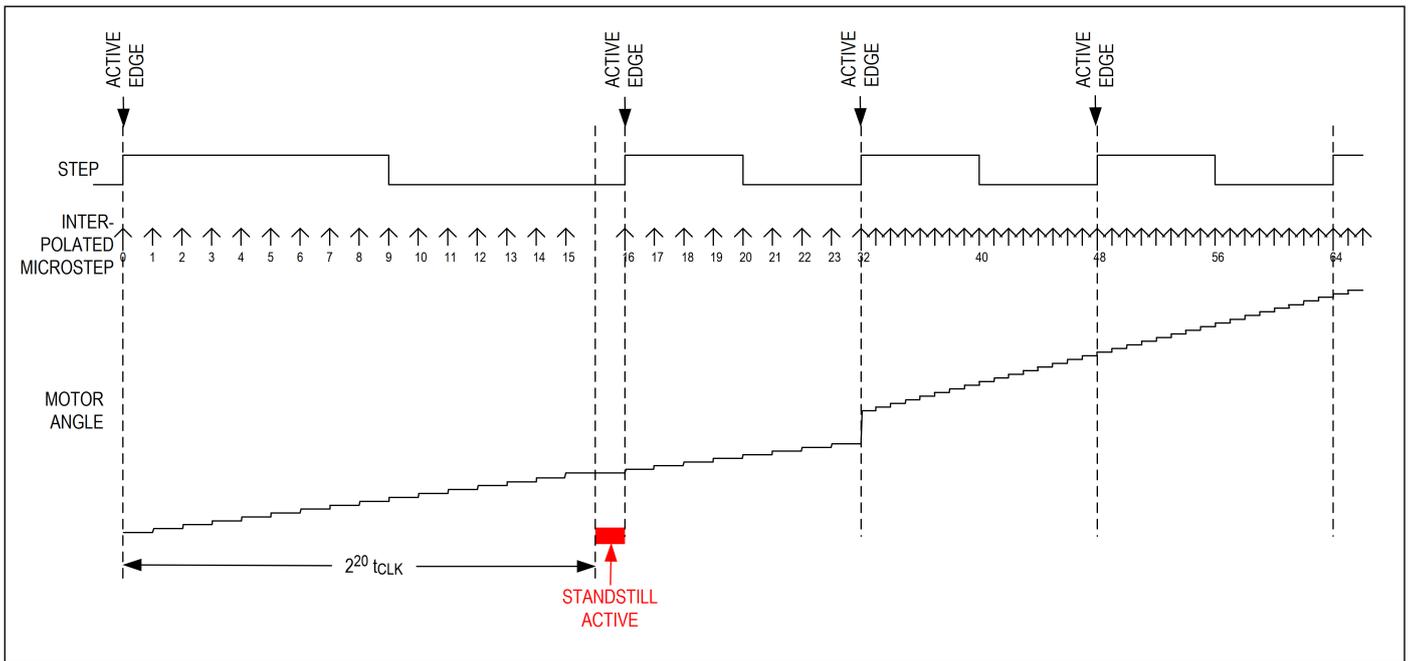


図 12. ステップ信号周波数が増加する場合の MicroPlyer マイクロステップ補間 (例: 16~256)

図 12 では、最初の STEP 信号サイクルは停止ビット `stst` を設定できるだけの長さになっています。このビットは次のアクティブ STEP エッジでクリアされます。その後、STEP 信号の外部周波数が増加します。高いレートで 1 サイクルが経過した後、MicroPlyer は補間されたマイクロステップ・レートをこのより高い周波数に適用します。低速レートでの最後のサイクルの間、MicroPlyer は 16 個のマイクロステップすべてを生成するわけではないため、レートが高くなったときの最初のサイクルと 2 番目のサイクルの間でモータ角にわずかな跳びが生じます。

**注:** TMC5271 では、停止検出タイマー長を設定可能です。DRV\_CONF レジスタの STANDSTILL\_TIME によって 8 種類のオプションの設定が可能です。

## StealthChop2

StealthChop2 は、ステッピング・モータ用の極めて静かな動作モードです。これは電圧モード PWM をベースとします。停止時や低速時、モータは完全に無雑音です。したがって、StealthChop2 で動作するステッピング・モータ・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を動作させてください。モータ性能は多くの場合、機械的な負荷がある方が良好です。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に適用します。したがって、StealthChop2 では、高いモータ・ダイナミクスが可能で、モータを非常に小さな電流までパワーダウンできます。最善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 (表 7 参照)。まず、停止状態からモータを始動します。ただし、公称稼働電流 (AT#1) を供給します。次に、モータを例えば、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 13 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 7. 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	<ul style="list-style-type: none"> <li>モータは停止状態にあり、実際の電流スケール (CS) は稼働時の電流 (IRUN) と同じ</li> <li>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。</li> <li>V<sub>S</sub>ピンは動作レベル</li> </ul>	$\leq 2^{20} + 2 \times 2^{18} t_{CLK}$ 、 $\leq 130\text{ms}$ (内部クロック使用)
AT#2	PWM_GRAD_AUTO	<ul style="list-style-type: none"> <li>大量の逆起電力が生じ、最大稼働電流に達する可能性のある速度にモータを駆動。</li> <li><math>1.5 \times \text{PWM\_OFS\_AUTO} \times (\text{IRUN} + 1)/32 &lt; (\text{PWM\_SCALE\_SUM}/4) &lt; 4 \times \text{PWM\_OFS\_AUTO} \times (\text{IRUN} + 1)/32</math></li> <li><math>\text{PWM\_SCALE\_SUM} &lt; 1023</math></li> </ul> <b>ヒント</b> ：標準的な 50 極ステッピング・モータの場合、代表的な範囲は 60RPM~300RPM です。	$\pm 1$ の変更に 8 つのフル・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、TMC5271-EVAL-KIT を用います。初期調整パラメータを求めるには、ファームウェアの初期化用の PWM\_GRAD と PWM\_OFS にアプリケーション固有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行ったことを示します。

**注意事項**：適切な調整を行わずに StealthChop2 で動作させると、特に、低抵抗モータで高減速度設定の場合、減速ランプ中にモータ電流が増加する原因となります。自動調整手順に従い、TMC5271-EVAL-KIT キット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・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。

FSR、FSR\_IREF、GLOBALSCALER、または V<sub>S</sub> 電圧を用いて電流範囲を変更すると、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。

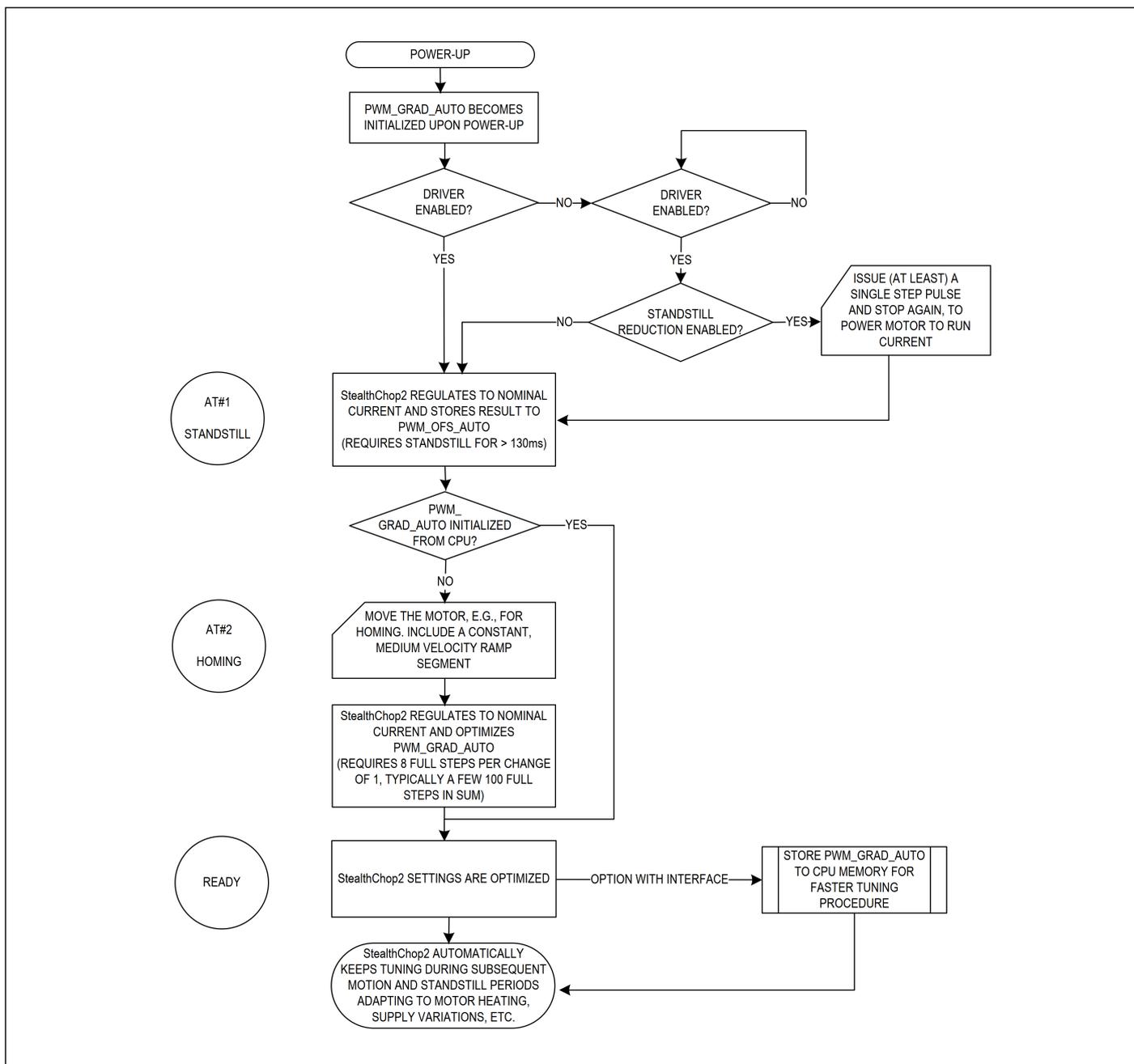


図 13. StealthChop2 の自動調整手順

## StealthChop2 のオプション

モータ電流を特定のレベルに一致させるために、実際のモータ速度に応じて実効的な PWM 電圧がスケールアップされます。目標電流でモータを駆動するのに必要な電圧レベルには、モータ抵抗、その逆起電力（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。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・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ตอบสนอง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介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。周波数と設定値の例および推奨値を表 8 に示します。

表 8. StealthChop2 の PWM 周波数の選択

CLOCK FREQUENCY $f_{CLK}$ (MHz)	PWM_FREQ = %00 $f_{PWM} = 2/1024 f_{CLK}$ (kHz)	PWM_FREQ = %01 $f_{PWM} = 2/683 f_{CLK}$ (kHz)	PWM_FREQ = %10 $f_{PWM} = 2/512 f_{CLK}$ (kHz)	PWM_FREQ = %11 $f_{PWM} = 2/410 f_{CLK}$ (kHz)
20	39.1*	58.1	78.1	97.6
18	35.2*	52.7	70.3	87.8
16	31.3*	46.9*	62.5	78.0
12.5 (internal)	24.4*	36.6*	48.8*	61.0
12	23.5*	36.1*	46.9*	58.4
8	15.7	25.3*	31.3*	38.8*

\*推奨値

### 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・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のオン時間中にモータ電流を測定し、モータ電流が目標電流に一致するように、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、ドライバがモータの目標電流の変動による変化 ( $V_{REF}$  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はなりません。初期調整ステップ AT#2 では、PWM\_REG はモータ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。最高速の加速や減速ランプが必要な場合は、PWM\_REG の設定を最適化する必要があります。図 14 および図 15 に、PWM\_REG の最適な設定の場合と過小な設定の場合のモータ位相電流の例を示します。

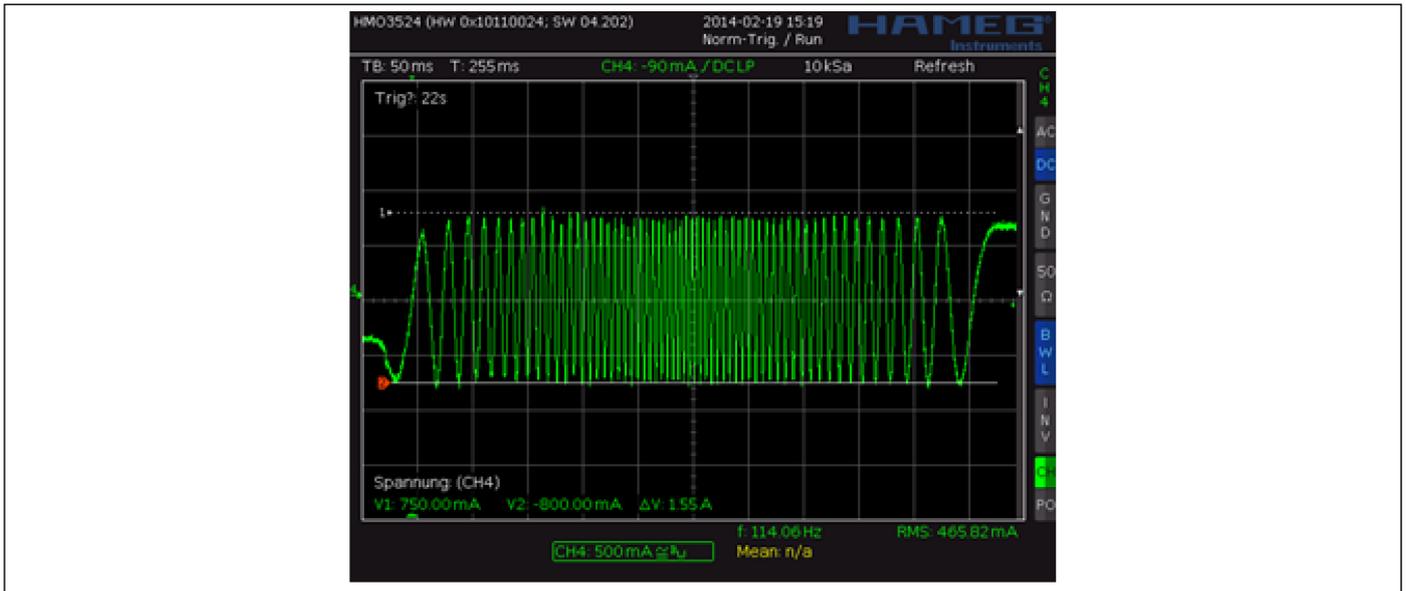


図 14. StealthChop2 : PWM\_REG の設定は良好

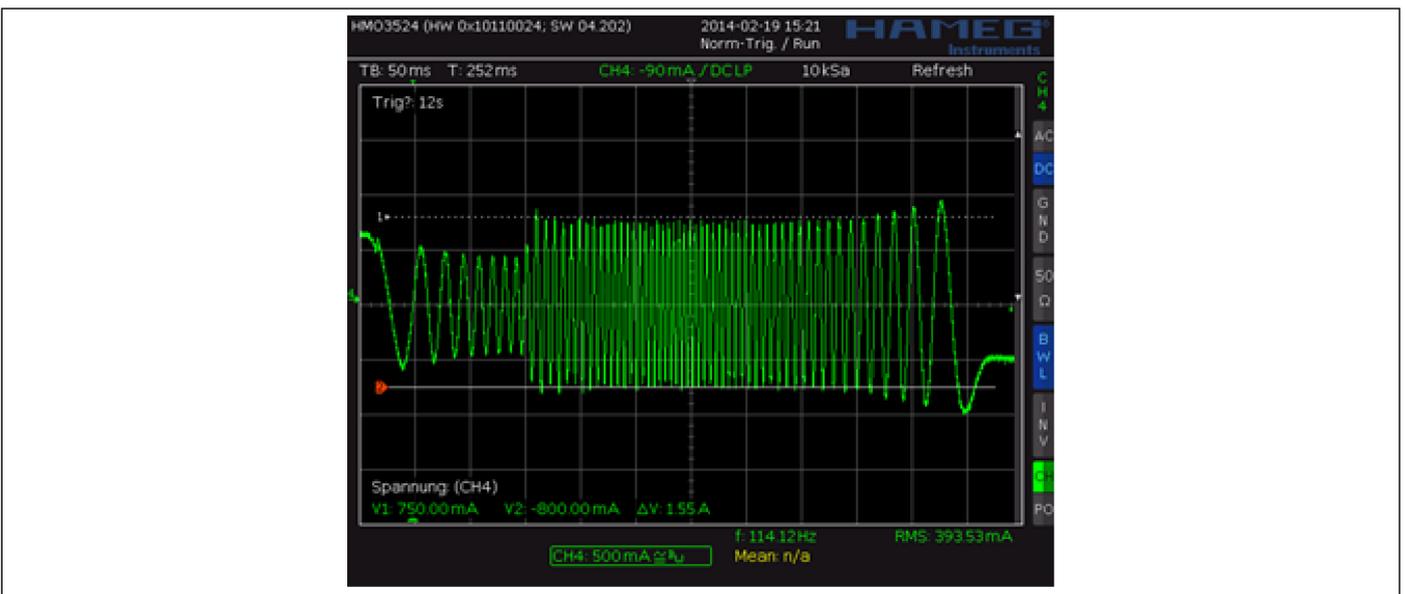


図 15. StealthChop2 : AT#2 の間の PWM\_REG の設定が過小

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、図 16 および 図 17 に示すように、加速フェーズ時のモータ電流のモニタリング時に調べることができます。

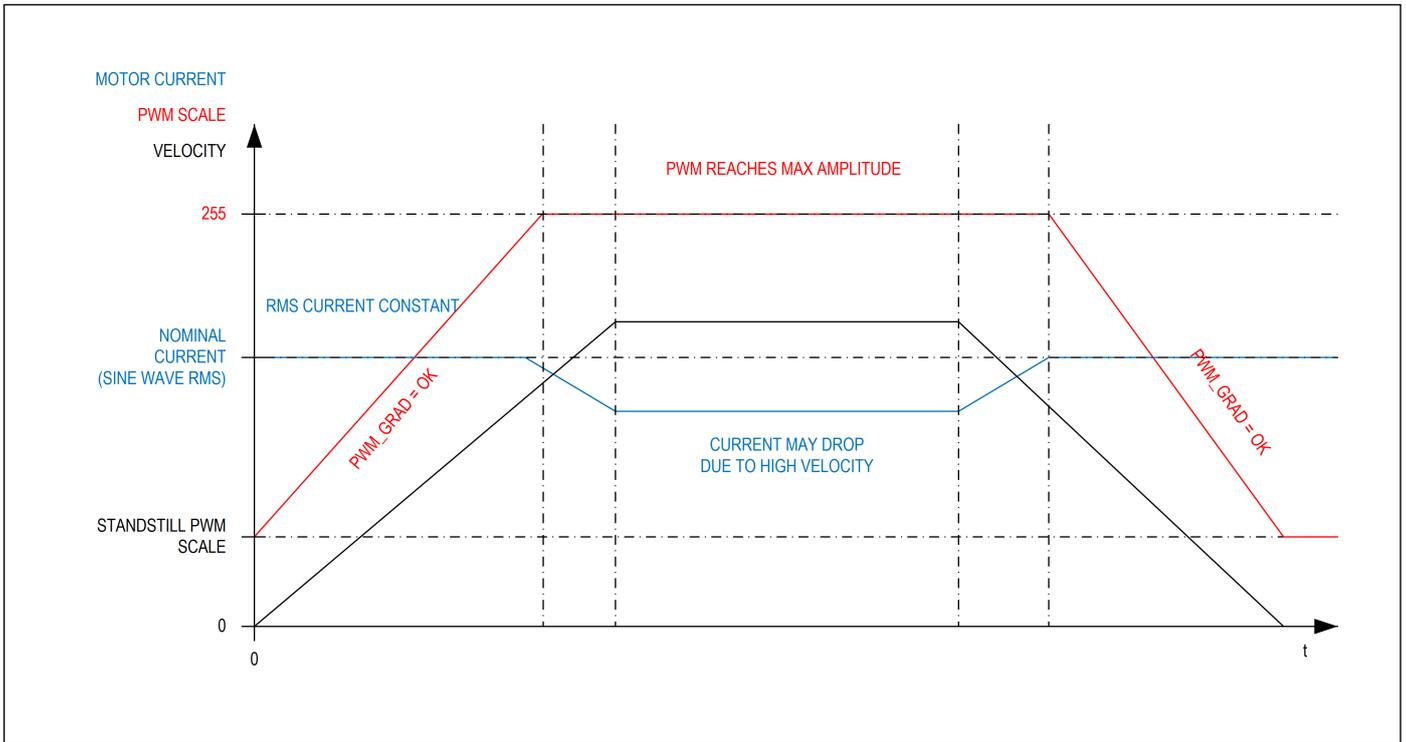


図 16. 正しく決定された PWM\_GRAD(\_AUTO)および PWM\_OFS(\_AUTO)

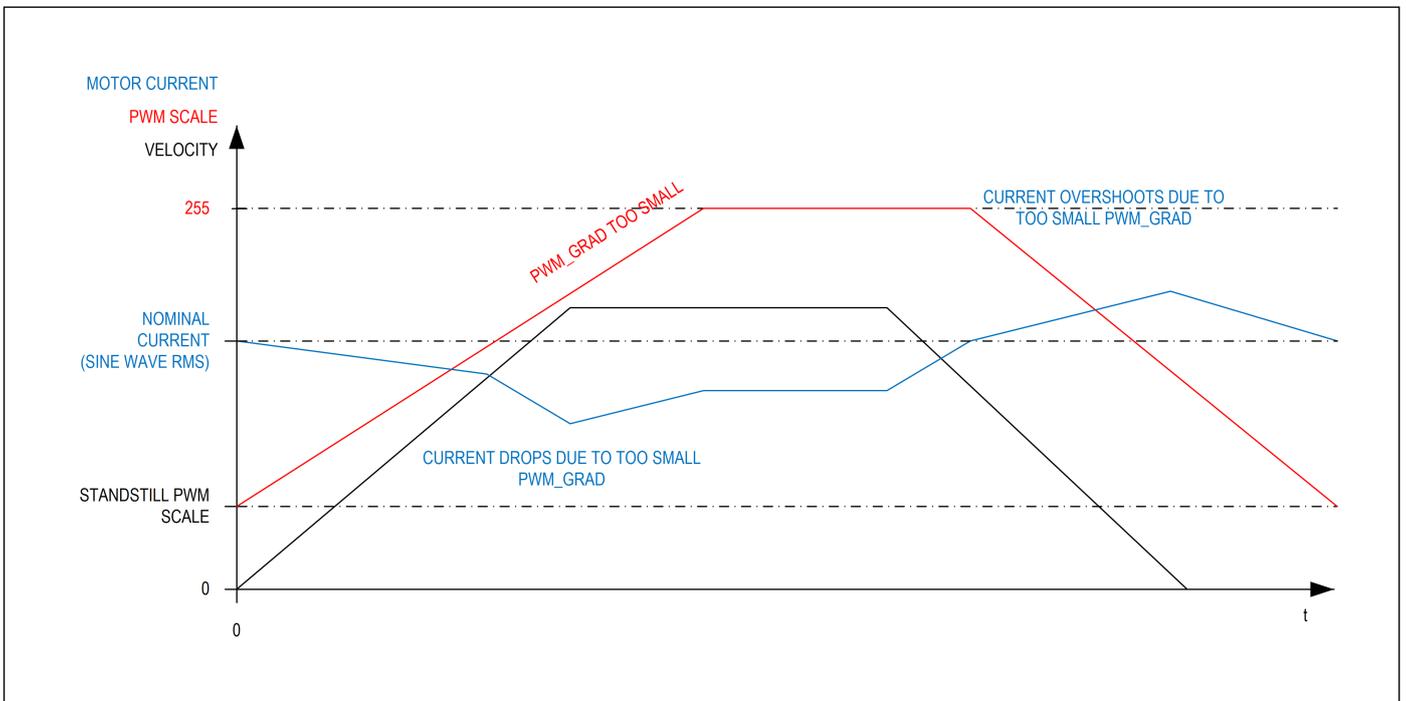


図 17. PWM\_GRAD の過小設定の例

## 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電流レギュレーションに下限値を設けます。チョッパがオンのフェーズ時 (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、停止パワーダウンを行うため) 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、モータ電流をゼロにできます。

StealthChop2 自動調整でのモータ・コイル下限電流値 (pwm\_meas\_sd\_enable = 0) は、次式で表せます。

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

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

$I_{\text{LOWERLIMIT}}$  は、最小公称 IRUN モータ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、必ず pwm\_meas\_sd\_enable = 1 に設定してください。

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

例：モータのコイル抵抗が  $5\Omega$ 、電源電圧が 16V とします。TBL = %01 および PWM\_FREQ = %00 の場合、 $t_{\text{BLANK}}$  は 24 クロック・サイクル、 $f_{\text{PWM}}$  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。

$$I_{\text{LOWERLIMIT}} = 24t_{\text{CLK}} \times \frac{2}{1024t_{\text{CLK}}} \times \frac{16\text{V}}{5\Omega} = \frac{24}{512} \times \frac{16\text{V}}{5\Omega} = 150\text{mA}$$

これは、関連する設定をすべて考慮すると、自動調整のモータ目標電流値を 150mA 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 を通じたモータ電流の変更の場合にもあてはまります。

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

$I_{\text{LOWERLIMIT}}$ ：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 でのモータ電流はこの下限値より大きくなくてはなりません。 $I_{\text{LOWERLIMIT}}$  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 の変化に対するドライバの対応能力も制限します。

電流下限値は、GLOBALSCALER の変化に対するドライバの対応能力も制限します。

電流下限値を無効にするには、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リング

非常に高い精度で安定したモータ動作を実現するために、純粋に速度ベースのフィードフォワード制御を用いることで、電流レギュレーションをオフに切り替えることができます。その原理を図 18 に示します。

速度ベースのスケーリングはオプションのモードで、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 振幅をスケーリングします。その後、電流は速度のみに基づいてスケーリングされます。

この機能は、pwm\_autoscale を 0 に設定した場合に使用できます。基本的な考え方は、モータ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・モータには一定のコイル抵抗があるため、目標電流を得るには、基本式  $I = U/R$  に基づき、一定の電圧振幅が必要です。ここで、 $R$  はコイル抵抗、 $U$  は PWM 値でスケーリングされた電源電圧です。

PWM\_OFS の初期値は次式で計算できます。

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

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

実効的な PWM 電圧  $U_{\text{PWM}}$  ( $1/\sqrt{2}$  × ピーク値) は、10 ビットの分解能と PWM\_SCALE\_SUM 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、次のように求められます。

$$U_{\text{PWM}} = V_S \times \frac{\text{PWM\_SCALE\_SUM}}{256} \times \frac{248}{256} \times \frac{1}{\sqrt{2}} = V_S \times \frac{\text{PWM\_SCALE\_SUM}}{374}$$

モータ速度の増加時には、モータの逆起電力の電圧が増加します。逆起電力の電圧は、モータ速度に比例します。これにより、コイル抵抗に有効な PWM 電圧が低下し、したがって電流も減少します。TMC5271 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とは無関係に自動的に計算されます。

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

CS\_ACTUAL には、IHOLD および IRUN の定義に従い、あるいは CoolStep の定義に従い、実際の電流スケールが考慮されています。

TSTEP は、256 マイクロステップ分解能に等価なマイクロステップ周波数、f<sub>clk</sub> はドライバに供給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、逆起電力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。したがって、PWM\_GRAD の第一近似は次のように計算されます。

$$\text{PWM\_GRAD} = C_{\text{BEMF}} \left[ \frac{\text{V}}{\frac{\text{rad}}{\text{s}}} \right] \times 2\pi \times \frac{f_{\text{clk}} \times 1.46}{V_S \times \text{MSPR}}$$

C<sub>BEMF</sub> は、モータの逆起電力定数で、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。

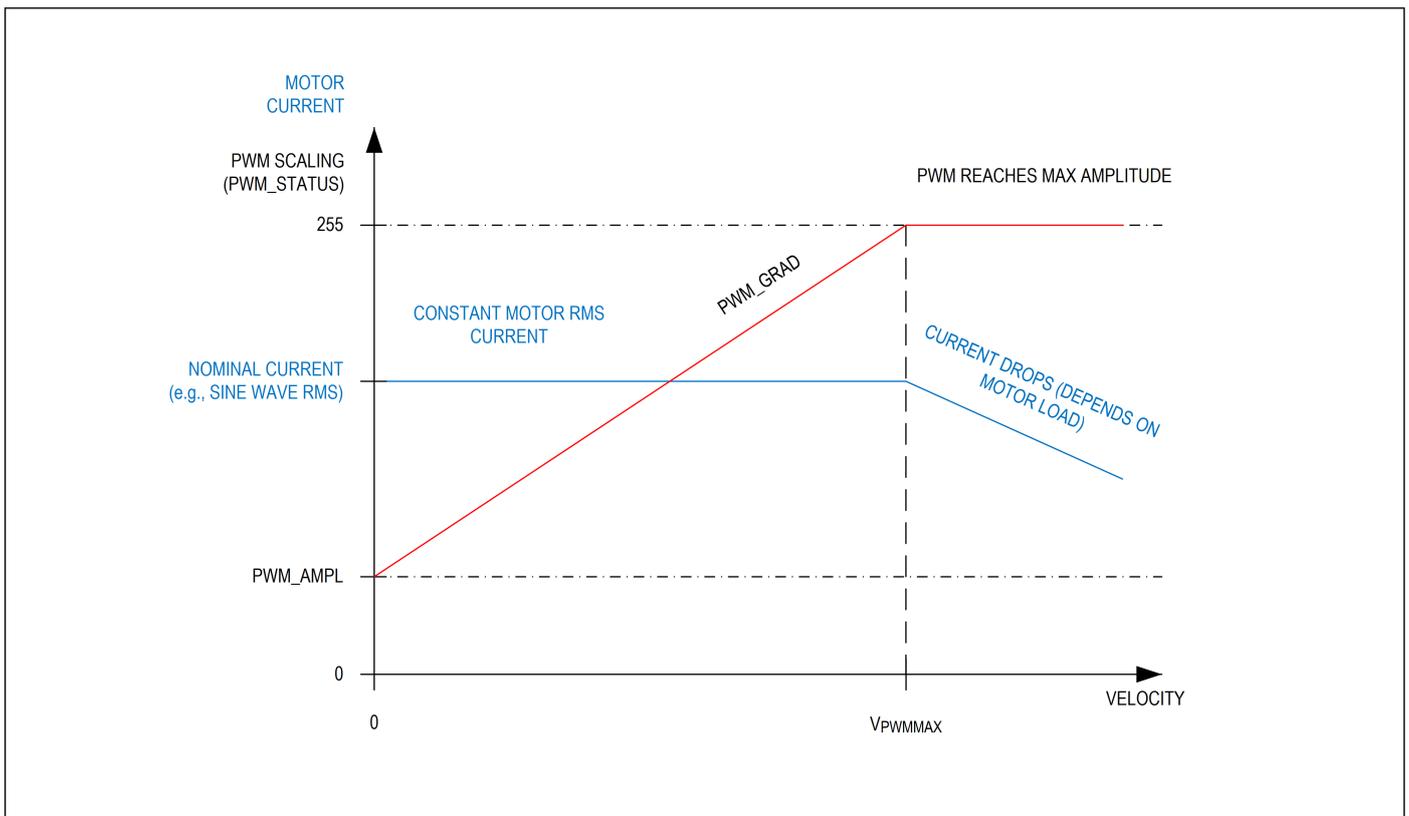


図 18. 速度ベースの PWM スケーリング (pwm\_autoscale = 0)

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。

### モータの逆起電力 (BEMF) 定数の解釈

BEMF 定数は、モータ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、BEMF 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 のモータの  $C_{BEMF}$  は 1V/rad/s です。そのようなモータを 1rps (1rps = 毎秒 1 回転 = 6.28rad/s) で回転させると、発生する BEMF 電圧は 6.28V となります。したがって、BEMF 定数は次のように計算できます。

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

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

Holding torque はモータ固有の保持トルクで、例えば、両方のコイルで電流が  $I_{COILNOM}$  のときに達するトルクです。トルクの単位は [Nm] で、1Nm = 100Ncm = 1000mNm です。

BEMF 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・ポジションを前提とするため、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ことができます。無騒音動作と最大の動的性能を組み合わせるため、TMC5271 では、速度スレッシュホールドに基づき StealthChop2 と SpreadCycle を組み合わせることができます。この場合、図 19 に示すように、StealthChop2 は低速時のみ有効です。

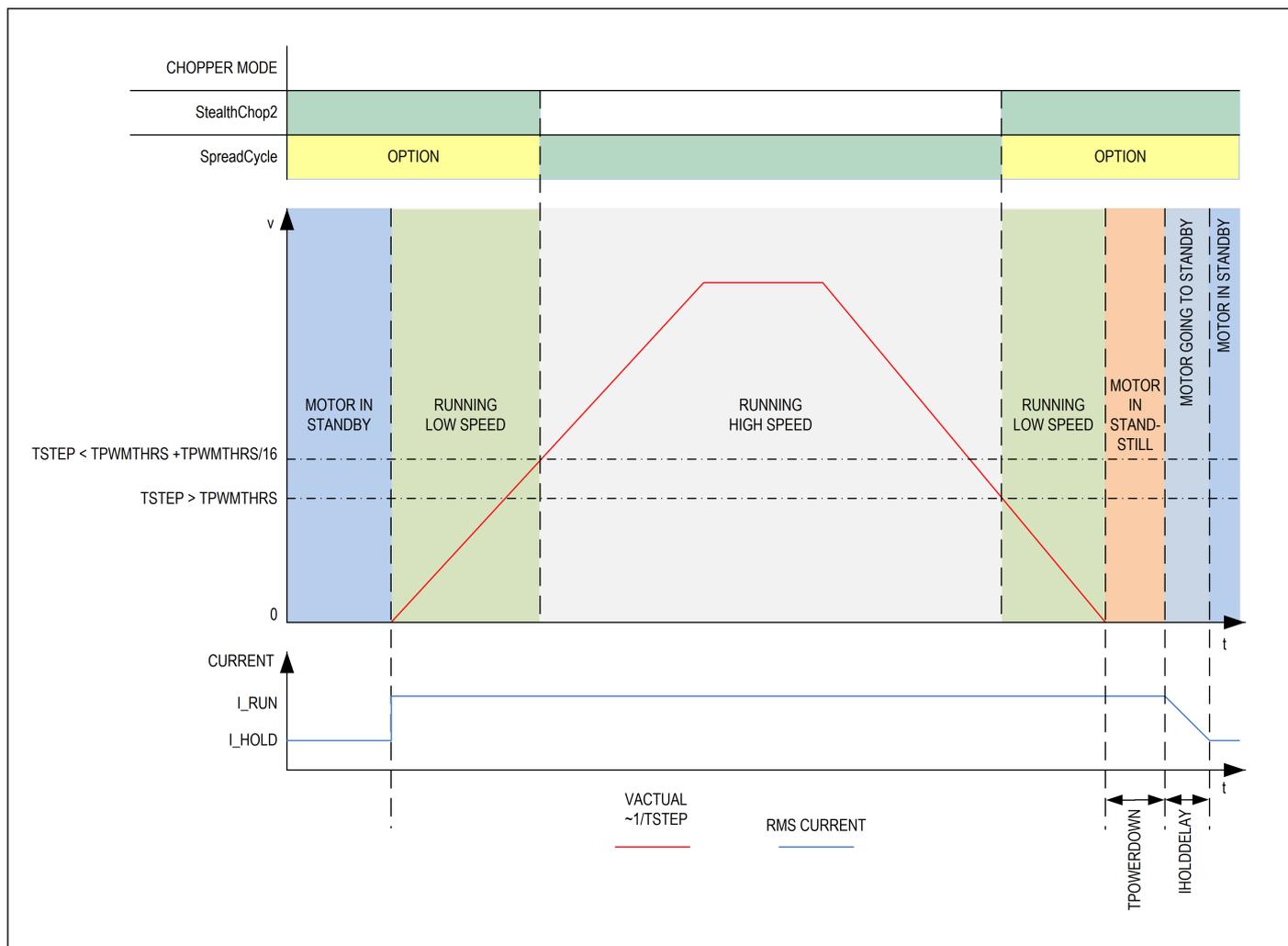


図 19. SpreadCycle への切り替えオプションのための TPWMTHRS

最初のステップとして、チョッパ原理である StealthChop2 と SpreadCycle の両方をパラメータ化し、個別に最適化します (StealthChop2 のセクションおよび SpreadCycle および一般的なチョッパのセクションを参照)。

次のステップは、スイッチオーバー速度を定義することです。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 :

自動補償 (sg\_angle\_offset) を行わないと、高速の切り替え時にジャークが発生します。モータの逆起電力 (速度と共に増加) がモータ電圧とモータ電流の間の位相を最大で 90°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が高速の場合、このジャークの発生時の強度が大きくなります。ジャークが大きい場合、一時的な過電流状態 (モータ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 1~数 10RPM) には、ほとんどのモータでジャークは完全に無視できます。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。これは、速度が切り替えスレッシュホールド未満に戻るまで、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、速度スレッシュホールド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。

モータがストールしたりモータ速度が突然変化したりした場合、ドライバ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、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、モータのストールといった逆起電力の突然の変化に対してドライバは遅れて反応します。

StealthChop2 での動作時にモータ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。直前のモータ速度、およびモータのコイル抵抗に応じて、モータ電流は数 10ms の時間にわたり大きく増加します。低速時、逆起電力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。

ドライバ段を、モータ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。これは過電流スレッシュホールドを 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の情報が、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、これは、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・コイルが不通となっていることを示します。
- OLA と OLB が時々セットされる場合は、モータ・コイルの抵抗値が約 5% 以上変動していることを示します。
- 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。

**注：**一方のコイルに開放負荷状態が存在すると、電流レギュレーションは、他方のコイルの目標電流を超え、過電流検出動作ポイントまで達する可能性があります。この理由は、測定上の制限による特定の状態で電流レギュレーションでは、コイルの電流をより高い目標電流でのみ安定化させるためです。これが問題となるアプリケーションでは、最初に SpreadCycle で開放負荷となっていないかどうかチェックしてください。

### PWM\_SCALE\_SUM からのモータ状態に関する情報

自動スケールリングで PWM\_SCALE\_SUM を読み出すことで、モータ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 0 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。短時間の始動を確保するために、モータ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・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

表 9 に、StealthChop2 チョップパー・モードに関連したすべてのパラメータを示します。

表 9. StealthChop2 を制御するパラメータ

パラメータ	説明	設定	注
en_pwm_mode	StealthChop2 の使用の一般的なイネーブル（GCONF レジスタ）。デフォルト = 0。	0	StealthChop2 ディスエーブル。SpreadCycle アクティブ。
		1	StealthChop2 イネーブル（速度スレッシュホールドに依存）。初期イネーブル時、モータは停止状態となっている必要があります。図 48 の自動調整フローチャートも参照してください。
pwm_meas_sd_enable	低速減衰フェーズ時の電流測定の制御。一般的なほとんどの使用で 1 を推奨。 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	電流下限値を無効化する他に低速減衰フェーズ時に電流を測定。
pwm_dis_reg_stst	このオプションは電流レギュレーションをディスエーブルし、停止時のレギュレーション・ノイズをなくします。 ディスエーブルすると、電源電圧が変化している場合やモータが冷却している場合に、モータ電流が停止時間中の長い時間にわたり変動する可能性があります。 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。目標のスレッシュホールド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 to 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	このレジスタは、SpreadCycle から StealthChop2 への切り替わり時に、PWM 振幅をより低い値にクリップします。ドライバは SpreadCycle 動作の間、それ以前に使用した StealthChop2 の振幅を保存します。両モード間の自動切り替えで最善の結果を得るには、これを 15 に設定し、SG4_THRS.sg_angle_offset ビットをセットします。 デフォルト = 12。	0 to 15	8 ビットの振幅制限の上位 4 ビット。

表 9. StealthChop2 を制御するパラメータ (続き)

パラメータ	説明	設定	注
pwm_autoscale	電流測定を用いる自動電流スケールをイネーブル。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ル。
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。最適な電流レギュレーションを行うには良好な結果を与える最小設定を使用します (最大の PWM 分解能)。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。ただし、値が高すぎる場合は、レギュレーション・ノイズおよびオーバーシュートが生じます。 デフォルト = 4。	1 to 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および初期化の値を与えます。 デフォルト = 0x1D。	0 to 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がディスエーブルされます。
PWM_GRAD	ユーザ定義の PWM 振幅 (勾配)。PWM_GRAD_AUTO の自動調整に対して、速度ベースのスケールおよび初期化の値を与えます。これにより先に決定した PWM_OFS のプリロードが可能になります。 デフォルト = 0。	0 to 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。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。	0 to 1023	-
FREEWHEEL	モータ電流設定値がゼロ ( $I_{HOLD} = 0$ ) 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・オプションにより、モータ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。
		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 スケール補正のリード・バック。調整時にはほぼ 0 に安定化します。	-255 to 255	読出し専用。SpreadCycle での動作時、スケール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。	0 to 255	読出し専用。
TOFF	モータ・ドライバの一般的なイネーブル。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 to 15	ドライバ・イネーブル。

表 9. StealthChop2 を制御するパラメータ (続き)

パラメータ	説明	設定	注
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。設定値を上げると、pwm_meas_sd_enableを設定しない限り、StealthChop2のデューティサイクルが低下するため、最小電流が増加します。  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、モータ速度やモータ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・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。図20に、いくつかのチョッパー減衰フェーズを示します。

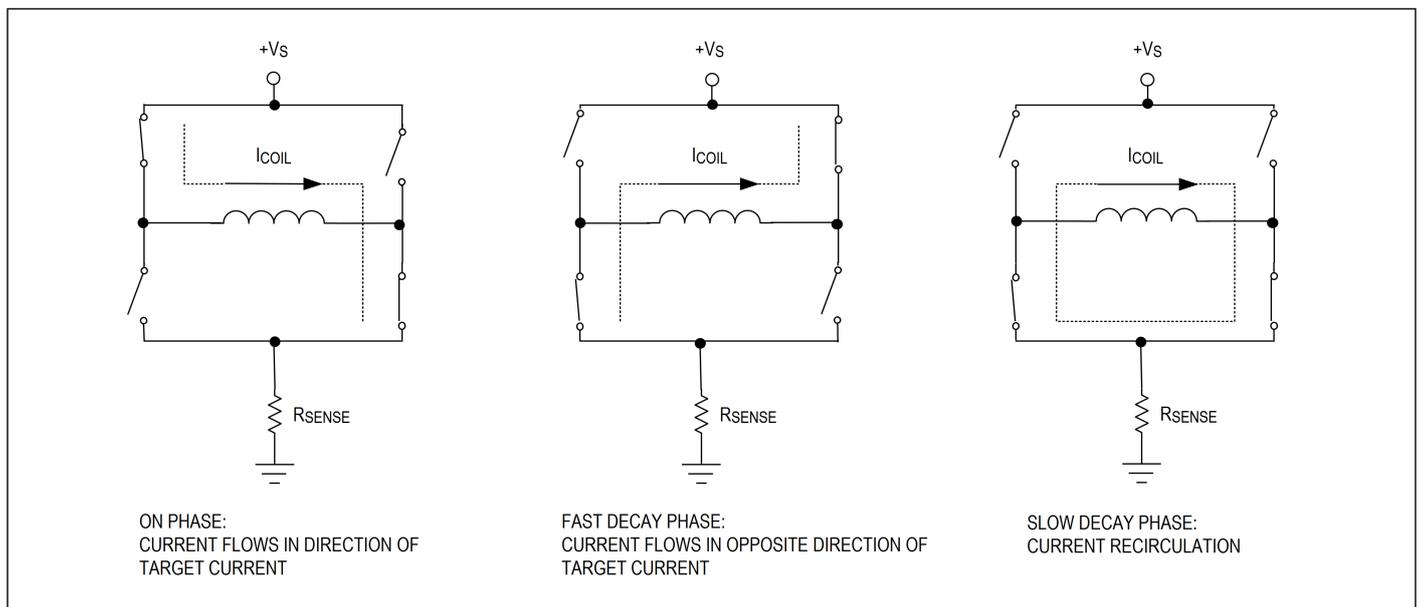


図 20. 代表的なチョッパー減衰フェーズ

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。

コイル電流の切り替わり時、寄生容量の充放電により R<sub>DS(ON)</sub>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マスクされている時間のことです。

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。高性能チョッパー・アルゴリズム SpreadCycle (chm=0) と、定評のある定オフ時間チョッパー・モード (chm=1) 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。表10に、これら2つのチョッパー・モードに関連したパラメータを示します。

チョッパー周波数は、チョッパー・モータ・ドライバにとって重要なパラメータです。周波数が低すぎる場合は可聴ノイズが発生します。周波数を上げると、モータの電流リップルは減りますが、周波数が高すぎる場合は磁気損失が生じる可能性があります。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、適切な妥協点を見つける必要があります。ほとんどのモータは 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 10. 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	注
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、モータをイネーブルすることが必要です。StealthChop2 のみで動作する場合、任意の設定が使用できます。  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 to 15	オフ時間の設定値 $N_{CLK} = 24 + 32 \times T_{OFF}$ (1 に設定すると 28 クロックの最小ブランキング時間 TBL 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	$20 t_{CLK}$ 制限事項：この設定を使用できるのは、10MHz 以下の外部クロックを組み合わせた場合のみです
		1	$28 t_{CLK}$
		2	$36 t_{CLK}$
		3	$54 t_{CLK}$
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チョッパー

## SpreadCycle チョッパー

SpreadCycle 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、モータやドライバを低消費電力化するのに重要です。

以下に、低速減衰時間 TOFF の開始値の計算例を示します。

- 目標チョッパー周波数：25kHz
  - $t_{OFF} = 1/25\text{kHz} \times 50/100 \times 1/2 = 10\mu\text{s}$
  - 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 =  $(t_{OFF} \times f_{CLK} - 12)/32$
- クロックが 12MHz の場合は、TOFF = 3.4 という結果が得られ、TOFF は 3 または 4 に設定する必要があります。
- クロックが 16MHz の場合は、TOFF = 4.6 という結果が得られ、TOFF は 4 または 5 に設定する必要があります。

**ヒント:** モータ速度を最大にする場合は、TOFF を 1 または 2 に設定し、また短い 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・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、チョッパは、立上がりおよび立下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・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、チョッパ周波数は低下します。モータのインダクタンスは、チョッパがモータ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。ブランキング中は電流コンパレータがディスエーブルされているためです。

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が低すぎるとモータ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が高すぎる場合は、チョッパ周波数が低下し、チョッパ・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。

この設定はモータには依存しません。大電流のモータは、通常、コイル抵抗も低いためです。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のノイズとモータの消費電力が増加します。高速減衰時間がブランキング時間よりわずかに長くなると、設定は最適なものとなります。これが困難な場合は、オフ時間の設定を短くすることができます。

電源電圧に比較してコイル抵抗が高い場合などのように、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。内蔵のヒステリシス・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・クロックごとにステップ単位で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。図 21 に示すように、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、そのサイクルの間デクリメントを行い、チョッパ・サイクルが終了するまで、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は安定化します。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。関連するヒステリシス・パラメータを表 11 に示します。

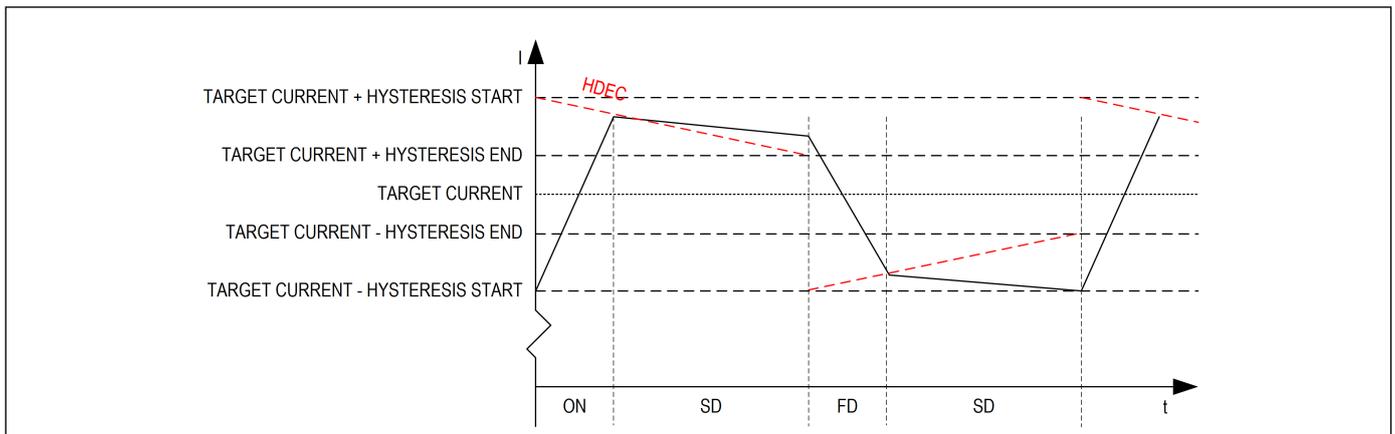


図 21.1 チョッパ・サイクルの間のコイル電流を示す、SpreadCycle チョッパの概略図

表 11. SpreadCycle モードのヒステリシス・パラメータ

パラメータ	説明	設定	注
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 to 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。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 to 2	-3~-1 : 負の HEND。
		3	0 : HEND はゼロ。
		4 to 15	1~12 : 正の HEND。

HSTRT = 0 かつ HEND = 0 の場合でも、TMC5271 はアナログ回路を介して最小のヒステリシスを設定します。以下に、ヒステリシスが最小の例を示します。

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値を用いない場合、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 (例えば 4) を、HSTRT 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電流リップルおよび消費電力の原因となるほど長すぎるとはいけません。その基本的な原理を図 22 に示します。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。

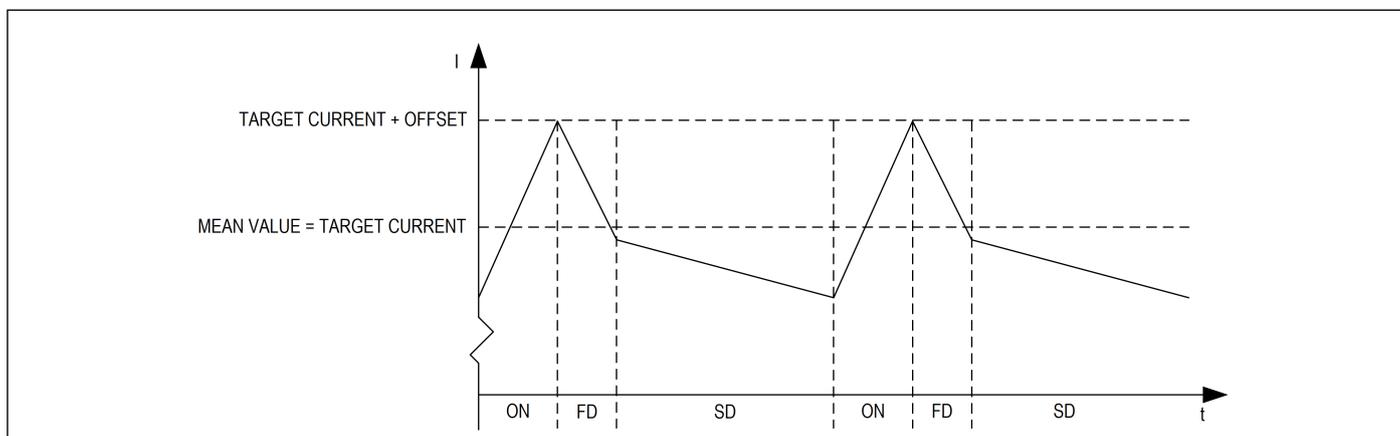


図 22. オフセットを持つ一般的な定オフ時間チョッパーが示すコイル電流

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、図 23 に示すように、高速減衰フェーズでは、モータ電流の絶対値が目標電流より小さくなるためです。ゼロ・オフセットが低すぎると、モータ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。表 12 にこのモードに関連するパラメータを示します。

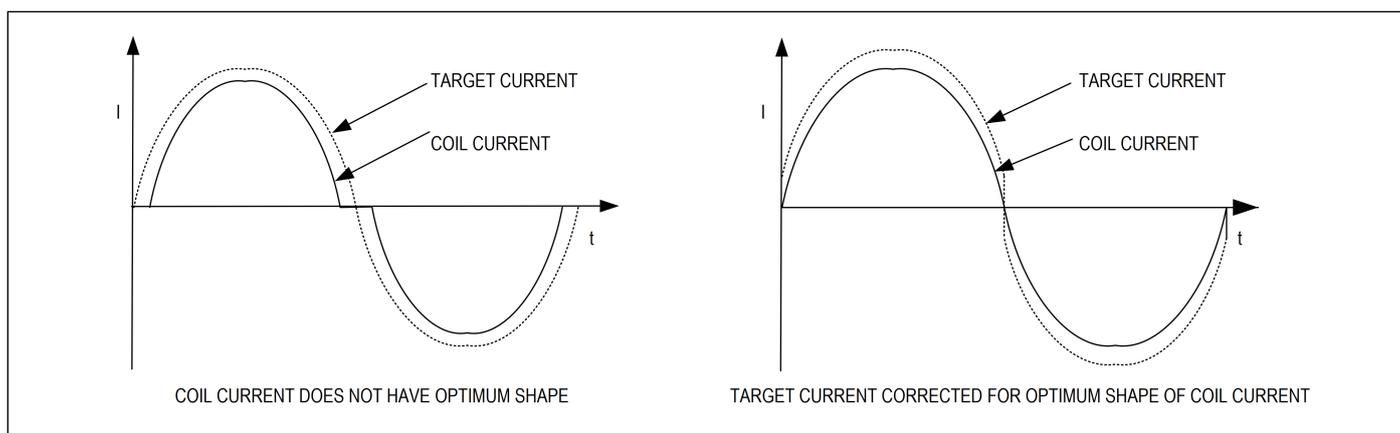


図 23. 一般的なチョッパのゼロ交差とサイン波オフセットを使用した補正

表 12. 定オフ時間チョッパ・モードを制御するパラメータ

パラメータ	説明	設定	注
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・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 to 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 to 2	負のオフセット: -3~-1
		3	オフセットなし: 0
		4 to 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

## 内蔵電流検出機能 (ICS)

TMC5271 には、電力散逸のない電流検出機能が内蔵されています。この機能により、外部電流検出機能を用いる場合に通常必要になる大きな外付け電力抵抗が不要になります。この ICS により、外部センス抵抗に基づく従来型のアーキテクチャと比較して、大幅な省スペースと省電力を実現できます。最適性能を実現するために、ICS では各パワー-MOSFET の  $R_{DS(ON)}$  を個別に測定し、最良の結果が得られるよう個々の MOSFET の温度を考慮します。

## モータ電流の設定

表 13 に、TMC5271 のモータ電流設定に関連するパラメータを示します。

表 13. モータ電流を制御するパラメータ

パラメータ	説明	設定	注
IRUN	モータ動作中の電流スケーリング。内蔵のサイン波テーブルから取得したコイル電流値をスケーリングします。高精度のモータ動作のためには、電流スケーリング係数を 16~31 の範囲にします。電流値をデジタル的にスケーリング・ダウンするとマイクロステップが粗くなり、実効的なマイクロステップ分解能が低下するためです。この設定は、CoolStep で設定される最大電流値も制御します。  デフォルト = 31。	0 to 31	スケーリング・ファクタ 1/32、2/32、~32/32。
IHOLD	IRUN と同じですが、停止時のモータ用です。  IHOLD は通常は IRUN より大幅に小さい値で、モータ・ポジションを維持するのに必要な最小限の値までスケール・ダウンできます。  デフォルト = 8。		
IHOLDDELAY	動作電流からホールド電流への滑らかな電流減少を可能にします。IHOLDDELAY は、TZEROWAIT の時間が経過してからモータがパワーダウンするまでのクロック・サイクル数を、2 <sup>18</sup> クロックのインクリメントで制御します。  0 = 即時にパワーダウン。 1~15 : 2 <sup>18</sup> クロックの倍数で表した電流ステップあたりの電流低減遅延。  例 : IRUN = 31 および IHOLD = 16 の場合、ホールド電流を低減するには、15 電流ステップが必要です。したがって、IHOLDDELAY を 4 に設定すると、パワーダウン時間は 4 × 15 × 2 <sup>18</sup> クロック・サイクルとなります。これは 16MHz の場合約 1 秒になります。  デフォルト = 1。	0	IHOLD への即時パワーダウン。
		1 to 15	電流デクリメントあたり、1 × 2 <sup>18</sup> ~15 × 2 <sup>18</sup> クロック。
IRUNDELAY	開始が検出されてからモータが起動するまでのクロック・サイクル数を制御します。  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・トルクを達成するには、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、モータの磁界を構成するために必要なエネルギーから生じる電源のピーク電流を防止できます。  デフォルト = 4。	0	所定ポジションに達した場合に即時に IRUN にパワーアップ。
		1 to 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。
TPOWERDOWN	TPOWERDOWN は、モータの停止 (stst) 後からモータ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 : StealthChop2 の PWM_OFFS_AUTO を自動調整するには、少なくとも 2 に設定することが必要です。  デフォルト = 10。	0 to 255	0~((2 <sup>8</sup> ) - 1) × 2 <sup>18</sup> t <sub>CLK</sub> 。
FSR_IREF	リファレンス電流 I <sub>REF</sub> (R <sub>REF</sub> に基づく) をスケーリングします。これは外部抵抗 R <sub>REF</sub> をスケーリングするのと同じです。  これと FSR_Mx を使用すると電流スケーリングの微調整が可能です。	0 to 3	0 : 25%の I <sub>REF</sub> 。 1 : 50%の I <sub>REF</sub> 。 2 : 75%の I <sub>REF</sub> 。 3 : 100%の I <sub>REF</sub> 。

表 13. モータ電流を制御するパラメータ (続き)

パラメータ	説明	設定	注
FSR	フルスケール・レンジ。 フルスケール・ビットにより、基本的に、ドライバの $R_{DS(ON)}$ の電流検出をモータ電流範囲に適応させることができます。 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。 また、フルスケール・ビットは、過電流保護スレッシュホールドも定義します (この値はピーク電流設定です)。	0 to 3	$R_{DS(ON)}$ および過電流保護スレッシュホールドの値は、電気的特性のセクションの出力仕様および保護回路の項に記載されています。
GLOBALSCALER_A	モータ電流のグローバル・スケーリング。微調整して特定のモータに合わせるには、この値を電流スケーリングに乗じます。  位相 A をスケーリングします。  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。  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。 SpreadCycle のみを使用する場合にのみ、値を変更して使用できます。	0 to 255	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 = フルスケール電流の 32/256~255/256。
GLOBALSCALER_B	モータ電流のグローバル・スケーリング。微調整して特定のモータに合わせるには、この値を電流スケーリングに乗じます。  位相 B をスケーリングします。  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。  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。 SpreadCycle のみを使用する場合にのみ、値を変更して使用できます。	0 to 255	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 = フルスケール電流の 32/256~255/256。

## フルスケール電流の設定

フルスケール電流  $I_{FS}$  は最大の絶対電流レベル設定で、外部リファレンス抵抗  $R_{REF}$  および  $DRV\_CONF$  レジスタの 2 つのパラメータによって定義されます。 $I_{FS}$  の適用は、可能な限り最高の電流制御分解能を活用するために必要です。最大 16 通りからなるフルスケール電流レンジを設定することで、同じ外部抵抗  $R_{REF}$  を使って異なるモータ・サイズやアプリケーションに適用できます。

$R_{REF}$  には、精度 1% の標準的な低電力抵抗で十分です。外付け抵抗  $R_{REF}$  の範囲は 10k $\Omega$ ~60k $\Omega$  です。

そのため、 $I_{REF}$  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 2 つの 2 ビット・パラメータ  $FSR$  および  $FSR\_IREF$  は、ドライバ段の標準的なオン抵抗を定義し ( $FSR$ )、更に、 $I_{REF}$  をスケーリングして  $R_{REF}$  に基づくフルスケール・レンジを制御します ( $FSR\_IREF$ )。

$FSR\_IREF$  は、表 14 に示すように、 $KIFS\_IREF$  スケーリング係数を定義するために 4 種類の設定が可能です。

表 14.  $FSR\_IREF$  設定に基づく  $KIFS\_IREF$  スケーリング係数

$FSR\_IREF$ BITS 1:0	$KIFS\_IREF$
11	1.00
10	0.75
01	0.50
00 (default)	0.25

下記の式は、フルスケール電流  $I_{FS}$  を、 $R_{REF}$  抵抗および  $DRV\_CONF$  のパラメータの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 FSR に依存します。

$I_{REF}$  のスケール係数は FSR\_IREF に依存します。

$$I_{FS}(RMS) = \frac{(K_{IFS}(KV) \times KIFS\_IREF)}{R_{REF}(k\Omega)}$$

$$I_{FS}(PEAK) = \frac{(K_{IFS}(KV) \times KIFS\_IREF)}{R_{REF}(k\Omega)} \times \sqrt{2}$$

この2つの式を用い、 $R_{REF} = 10k\Omega$  および  $R_{REF} = 36k\Omega$  の場合のフルスケール・レンジの値を、表 15 および表 16 に示します。

注：

- FSR と FSR\_IREF の組み合わせによっては、フルスケール・レンジが同じになる場合があります。電流制御精度を最高のものにするには、できるだけ大きな FSR\_IREF を選択することが常に推奨されます。
- 最適な電流レギュレーション性能と適切な電流レベルを実現するには、対称の基板レイアウトを TMC5271 に用いることが必要です。

表 15.  $I_{FS}$  フルスケール・レンジ設定 (標準的な  $R_{REF} = 10k\Omega$ )

REGISTER CONFIGURATION		KIFS_IREF	$K_{IFS}$ (A x k $\Omega$ )	$I_{FS}$ (A RMS)	$I_{FS}$ (A PEAK)	TYPICAL $R_{DS(ON)}$ (LS) ( $\Omega$ )
FSR BITS 1:0	FSR_IREF BITS 1:0					
11	11	1.0	16	1.60	2.26	0.055
	10	0.75		1.20	1.69	
	01	0.5		0.80	1.13	
	00 (default)	0.25		0.40	0.56	
10	11	1.0	12.06	1.21	1.70	0.070
	10	0.75		0.9	1.28	
	01	0.5		0.60	0.85	
	00 (default)	0.25		0.30	0.43	
01	11	1.0	8.16	0.82	1.15	0.100
	10	0.75		0.61	0.86	
	01	0.5		0.41	0.58	
	00 (default)	0.25		0.20	0.29	
00 (default)	11	1.0	4.1	0.41	0.58	0.185
	10	0.75		0.31	0.43	
	01	0.5		0.21	0.29	
	00 (default)	0.25		0.10	0.14	

表 16.  $I_{FS}$  フルスケール・レンジ設定 ( $R_{REF} = 36k\Omega$  の場合の例)

REGISTER CONFIGURATION		KIFS_IREF	$K_{IFS}$ (A x k $\Omega$ )	$I_{FS}$ (A RMS)	$I_{FS}$ (A PEAK)	TYPICAL $R_{DS(ON)}$ (LS) ( $\Omega$ )
FSR BITS 1:0	FSR_REF BITS 1:0					
11	11	1.0	16	0.444	0.628	0.055
	10	0.75		0.333	0.471	
	01	0.5		0.222	0.314	
	00 (default)	0.25		0.111	0.157	
10	11	1.0	12.06	0.335	0.474	0.070
	10	0.75		0.251	0.355	
	01	0.5		0.168	0.237	

表 16. I<sub>FS</sub> フルスケール・レンジ設定 (R<sub>REF</sub> = 36kΩ の場合の例) (続き)

	00 (default)	0.25		0.084	0.118	
01	11	1.0	8.16	0.227	0.321	0.100
	10	0.75		0.170	0.240	
	01	0.5		0.113	0.160	
	00 (default)	0.25		0.057	0.080	
00 (default)	11	1.0	4.1	0.114	0.161	0.185
	10	0.75		0.085	0.120	
	01	0.5		0.057	0.080	
	00 (default)	0.25		0.028	0.040	

### 速度ベースのモード制御

TMC5271 では、最適なモータ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。いくつかの速度スレッシュホールドでは、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。

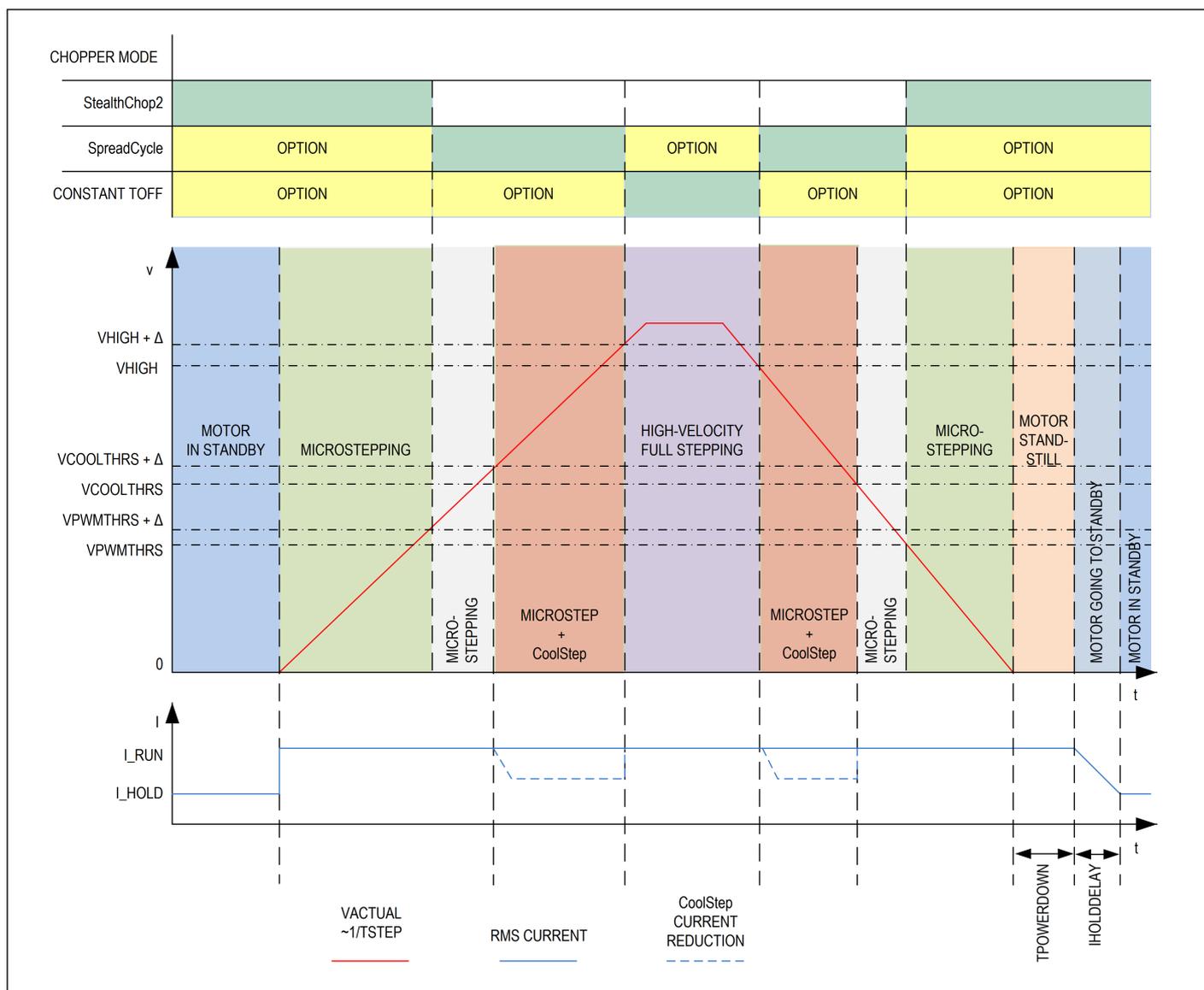


図 24. 速度依存モードの選択

図 24 に、使用可能なすべてのスレッシュホールドと必要な順序を示します。VPWMTHRS、VHIGH、VCOOLTHRS は、TPWMTHRS、THIGH、TCOOLTHRS の設定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、スレッシュホールドをそれに合わせて変化させる必要はありません。スレッシュホールドは、マイクロステップの設定値とは無関係に、同じモータ速度を表します。TSTEP は、これらのスレッシュホールド値と比較されます。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に、 $1/16TSTEP$  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、スレッシュホールドとして設定した  $1/16$  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。

自動速度スレッシュホールド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、ソフトウェアで活性化したり非活性化したりする必要はありません。

表 17. 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	注
stst	各動作モードでモータ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 <sup>20</sup> 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、読み出し専用。
TPOWER DOWN	これは、モータの停止 (stst) 後からモータ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 ( $f_{CLK} = 16\text{MHz}$ 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 to 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 to 1048575	ステータス・レジスタ、読み出し専用。 $t_{CLK}$ 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ます (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 to 1048575	StealthChop2 の動作用上限速度スレッシュホールドを制御するための設定値。
TCOOLTHRS	TCOOLTHRS $\geq$ TSTEP $\geq$ THIGH : <ul style="list-style-type: none"> <li>StallGuard2/4 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は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 to 1048575	CoolStep と StallGuard2/4 の動作用下限速度スレッシュホールド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 TFD = 0 (低速減衰のみの定オフ時間) の場合に chm = 1 に切り替わります。</li> <li>チョッパー同期はオフに切り替わります (SYNC = 0)</li> <li>vhighfs が設定されていると、モータはフル・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。</li> </ul> デフォルト : 0。	0 to 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用の上限スレッシュホールドを制御するための設定値。
small_hysteresis	TSTEP (下限速度スレッシュホールド) と $(TSTEP \times 15/16) - 1$ または $(TSTEP \times 31/32) - 1$ (上限速度スレッシュホールド) 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。 内蔵モーション・コントローラと共に使用する場合に推奨。
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° の位置でのみ生じます。フル・ステップの目標電流は、45° ポジション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・ステップへの切り替わりなし。
		1	高速でのフル・ステップ。

表 17. 速度ベースのモード制御パラメータ（続き）

パラメータ	説明	設定	注
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	高速時の一般的な定時間オフ・チョッパー。
en_pwm_mode	StealthChop2 電圧 PWM イネーブル・フラグ（速度スレッシュホールドに依存）。停止時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 なし。
		1	StealthChop2 が設定されていて TSTEP > TPWMTHRS の場合に StealthChop2 を活性化。

## 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た動作プロファイルを、自動で計算します。TMC5271 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・モータのトルク曲線に合わせることができます。ポジション・モードでは、加速度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

TMC5271 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。表 18 に、すべてのランプ・ジェネレータ・パラメータとその単位、および  $f_{CLK}$  への依存性を示します。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。

$v[TMC5271]$  および  $a[TMC5271]$  は、TMC5271 の内部単位です。これらは、TMC5271 の速度/加速度レジスタに書込む必要のある値です。計算機ツールおよび評価用ツールは製品ウェブサイトから入手できます。

表 18. 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
$f_{CLK}$	Hz	TMC5271 のクロック周波数。
s	s	秒。
$\mu S$	Microstep	—
FS	Full step	—
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—full-step count	—	一回転あたりのモータのフル・ステップ、例えば 200。
$\mu step\ velocity\ v[Hz]$	microsteps/s	$v[Hz] = v[TMC5271] \times (f_{CLK}[Hz] / 2^{24})$ .
$\mu step\ acceleration\ a[Hz/s]$	microsteps/s <sup>2</sup>	$a[Hz/s] = a[TMC5271] \times f_{CLK}[Hz]^2 / 2^{41}$ .
Rotations per second $v[rps]$	rotations/s	$v[rps] = v[microsteps/s] / USC / FSC$ .
rps acceleration $a[rps/s^2]$	rotations/s <sup>2</sup>	$a[rps/s^2] = a[microsteps/s^2] / USC / FSC$ .

表 18. ランプ・ジェネレータのパラメータと単位の関係 (続き)

パラメータ/記号	単位	説明
Ramp steps[microsteps] = rs	microsteps	$rs = (v[TMC5271])^2/a[TMC5271]/2^6$ 線形加速度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	-	$TSTEP = f_{CLK}/f_{256STEP} = f_{CLK}/(f_{STEP} \times 256/USC)$ $= 2^{24}/(VACTUAL \times 256/USC)$ .  速度スレッシュホールド用の時間基準は、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 ( $f_{256STEP}$ ) を基準にしています。
Ramp generator update rate	Hz	$f_{UPDATE} = f_{CLK}/512$ . 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ることがあります。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、シーケンサ入力のマイクロステップ分解能は減少できます。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

ポジショニング・モードの場合、ランプ・ジェネレータは、3 つの加速フェーズと 3 つの減速フェーズに加えて、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ステッピング・モータのトルクが減少するため、高速時の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、システムの減速能力は加速能力より高いことが明らかです。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。これにより、時間が重要なアプリケーションでのモータの動作速度を最大化できます。

目標のポジションおよびランプ・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、目標に達するための最善の (最も速い) 方法を常に使用しますが、ユーザが設定した加速度の制約は守ります。その結果、動作が自動的に停止し、ゼロを交差し、再度逆に動作する可能性も生じます。例えば、最終的な減速フェーズ時に目標ポジションが再度変更になり、より近いポジションに引きつけられた場合、設定された減速度値では直接新しい目標ポジションに達することはできません。この場合は、`second_move` フラグ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。

図 25~図 31 に代表的な (折れ曲がりの) ケースの例を示します。

#### 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、常に低い値 (1000~10) に設定する必要があります。TSTOP = 0 には設定しないでください。設定されたマイクロステップ目標ポジション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 0 に設定します。

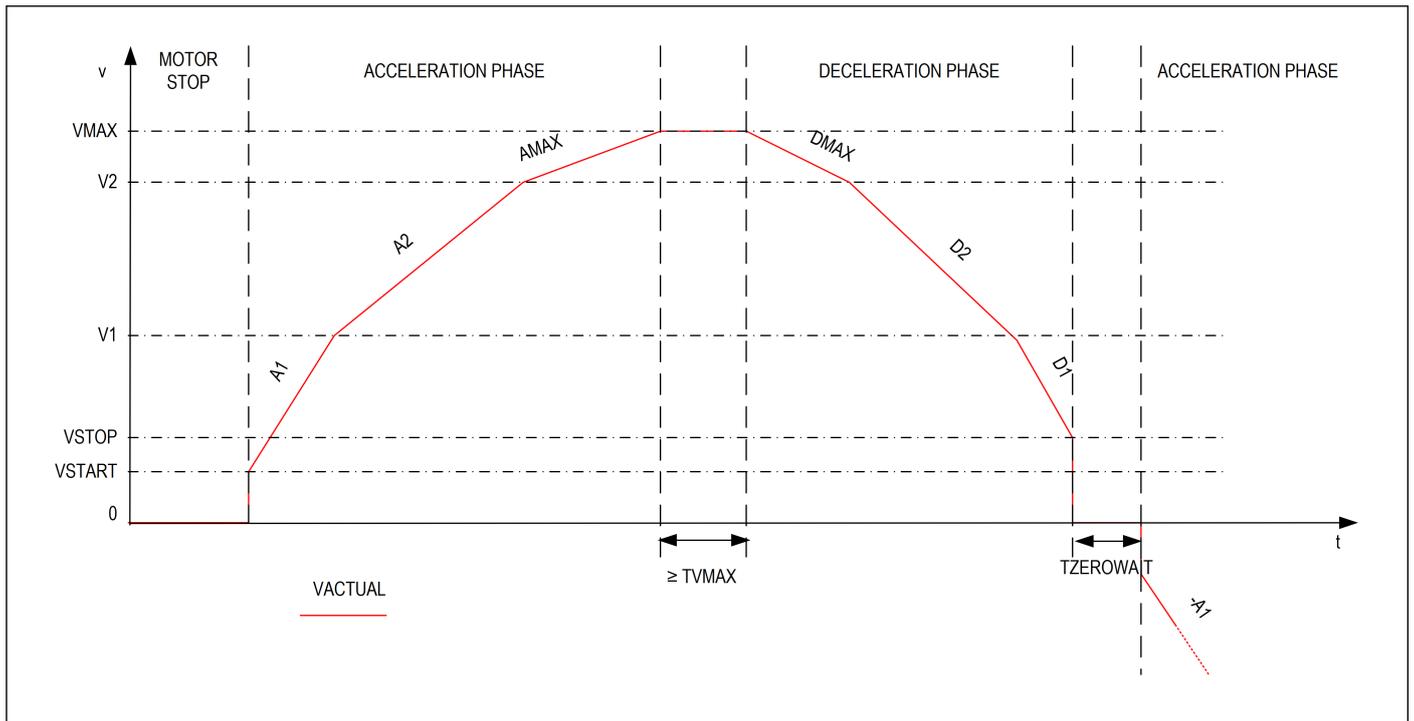


図 25. 2 回目の動作が負の方向に向かうランプ・ジェネレータ速度パターン

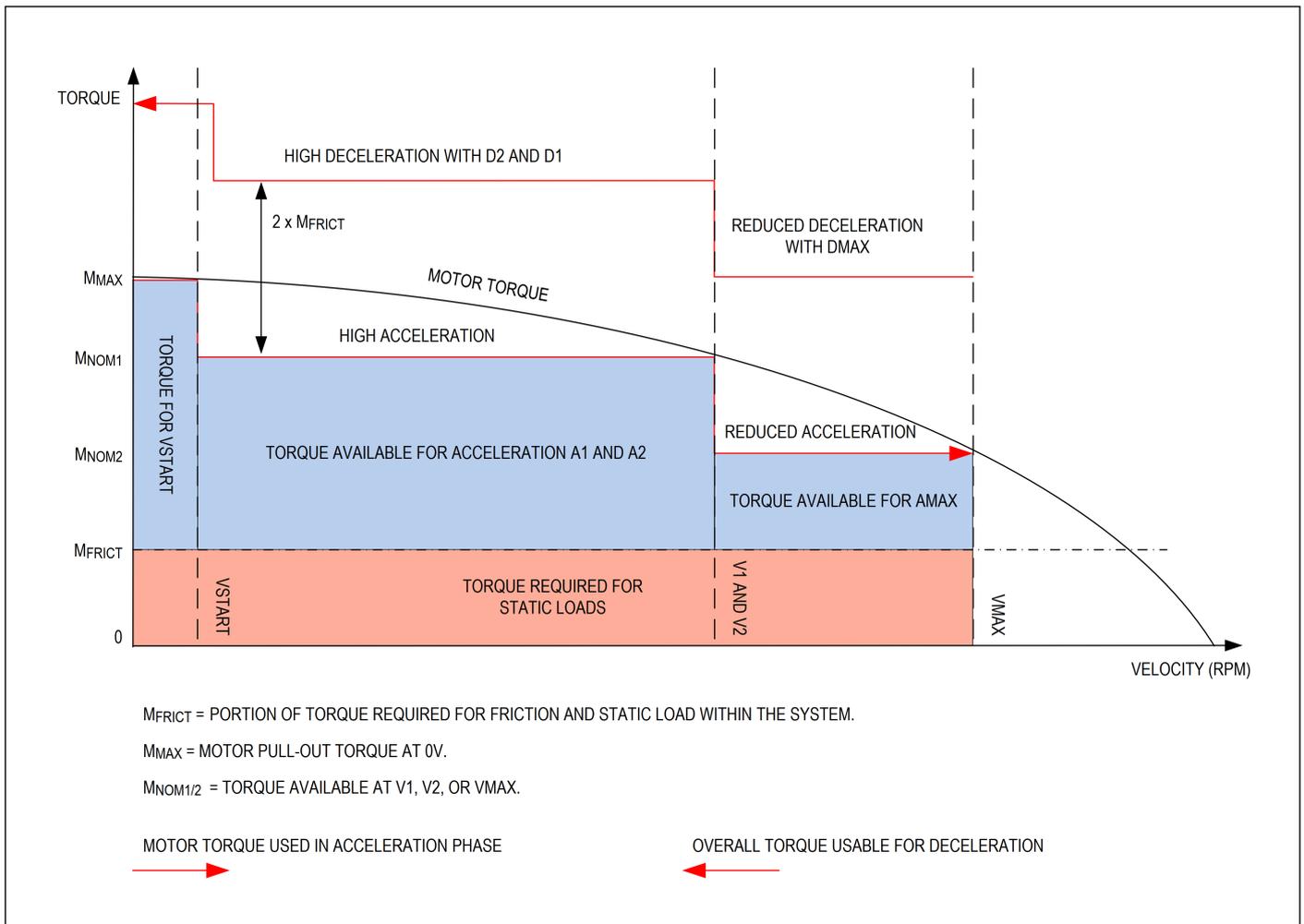


図 26. ランプ・ジェネレータを用いた最適なモータ・トルク使用法を示す図

## 8 点ランプ

### 始動速度および停止速度

必要な場合、短時間の運動を加速するためにモータの低速度範囲をスキップすることができます。これを行うには、VSTART および VSTOP のレベルを増加します。標準的な範囲は最大で数 10Hz のフル・ステップです。

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。モータがこれに追従できないと思われる場合、TZEROWAIT (ポジショニング・モードでのみ使用可能) を設定して後続の動作の時間遅延を設定できます。遅延時間が有効であることは、t\_zerowait\_active フラグで示されます。目標ポジションに達すると、position\_reached フラグが活性化されます。

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・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ポジション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) に基づいた定速セグメントを強制的に挿入できます。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。最小の VMAX' は VSTOP と同じです。図 27~図 31 に、擬似 S 字構成を実現するための様々なポジション間距離に基づいた速度プロファイルを示します。

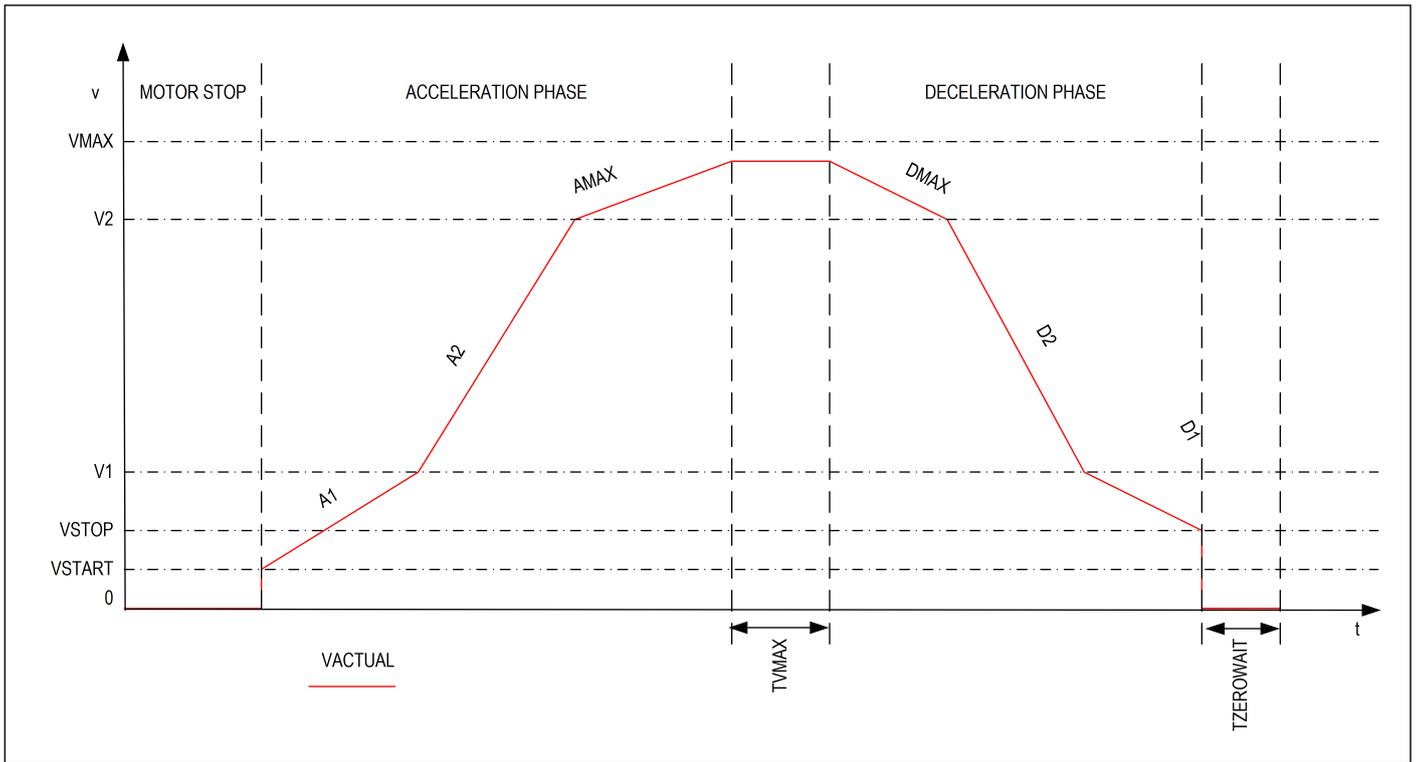


図 27. 距離が短すぎるために VMAX に到達できない 8 点ランプ

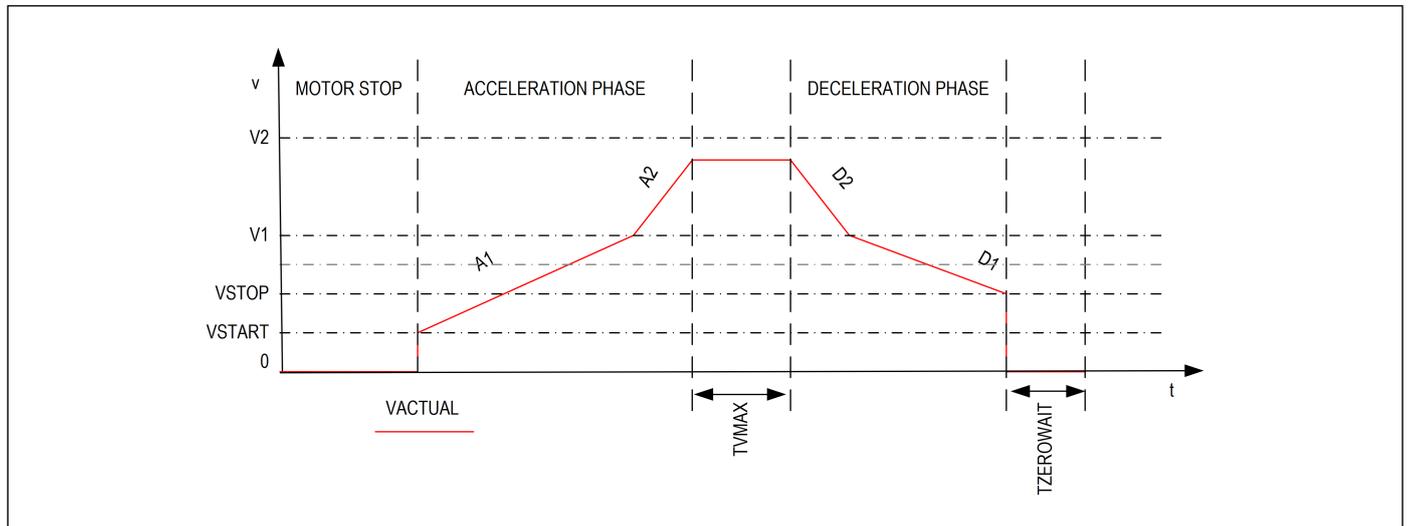


図 28. 距離が短いために V2 に達せず AMAX フェーズと DMAX フェーズがないプロファイル

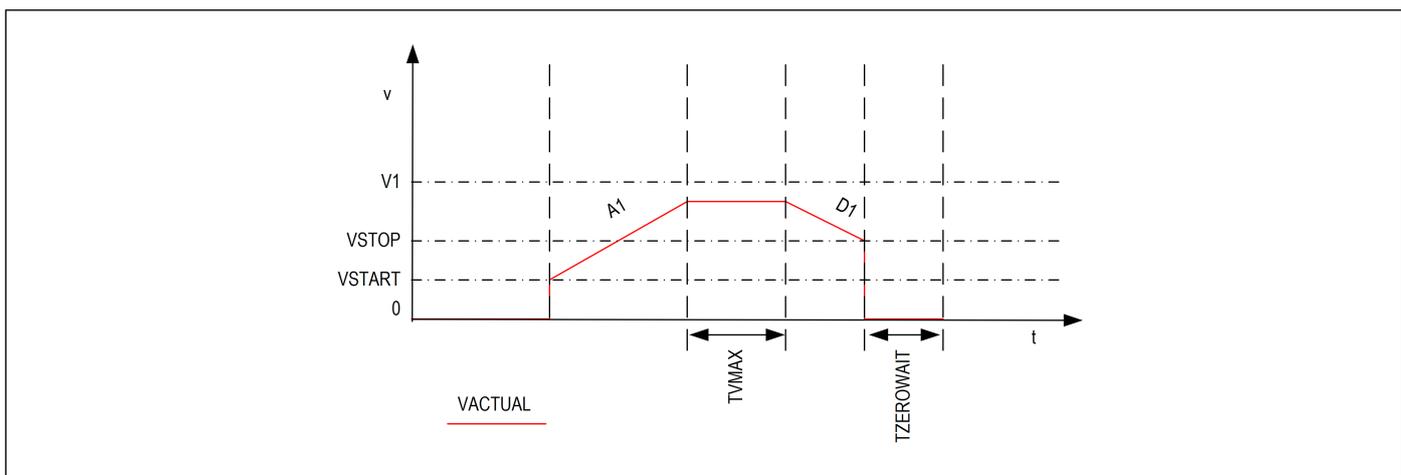


図 29. 距離が短いために V1 に達しないプロフィール

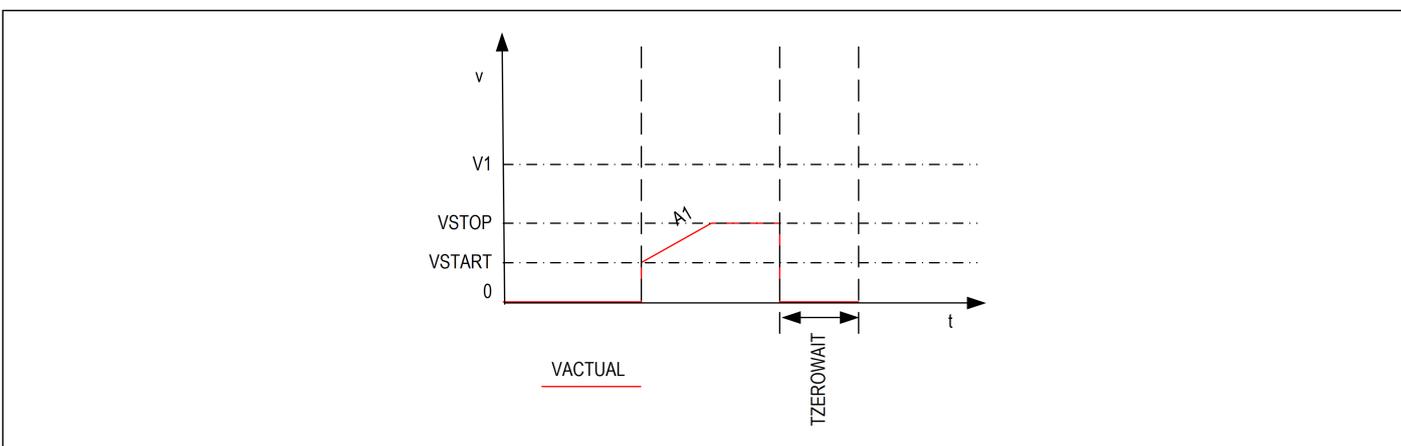


図 30. 距離が短いために TVMAX を維持できないプロフィール

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。

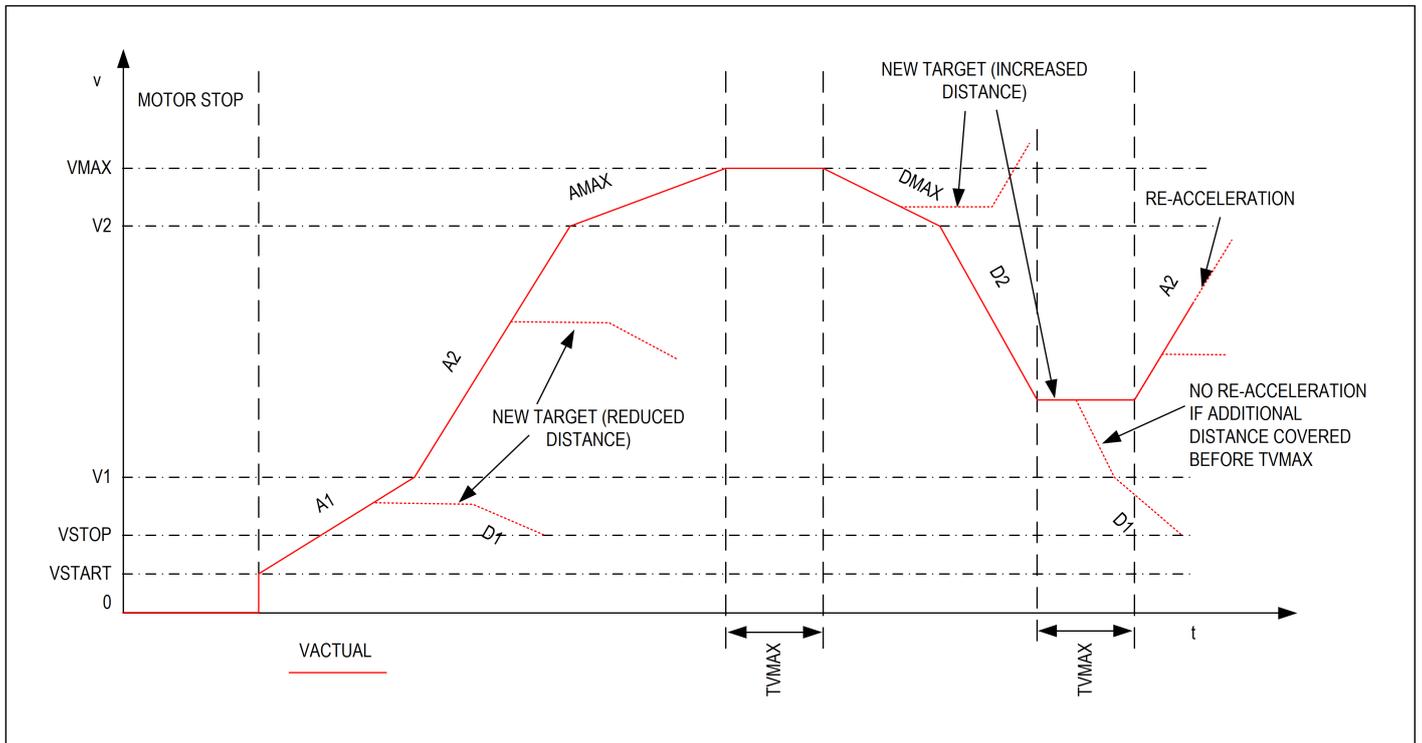


図 31. オンザフライの目標ポジション変更があった場合の 8 点ランプの例

## 速度モード

使いやすさを実現するため、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は使いません。速度モードでは AMAX および VMAX のみが関係します。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。vzero フラグはモータの停止を通知します。velocity\_reached フラグは、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、目標ポジション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ことを要求します。

加速設定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。

1. 速度モードに切り替えます。VMAX を 0 に設定し、AMAX を目的の減速値に設定します。これにより、線形ランプを使用してモータが停止されます。まだ VMAX に達していない場合、TVMAX フェーズがトリガされる場合があります（設定されている場合）。その代わりにソフトストップを用いると TVMAX 設定は無視されます。
2. ポジショニング・モードで停止を行うには、仮想スイッチでソフトストップを用いることを推奨します。
3. 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。1 または 2 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。
4. 停止スイッチを活性化します。これは、ハードウェア入力（例えば、停止スイッチ入力に用いたワイヤード OR）によって行うことができます。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) を用いてスイッチの活性化をシミュレートします。
5. 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・コマンドはモータをユーザ定義の速度で動作させ、また、目標ランプ・ジェネレータは、余すところなく有効な動作範囲を確保します。

ジョイスティック制御を行うには、以下のステップに従います。

1. 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。更に、仮想停止スイッチ (VIRTUAL\_STOP\_L および VIRTUAL\_STOP\_R) によって機械的な制限を強制的に適用することもできます。
2. 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、VMAX = 0 に設定することで停止動作を停止できます。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。
3. 加速設定を変更しない場合、XTARGET を書き換える必要はなく、VMAX を変更するだけで済みます。
4. 目標ポジションに達したためにランプ・コントローラが減速する場合、またはソフトストップを用いる場合は、DMAX、D1、D2、VSTOP のみを使用します。

### 速度スレッシュホールド

ランプ・ジェネレータは、図 32 に示すように、実際速度 VACTUAL と合わせていくつかの速度スレッシュホールドを提供します。様々な範囲を用いることで、モータを最適なステップ・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、スレッシュホールド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、THIGH と TCOOLTHRS の設定で決められます。TSTEP は、これらのスレッシュホールド値と比較されます。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に、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として設定された値の 1/16 または 1/32 だけ高くなります。StealthChop2 のスレッシュホールド 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スレッシュホールドは、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。

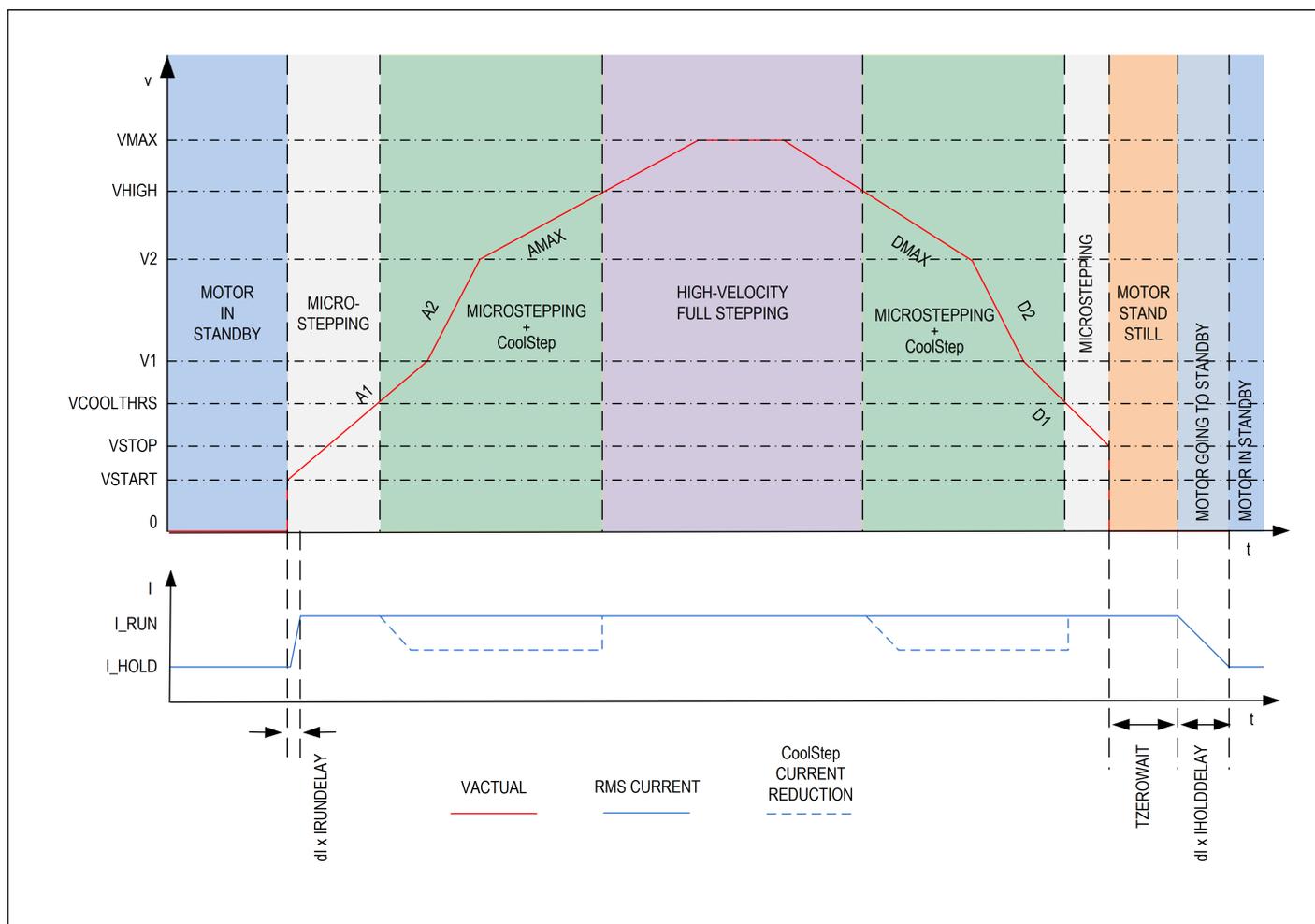


図 32. ランプ・ジェネレータの速度依存モータ制御

## 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、または電子機械リファレンス・スイッチによって検出できる物理的なハード・ストップを用いて見出すことができます。

線形駆動の場合、図 33 に示すように、機械的な動作範囲を超えることはできません。この条件は、左リファレンス・スイッチおよび右リファレンス・スイッチの停止スイッチ機能を有効化することで、異常な状態が生じた場合にも確実に満たすことができます。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。モータを停止する方法には次の 2 通りがあります。

1. スイッチを用いることで突然の停止が可能です。これは緊急時や StallGuard2/4 に基づくホーミングの場合に有用です。
2. ソフト停止機能 (ビット `en_softstop = 1`) を用いることで、減速設定 (DMAX、V2、D2、V1、D1) によりモータをゼロまで緩やかに減速することができます。

**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。

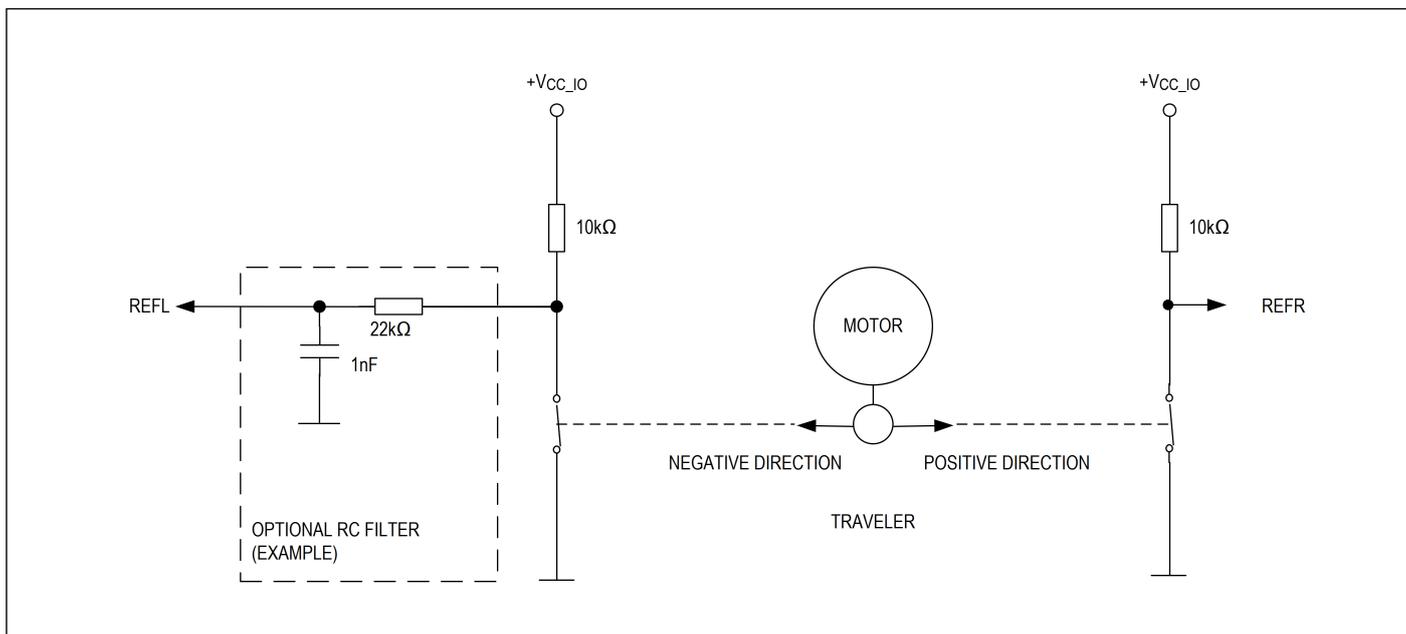


図 33. リファレンス・スイッチの使用 (例)

スイッチの極性をプログラミングするか、プルアップ抵抗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、TMC5271 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・プロシージャを実行するには、以下の手順に従います。

1. 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ます。
2. 目的のスイッチ・イベントでのポジション・ラッチを活性化し、アクティブ・スイッチでのモータの（ソフト）ストップを活性化します。StallGuard2/4 ベースのホーム・プロシージャでは、ハード・ストップ (`en_softstop = 0`) を用いる必要があります。
3. 動作ランプをスイッチの方向に起動します（左スイッチの場合はより負のポジションに移動させ、右スイッチの場合はより正のポジションに移動させます）。この動作はポジション・ランピング・コマンドを用いることでタイムアウトできます。
4. スwitchに達すると直ちに、ポジションがラッチされモータは停止します。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、モータが再度停止状態になるまで待ちます。
5. 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。StallGuard2/4 ベースのホーム・プロシージャ時、またはハード・ストップの使用時には、`XACTUAL` はホーム・ポジションで正確に停止するため、差はありません (0)。
6. 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・プロシージャは終了です。ポジション 0 に移動させると、モータは正確にスイッチ点に戻ります。StallGuard2/4 をホーム・プロシージャに用いた場合は、`RAMP_STAT` の読出しと再書き込みを行うことで、StallGuard2/4 の停止イベント `event_stop_sg` がクリアされ、モータはストップ条件から解放されます。

3 つ目のスイッチによるホーム・プロシージャを行うには、次の手順に従います。

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・スイッチをします。TMC5271 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チショットできます。また、この機能は、割込み出力も可能です。

1. ラッチ機能を活性化します (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り込み出力をトリガできます (割り込みが 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)。
2. 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。
3. スwitchがトリガされたら、XLATCH を読み出します。これはスイッチ・イベントが生じたポジションを示します。
4. スwitch・イベントの検出後、モータを停止させ、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、上述のホーム・イン・プロセスで説明しています)。

### 仮想リファレンス・スイッチ

TMC5271 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 (StallGuard2/4 ホーム・イン)、物理的な移動範囲を安全に制限しています。これを図 34 に示します。仮想停止スイッチが活性化するのは、実際のモータ・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。これを行うには、virtual\_stop\_enc を 1 に設定します。

仮想停止の値 (VIRTUAL\_STOP\_R、VIRTUAL\_STOP\_L) は、符号付き 32 ビットの移動範囲のオーバーフロー/アンダーフロー・レンジ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が減速できるようにします。

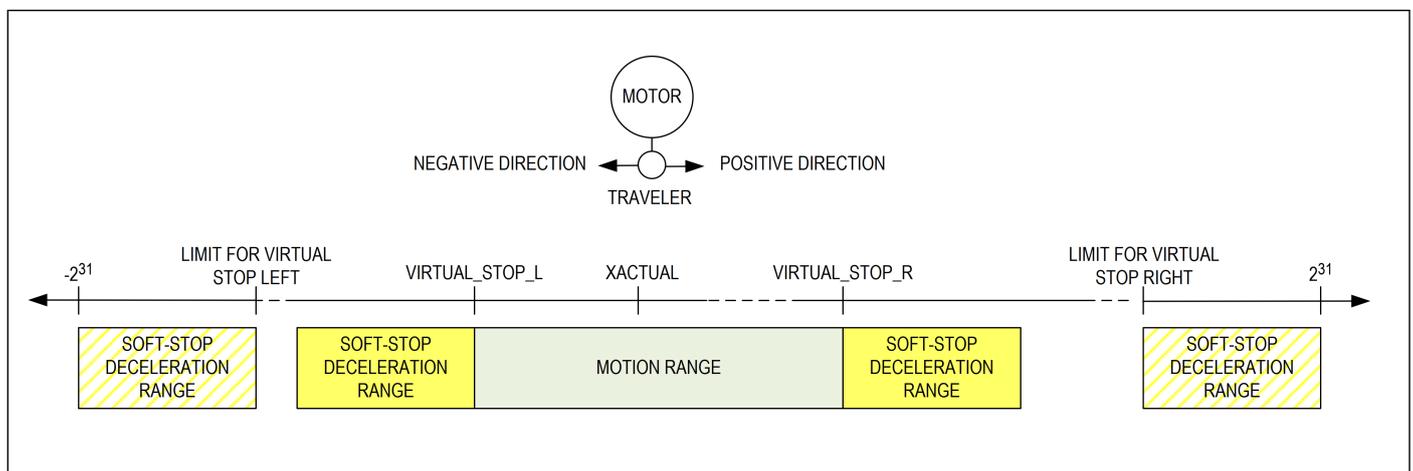


図 34. 仮想停止スイッチと制限の図解

### 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 マイクロ秒未満で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で 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と仮定すると、これは目標ポジションに達するまでに大きな遅延を生む原因となります。ランプを短時間で終了させるには、VSTOP を 100~1000 の最小範囲に設定します (100 の場合は約 10ms 未満、1000 の場合は約 1ms 未満になります)。

## ポジション比較機能

ポジション比較機能を用いると、外部イベントをモータの動作と同期してトリガできます。この機能は、 $X_{ACTUAL} = X_{COMPARE}$  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、 $X_{ACTUAL}$  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  $2^{24} - 1$  ステップ）。オプションを表 19 に示します。

表 19.  $X_{COMPARE\_REPEAT}$  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	初めて $X_{COMPARE}$ に達したときに、出力がアクティブになります。そのポジションを離れると1クロック・サイクルの間ディスエーブルになり、次のステップで直ちに再度活性化します。ディスエーブルするには $X_{COMPARE\_REPEAT}$ を 0 に設定します。
>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~ $2^{24} - 1$ マイクロステップ。

ポジション比較条件  $X_{ACTUAL} = X_{COMPARE}$  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、その後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。

## エンコーダを用いたクローズド・ループ・ポジション制御

この機能は、エンコーダ・ポジション・フィードバックを基本とする単純なクローズド・ループ・ポジション・レギュレーションです。これは P レギュレータです。この機能を有効化するには、比例係数  $P$  を 0 より大きい値に設定します。 $P = 0$  の場合、この機能は無効化されます。また、モーション・コントローラをイネーブルし、ステップ/方向モードにはしないことも必要です ( $GCONF[13] SD = 0$ )。

次式のとおり、誤差が  $X_{ACTUAL}$  と  $X_{ENC}$  の差として計算されます。

$$\text{error} = X_{ACTUAL} - X_{ENC}$$

$\text{tolerance}$  パラメータはこの誤差に関係します。 $\text{tolerance}$  の値未満の誤差は無視されます。P レギュレータ出力の制限は  $V_{MAX}$  パラメータによって設定されます。

$RAMP\_STAT$  レジスタのポジション到達フラグ  $\text{pos\_reached}$  がセットされると、 $\text{en\_tol\_on\_pos\_reached}$  ビットが P レギュレータの動作を制御します。

0 にセットすると P レギュレータは常にアクティブになります。関連するパラメータは  $POSITION\_P\_CTRL$  レジスタ (0x2D) にあります。これらを表 20 に示します。

表 20. クローズド・ループ・ポジション制御に関連するパラメータ

パラメータ	説明	範囲	注
$\text{en\_tol\_on\_pos\_reached}$	モータが目標ポジションに達した後に $\text{tolerance}$ を有効化します。 0: P コントローラは $\text{tolerance}$ 未満の絶対誤差を無視します。 1: P コントローラは、モーション・コントローラの $\text{pos\_reached}$ フラグ ( $RAMP\_STAT[9]$ ) が 1 に設定されている場合にのみ、 $\text{tolerance}$ 未満の絶対誤差を無視します。	0/1	デフォルト = 1

表 20. クローズド・ループ・ポジション制御に関連するパラメータ (続き)

パラメータ	説明	範囲	注
tolerance	P コントローラの許容誤差の設定 tolerance 未満の誤差は無視されます。 デフォルト = 0。	0 to 255	-
P	ポジション P レギュレータの比例パラメータ。 P = 0 の場合、P レギュレータ機能は無効化されます。 P > 0 の場合、P レギュレータ機能は有効化されます。 係数が 1 の場合、ポジションの偏差が +5 または -5 であると補正速度が +5 または -5 になることを意味します。 デフォルト = 0。	UINT 10.0	端数はなし

## StallGuard2 の負荷測定

異なるモータ制御方式を適合させるために、TMC5271 には 2 つの基本チョッパ・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。モータの負荷が増加するにつれ、図 35 に示すように、StallGuard2 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、StallGuard2 のスレッシュホールド (SGTHRS) を設定します。表 21 に関連するパラメータの一覧を示します。

**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。

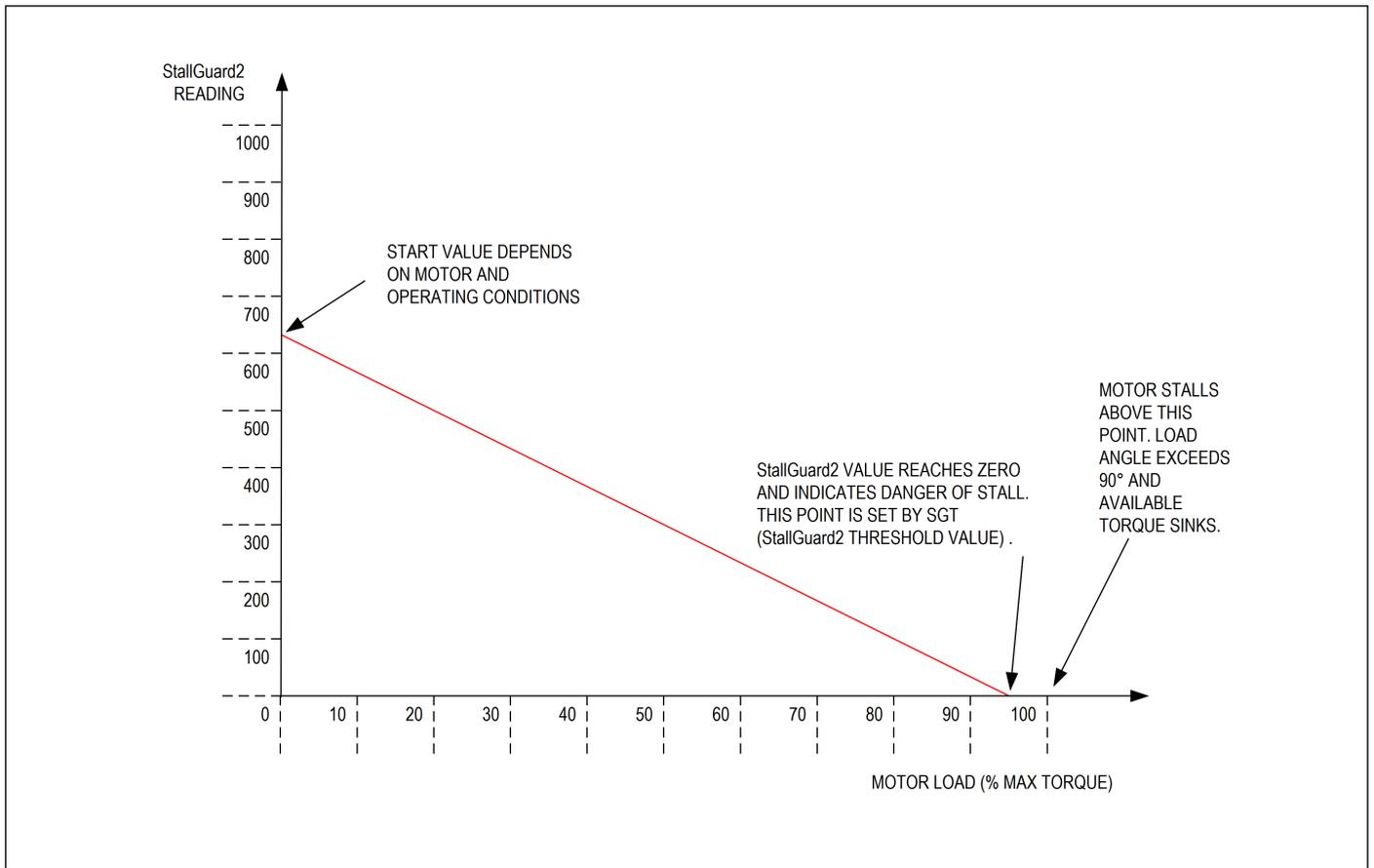


図 35. StallGuard2 の機能原理

表 21. StallGuard2 関連のパラメータ

パラメータ	説明	設定	注
SGT	SGT を用いると、正または負のオフセットを StallGuard2 の結果に追加できます。この値が高いと StallGuard2 の結果の許容可能ダイナミック・レンジが増加し、感度が低下します。 StallGuard2 による調整の標準的な目標は、ダイナミック・レンジを最大としながらもモータがストールする前に確実に StallGuard2 の結果が安全にゼロ値になるような方法で、StallGuard2 の結果をシフトさせることです。	0	中立値
		+1 to +63	低感度
		-1 to -64	高感度
sfilt	測定精度を向上させるには StallGuard2 のフィルタをイネーブルします。これをセットすると、測定頻度がモータの電気的周期 (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

表 21. StallGuard2 関連のパラメータ (続き)

パラメータ	説明	設定	注
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・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 to 1023	0 : 最大負荷 低値 : 高負荷 高値 : 低負荷

### StallGuard2 のスレッシュホールド SGT の調整

StallGuard2 の値 SG\_RESULT は、モータ固有の特性と、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 のスレッシュホールド SGT を特定のモータ・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。

#### StallGuard2 SGT を調整するための初期手順 :

- モータ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がストールした場合は、SGT を小さくします。モータ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 は符号付きです。そのため、SGT は負または正の値をもつことができます。
- 次に、sg\_stop をイネーブルし、ストール時に必ずモータ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 をクリア (1 を書き込んでクリア) して、モータを再始動します。
- 最適設定に達するのは、モータ・ストールの直前の増加した負荷に対して SG\_RESULT が 0~約 100 の範囲にある場合です。負荷がない場合、SG\_RESULT の増加は 100 以上になります。ほとんどの場合、SGT は特定の動作速度または速度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 設定は、StallGuard2 測定をモータ内の抵抗性負荷に対して補償するための 1 つの要素です。停止時あるいは極めて低い速度時は、抵抗性損失がモータのエネルギー・バランスの主要素になります。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に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を動作させます。この速度範囲では、SG\_RESULT はモータ負荷にはそれほど依存しません。モータは十分な逆起電力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。
- sfilt をスイッチ・オンします。次に、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・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、StallGuard2 が良好な結果をもたらすことのできる下限速度スレッシュホールドに一致するよう、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、モータの逆起電力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加します。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にはいくつかの方法があります。

### 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、図 36 に示すように、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スレッシュホールドを用います。ストール検出がディスエーブルされるのは、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。

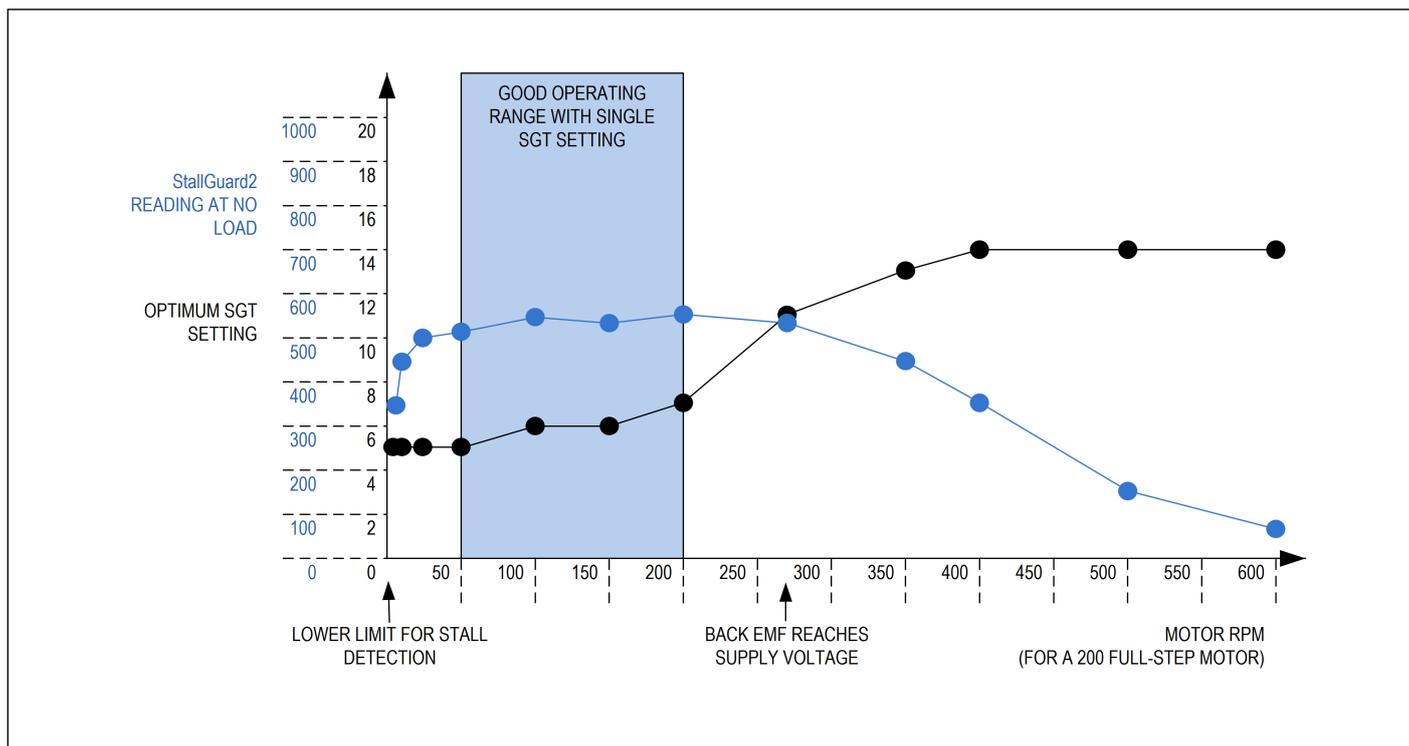


図 36. サンプル・モータを用いた SGT の最適設定値と StallGuard2 の読出し値

### 高トルク・リップルと共振を伴う小型モータ

保持トルクの高いモータでは、モータ電流の変化に伴う 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。こうしたモータでは、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・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では、温度上昇に伴いモータ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、モータ効率の低下と同様に、温度上昇に対して SG\_RESULT を直線的に低下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・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。

$$\text{StallGuard2 measurement error} = \pm \max(1, |\text{SGT}|)$$

## 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 を用いる場合では、より高精度の測定を行うことがフル・ステップごとに更新を行うこと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、StallGuard2 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・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・スレッシュホールドを決定する必要があります。そのため、モータ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・スレッシュホールドは、パラメータの浮遊を許容できるように、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、最小の読出し値が表示されます。ステップ・ロスの後、モータは振動し、SG\_RESULT の読出し値は増加します。

## StallGuard2 動作の制限

StallGuard2 は、モータ速度が極端な場合、信頼できる動作ができません。モータ速度が非常に低い (多くのモータでは 1rps 未満) 場合、生成される逆起電力は低く、測定が不安定で環境条件 (温度など) に依存するようになります。前述の自動調整手順は、モータ温度などの条件の変動を補正するのに役立ちます。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・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、モータの逆起電力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。どちらの機能も同様です。どちらも、低負荷時の高い値から高負荷時の低い値へと変化する負荷値を供給します。StallGuard2 はストール検出時にゼロの読出し値を示すよう調整されるのに対して、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 37 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。最大モータ負荷に近づくにつれ、値はモータ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の感度をチェックします。表 22 に、StallGuard4 に関連するパラメータの一覧を示します。

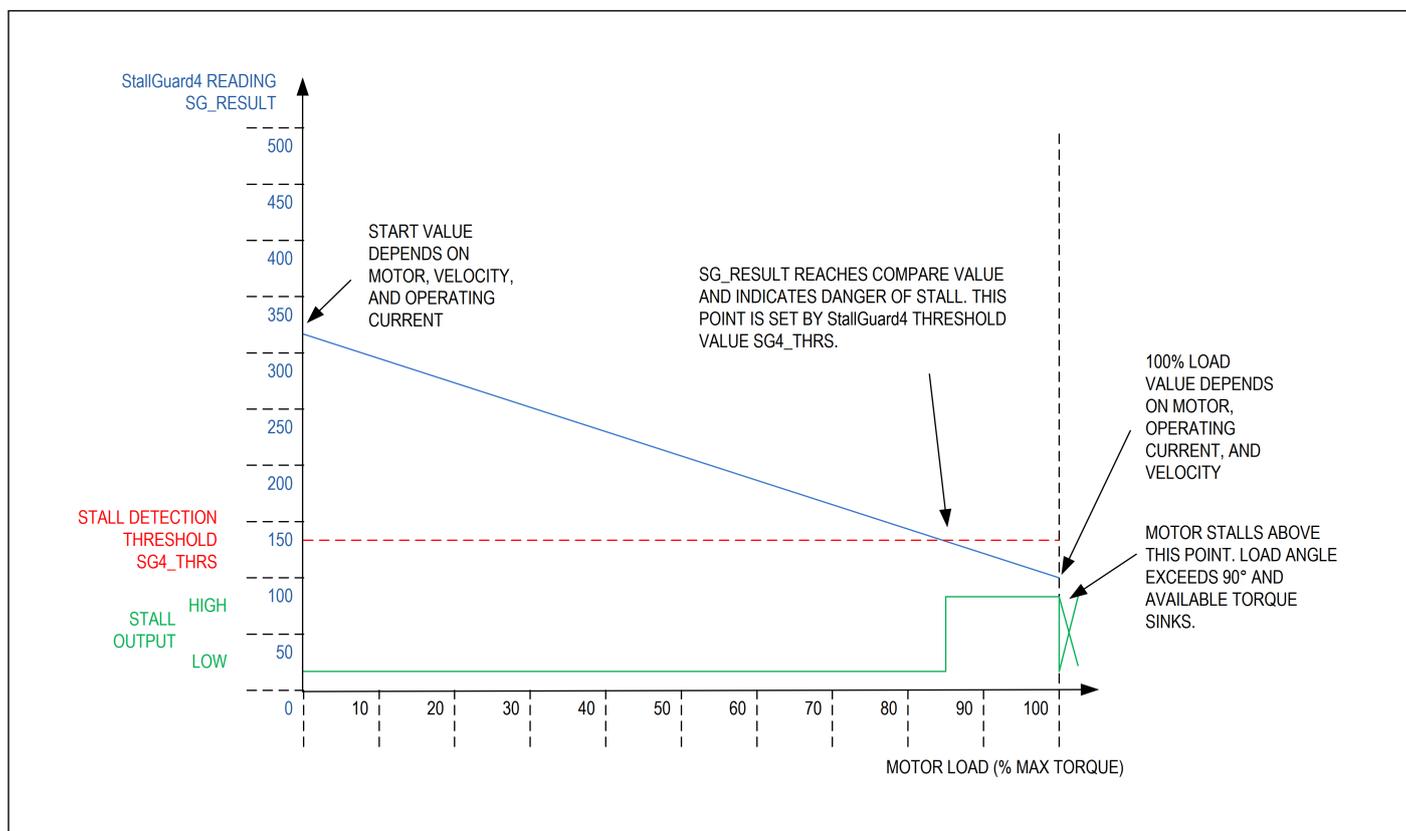


図 37. StallGuard4 の動作モード

表 22. StallGuard4 関連のパラメータ

パラメータ	説明	設定	注
SG4_THRS	この値はストール検出のための StallGuard4 スレッシュホールド・レベルを制御します。モータ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。  デフォルト = 0。	0 to 255	この値は SG4_RESULT と比較されます。 SG4_RESULT がこの値より低下した場合、ストール出力がアクティブになります。
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、実際のチョッパー・モードのようなイネーブル条件や VCOOLTHRS のような速度スレッシュホールドとは独立に生成されます。結果は SG4_IND_x の測定値から計算され、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 to 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) でのみ使用できます。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 to 255	低値：最大負荷。 高値：低/無負荷。

表 22. StallGuard4 関連のパラメータ (続き)

パラメータ	説明	設定	注
sg4_filt_en	0: フィルタを使用しない動作。SG4_RESULT はフル・ステップごとに更新されます。 1: フィルタを使用した動作。SG4_IND_x が使用可能です。SG4_RESULT は最後の 4 つの SG4_IND_x 測定値の平均値です。	0/1	0: フィルタ・オフ (デフォルト)。 1: フィルタを使用した動作。SG4_IND_x の値が使用可能です。
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 (デフォルト)。 1: StealthChop2 と SpreadCycle の切り替えを最適化。

## StallGuard4 の調整

StallGuard4 の値 SG4\_RESULT は、モータ固有の特性と、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 のスレッシュホールド SG4\_THRS を特定のモータ・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。

StallGuard4 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。

1. モータ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。
2. モータに印加する機械的負荷を徐々に増加します。モータ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)。
3. 次に、DIAGx 出力を介して StallGuard4 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を停止させます。ストール時にはモータが確実に停止するようにします。ストールが検出される前にモータがストールする場合は、SG4\_THRS を増加します。
4. ストールが確実に検出され、ストールの瞬間に DIAGx 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速度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。

診断出力 DIAGx は、SG4\_RESULT が  $2 \times \text{SG4\_THRS}$  を下回ると StallGuard4 によってパルス駆動されます。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。StallGuard4 が良好な結果をもたらすことのできる下限速度スレッシュホールドに一致するよう、TCOOLTHRS を設定してください。

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にはいくつかの方法があります。

## StallGuard4 の更新レート

StallGuard4 の測定値 SG4\_RESULT は、モータのフル・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。

1. 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。
2. 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、動作が不明確なモータでの CoolStep の使用に最適です。フィルタリング・モードでは、負荷の突然の増加 (ハード・モータ障害) に対する感度は減少します。

## モータ・ストールの検出

モータ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速度または速度範囲を用いてストール・スレッシュホールドを決定する必要があります。更に、モータ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・スレッシュホールドは、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、固定スレッシュホールド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、モータ速度が極端な場合、信頼できる動作ができません。モータ速度が非常に低い（多くのモータでは 1rps 未満）場合、生成される逆起電力は低く、測定が不安定で環境条件（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・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、モータの逆起電力が電源電圧を超えることで特徴づけられます。

## StallGuard2 および StallGuard4 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/StallGuard4 は、動作するには特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、確実に開始点がハード・ストップから十分離れているようにしてください。SGT（StallGuard2）または SG4\_THRS（StallGuard4）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活性化します（SW\_MODE レジスタの sg\_stop ビットをセット）。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を停止させます。ストップ条件は、DRV\_STATUS レジスタの StallGuard2/StallGuard4 フラグによっても示されます。モータ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/StallGuard4 をディスエーブルできます。あるいは、RAMP\_STAT レジスタの event\_stop\_sg フラグをクリアすることで、モータ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 レジスタの event\_stop\_sg フラグをクリアすることにより、TZEROWAIT の時間の経過後、モータを再始動できます。

## CoolStep の負荷適応型電流スケールリング

CoolStep は、モータの機械的な負荷に基づきステッピング・モータに対してスマートなエネルギー最適化を自動的に行い、これらのモータを環境フレンドリーなものにします。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表 23 に示す 2 つのパラメータです。CoolStep に関連するその他のパラメータを表 24 に示します。

表 23. CoolStep の重要パラメータ

パラメータ	説明	範囲	注
SEMIN	下限スレッシュホールドを設定する 4 ビットの符号なし整数。SG_RESULT がこのスレッシュホールドを下回る（負荷が大きいかを示す）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。	0	CoolStep をディスエーブル（デフォルト）
		1 to 15	スレッシュホールドは SEMIN × 32
SEMAX	上限スレッシュホールドを制御する 4 ビットの符号なし整数。このスレッシュホールド以上の値の 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。上限スレッシュホールドは、(SEMIN + SEMAX + 1) × 32 です。  デフォルト = 0。	0 to 15	スレッシュホールドは (SEMIN + SEMAX + 1) × 32

図 38 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。
- 青線はモータに加えられた機械的負荷を示します。
- 赤線はモータ・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 が電流を増加します。負荷が減少すると、図 38 に示すように、SG\_RESULT が (SEMIN + SEMAX + 1) × 32 を上回り、電流は減少します。

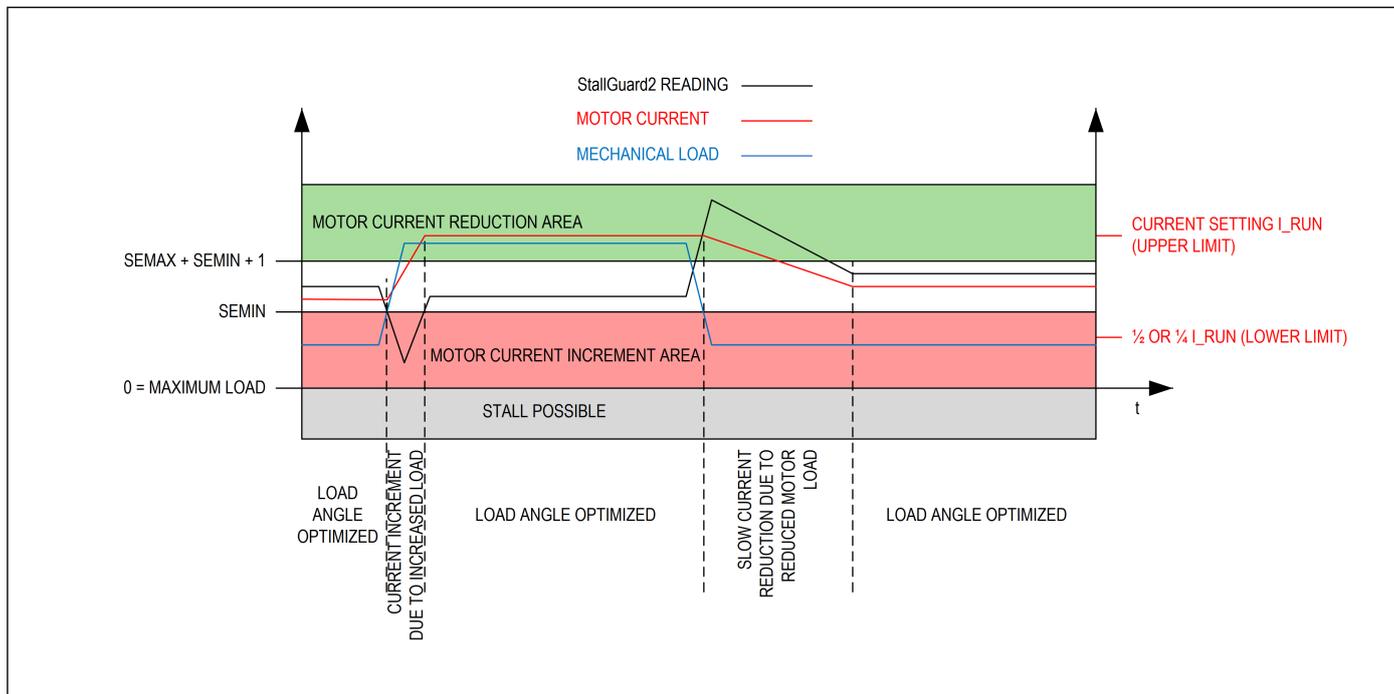


図 38. モータ電流を負荷に適合させる CoolStep

表 24. その他の CoolStep パラメータとステータス情報

パラメータ	説明	範囲	注
SEUP	電流のインクリメント・ステップを設定します。SEMIN で設定された下限スレッショルド未満の新しい StallGuard2 または StallGuard4 の値が測定されると、モータ電流は必ずこの設定値だけインクリメントされます。  デフォルト = 0。	0 to 3	CS 値 CS_ACTUAL のステップ幅は 1、2、4、8 です。
SEDN	モータ電流の各電流デクリメントに必要な、上限スレッショルドを上回る StallGuard2/4 読出し値の数を設定します。  デフォルト = 0。	0 to 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リングして、CoolStep 動作の下限モータ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電流を十分上回る電流で動作するようにしてください。  デフォルト = 0。	0	0 : IRUN の 1/2 (StealthChop2 と共に使用する場合は IRUN ≥ 16 であることが必要)。
		1	1 : IRUN の 1/4 (StealthChop2 と共に使用する場合は IRUN ≥ 28 であることが必要)。

表 24. その他の CoolStep パラメータとステータス情報 (続き)

パラメータ	説明	範囲	注
TCOOLTHRS	CoolStep に切り替えるための下限速度スレッシュホールド。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適応できます。  デフォルト = 0。	1 to 2 <sup>20</sup> - 1	スレッシュホールド値を TSTEP と比較することで、下限 CoolStep 速度を指定します。
THIGH	CoolStep 用の上限速度スレッシュホールド値。この速度を超えると CoolStep はディスエーブルされます。StallGuard2/4 が安定した結果を与える速度範囲に合わせてください。  デフォルト = 0。	1 to 2 <sup>20</sup> - 1	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。
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 to 31	1/32、2/32、~32/32、読出し専用。

### CoolStep の調整

CoolStep を SpreadCycle と併用して調整する前に、まず StallGuard2 のスレッシュホールド・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。下限スレッシュホールドに達することは、高速応答を必要とする可能性のある、より厳しいイベントであるためです。応答が遅すぎる場合、モータはストールする可能性があります。反対に、上限スレッシュホールド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。

#### 注意:

CoolStep がモータ電流を増やすと、モータ・ストールが誤検出されることがあります。最良の結果を得るには、StallGuard2 ベースのホーミング時に CoolStep をディスエーブルします。

Stallguard2 を CoolStep と組み合わせたい場合は、必要に応じ CoolStep の下限閾値 SEMIN を増やしてみてください。

### 応答時間

モータ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ブ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。

ヒント: 最も一般的で最も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、それに応じた速度スレッシュホールドを設定することです。加速値および減速値は通常、高ダイナミック値を示すため、これらのフェーズではフル・モータ電流が必要ですが、時間が短いために全体的な消費電力への影響はごくわずかです。

### 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 時のモータ負荷は測定できないため、下限速度スレッシュホールドが設けられています。このスレッシュホールドは、アプリケーション固有のデフォルト値に設定する必要があります。このスレッシュホールド未満では、IRUN または I HOLD を用いた通常の電流設定が有効です。上限スレッシュホールドは 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。

どちらのスレッシュホールドも StallGuard2 と StallGuard4 の調整プロセスの結果として設定できます。

## 診断出力およびステータス出力

2つの診断出力 DIAG0 および DIAG1 は、その設定に応じてホストに一定範囲のステータス信号および割り込み信号を供給し、特定のドライバやモーション・コントローラの状態をモニタしそれに応答します。GCONF のビットである `diag0_int_pushpull` および `diag1_poscomp_pushpull` は、オープン・コレクタ（アクティブ・ロー）出力信号を選択するか（デフォルト）、アクティブ・ローのプッシュプル出力信号を選択することができます。オープン・コレクタ出力を用いる場合、4.7kΩ~33kΩ の範囲の外部プルアップ抵抗が必要です。

図 39 に DIAG0 の回路を示します。DIAG0 は、リセット状態になるとローに駆動されます。GCONF のビット `diag0_sel_nError_Ramp` を用いると、内蔵のモーション・コントローラまたはドライバのステータス・フラグおよびエラー・フラグに関連する信号およびフラグの最初の選択を行うことができます。様々な設定ビットにより、特定のフラグおよび割り込み信号を DIAG0 に対して有効にできます。GCONF のビット `diag0_stall_step` を用いると、モーション・コントローラの内部ステップ信号を DIAG0 のデバッグ出力としてマッピングすることができます。このステップ信号は、モーション・コントローラが生成するマイクロステップごとに、アクティブ・エッジ（立上がりおよび立下がり）を示します。

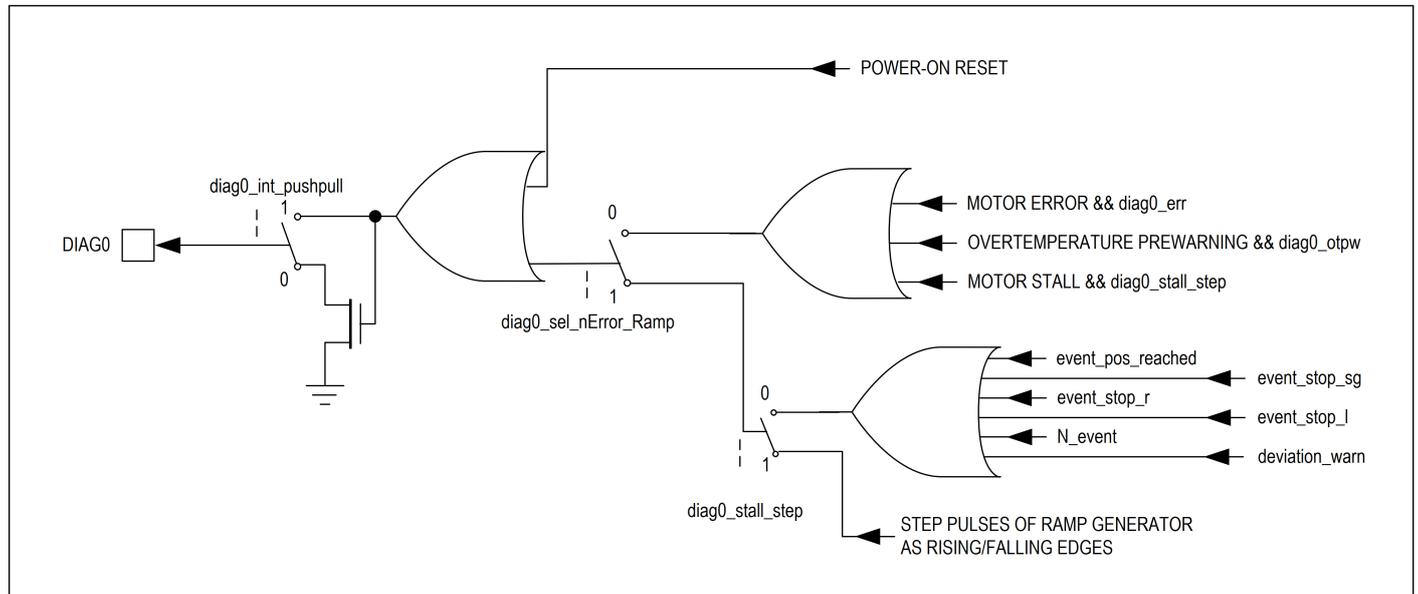


図 39. DIAG0 の出力回路

図 40 に DIAG1 の回路構造を示します。DIAG1 に対して GCONF のビット `diag1_sel_nStallIndex_xcomp` を用いると、内蔵のモーション・コントローラのポジション比較機能またはモータ・ドライバのストール・フラグおよびインデックス・フラグに関連する信号およびフラグの最初の選択を行うことができます。設定ビットにより、特定のフラグおよび割り込み信号を DIAG1 に対して有効にできます。GCONF のビット `diag1_stall_dir` を用いると、モーション・コントローラの内部方向信号を DIAG1 のデバッグ出力としてマッピングすることができます。

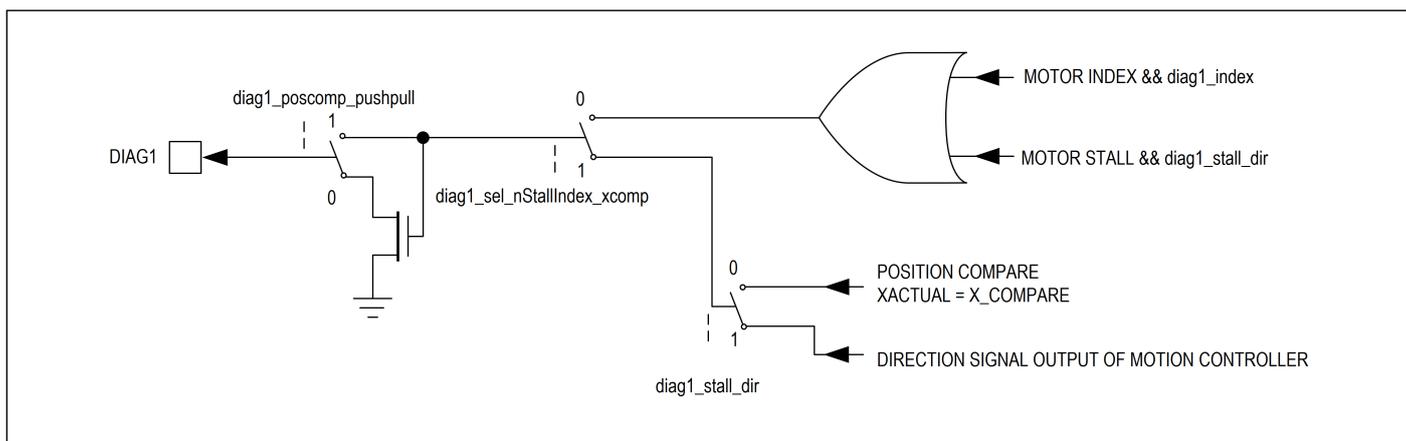


図 40. DIAG1 の出力回路

## DcStep

DcStep はステッピング・モータ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、ステッピング・モータはランプ・ジェネレータから指示された目標速度で動作できます。モータが過負荷になった場合には、モータが負荷を駆動できるまで速度を落とします。したがって、ステッピング・モータ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。

このモードは、内蔵モーション・コントローラを用いている場合にのみ使用できます。外部のステップおよび方向モードでは、DcStep を使用することはできません。

## 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、図 41 に示すように、利用可能な最大モータ・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、モータとアプリケーションの慣性重量を用いて、モータ・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速度まで増加できます。

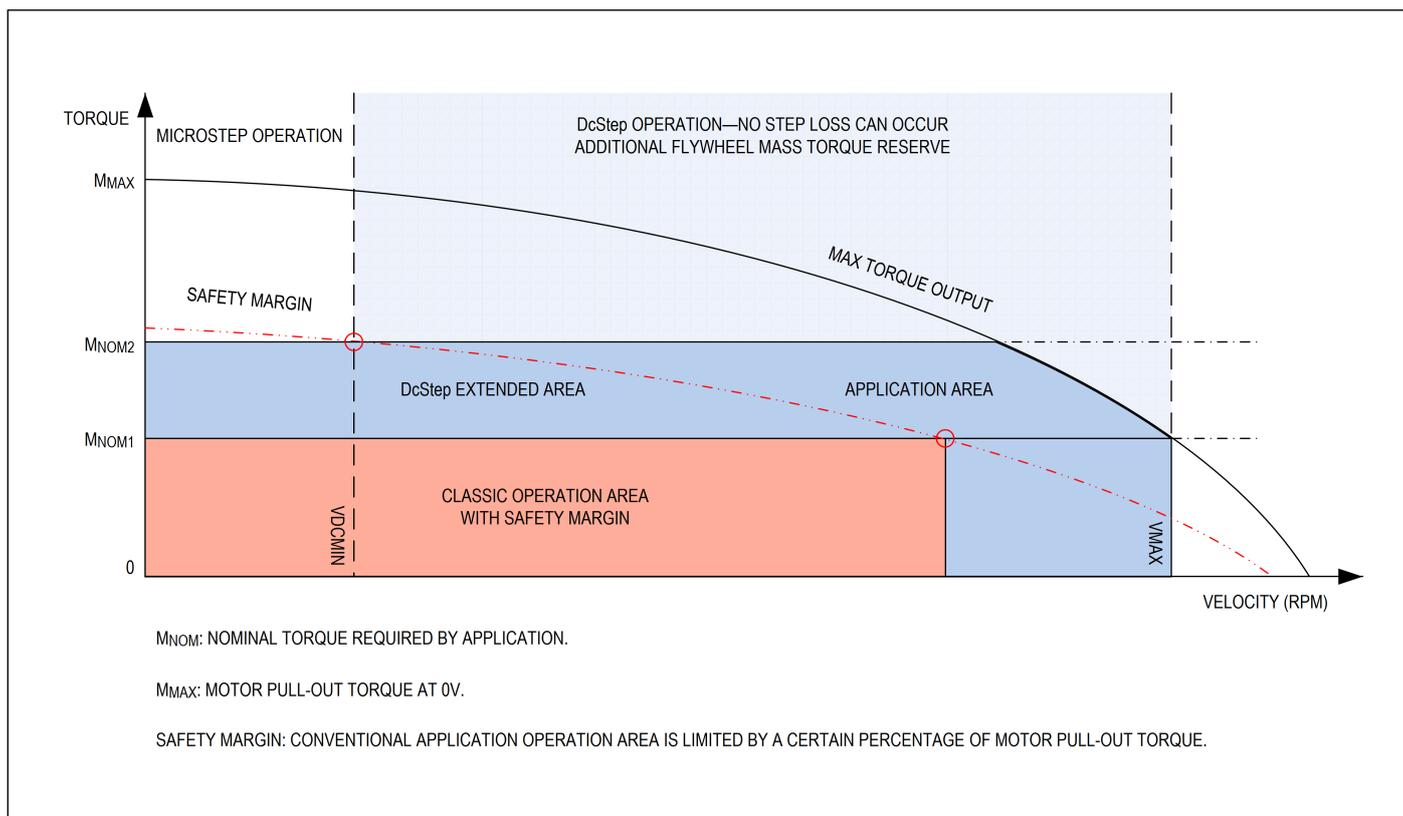


図 41. DcStep により拡大されるアプリケーション動作領域

## 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  $T_{OFF}$  チョッパーを用いて、ランプ・ジェネレータの目標速度  $V_{MAX}$ 、あるいはモータが過負荷の場合には減速した速度でモータを動作させます。図 42 に過負荷状態の様子を示します。これには、最小動作速度  $V_{DCMIN}$  を設定する必要があります。 $V_{DCMIN}$  は、DcStep がモータ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する必要があります。速度が  $V_{DCMIN}$  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  $V_{DCMIN}$  未満になった場合には、StallGuard2/4 によって処理されます。

**注：** 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ことが必要です。コサイン波の極性は、0~511 では正、512~1023 では負であることが必要です。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、サイン波ルックアップ・テーブルのセクションを参照してください。

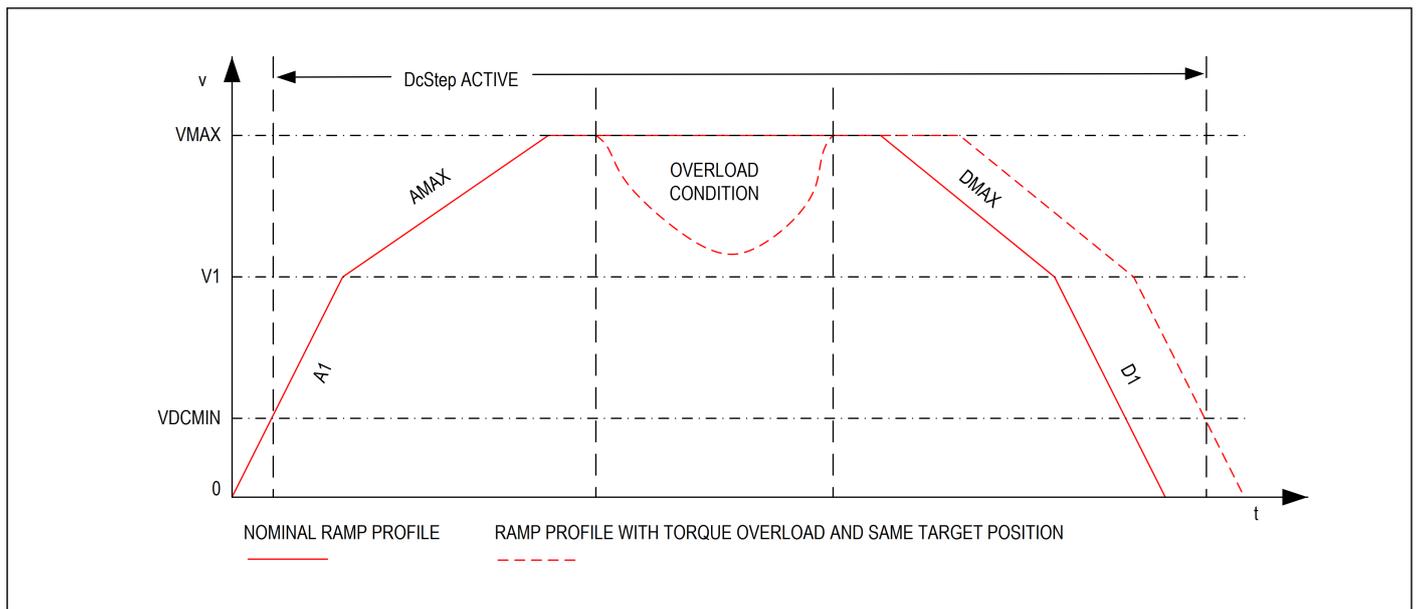


図 42. 過負荷状態時の DcStep の速度プロファイル

### 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。モータがブロックされた場合や、モータ動作がそれ以上確実に検知できないようなモータ依存最小速度を下回るほど減速した場合は、モータがストールする可能性、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モータ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、モータがストールすると `VACTUAL` はゼロに設定されます。モータは、`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` はアクティブなストップ条件を示します。モータは、`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（0~255 に対応）のみが生じるので、この値の範囲は、0~255 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。表 25 に DcStep のストール検出に関連するパラメータを示します。

フライホイール負荷がモータ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が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 モードにおけるストール検出のパラメータ

パラメータ	説明	範囲	注
<code>vhighfs</code> and <code>vhighchm</code>	CHOPCONF におけるこれらのチョッパ設定フラグは、DcStep 動作用に設定されている必要があります。 <code>VDCMIN</code> を超過すると直ちに、チョッパは <code>T<sub>OFF</sub></code> が一定のチョッパに切り替わり、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。
<code>TOFF</code>	DcStep では、多くの場合、CHOPCONF のオフ時間値が大きい方がメリットが得られます。2 より大きな設定値を推奨します。	2 to 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。

表 25. DcStep モードにおけるストール検出のパラメータ (続き)

パラメータ	説明	範囲	注
VDCMIN	これは、内蔵ランプ・ジェネレータを使用する場合の DcStep 動作の下限スレッシュホールドです。このスレッシュホールド未満では、モータ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。  StealthChop2 を活性化することでも DcStep をディスエーブルできます。	0 to 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・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。この値が高いほど、トルクと速度が高くなります。値を低くすると、VDCMIN で設定された下限速度まで動作を低下できます。  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、最大モータ温度、最大電源電圧、最低モータ温度など) で再チェックします。	0 to 1023	この設定に対する下限値は、クロック・サイクル単位の t <sub>BLANK</sub> (TBL で定義) + n で、n は 1~100 の範囲です (標準的なモータの場合)。
DC_SG	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。  ストールは、モータのハード・ストップを発行することで、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。この方法で、ストールが生じるとモータは停止します。	0 to 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速度の測定

DcStep には、モータ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、モータ速度を低減する機能があります。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。

$$VACTUAL_{DCSTEP} = \frac{XACTUAL(TIME2) - XACTUAL(TIME1)}{TIME2 - TIME1} \times \frac{2^{24}}{f_{CLK}}$$

DcStep 動作における実際のモータ速度の例は次のとおりです。

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

TMC5271 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 (リセット状態) 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・モータにとっては良い開始点となっています。テーブルをモータ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。

- マイクロステップ-低コスト・モータで著しく改善
- モーター-滑らかで静かに動作
- 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・ノイズ-実際のモータ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

TMC5271 では、サイン波をモータごとに個別に設定できます。各モータに対して、位相 A と位相 B の間の位相シフトを ±45° の範囲で調整できます。更に、各モータのモータ位相ごとに個別の電流スケーラがあります (内蔵電流検出機能のセクションを参照)。

## 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  $0^\circ \sim 90^\circ$  に割り当てます。これが  $360^\circ$  まで対称的に拡張されます。テーブルの読出し時には、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 ビットずつを使用して、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  $W_x$  を加算するかどうか、すなわち  $W_x + 1$  とするかどうかを制御します。 $W_x$  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  $W_x$  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  $X1 \sim X3$  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、その基本勾配は  $W1$  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は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  $-248 \sim +248$  の範囲となるよう調整する必要があります。図 43 に例を示します。

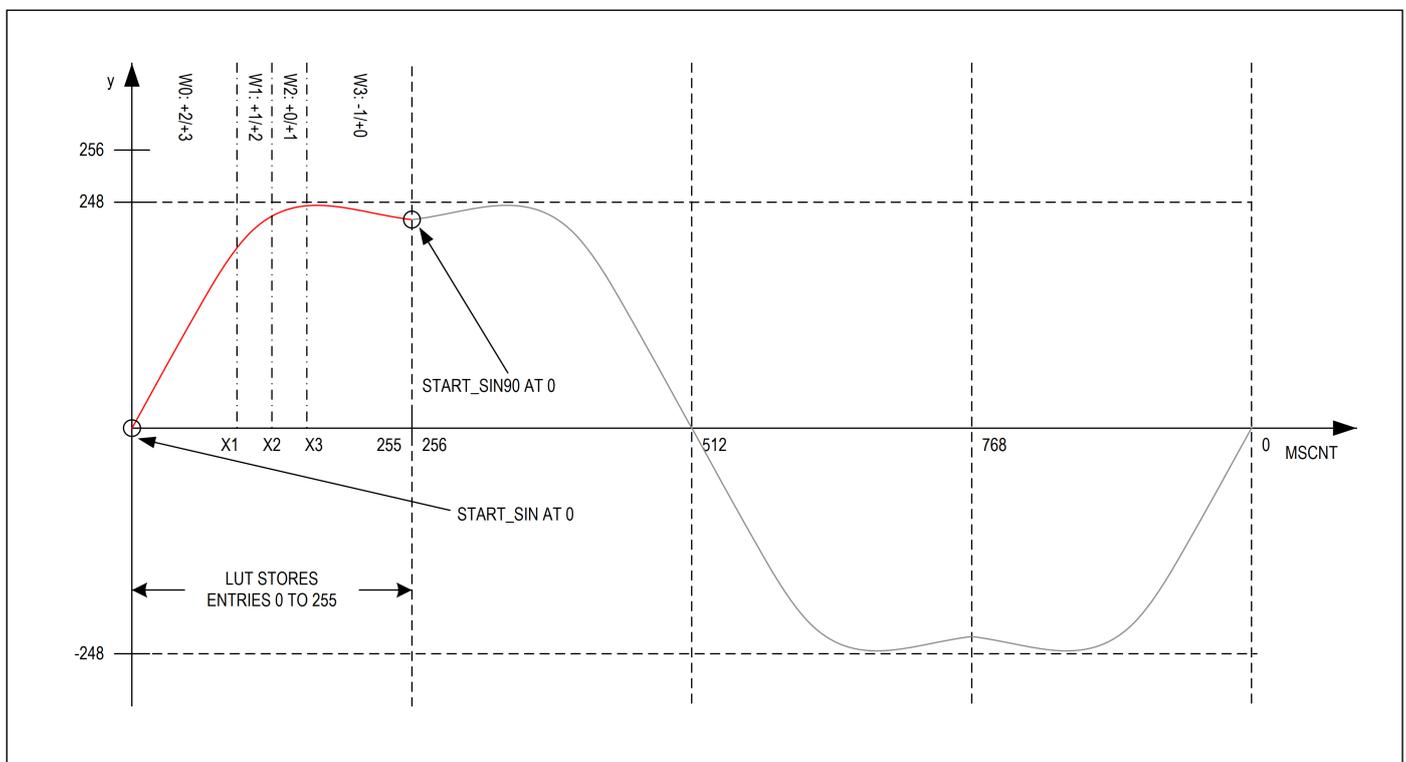


図 43. LUT プログラムの例

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・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のマッチング：

図 44 に示すように、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・コイル用の2つめの波形の開始点は `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。モータの許容誤差に対しては、数ステップから最大で数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。

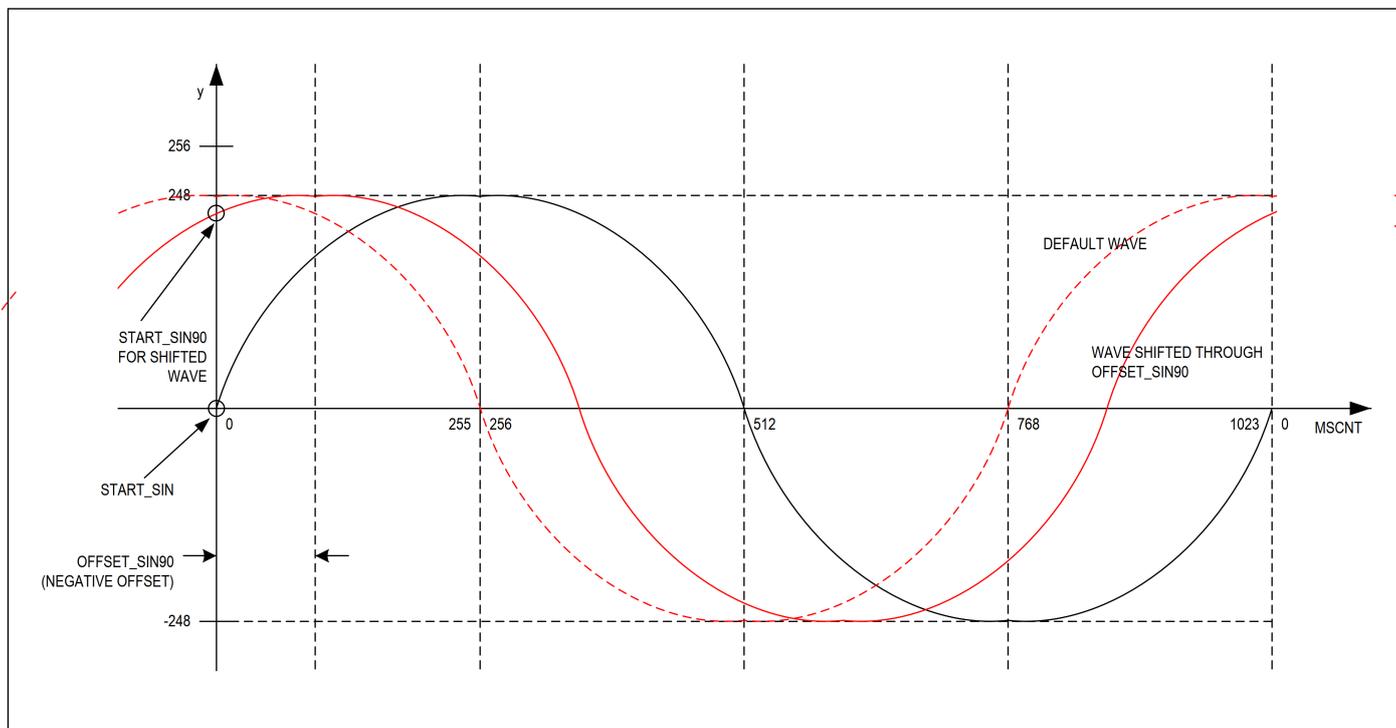


図 44. `OFFSET_SIN90` を使用したコサイン波のシフト

デフォルトのテーブルは、マイクロステップ・テーブルの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] = %10101010101010101010101010101000 = 0xAAAAB554
MSLUT[1] = %01001010100101010101010010101010 = 0x4A9554AA
MSLUT[2] = %00100100010010010010010010010010 = 0x24492929
MSLUT[3] = %00010000000100000100001000100010 = 0x10104222
MSLUT[4] = %11111011111111111111111111111111 = 0xFBFFFFFF
MSLUT[5] = %10110101101110110111011011110110 = 0xB5BB777D
MSLUT[6] = %01001001001010010101010101010110 = 0x49295556
```

```
MSLUT[7] = %00000000010000000100001000100010 = 0x00404222
```

```
MSLUTSEL = 0xFFFF8056:
```

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

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

```
MSLUTSTART = 0x00F70000:
```

```
START_SIN = 0, START_SIN90 = 247
```

モータ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。オフセットが-17まで、または+17までの場合は、START\_SIN90 = 246 を用います。START\_SIN は常に 0 です。

### TriCoder-逆起電力センサーレス停止ステップロス検出

TriCoder 機能は、モータの逆起電力 (BEMF) を利用する、センサーレスの停止ステップロス検出機能です。

BEMF デコーダは、モータがディスエーブル時 (モータ・コイルに有効な電流が加えられていないとき) のモータ動作を検出できます。この機能は、消費電力を節約する要求があるために保持電流を印加できないながらも、通常一定量のコギング・トルクまたは摩擦がモータを定位置に保持するようなデバイスに対して、特に有用です。モータが外部の力によって回転するような場合に、その動きが探知されます。その結果を利用して、モータが動いた場合の新しいホーミング・シーケンスをトリガしたり、ステップ数の追跡を行い続けその後それを補正したりできます。

それ以外に、モータを入力デバイスとして用いて、手動のティーチインを行う、またはユーザとのインタラクションを行う、という使い方があります。システムでは、次の精度が可能です。

- この原理では、モータによって生成される一定の最小 BEMF 電圧が動作を検知する必要があります。モータによっては、この必要な BEMF レベルは、数 RPM の範囲で容易に到達でき、したがって動作の発生は容易に検知できます。
- 動作開始時または終了時には、1 ステップまたは数ステップが失われたり、正しくカウントされなかったりする可能性があります。これと実際のモータおよびアプリケーションの機構との関連性をチェックする必要があります。
- システムは、1 フル・ステップの倍数でカウントを行います。適合したエンコーダ・スケーリング係数を設定することで、変更されたモータ・ポジションを直接追跡できます。

### TriCoder BEMF デコーダの動作原理

TriCoder BEMF デコーダは、モータがディスエーブルの間、または、ローサイド・ドライバを介した受動ブレーキングでモータが停止させられた場合、モータの動作を追跡できます。モータの動作を検知するために、ドライバ IC は、モータ・コイルの電圧をチェックし、それをプログラマブルなヒステリシス・スレッシュホールドと比較します。主要動作モードを図 45 に示します。

ディテクタ回路は、プログラマブルなヒステリシス・スレッシュホールドと帯域幅制限の組み合わせを使用し、モータ・ラインに結合した電圧スパイクなどによる誤ったトリガを回避します。

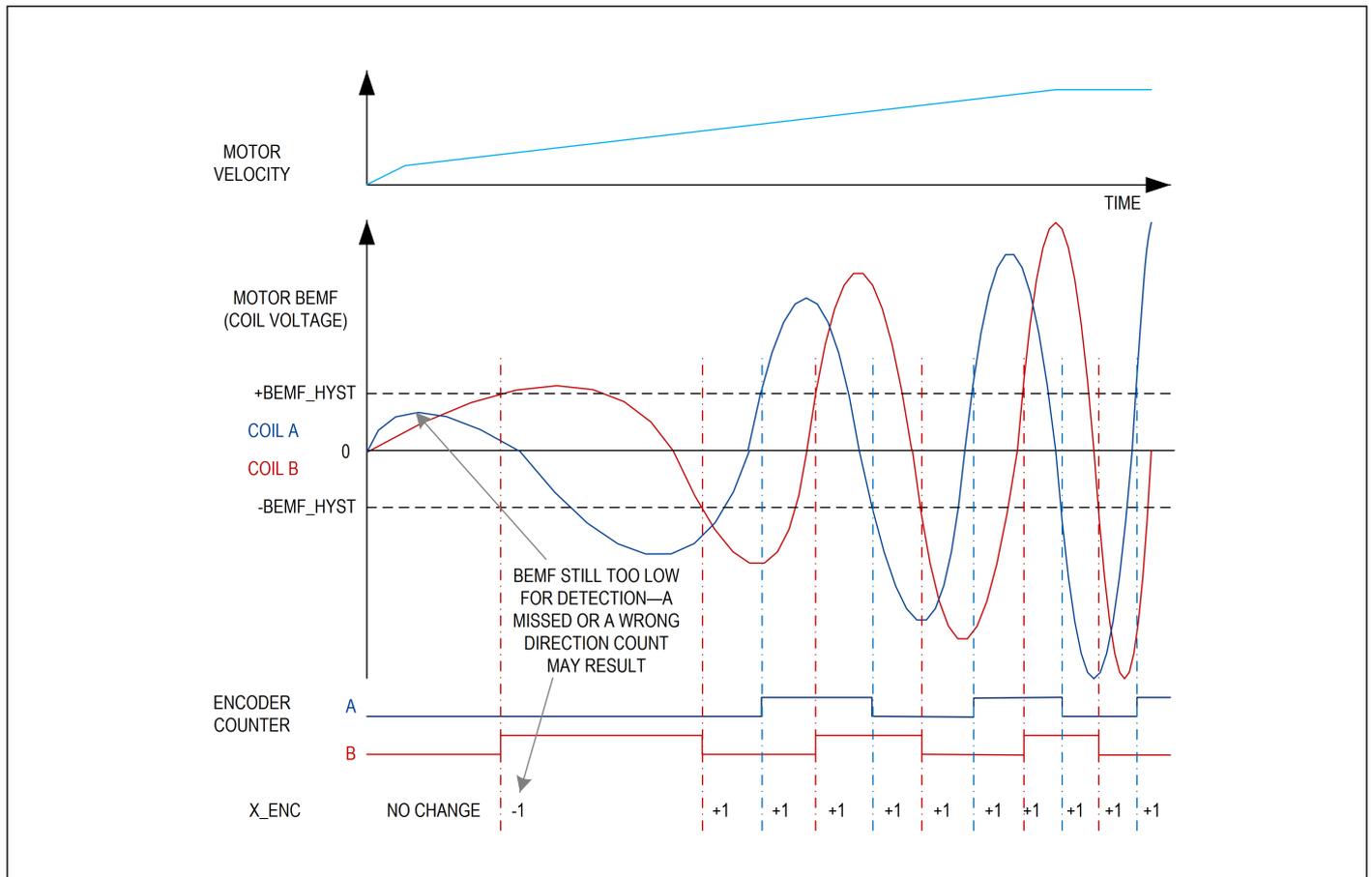


図 45. モータ動作の検知

### TriCoder 動作に対するモータおよび速度の条件

特定のヒステリシス設定を用いて検知できる最低速度を決定するには、モータの逆起電力の電圧をチェックします。

$$C_{BEMF} \left[ \frac{V}{\frac{\text{rad}}{s}} \right] = \frac{\text{HoldingTorque}[\text{Nm}]}{2 \times \text{ICOILNOM}[\text{A}]}$$

$$U_{BEMFopen}[V] = \frac{\text{HoldingTorque}[\text{Nm}]}{2 \times \text{ICOILNOM}[\text{A}]} \times \frac{2\pi \times \text{Velocity}[\text{RPM}]}{60}$$

受動ブレーキング・モードでは、モータはコイルを短絡することで電氣的に制動されます。

したがって、ある特定のモータ速度に対するエンコーダの感度が低下します。最初に生成される逆起電力電圧はモータ・コイルの抵抗から生じるモータ電流に変換される必要があるためです。

この電流が再び、ドライバ内のローサイド MOSFET スイッチの  $R_{DS(ON)}$  の電圧に転送されます。このモードで最高の結果を得るには、電力段の電流設定として最低の値を選択し、コイル抵抗に対する  $R_{DS(ON)}$  の比率を増加させます。

$$U_{BEMFbrake}[V] = \frac{\text{HoldingTorque}[\text{Nm}]}{2 \times \text{ICOILNOM}[\text{A}]} \times \frac{2\pi \times \text{Velocity}[\text{RPM}]}{60} \times \frac{R_{DS(ON)}}{R_{DS(ON)} + R_{COIL}}$$

## イネーブルまでの時間

モータを動作させるには、モータがフリーホイーリング・モードまたは受動ブレーキング・モードでなくてはなりません。モータの残留コイル電流が確実になくなるよう、それ以前の動作に応じて一定の遅延時間が必要となることがあります。モータ停止の検知の後、モータ電流は 0 までランプ・ダウンします (IHOLDDELAY の設定により制御)。最低動作速度に必要な停止検知時間 STANDSTILL\_TIME を設定してください。高速の IHOLDDELAY 設定 (0 は即時パワーダウン) でインダクタンスが高いモータを用いる場合、残留モータ電流を電源に戻すことでこれを減衰させるために、数 100 マイクロ秒から数ミリ秒の追加時間が必要となることがあります。残留電流はモータの BEMF のように見えることがあり、そうでない場合は、ステップ検出をもたらすこともあります。

## 関連設定

### 1. 設定 : BEMF デコーダのイネーブル

モータがフリーホイーリング・モードになっている場合は常に、BEMF デコーダを自動的にイネーブルするために以下の設定が必要です。

- ENCMODE\_NBEMF\_ABN\_SEL : 外部エンコーダ入力をディスエーブルするには 0 に設定します。モータは停止状態 (stst=1) であることが必要です。
- PWM\_CONF\_FREEWHEEL : 1 (フリーホイール) または 2 (受動ブレーキング) に設定します。

StealthChop2 :

- IHOLD : 停止時にフリーホイーリング・オプションをイネーブルするには 0 に設定します。CS\_ACTUAL は 0 に到達する必要があるあります。

SpreadCycle :

- モータをディスエーブルしフリーホイーリングを可能にするには、TOFF = 0 に設定します。

低消費電力モード :

- 低消費電力モードの場合、この間に BEMF エンコーダをイネーブルするには、ENCMODE\_NBEMF\_ABN\_SEL = 0 および ENCMODE\_QSC\_ENC\_EN = 1 に設定すれば十分です。

### 2. 動作設定

- ENCMODE\_BEMF\_BLANK\_TIME : エンコーダ機能が活性化するまでにこの時間が経過している必要があります。
- ENCMODE\_BEMF\_HYST (0~7) : ヒステリシスが 10、25、50、75、100、150、200、250mV (代表値)。
- ENCMODE\_BEMF\_BLANK\_TIME :  $2^{14}$  クロック・サイクルの倍数でカウントします。この時間により、直前のモータ動作の結果生じたモータ・コイル電流が、モータの BEMF をモニタする前に 0 まで減衰できます。モータが停止状態になった直後に誤った動作検出が発生する場合は、この時間を増加させます。
- DRV\_CONF\_Mx\_STANDSTILL\_TIME : 停止検知までの時間を短縮して、IHOLD\_IRUN\_IHOLDDELAY 設定で制御されるモータ電流の 0 へのランプ・ダウンを速めます。デフォルトは  $2^{20}$  クロック・サイクルです。
- ENCMODE\_BEMF\_FILTER\_SEL : デコードされる BEMF 信号のためのデジタル・フィルタ時間。
  - 0 :  $f_{CLK}/24600$
  - 1 :  $f_{CLK}/12300$
  - 2 :  $f_{CLK}/6150$
  - 3 :  $f_{CLK}/2870$
  - この設定は A チャンネルおよび B チャンネルの入力帯域幅を制限し、それにより、エンコーダのカウント・レートは結果として生じる周波数 (内部  $f_{CLK}$  が約 12.5MHz の場合、2kHz、4kHz、8kHz、16kHz) の 4 倍になります。予測される最大カウント・レートを十分に上回るよう設定してください。
- ENC\_CONST : X\_ENC カウンタがモータのマイクロステップ分解能に一致するよう設定します。例えば、256 マイクロステップ設定の場合は 256 です。この設定により、モータのフル・ステップあたり X\_ENC が 256 だけ増加/減少します。
- X\_ENC : エンコーダが活性化する前に 0 または X\_ACTUAL に設定できます。エンコーダの動作中にはステップ差または更新されたモータ・ポジションを読み出します。

### ABN インクリメンタル・エンコーダ・インターフェース

TMC5271 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。

インクリメンタル・エンコーダ・インターフェースを使用するには、これがアクティブにイネーブルされる必要があります。リセットおよびパワーアップ後は、センサレス・フル・ステップ・エンコーダがデフォルトで選択されます。インクリメンタル・エンコーダ・インターフェースをイネーブルするには、ENCMODE レジスタの nBEMF\_ABN\_SEL ビットをセットします。

## N 信号

N 信号を使用すると、ポジション・カウンタをクリアするか XACTUAL 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ポジションをクリアあるいはラッチするトリガを発生するには、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・ポジション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 (`clr_once` フラグ)。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 `diag0_nint_step` ビットを用いて、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・ポジション `X_ENC` 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。

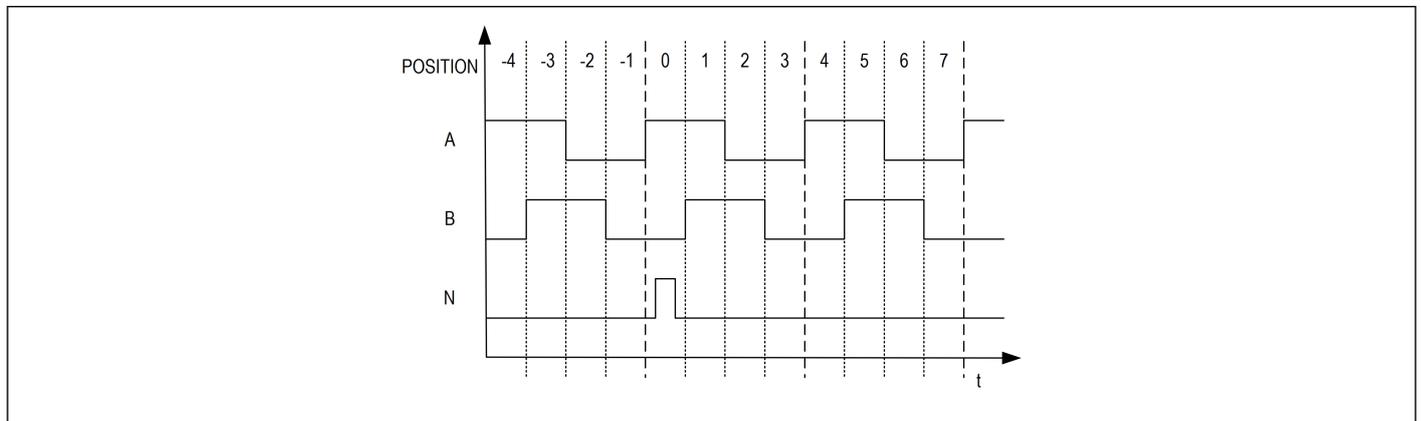


図 46. インクリメンタル・エンコーダの ABN 信号の概要

## X\_ENC エンコーダ・カウンタ

X\_ENC エンコーダ・カウンタは、電流エンコーダを読み出し可能なポジションに保持します。信号 A、B、N の処理に関する様々なモードは、図 46 に示すように、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。

## ENC\_STATUS レジスタ

ENC\_STATUS レジスタは、N チャンネル信号でのエンコーダ・クリアのイベントのステータスを保持します。ENC\_LATCH レジスタは、N 信号イベント時の実際のエンコーダ・ポジションを常時保存します。

## ENC\_CONST エンコーダ定数

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ともに、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ピング・モータでは、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、モータとエンコーダの方向に一致するようカウント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、 $(2^{16} - (\text{FACTOR} + 1)) \cdot (2^{16} - \text{FRACTION})$  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： $(2^{16} - (25 + 1)) \cdot (10000 - 6000) = (2^{16} - 26) \cdot (4000) = 0xFFE6.0x0FA0$
- 負のエンコーダ定数は、次式を用いて計算できます。 $(2^{16} - (\text{FACTOR} + 1)) \cdot (10000 - \text{DECIMALS})$

### モータの分解能に合うエンコーダ設定

表 26 において、モータ・パラメータに対するエンコーダの設定例は次のとおりです。

1. USC = 256 マイクロステップ。
2. FSC = 200 フル・ステップ・モータ。
3. Factor = FSC × USC / エンコーダ分解能。

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

エンコーダ分解能	必要なエンコーダ係数	注
200	256	モータ・ステップを 1/256 マイクロステップに一致させるには、このエンコーダ係数を BEMF エンコーダに使用します。
360	142.2222 = $9320675.5555/2^{16}$ = $1422222.2222/10000$	実現可能な正確な一致はなし。
500	102.4 = $6710886.4/2^{16}$ = $1024000/10000$	10 進数設定で正確に一致。
1000	51.2	10 進数設定で正確に一致。
1024	50	-
4000	12.8	10 進数設定で正確に一致。
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 にプログラムする必要があります。そのため次のように設定します。

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

### チップの緊急停止およびパワーダウン・モード

TMC5271 には、ピンおよびレジスタで制御可能な緊急停止モードおよびパワーダウン・モードが備わっています。

表 27 にこれらのモードの概要を示します。IC の最大消費電力は、IC 自体にのみ関係するもので、実際のモータ電流は考慮しません。

表 27. チップのモード比較

モード	開始条件	終了条件	レジスタの内容	IC の最大消費電力	SPI/UART	モータ電力
SPI/UART を介した緊急停止	GCONF レジスタ (0x0) : drv_enn = 1	GCONF レジスタ (0x0) : drv_enn = 0	変化なし	~6mA	アクティブ	ドライバは直ちにディスエーブル

表 27. チップのモード比較 (続き)

モード	開始条件	終了条件	レジスタの内容	ICの最大消費電力	SPI/UART	モータ電力
停止オプション付きの設定可能な外部緊急停止	<p>このモードを有効化するには、次のように設定します。 GCONF レジスタ (0x0) : stop_enable = 1</p> <p>このモードをトリガし保持するには、ピン A1 を 1 にプルアップし保持します。</p> <p>このモードを設定するには、次のように設定します。 IHOLD_IRUN レジスタ (0x12) : IHOLD および IHOLD_DELAY。 PWMCONF レジスタ (0x3C) : FREEWHEEL。</p>	ピン A1 を 0 にプルダウン	変化なし	~6mA	アクティブ	ドライバはアクティブですが、モータは設定された保持電流、または StealthChop2 のフリーホイール・オプションで停止します
ハード外部緊急停止	<p>このモードを有効化するには、次のように設定します。 DRV_CONF レジスタ (0x5) : en_em_disable = 1。</p> <p>このモードを設定するには、次のように設定します。 DRV_CONF レジスタ (0x5) : sel_em_stop_src。</p> <p>このモードをトリガするには、次のように設定します。REFL ピンおよび REFR ピン = 1。</p>	ピン (REFL および REFR = 0)	変化なし	~6mA	アクティブ	ドライバは直ちにディスエーブル
低消費電力自己消費モード	<p>このモードを有効化するには、GCONF レジスタ (0x0) : qsc_sts_ena = 1。</p> <p>このモードをトリガするには、一度モータ・ドライバを停止状態にしてから IC を自己消費モードにします。 モード・アクティブ・インジケータ-I0IN レジスタ (0x4) : ビット 15 qsc = 1 かどうかをチェックします。 自己消費モードの設定を更に行うには、ENCMODE レジスタ (0x2F) : qsc_enc_en = 1 とすると自己消費モードの間のエンコーダをイネーブルします。</p>	GCONF レジスタ (0x0) : qsc_sts_ena = 0	変化なし	~750μA	アクティブ	ドライバは直ちにディスエーブル
リセット	SLEEPN のロー・パルス > 30μs	SLEEPN = 1	デフォルトにリセット	~1μA (リセット時)	非アクティブ	SLEEPN の入力フィルタ遅延 (~50μs) 後にドライバはディスエーブル
スリープ・モード	SLEEPN が永久にローにプルダウン	SLEEPN = 1	デフォルトにリセット	~1μA	非アクティブ	SLEEPN の入力フィルタ遅延 (~50μs) 後にドライバはディスエーブル

**SPI または UART を使用した緊急停止**

TMC5271 には、ドライバ段のすべてのパワー-MOSFET をスイッチ・オフする、レジスタ制御オプションがあります。これにより、モータをフリーホイール状態にすることができます。これを行うには、GCONF レジスタのビット drv\_enn を 1 に設定します。

## 停止オプションを用いた緊急停止

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 A がモータ停止および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 のビット stop\_enable をセットすると、このオプションを活性化できます。

ピン A がハイにプルアップされ、それがハイにとどまる限り、モータは停止し、IHOLD および IHOLD\_DELAY (IHOLD\_IRUN レジスタ内)、および StealthChop2 を使用している場合は StealthChop2 の停止オプション (PWMCONF レジスタの FREEWHEEL パラメータ) で設定される、パワーダウン状態になります。ピン A が再度ローにプルダウンされると、モータ・ドライバはデフォルトの動作モードに戻ります。

## ハード緊急停止

このモードは、外部制御ピンによってトリガされた場合にモータ・ドライバを直ちにスイッチ・オフする、特別なハード・ストップ・モードです。高速時にハード・ストップを行うと高い BEMF 電圧が生じデバイスを破壊する可能性があるため、注意が必要です。

このモードは、DRV\_CONF レジスタ (0x5) の en\_em\_disable ビットおよび sel\_em\_stop\_src ビットでイネーブルおよび設定ができます。その後、REFL ピンと REFR ピンを用いてこのモードをトリガできます。ハード・ストップをトリガするには、REFL ピンと REFR ピンのどちらも、ハイ (1) に短絡する必要があります。

## 低消費電力自己消費モード

低消費電力自己消費モードは TMC5271 の特別なモードで、通信インターフェースをオン状態にしたまま、かつ、レジスタの内容を保持したまま、消費電力を最小限に抑えます。チップの消費電力は約 750µA に低減します。

このモードを開始するには、GCONF (0x0) レジスタの qsc\_sts\_ena ビットを 1 に設定します。他の停止モードやパワーダウン・モードと同様、このモードになるとモータ・ドライバ段は完全にディスエーブルされます。再度イネーブルするには、qsc\_sts\_ena が再び 0 に設定された後の 2 回目のレジスタ・アクセスで GCONF レジスタの drv\_enn ビットをセットする必要があります。

デフォルトでは、ポジション・トラッキングができるよう、このモードの間、インクリメンタル・エンコーダ・ユニットはイネーブルされたままです。ENCMODE (0x2F) レジスタの qsc\_enc\_en ビットを用いると、このエンコーダを使用しない場合にスイッチ・オフでき、余分なエネルギーを節約できます。

低消費電力自己消費モードの間、次のレジスタへの書込みはできません。

- 0x01 – GSTAT
- 0x10 – X\_COMPARE
- 0x18 – XACTUAL
- 0x2B – RAMP\_STAT

ENCMODE[15] qsc\_en\_en が 1 に設定されている場合、低消費電力自己消費モードの間、次のレジスタには書込みのみができます。

- 0x2E – X\_ENC
- 0x31 – ENC\_STATUS

## 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 35µs (代表値) を超える短パルスを 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)。この持続時間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 を再設定する間、GCONF レジスタのビット 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、SLEEPN を V<sub>s</sub> (これは高電圧ピンです) に接続します。

## 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用いる場合、CLK 入力ピンを IC の近くで GND に直接接続します。内部クロックは通常、12.5MHz の周波数で動作します。詳細については、[電気的特性](#)のセクションを参照してください。

### 外部クロックの使用

専用の外部クロックを用いることで、TMC5271 内のすべての速度演算および加速度演算に用いられるタイム・ベースを正確に計算できます。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。クロック信号に必要な最小および最大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。クロックのデューティサイクル条件は、特に 20MHz に近いクロック周波数で満たされる必要があります。

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ブ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。IOIN レジスタの ext\_clk ビットを読み出すと、現在どのクロック・ソース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、内部クロックに滑らかにかつ自動的に切り替わり、ドライバの損傷を防止します。TMC5271 が自己消費パワーダウン・モードになっている場合、クロック・ソースは内部クロックに切り替わります。

### 保護およびドライバ診断

TMC5271 のドライバでは、GND への短絡保護や低電圧検出など、診断機能、ステータス情報、保護機能をフルセットで利用できます。開放負荷状態の検出では、モータ・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。

### 過熱保護およびサーマルシャットダウン

TMC5271 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 (DRV\_STATUS のフォルト・フラグ) 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) 未満に低下するまでスリープ状態となります。その後、ドライバは自動的に再イネーブルされます。

更に、内蔵 ADC がチップの平均温度を検知します (ただし、ドライバ段はこれよりはるかに高い温度になっている可能性があります)。これは、ADC\_TEMPERATURE パラメータとして IOIN[8:1]レジスタから読み出せます。この測定機能は、温度測定が不要な場合には、IOIN[9]レジスタの ADC\_EN 制御ビットでディスエーブルして消費電力を削減できます。ADC\_TEMPERATURE の値は、次の式を用いて摂氏温度に変換できます。

$$T(^{\circ}\text{C}) = (2.03 \times \text{ADC\_TEMPERATURE}) - 259$$

熱は主としてモータ・ドライバ段で発生します。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緊急時のみの措置です。シャットダウン・レベルまで温度が上昇することは、設計によって回避する必要があります。

### モータ温度の測定

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。所定のモータ電流に対して、デューティサイクルはモータ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\_SUM の値 (PWM\_SCALE レジスタ内) を用いて実際のモータ温度を推定し、モータ温度の時間変化をモニタできます。この測定は、モータ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、モータ温度は短時間では変化しません。

モータ温度を推定するには次の式が役立ちます。既知の条件でパワーアップした後、最初にリファレンス温度測定 (Temp\_Reference) を行う必要があります。係数 100/0.39 は、銅の温度係数で定義されているものです。

$$\text{Temperature} = \text{Temp\_Reference} + \left( \frac{\text{PWM\_SCALE\_SUM} - \text{PWM\_SCALE\_SUM\_reference}}{\text{PWM\_SCALE\_SUM\_reference}} \times \frac{100}{0.39} \right)$$

## 短絡保護

TMC5271 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、GND および  $V_S$  への短絡状態に対して保護されます。大半の短絡状態は、システムのグランドに接続された導電素子に接触した場合など、モータ・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をスイッチ・オフすることで) ESD 放電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、短絡は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、表 28 に示すように、様々な過電流保護スレシヨルドでローサイドの短絡保護がトリガします。

表 28. フルスケール電流設定に基づく過電流保護スレシヨルド

FULL-SCALE CURRENT SETTING FSR (BITS)	OVERCURRENT PROTECTION THRESHOLD (A)
11	3.00
10	2.25
01	1.50
00	0.75

## 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。TMC5271 は、目的のモータ・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。低電圧状態、高モータ速度設定、あるいは短絡および過熱状態も、開放負荷フラグをトリガする原因となり、モータ・トルクに影響を受ける可能性があることをユーザに通知する可能性があります。モータ停止時には、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、低速あるいは公称モータ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 フラグおよび olb フラグは情報を提供するのみで、何らかのドライバ動作を引き起こすものではありません。

## 低電圧ロックアウト保護

TMC5271 は、 $+V_S$  および  $+V_{CC10}$  に対する UVLO 保護機能を備えています。 $+V_S$  の UVLO 状態は 1.9V (最大値) 未満でトリガされます。 $+V_{CC10}$  の UVLO 状態は 1.3V (最大値) 未満でトリガされます。

$+V_S$  の UVLO 状態は、GSTAT レジスタから vm\_uvlo フラグとして読み出せます。これは書き込んでクリアするフラグです。クリアするには 1 にアクティブに設定する必要があります。UVLO 状態はまた、ピン設定に応じて DIAG0 ピンでも示されます。

$+V_{CC10}$  の UVLO 時には、IC との通信は不可能です。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。TMC5271 のモータ位相出力ピンは、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。これはモータのホット・プラグに対する保護ではありません。

## クイック設定ガイド

次の図 47~図 56 に示すガイドは、初期設定をセットアップし、最小セットの測定を行い、ドライバの調整に対する決定を行うための、実用的なツールとなることを意図したものです。これらは、先進的な機能やオプションのすべてを対象としたものではなく、モータを滑らかに回転させるための基本的な機能セットに焦点を置いています。ひとたびモータが動作すれば、ユーザはその他の機能やより高度な機能を詳細に調べる判断ができます。1 つのモータ・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

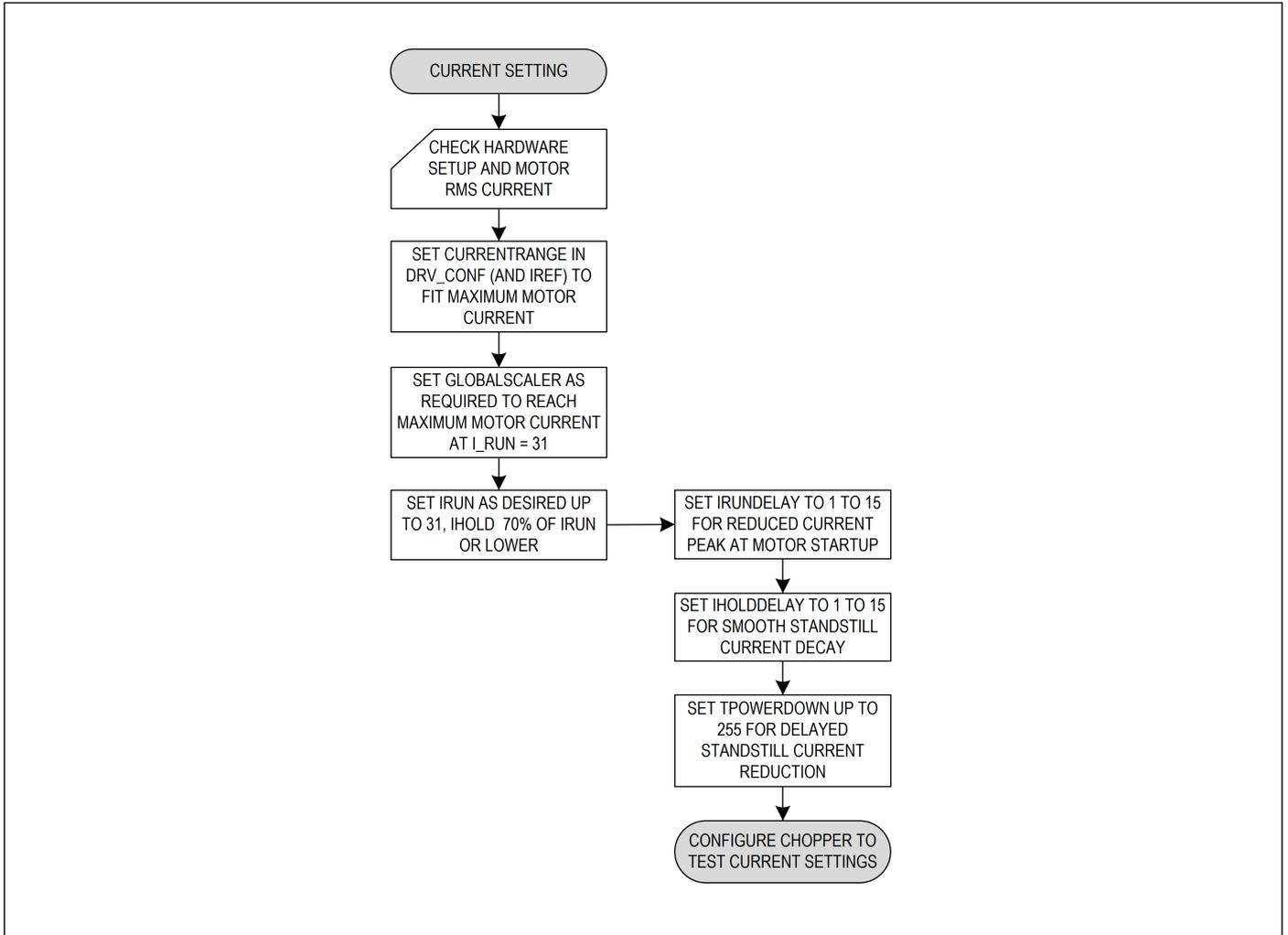


図 47. クイック設定ガイド-電流設定

StealthChop2 設定

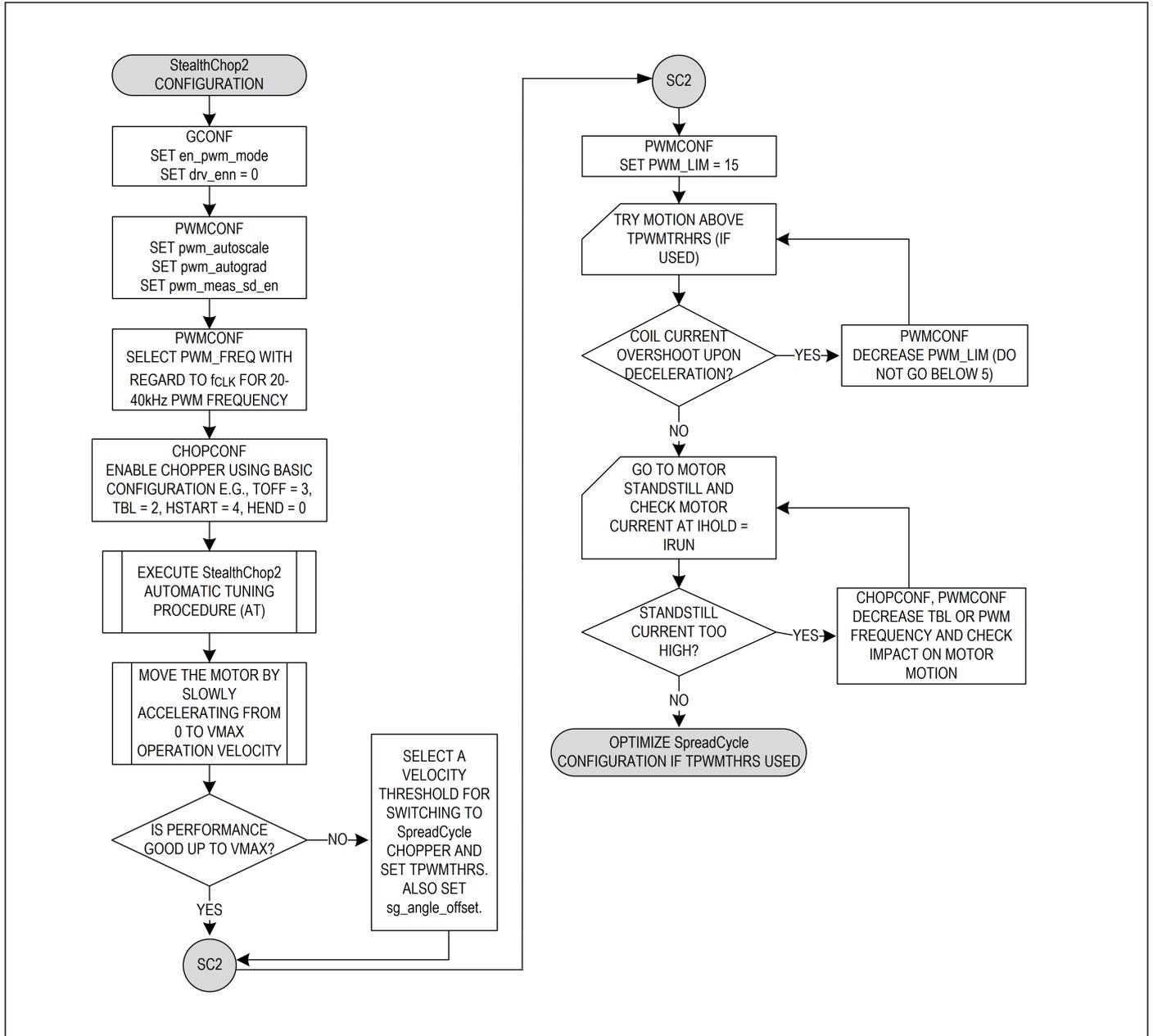


図 48. クイック設定ガイド–StealthChop2 設定

## SpreadCycle 設定

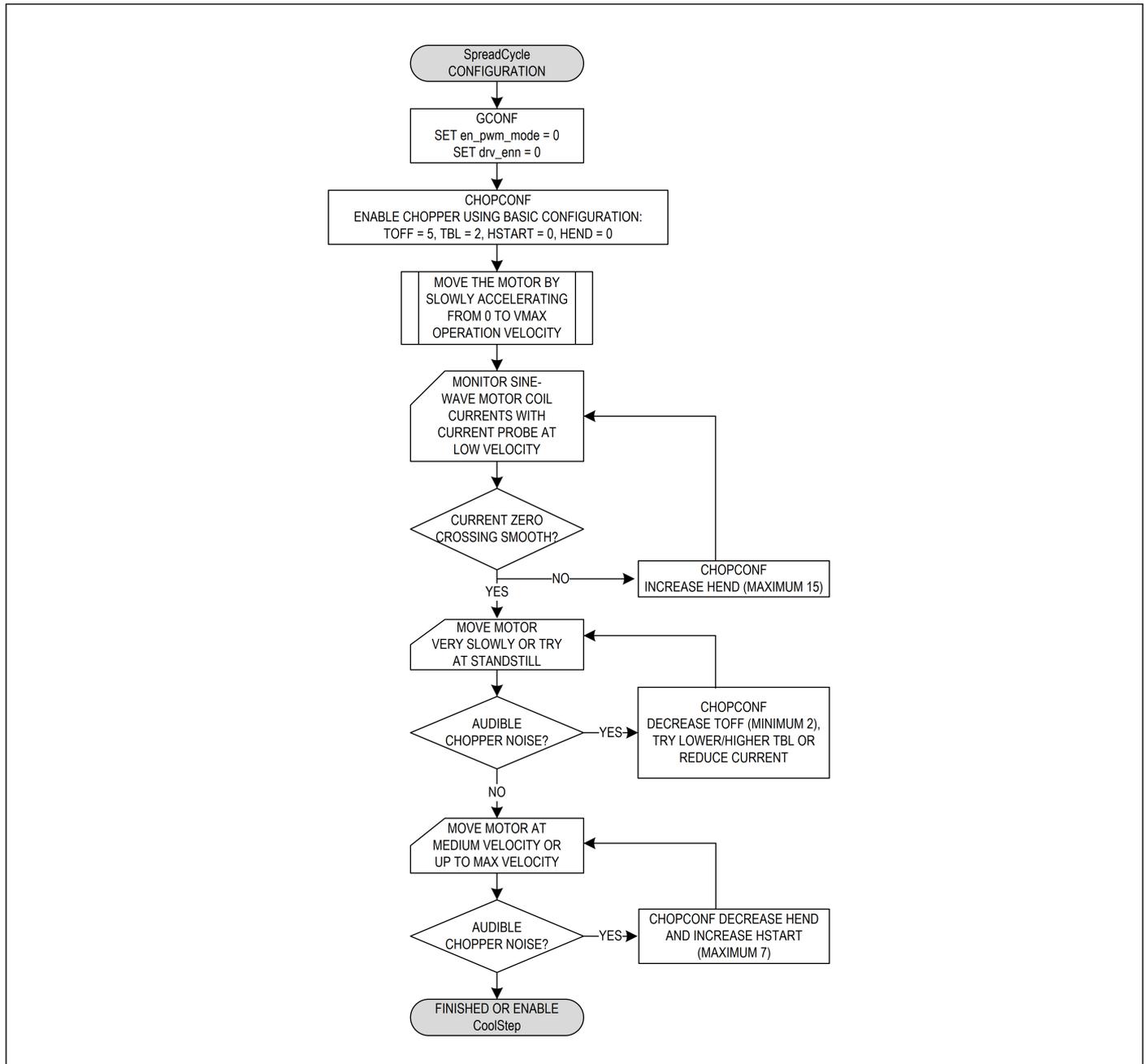


図 49. クイック設定ガイド–SpreadCycle

CoolStep を StealthChop2 と組み合わせてイネーブル

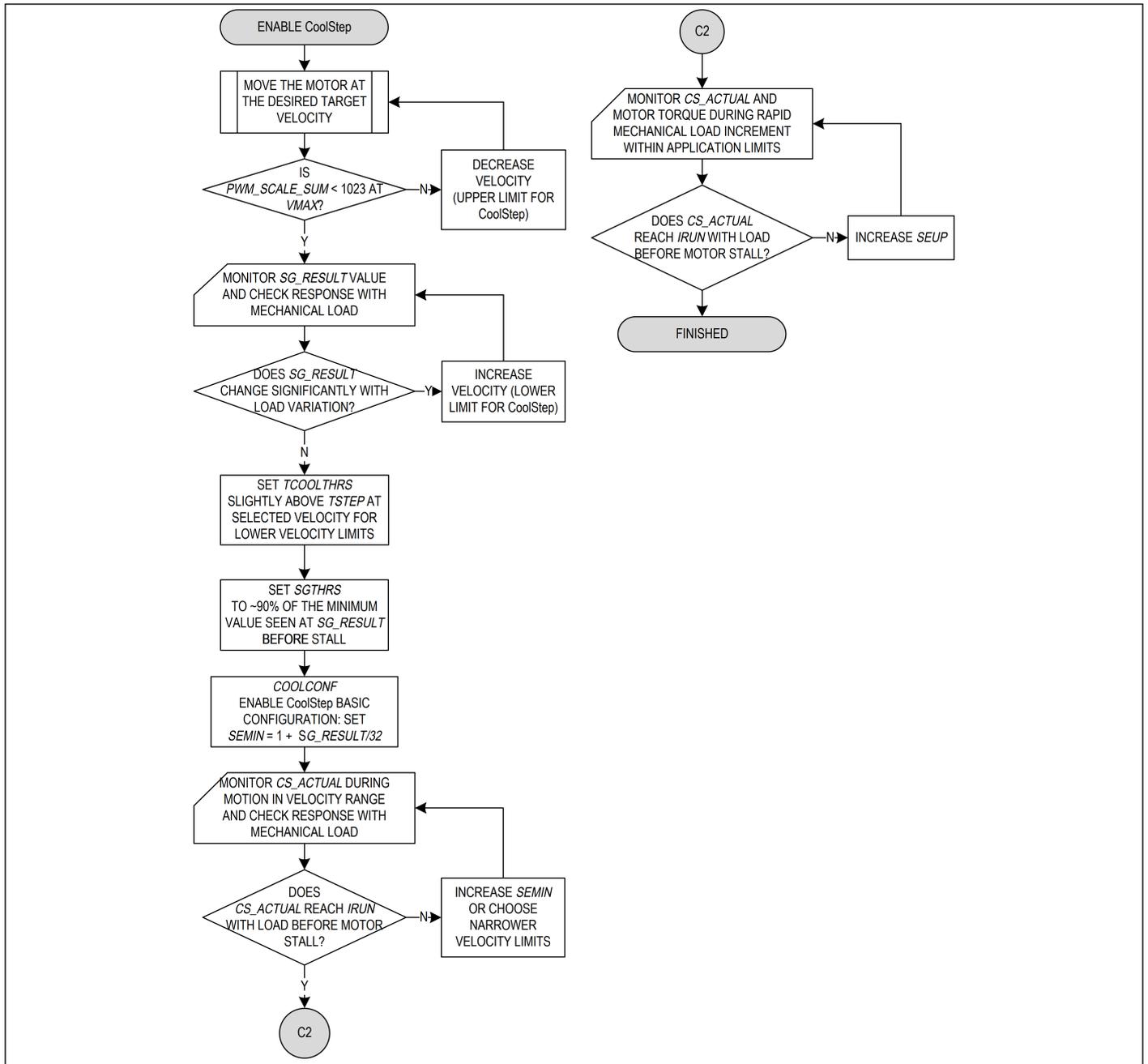


図 50. クイック設定ガイド—StealthChop2 を組み合わせた CoolStep

CoolStep を SpreadCycle と組み合わせてイネーブル

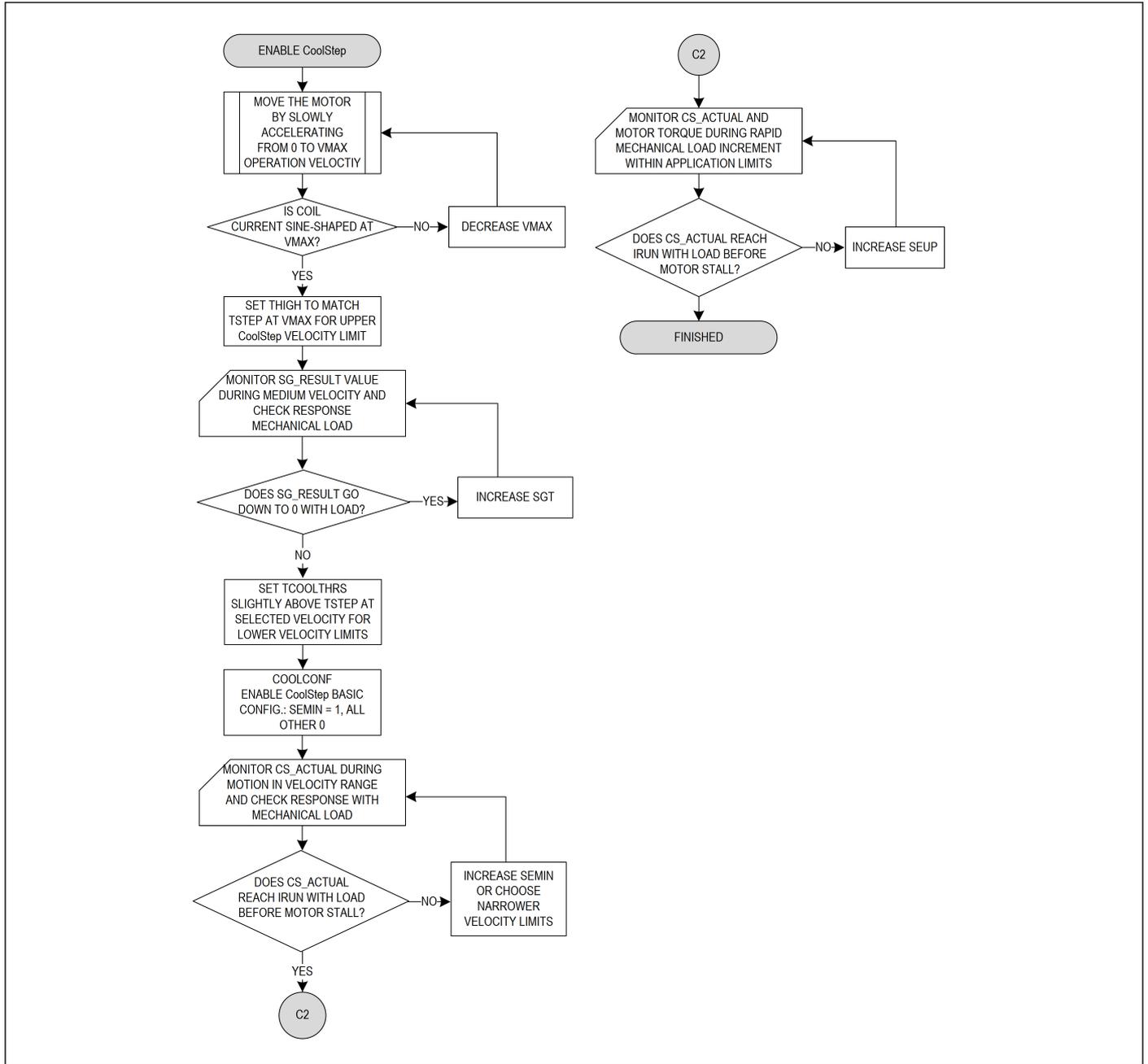


図 51. クイック設定ガイド—SpreadCycle を組み合わせた CoolStep

## モーション・コントローラを用いたモータの作動

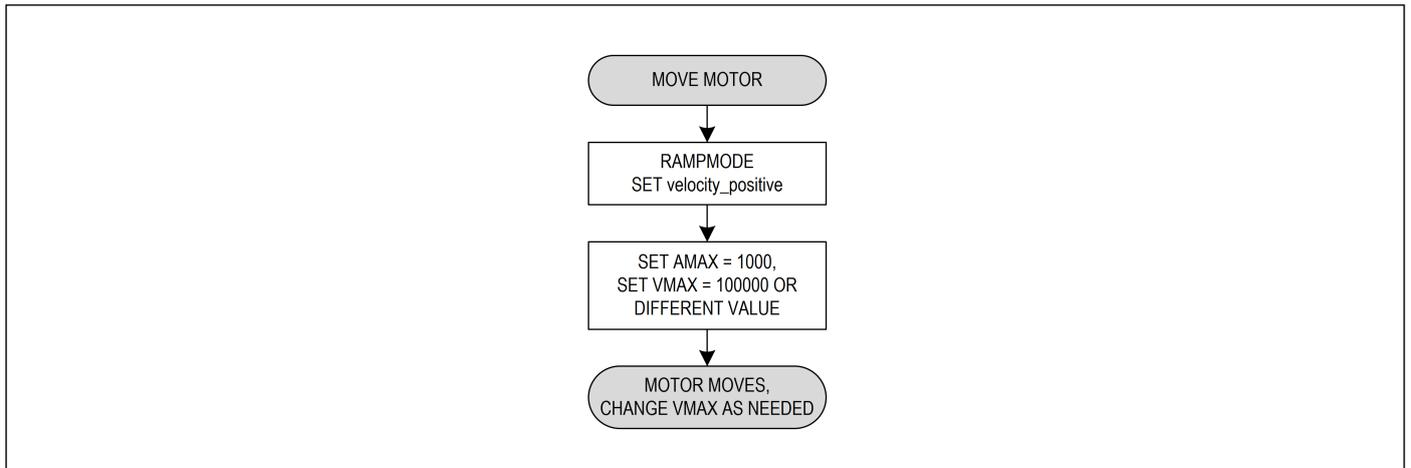


図 52. クイック設定ガイド-速度モードでのモータの作動

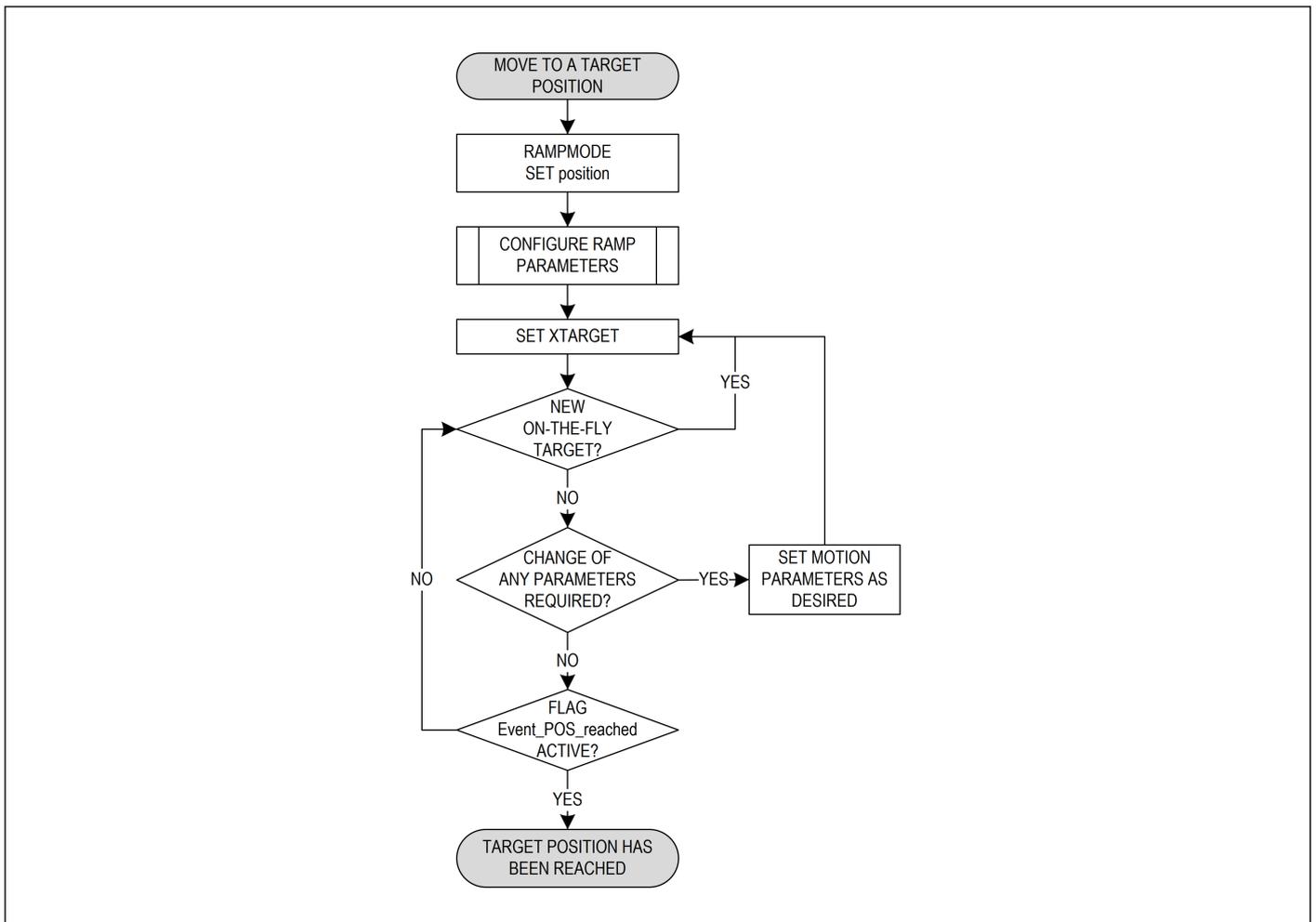


図 53. クイック設定ガイド-目標ポジションへのモータの作動

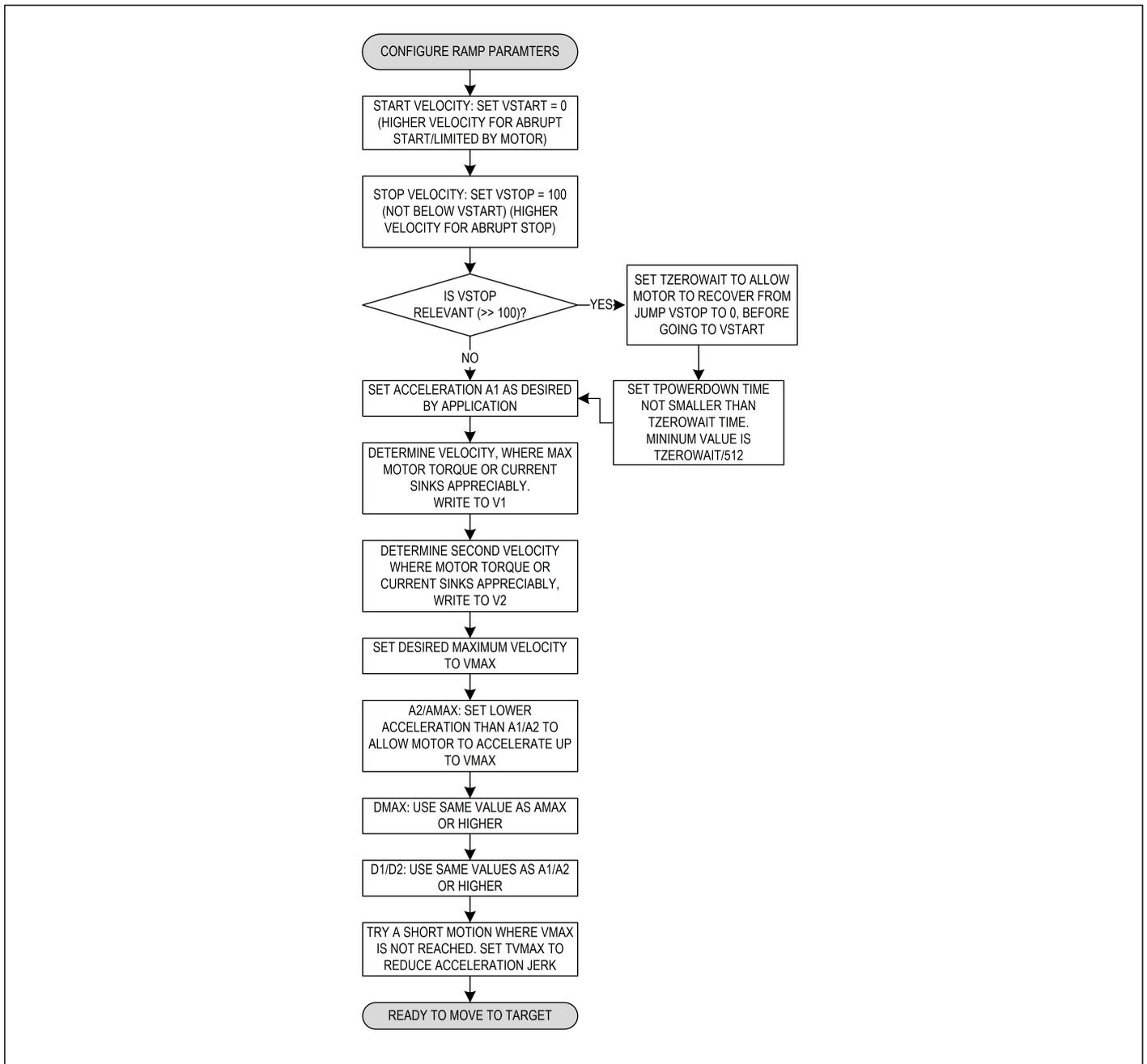


図 54. クイック設定ガイド—モーション・ランプ・パラメータの設定

## DcStep 動作のイネーブル

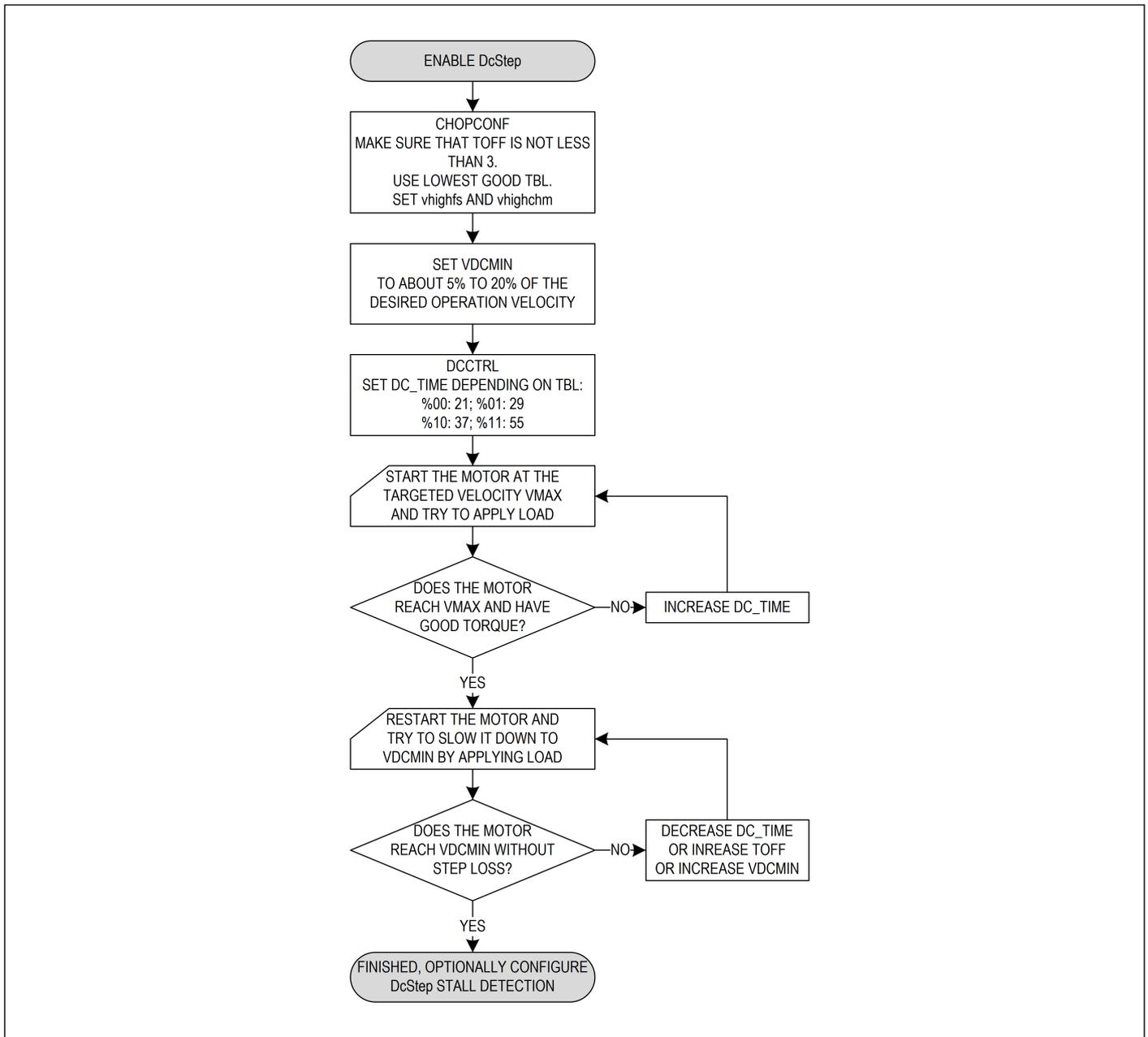


図 55. クイック設定ガイド-DcStep

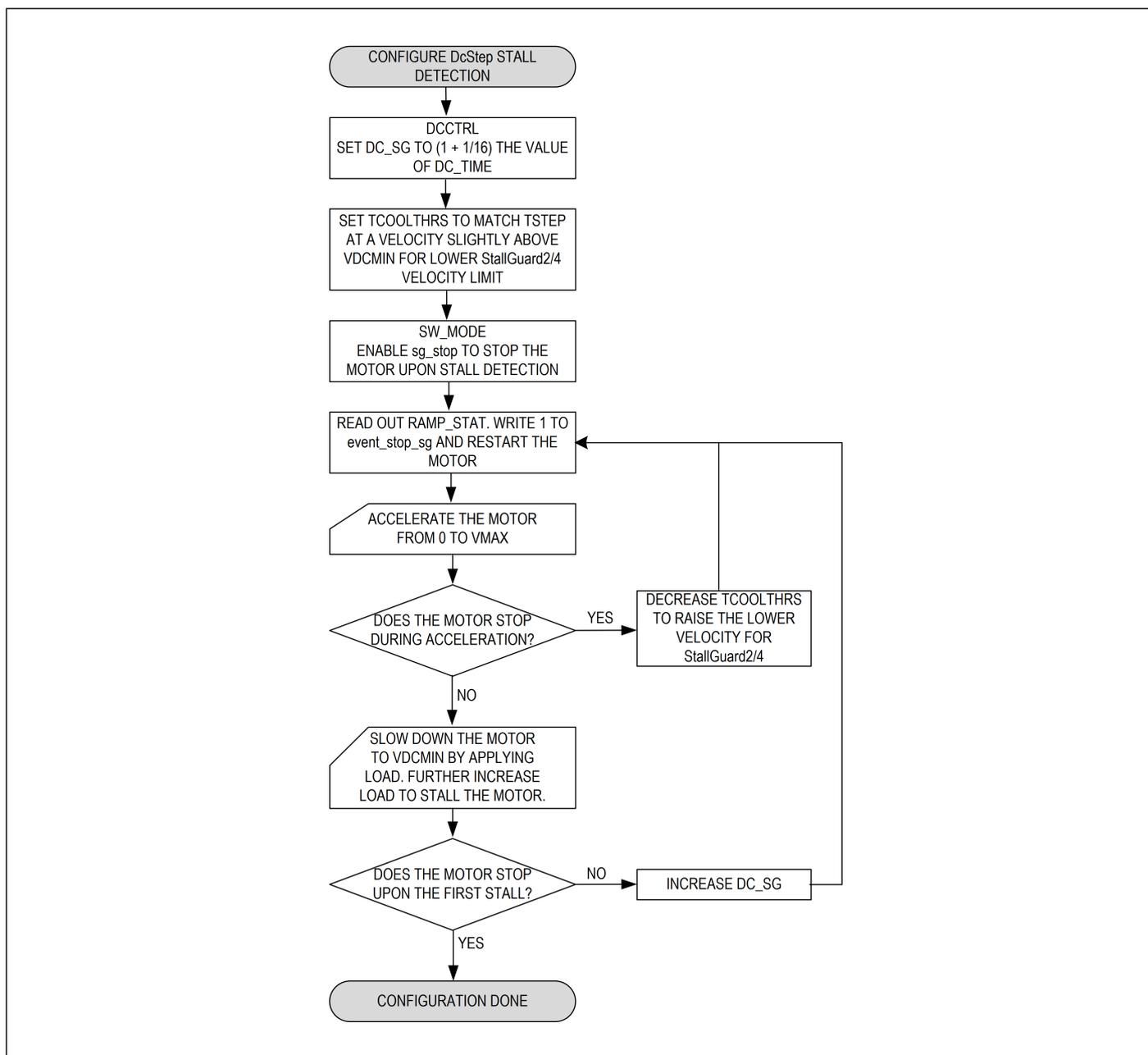


図 56. クイック設定ガイド–DcStep でストール検出を使用

## モータの適合化

モータは、特に低電圧動作の場合、アプリケーションの機構および使用可能な電源電圧と電流に十分見合ったものとなるよう慎重に選択する必要があります。したがって、所定のモータの電源電圧条件を理解することが重要です。特定のトルクを生み出すことと、必要な速度においてこのトルクを提供する能力は、どちらもモータ固有の電圧を必要とします。これら 2 つの要素が合わさって大きな影響力を持ちます。

ステッピング・モータに関連する主なパラメータは以下のとおりです。

- 公称（実効値）コイル電流  $I_{COILNOM}$  [A]
- 公称コイル抵抗  $R_{COIL}$  [ $\Omega$ ]
- 定格コイル電圧  $U_N = R_{COIL} \times I_{COILNOM}$  [V]（場合によっては  $I_{COILNOM}$  の代わりに指定）
- $I_{COILNOM}$  での保持トルク Holding Torque [Nm]
- 逆起電力（BEFM）定数  $C_{BEMF}$  [V/rad/s]。モータの BEMF の詳細については、[モータの逆起電力（BEMF）定数の解釈](#)のセクションにも記載されています。

仕様規定されたモータ・トルクが両モータ・コイルの実効値電流  $I_{COIL}$  で実現され、必要な磁界強度を生成します。本質的に、電流の低下に比例してトルクも低下します。例えば、70%のトルクは 70%の電流で実現されます。70%に低減するだけでも大幅にエネルギーを節約できます。消費電力は電流の 2 乗に比例するためです。したがって、トルク・リザーブの大きなモータほど、高い効率を実現できます。

以上を念頭に、ドライバの電力段抵抗 ( $LS + HS$ ) を考慮し、モータ停止および低速動作に必要な電源電圧  $U_{BAT}$  を次式で計算します。

$$U_{BAT} = (R_{COIL} + R_{DS(ON), LS} + R_{DS(ON), HS}) \times I_{COIL} \times \sqrt{2}$$

$I_{COIL}$  は、必要なトルクを与えるモータ電流の実効値です。

更に、より高速の動作（毎秒数回を超える電氣的回転）では、モータ固有の逆起電力定数  $C_{BEMF}$  も考慮する必要があります（以下の説明を参照）。この場合、所定のモータおよび最大速度[RPM]の実現可能な最低電源電圧は、次式で計算できます。

$$U_{BAT} = \left( (R_{COIL} + R_{DS(ON), LS} + R_{DS(ON), HS}) \times I_{COIL} + \frac{\text{HoldingTorque[Nm]} \times \frac{2\pi \times \text{Velocity[RPM]}}{60}}{2 \times I_{COILNOM}} \right) \times \sqrt{2}$$

**モータをバッテリー動作に適合**：ユーザがコイルの巻線を適応させることは、ほとんどのモータ・サプライヤに対して可能です。これにより、バッテリー動作の低モータ電圧とより大きなモータ電流とのトレード・オフが可能となります。短く太いコイル巻線を使用するモータは、長く細いコイル巻線を使用する同じモータに比べより低い電圧で動作できますが、同じトルクを実現するのに必要な電流は大きくなります。

## レジスタ・マップ

## TMC5271

ADDRESS	NAME	MSB							LSB
<b>General Configuration Registers</b>									
0x00	<a href="#">GCONF[31:24]</a>	-	diag1_sel_nStallIn dex_xcomp	diag0_sel_nError_ Ramp	-	-	-	-	-
	<a href="#">GCONF[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">GCONF[15:8]</a>	qsc_sts_ena	drv_enn	SD	direct_mode	stop_ena ble	small_hy steresis	diag1_poscomp_p ushpull	diag0_int _pushpul l
	<a href="#">GCONF[7:0]</a>	diag1_in dex	diag1_st all_dir	diag0_st all_step	diag0_ot pw	diag0_er ror	shaft	multistep _filt	en_pwm _mode
0x01	<a href="#">GSTAT[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">GSTAT[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">GSTAT[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">GSTAT[7:0]</a>	-	-	-	vm_uvlo	register_ reset	uv_ldo	drv_err	reset
0x02	<a href="#">IFCNT[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">IFCNT[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">IFCNT[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">IFCNT[7:0]</a>	IFCNT[7:0]							
0x03	<a href="#">NODECONF[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">NODECONF[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">NODECONF[15:8]</a>	-	-	-	-	SENDDelay[3:0]			
	<a href="#">NODECONF[7:0]</a>	NODEADDR[7:0]							
0x04	<a href="#">IOIN[31:24]</a>	VERSION[7:0]							
	<a href="#">IOIN[23:16]</a>	-	-	SPI_FLT_SEL[1:0]	-	SILICON_RV[2:0]			
	<a href="#">IOIN[15:8]</a>	qsc	-	OUTPUT	EXT_RE S_DET	EXT_CL K	-	ADC_EN	ADC_TE MPERAT URE[7]
	<a href="#">IOIN[7:0]</a>	ADC_TEMPERATURE[6:0]							
0x05	<a href="#">DRV_CONF[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">DRV_CONF[23:16]</a>	-	-	-	-	-	STANDSTILL_TIME[2:0]		
	<a href="#">DRV_CONF[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">DRV_CONF[7:0]</a>	-	-	-	en_em_d isable	FSR_IREF[1:0]		FSR[1:0]	
0x06	<a href="#">GLOBAL SCALER[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">GLOBAL SCALER[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">GLOBAL SCALER[15:8]</a>	GLOBALSCALER_B[7:0]							
	<a href="#">GLOBAL SCALER[7:0]</a>	GLOBALSCALER_A[7:0]							

ADDRESS	NAME	MSB							LSB	
0x07	<a href="#">RAMPMODE[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">RAMPMODE[23:16]</a>	-	-	-	-	-	-	-	-	
	<a href="#">RAMPMODE[15:8]</a>	-	-	-	-	-	-	-	-	
	<a href="#">RAMPMODE[7:0]</a>	-	-	-	-	-	-	-	RAMPMODE[1:0]	
0x08	<a href="#">MSLUT_ADDR[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">MSLUT_ADDR[23:16]</a>	-	-	-	-	-	-	-	-	
	<a href="#">MSLUT_ADDR[15:8]</a>	-	-	-	-	-	-	-	-	
	<a href="#">MSLUT_ADDR[7:0]</a>	-	-	-	MSLUT_ADDR[4:0]					
0x09	<a href="#">MSLUT_DATA[31:24]</a>	MSLUT_DATA[31:24]								
	<a href="#">MSLUT_DATA[23:16]</a>	MSLUT_DATA[23:16]								
	<a href="#">MSLUT_DATA[15:8]</a>	MSLUT_DATA[15:8]								
	<a href="#">MSLUT_DATA[7:0]</a>	MSLUT_DATA[7:0]								
<b>Position Compare</b>										
0x10	<a href="#">X_COMPARE[31:24]</a>	X_COMPARE[31:24]								
	<a href="#">X_COMPARE[23:16]</a>	X_COMPARE[23:16]								
	<a href="#">X_COMPARE[15:8]</a>	X_COMPARE[15:8]								
	<a href="#">X_COMPARE[7:0]</a>	X_COMPARE[7:0]								
0x11	<a href="#">X_COMPARE_REPEAT [31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">X_COMPARE_REPEAT [23:16]</a>	X_COMPARE_REPEAT[23:16]								
	<a href="#">X_COMPARE_REPEAT [15:8]</a>	X_COMPARE_REPEAT[15:8]								
	<a href="#">X_COMPARE_REPEAT [7:0]</a>	X_COMPARE_REPEAT[7:0]								
<b>Velocity Dependent Configuration Registers</b>										
0x12	<a href="#">IHOLD_IRUN[31:24]</a>	-	-	-	-	IRUNDELAY[3:0]				
	<a href="#">IHOLD_IRUN[23:16]</a>	-	-	-	-	IHOLDDELAY[3:0]				
	<a href="#">IHOLD_IRUN[15:8]</a>	-	-	-	IRUN[4:0]					
	<a href="#">IHOLD_IRUN[7:0]</a>	-	-	-	IHOLD[4:0]					
0x13	<a href="#">TPOWERDOWN[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TPOWERDOWN[23:16]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TPOWERDOWN[15:8]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TPOWERDOWN[7:0]</a>	TPOWERDOWN[7:0]								
0x14	<a href="#">TSTEP[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TSTEP[23:16]</a>	-	-	-	-	TSTEP[19:16]				
	<a href="#">TSTEP[15:8]</a>	TSTEP[15:8]								
	<a href="#">TSTEP[7:0]</a>	TSTEP[7:0]								
0x15	<a href="#">TPWMTHRS[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TPWMTHRS[23:16]</a>	-	-	-	-	TPWMTHRS[19:16]				
	<a href="#">TPWMTHRS[15:8]</a>	TPWMTHRS[15:8]								
	<a href="#">TPWMTHRS[7:0]</a>	TPWMTHRS[7:0]								
0x16	<a href="#">TCOOLTHRS[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TCOOLTHRS[23:16]</a>	-	-	-	-	TCOOLTHRS[19:16]				

ADDRESS	NAME	MSB							LSB
	<a href="#">TCOOLTHRS[15:8]</a>	TCOOLTHRS[15:8]							
	<a href="#">TCOOLTHRS[7:0]</a>	TCOOLTHRS[7:0]							
0x17	<a href="#">THIGH[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">THIGH[23:16]</a>	-	-	-	-	THIGH[19:16]			
	<a href="#">THIGH[15:8]</a>	THIGH[15:8]							
	<a href="#">THIGH[7:0]</a>	THIGH[7:0]							
<b>Ramp Generator Registers</b>									
0x18	<a href="#">XACTUAL[31:24]</a>	XACTUAL[31:24]							
	<a href="#">XACTUAL[23:16]</a>	XACTUAL[23:16]							
	<a href="#">XACTUAL[15:8]</a>	XACTUAL[15:8]							
	<a href="#">XACTUAL[7:0]</a>	XACTUAL[7:0]							
0x19	<a href="#">VACTUAL[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">VACTUAL[23:16]</a>	VACTUAL[23:16]							
	<a href="#">VACTUAL[15:8]</a>	VACTUAL[15:8]							
	<a href="#">VACTUAL[7:0]</a>	VACTUAL[7:0]							
0x1A	<a href="#">AACTUAL[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">AACTUAL[23:16]</a>	AACTUAL[23:16]							
	<a href="#">AACTUAL[15:8]</a>	AACTUAL[15:8]							
	<a href="#">AACTUAL[7:0]</a>	AACTUAL[7:0]							
0x1B	<a href="#">VSTART[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">VSTART[23:16]</a>	-	-	-	-	-	VSTART[17:16]		
	<a href="#">VSTART[15:8]</a>	VSTART[15:8]							
	<a href="#">VSTART[7:0]</a>	VSTART[7:0]							
0x1C	<a href="#">A1[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">A1[23:16]</a>	-	-	-	-	-	A1[17:16]		
	<a href="#">A1[15:8]</a>	A1[15:8]							
	<a href="#">A1[7:0]</a>	A1[7:0]							
0x1D	<a href="#">V1[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">V1[23:16]</a>	-	-	-	-	V1[19:16]			
	<a href="#">V1[15:8]</a>	V1[15:8]							
	<a href="#">V1[7:0]</a>	V1[7:0]							
0x1E	<a href="#">A2[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">A2[23:16]</a>	-	-	-	-	-	A2[17:16]		
	<a href="#">A2[15:8]</a>	A2[15:8]							
	<a href="#">A2[7:0]</a>	A2[7:0]							
0x1F	<a href="#">V2[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">V2[23:16]</a>	-	-	-	-	V2[19:16]			
	<a href="#">V2[15:8]</a>	V2[15:8]							
	<a href="#">V2[7:0]</a>	V2[7:0]							
0x20	<a href="#">AMAX[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">AMAX[23:16]</a>	-	-	-	-	-	AMAX[17:16]		
	<a href="#">AMAX[15:8]</a>	AMAX[15:8]							

ADDRESS	NAME	MSB							LSB	
	<a href="#">AMAX[7:0]</a>	AMAX[7:0]								
0x21	<a href="#">VMAX[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">VMAX[23:16]</a>	-	VMAX[22:16]							
	<a href="#">VMAX[15:8]</a>	VMAX[15:8]								
	<a href="#">VMAX[7:0]</a>	VMAX[7:0]								
0x22	<a href="#">DMAX[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">DMAX[23:16]</a>	-	-	-	-	-	-	-	DMAX[17:16]	
	<a href="#">DMAX[15:8]</a>	DMAX[15:8]								
	<a href="#">DMAX[7:0]</a>	DMAX[7:0]								
0x23	<a href="#">D2[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">D2[23:16]</a>	-	-	-	-	-	-	-	D2[17:16]	
	<a href="#">D2[15:8]</a>	D2[15:8]								
	<a href="#">D2[7:0]</a>	D2[7:0]								
0x24	<a href="#">D1[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">D1[23:16]</a>	-	-	-	-	-	-	-	D1[17:16]	
	<a href="#">D1[15:8]</a>	D1[15:8]								
	<a href="#">D1[7:0]</a>	D1[7:0]								
0x25	<a href="#">VSTOP[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">VSTOP[23:16]</a>	-	-	-	-	-	-	-	VSTOP[17:16]	
	<a href="#">VSTOP[15:8]</a>	VSTOP[15:8]								
	<a href="#">VSTOP[7:0]</a>	VSTOP[7:0]								
0x26	<a href="#">TVMAX[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TVMAX[23:16]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TVMAX[15:8]</a>	TVMAX[15:8]								
	<a href="#">TVMAX[7:0]</a>	TVMAX[7:0]								
0x27	<a href="#">TZEROWAIT[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TZEROWAIT[23:16]</a>	-	-	-	-	-	-	-	-	
	<a href="#">TZEROWAIT[15:8]</a>	TZEROWAIT[15:8]								
	<a href="#">TZEROWAIT[7:0]</a>	TZEROWAIT[7:0]								
0x28	<a href="#">XTARGET[31:24]</a>	XTARGET[31:24]								
	<a href="#">XTARGET[23:16]</a>	XTARGET[23:16]								
	<a href="#">XTARGET[15:8]</a>	XTARGET[15:8]								
	<a href="#">XTARGET[7:0]</a>	XTARGET[7:0]								
<b>Ramp Generator Driver Feature Control Registers</b>										
0x29	<a href="#">VDCMIN[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">VDCMIN[23:16]</a>	-	VDCMIN[14:8]							
	<a href="#">VDCMIN[15:8]</a>	VDCMIN[7:0]								
	<a href="#">VDCMIN[7:0]</a>	Reserved[7:0]								
0x2A	<a href="#">SW_MODE[31:24]</a>	-	-	-	-	-	-	-	-	
	<a href="#">SW_MODE[23:16]</a>	-	-	-	-	-	-	-	-	
	<a href="#">SW_MODE[15:8]</a>	-	virtual_St ep_enc	en_virtua l_stop_r	en_virtua l_stop_l	en_softst op	sg_stop	en_latch _encoder	latch_r_i nactive	

ADDRESS	NAME	MSB						LSB	
	<a href="#">SW_MODE[7:0]</a>	latch_r_active	latch_l_inactive	latch_l_active	swap_lr	pol_stop_r	pol_stop_l	stop_renable	stop_lease
0x2B	<a href="#">RAMP_STAT[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">RAMP_STAT[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">RAMP_STAT[15:8]</a>	status_virtual_stop_r	status_virtual_stop_l	status_s_g	second_move	t_zero_wait_active	vzero	position_reached	velocity_reached
	<a href="#">RAMP_STAT[7:0]</a>	event_pos_reached	event_stop_sg	event_stop_r	event_stop_l	status_latch_r	status_latch_l	status_stop_r	status_stop_l
0x2C	<a href="#">XLATCH[31:24]</a>	XLATCH[31:24]							
	<a href="#">XLATCH[23:16]</a>	XLATCH[23:16]							
	<a href="#">XLATCH[15:8]</a>	XLATCH[15:8]							
	<a href="#">XLATCH[7:0]</a>	XLATCH[7:0]							
0x2D	<a href="#">POSITION_P_CTRL[31:24]</a>	-	-	-	en_tol_on_pos_reached	-	-	-	-
	<a href="#">POSITION_P_CTRL[23:16]</a>	tolerance[7:0]							
	<a href="#">POSITION_P_CTRL[15:8]</a>	-	-	-	-	-	-	P[9:8]	
	<a href="#">POSITION_P_CTRL[7:0]</a>	P[7:0]							
<b>Encoder Registers</b>									
0x2E	<a href="#">X_ENC[31:24]</a>	X_ENC[31:24]							
	<a href="#">X_ENC[23:16]</a>	X_ENC[23:16]							
	<a href="#">X_ENC[15:8]</a>	X_ENC[15:8]							
	<a href="#">X_ENC[7:0]</a>	X_ENC[7:0]							
0x2F	<a href="#">ENCMODE[31:24]</a>	-	-	BEMF_FILTER_SEL[1:0]		-	-	-	-
	<a href="#">ENCMODE[23:16]</a>	BEMF_BLANK_TIME[7:0]							
	<a href="#">ENCMODE[15:8]</a>	qsc_enc_en	bemf_hyst[2:0]			nBEMF_ABN_SEL	enc_sel_decimal	latch_x_act	clr_enc_x
	<a href="#">ENCMODE[7:0]</a>	pos_neg_edge[1:0]		clr_once	clr_cont	ignore_AB	pol_N	pol_B	pol_A
0x30	<a href="#">ENC_CONST[31:24]</a>	ENC_CONST[31:24]							
	<a href="#">ENC_CONST[23:16]</a>	ENC_CONST[23:16]							
	<a href="#">ENC_CONST[15:8]</a>	ENC_CONST[15:8]							
	<a href="#">ENC_CONST[7:0]</a>	ENC_CONST[7:0]							
0x31	<a href="#">ENC_STATUS[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">ENC_STATUS[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">ENC_STATUS[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">ENC_STATUS[7:0]</a>	-	-	-	-	-	-	deviation_warn	n_event
0x32	<a href="#">ENC_LATCH[31:24]</a>	ENC_LATCH[31:24]							
	<a href="#">ENC_LATCH[23:16]</a>	ENC_LATCH[23:16]							

ADDRESS	NAME	MSB							LSB
	<a href="#">ENC_LATCH[15:8]</a>	ENC_LATCH[15:8]							
	<a href="#">ENC_LATCH[7:0]</a>	ENC_LATCH[7:0]							
0x33	<a href="#">ENC_DEVIATION[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">ENC_DEVIATION[23:16]</a>	-	-	-	-	ENC_DEVIATION[19:16]			
	<a href="#">ENC_DEVIATION[15:8]</a>	ENC_DEVIATION[15:8]							
	<a href="#">ENC_DEVIATION[7:0]</a>	ENC_DEVIATION[7:0]							
0x34	<a href="#">VIRTUAL_STOP_L[31:24]</a>	VIRTUAL_STOP_L[31:24]							
	<a href="#">VIRTUAL_STOP_L[23:16]</a>	VIRTUAL_STOP_L[23:16]							
	<a href="#">VIRTUAL_STOP_L[15:8]</a>	VIRTUAL_STOP_L[15:8]							
	<a href="#">VIRTUAL_STOP_L[7:0]</a>	VIRTUAL_STOP_L[7:0]							
0x35	<a href="#">VIRTUAL_STOP_R[31:24]</a>	VIRTUAL_STOP_R[31:24]							
	<a href="#">VIRTUAL_STOP_R[23:16]</a>	VIRTUAL_STOP_R[23:16]							
	<a href="#">VIRTUAL_STOP_R[15:8]</a>	VIRTUAL_STOP_R[15:8]							
	<a href="#">VIRTUAL_STOP_R[7:0]</a>	VIRTUAL_STOP_R[7:0]							
<b>Motor Driver Registers</b>									
0x36	<a href="#">MSCNT[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">MSCNT[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">MSCNT[15:8]</a>	-	-	-	-	-	-	MSCNT[9:8]	
	<a href="#">MSCNT[7:0]</a>	MSCNT[7:0]							
0x37	<a href="#">MSCURACT[31:24]</a>	-	-	-	-	-	-	-	CUR_B[8]
	<a href="#">MSCURACT[23:16]</a>	CUR_B[7:0]							
	<a href="#">MSCURACT[15:8]</a>	-	-	-	-	-	-	-	CUR_A[8]
	<a href="#">MSCURACT[7:0]</a>	CUR_A[7:0]							
0x38	<a href="#">CHOPCONF[31:24]</a>	diss2vs	diss2g	dedge	intpol	MRES[3:0]			
	<a href="#">CHOPCONF[23:16]</a>	TPFD[3:0]				vhighchm	vhighfs	-	TBL[1]
	<a href="#">CHOPCONF[15:8]</a>	TBL[0]	chm	-	disfdcc	fd3	HEND_OFFSET[3:1]		
	<a href="#">CHOPCONF[7:0]</a>	HEND_OFFSET[0]	HSTRT_TFD210[2:0]			TOFF[3:0]			
0x39	<a href="#">COOLCONF[31:24]</a>	-	-	-	-	-	-	-	sflt
	<a href="#">COOLCONF[23:16]</a>	-	sgt[6:0]						
	<a href="#">COOLCONF[15:8]</a>	seimin	sedn[1:0]		-	semax[3:0]			
	<a href="#">COOLCONF[7:0]</a>	-	seup[1:0]		-	semin[3:0]			
0x3A	<a href="#">DCCTRL[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">DCCTRL[23:16]</a>	DC_SG[7:0]							

ADDRESS	NAME	MSB							LSB
	<a href="#">DCCTRL[15:8]</a>	-	-	-	-	-	-	-	DC_TIME[9:8]
	<a href="#">DCCTRL[7:0]</a>	DC_TIME[7:0]							
0x3B	<a href="#">DRV_STATUS[31:24]</a>	stst	olb	ola	s2gb	s2ga	otpw	ot	stallguard
	<a href="#">DRV_STATUS[23:16]</a>	-	-	-	CS_ACTUAL[4:0]				
	<a href="#">DRV_STATUS[15:8]</a>	fsactive	stealth	s2vsb	s2vsa	-	-	-	SG_RESULT[9:8]
	<a href="#">DRV_STATUS[7:0]</a>	SG_RESULT[7:0]							
0x3C	<a href="#">PWMCONF[31:24]</a>	PWM_LIM[3:0]			PWM_REG[3:0]				
	<a href="#">PWMCONF[23:16]</a>	pwm_dis_reg_stst	pwm_meas_sd_enable	FREEWHEEL[1:0]	pwm_autograd	pwm_autoscale	PWM_FREQ[1:0]		
	<a href="#">PWMCONF[15:8]</a>	PWM_GRAD[7:0]							
	<a href="#">PWMCONF[7:0]</a>	PWM_OFS[7:0]							
0x3D	<a href="#">PWM_SCALE[31:24]</a>	-	-	-	-	-	-	-	PWM_SCALE_AUTO[8]
	<a href="#">PWM_SCALE[23:16]</a>	PWM_SCALE_AUTO[7:0]							
	<a href="#">PWM_SCALE[15:8]</a>	-	-	-	-	-	-	-	PWM_SCALE_SUM[9:8]
	<a href="#">PWM_SCALE[7:0]</a>	PWM_SCALE_SUM[7:0]							
0x3E	<a href="#">PWM_AUTO[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">PWM_AUTO[23:16]</a>	PWM_GRAD_AUTO[7:0]							
	<a href="#">PWM_AUTO[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">PWM_AUTO[7:0]</a>	PWM_OFS_AUTO[7:0]							
0x3F	<a href="#">SG4_THRS[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">SG4_THRS[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">SG4_THRS[15:8]</a>	-	-	-	-	-	-	SG_ANGLE_OFFSET	SG4_fit_en
	<a href="#">SG4_THRS[7:0]</a>	SG4_THRS[7:0]							
0x40	<a href="#">SG4_RESULT[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">SG4_RESULT[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">SG4_RESULT[15:8]</a>	-	-	-	-	-	-	-	SG4_RESULT[9:8]
	<a href="#">SG4_RESULT[7:0]</a>	SG4_RESULT[7:0]							
0x41	<a href="#">SG4_IND[31:24]</a>	sg4_ind_3[7:0]							
	<a href="#">SG4_IND[23:16]</a>	sg4_ind_2[7:0]							
	<a href="#">SG4_IND[15:8]</a>	sg4_ind_1[7:0]							
	<a href="#">SG4_IND[7:0]</a>	sg4_ind_0[7:0]							
<b>Microstep Look-Up Table</b>									
0x80	<a href="#">MSLUT0[31:24]</a>	MSLUT_0[31:24]							
	<a href="#">MSLUT0[23:16]</a>	MSLUT_0[23:16]							
	<a href="#">MSLUT0[15:8]</a>	MSLUT_0[15:8]							
	<a href="#">MSLUT0[7:0]</a>	MSLUT_0[7:0]							
0x81	<a href="#">MSLUT1[31:24]</a>	MSLUT_1[31:24]							
	<a href="#">MSLUT1[23:16]</a>	MSLUT_1[23:16]							

ADDRESS	NAME	MSB							LSB
	<a href="#">MSLUT1[15:8]</a>	MSLUT_1[15:8]							
	<a href="#">MSLUT1[7:0]</a>	MSLUT_1[7:0]							
0x82	<a href="#">MSLUT2[31:24]</a>	MSLUT_2[31:24]							
	<a href="#">MSLUT2[23:16]</a>	MSLUT_2[23:16]							
	<a href="#">MSLUT2[15:8]</a>	MSLUT_2[15:8]							
	<a href="#">MSLUT2[7:0]</a>	MSLUT_2[7:0]							
0x83	<a href="#">MSLUT3[31:24]</a>	MSLUT_3[31:24]							
	<a href="#">MSLUT3[23:16]</a>	MSLUT_3[23:16]							
	<a href="#">MSLUT3[15:8]</a>	MSLUT_3[15:8]							
	<a href="#">MSLUT3[7:0]</a>	MSLUT_3[7:0]							
0x84	<a href="#">MSLUT4[31:24]</a>	MSLUT_4[31:24]							
	<a href="#">MSLUT4[23:16]</a>	MSLUT_4[23:16]							
	<a href="#">MSLUT4[15:8]</a>	MSLUT_4[15:8]							
	<a href="#">MSLUT4[7:0]</a>	MSLUT_4[7:0]							
0x85	<a href="#">MSLUT5[31:24]</a>	MSLUT_5[31:24]							
	<a href="#">MSLUT5[23:16]</a>	MSLUT_5[23:16]							
	<a href="#">MSLUT5[15:8]</a>	MSLUT_5[15:8]							
	<a href="#">MSLUT5[7:0]</a>	MSLUT_5[7:0]							
0x86	<a href="#">MSLUT6[31:24]</a>	MSLUT_6[31:24]							
	<a href="#">MSLUT6[23:16]</a>	MSLUT_6[23:16]							
	<a href="#">MSLUT6[15:8]</a>	MSLUT_6[15:8]							
	<a href="#">MSLUT6[7:0]</a>	MSLUT_6[7:0]							
0x87	<a href="#">MSLUT7[31:24]</a>	MSLUT_7[31:24]							
	<a href="#">MSLUT7[23:16]</a>	MSLUT_7[23:16]							
	<a href="#">MSLUT7[15:8]</a>	MSLUT_7[15:8]							
	<a href="#">MSLUT7[7:0]</a>	MSLUT_7[7:0]							
0x88	<a href="#">MSLUT_START[31:24]</a>	OFFSET_SIN90[7:0]							
	<a href="#">MSLUT_START[23:16]</a>	START_SIN90[7:0]							
	<a href="#">MSLUT_START[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">MSLUT_START[7:0]</a>	START_SIN[7:0]							
0x89	<a href="#">MSLUT_SEL[31:24]</a>	X3[7:0]							
	<a href="#">MSLUT_SEL[23:16]</a>	X2[7:0]							
	<a href="#">MSLUT_SEL[15:8]</a>	X1[7:0]							
	<a href="#">MSLUT_SEL[7:0]</a>	W3[1:0]	W2[1:0]	W1[1:0]	W0[1:0]				

## レジスタの詳細

## GCONF (0x0)

BIT	31	30	29	28	27	26	25	24
Field	–	diag1_sel_n StallIndex_x comp	diag0_sel_n Error_Ramp	–	–	–	–	–
Reset	–	0x0	0x0	–	–	–	–	–
Access Type	–	Write, Read	Write, Read	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	qsc_sts_en a	drv_enn	SD	direct_mode	stop_enable	small_hyste resis	diag1_posc omp_pushp ull	diag0_int_p ushpull
Reset	0x0	0x1	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	diag1_index	diag1_stall_ dir	diag0_stall_ step	diag0_otpw	diag0_error	shaft	multistep_fil t	en_pwm_m ode
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明	デコード
diag1_sel_nS tallIndex_xcomp	30	DIAG1の出力設定	0x0: DIAG1 provides selected information on stall or index pulse as configured 0x1: DIAG1 is signaling X_COMPARE = X_ACTUAL or direction of motor 0 motion controller as configured
diag0_sel_nE rror_Ramp	29	DIAG0の出力設定	0x0: DIAG0 provides selected information on the driver status 0x1: DIAG0 provides motion controller status information
qsc_sts_ena	15	低消費電力を実現するために自己消費モードを有効化。これにより、drv_ennを1にリセットするとドライバがディスエーブルされます。  ドライバが停止状態になった場合にのみインエーブルできます。  デバイスが自己消費モードかどうかを調べるには、qscビット (0x4 [15]) をチェックします。	0x0: disable quiescent mode 0x1: enable quiescent mode
drv_enn	14	ドライバ段の制御	0x0: Enable driver 0x1: Disable driver

ビットフィールド	ビット	説明	デコード
SD	13	1に設定すると、内蔵ランプ・ジェネレータがオフに切り替わり、外部のS/D入力が使用されます。	0x0: Using internal motion controller 0x1: Switches off internal motion controller and using external STEP/DIR inputs
direct_mode	12	シリアル・インターフェースを介して直接モータ位相電流制御をイネーブル	0x0: Normal operation 0x1: Motor coil currents and polarity directly programmed through the serial interface. Register XTARGET (0x28) specifies signed coil A current (bits 8:0) and coil B current (bits 24:16). In this mode, the current is scaled by the IHOLD setting (0x12). Velocity-based current regulation of StealthChop2 is not available in this mode. The automatic StealthChop2 current regulation only works for low stepper motor velocities.
stop_enable	11	モータ・ハード・ストップ機能を有効化	0x0: Normal operation 0x1: Emergency stop. ENC_A1 stops the sequencer when tied high (no steps are executed by the sequencer: motor goes to standstill state).
small_hysteresis	10	ステップ周波数比較のためのヒステリシスの選択	0x0: Hysteresis for step frequency comparison is 1/16 0x1: Hysteresis for step frequency comparison is 1/32
diag1_poscomp_pushpull	9	DIAG1の出カタイプ設定	0x0: DIAG1 is open collector output (active low) 0x1: Enable DIAG1 push-pull output (active low)
diag0_int_pushpull	8	DIAG0の出カタイプ設定	0x0: DIAG0 is open collector output (active low) 0x1: Enable DIAG0 push-pull output (active low)
diag1_index	7	DIAG1の出力設定	0x0: Disable DIAG1 active on index position 0x1: diag1_index Enable DIAG1 active on index position (microstep look-up table position 0)
diag1_stall_dir	6	diag1_sel_nStallIndex_xcomp (0x0 [30]) に応じたDIAG1の出力設定	0x0: diag1_sel_nStallIndex_xcomp = 0 disables motor stall signal at DIAG1 Motor stall not indicated at DIAG1  diag1_sel_nStallIndex_xcomp = 1 enables DIAG1 to output position compare signal DIAG1 outputs position compare signal 0x1: diag1_sel_nStallIndex_xcomp = 0 enables DIAG1 active on motor stall (set TCOOLTHRS before using this feature)  diag1_sel_nStallIndex_xcomp = 1 enables DIAG1 as direction output for external STEP/DIR driver using the direction signal of ramp generator 0
diag0_stall_step	5	diag0_sel_nError_Ramp (0x0 [29]) に応じたDIAG0の出力設定	0x0: diag0_sel_nError_Ramp = 0 disables motor stall signal at DIAG0  diag0_sel_nError_Ramp = 1 enables motion controller interrupt flags at DIAG0 0x1: diag0_sel_nError_Ramp = 0 enables DIAG0 active on motor stall  diag0_sel_nError_Ramp = 1 enables DIAG0 as step output for external STEP/DIR driver using the direction signal of ramp generator 0

ビットフィールド	ビット	説明	デコード
diag0_otpw	4	DIAG0の出力設定	0x0: Disable DIAG0 active on driver overtemperature prewarning 0x1: Enable DIAG0 active on driver overtemperature prewarning (otpw)
diag0_error	3	DIAG0の出力設定 DIAG0は常にリセット・ステータスを示します。つまり、リセット状態時にはアクティブ・ローです。	0x0: Disable DIAG0 active on driver errors 0x1: Enable DIAG0 active on driver errors Over temperature (ot), short to GND (s2g)
shaft	2	モータの方向/方向符号の変更	0x0: Default motor direction 0x1: Inverse motor direction
multistep_filt	1	StealthChop2のステップ入力フィルタ処理をイネーブル	0x0: Step input filtering disabled 0x1: Enable step input filtering for StealthChop2 optimization with external step source (default = 1)
en_pwm_mode	0	StealthChop2モードを有効化	0x0: No StealthChop2 0x1: StealthChop2 voltage PWM mode enabled (depending on velocity thresholds). Switch from off to on state while in stand-still and at I <sub>HOLD</sub> = nominal I <sub>RUN</sub> current, only.

## GSTAT (0x1)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	7	6	5	4	3	2	1	0
Field	-	-	-	vm_uvlo	register_reset	uv_ido	drv_err	reset
Reset	-	-	-	0b0	0b1	0b0	0b0	0b1
Access Type	-	-	-	Write 1 to Clear, Read				

ビットフィールド	ビット	説明	デコード
vm_uvlo	4	V <sub>S</sub> の低電圧検出フラグ	0: Normal operation 1: VS undervoltage has occurred since last reset
register_reset	3	レジスタマップのデフォルト・リセット・フラグ	0: Normal operation 1: Indicates that the register map has been reset. All registers have been cleared to reset values.

ビットフィールド	ビット	説明	デコード
uv_ldo	2	LDOの低電圧状態フラグ	0: Normal operation 1: Indicates an undervoltage on the LDO. The driver is disabled during undervoltage. This flag is latched for information. This flag is also set when returning from quiescence mode due to resetting the LDO.
drv_err	1	ドライバ・エラー・フラグ	0: Normal operation 1: Indicates that the driver has been shut down due to overtemperature or short-circuit detection. Read DRV_STATUS for details. The flag can only be cleared when the temperature is below the limit again.
reset	0	リセット・フラグ	0: Normal operation 1: Indicates that the IC has been reset

## IFCNT (0x2)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	7	6	5	4	3	2	1	0
Field	IFCNT[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
IFCNT	7:0	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。

## NODECONF (0x3)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	SENDDelay[3:0]			
Reset	–	–	–	–	0x3			
Access Type	–	–	–	–	Write, Read			
BIT	7	6	5	4	3	2	1	0
Field	NODEADDR[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明	デコード
SENDDelay	11:8	SWUARTのノード設定	0x0: 8-bit times (not allowed with multiple nodes) 0x1: 8-bit times (not allowed with multiple nodes) 0x2: 24-bit times 0x3: 24-bit times 0x4: 40-bit times 0x5: 40-bit times 0x6: 56-bit times 0x7: 56-bit times 0x8: 72-bit times 0x9: 72-bit times 0xA: 88-bit times 0xB: 88-bit times 0xC: 104-bit times 0xD: 104-bit times 0xE: 120-bit times 0xF: 120-bit times
NODEADDR	7:0	これら8ビットは、このノードのUARTアドレスを設定します。このアドレスは、AD0、AD1、AD2の定義に従い1~6だけ増加します。  NODEADDR + ADxは255未満でなくてはなりません。	

## IOIN (0x4)

BIT	31	30	29	28	27	26	25	24
Field	VERSION[7:0]							
Reset								
Access Type	Read Only							
BIT	23	22	21	20	19	18	17	16
Field	–	–	SPI_FLT_SEL[1:0]		–	SILICON_RV[2:0]		
Reset	–	–	0x2		–	0b000		
Access Type	–	–	Write, Read		–	Read Only		
BIT	15	14	13	12	11	10	9	8
Field	qsc	–	OUTPUT	EXT_RES_DET	EXT_CLK	–	ADC_EN	ADC_TEMPERATURE[7]
Reset		–	0b1	0b0	0b0	–	0x1	
Access Type	Read Only	–	Write, Read	Read Only	Read Only	–	Write, Read	Read Only
BIT	7	6	5	4	3	2	1	0
Field	ADC_TEMPERATURE[6:0]							Reserved
Reset								
Access Type	Read Only							Read Only

ビットフィールド	ビット	説明	デコード
VERSION	31:24	0x60 (番号が同じ場合、完全なデジタル互換性があることを意味します。)	
SPI_FLT_SEL	21:20	SPIインターフェース用のフィルタ設定	0x0: No filter is applied 0x1: No filter is applied 0x2: 10ns filter 0x3: 20ns filter
SILICON_RV	18:16	シリコン・リビジョン番号	0x0: Engineering samples 0x1: Engineering samples 0x2: Mass production silicon revision 0x3 0x4 0x5 0x6 0x7
qsc	15	自己消費モードのフラグ	0x0: Normal mode 0x1: Device is in quiescence mode
OUTPUT	13	USELピンを介して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、複数のノードのデジタイゼーション・アドレス指定を行うことです。 注：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。 UARTリング・モードの場合は使用しません。	0x0: Low level, 0 0x1: High level, 1 (default reset value)

ビットフィールド	ビット	説明	デコード
EXT_RES_DET	12	外付け抵抗の検出	0x0: No external resistor detected 0x1: External resistor between IREF and GND
EXT_CLK	11	クロック・ソース・インジケータ・フラグ	0x0: The internal oscillator is used for generating the clock signal (12.5MHz) 0x1: The external oscillator is used for generating the clock signal
ADC_EN	9	ADCイネーブル制御	0x0: Disable ADC when temperature measurement is not needed to reduce power consumption 0x1: Enable temperature measurement of ADC (default)
ADC_TEMPERATURE	8:1	$T = (2.03 \times \text{ADC\_TEMPERATURE} - 259)^\circ\text{C}$	
Reserved	0		

## DRV\_CONF (0x5)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	STANDSTILL_TIME[2:0]		
Reset	-	-	-	-	-	0x0		
Access Type	-	-	-	-	-	Write, Read		
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	7	6	5	4	3	2	1	0
Field	-	-	-	en_em_disable	FSR_IREF[1:0]		FSR[1:0]	
Reset	-	-	-	0x0	0x0		0x0	
Access Type	-	-	-	Write, Read	Write, Read		Write, Read	

ビットフィールド	ビット	説明	デコード
STANDSTILL_TIME	18:16	ドライバが停止を示す信号を発するまで通過させる必要のあるCLKサイクル数を決定します。 $2^{(20-\text{STANDSTILL\_TIME})}$	0x0: $2^{20}$ 0x1: $2^{19}$ 0x2: $2^{18}$ 0x3: $2^{17}$ 0x4: $2^{16}$ 0x5: $2^{15}$ 0x6: $2^{14}$ 0x7: $2^{13}$

ビットフィールド	ビット	説明	デコード
en_em_disable	4	緊急ドライバ・オフ・イネーブル	0x0: Emergency stop disabled 0x1: Enables driver disable when REFL and REFR inputs are both triggered
FSR_IREF	3:2	リファレンス電流 $I_{REF}$ ( $R_{REF}$ に基づく) をスケールリングします。これは外部抵抗 $R_{REF}$ をスケールリングするのと同じです。  これとFSRを使用すると電流スケールリングの微調整が可能です。	0x0: 25% $I_{REF}$ 0x1: 50% $I_{REF}$ 0x2: 75% $I_{REF}$ 0x3: 100% $I_{REF}$
FSR	1:0	フルスケール・レンジ  フルスケール・ビットにより、基本的に、ドライバの $R_{DS(ON)}$ の電流検出をモータ電流範囲に適応させることができます。 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。 また、フルスケール・ビットは、過電流保護スレッシュホールドも定義します（この値はピーク電流設定です）。  $R_{DS(ON)}$ および過電流保護スレッシュホールドの値は、電気的特性の表の出力仕様および保護回路の項に記載されています。	0x0: $K_{IFS} = 4.10$ , typ. $R_{DS(ON),LS} = 0.19\Omega$ . 0x1: $K_{IFS} = 8.16$ , typ. $R_{DS(ON),LS} = 0.1\Omega$ . 0x2: $K_{IFS} = 12.06$ , typ. $R_{DS(ON),LS} = 0.07\Omega$ . 0x3: $K_{IFS} = 16.00$ , typ. $R_{DS(ON),LS} = 0.055\Omega$ .

## GLOBAL SCALER (0x6)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	GLOBALSCALER_B[7:0]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	GLOBALSCALER_A[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
GLOBALSCALER_B	15:8	<p>モータ電流のグローバル・スケーリング。ドライブを特定のモータ・タイプに合わせるために、この値が電流スケーリングに乘じられます。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。位相Bをスケーリングします。</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><b>ヒント：</b> 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。 GLOBALSCALER_BとGLOBALSCALER_Aで異なる値を用いることは、SpreadCycleモードの場合にのみ可能です。</p>
GLOBALSCALER_A	7:0	<p>モータ電流のグローバル・スケーリング。ドライブを特定のモータ・タイプに合わせるために、この値が電流スケーリングに乘じられます。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。位相Aをスケーリングします。</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><b>ヒント：</b> 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。 GLOBALSCALER_BとGLOBALSCALER_Aで異なる値を用いることは、SpreadCycleモードの場合にのみ可能です。</p>

#### RAMPMODE (0x7)

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
Field	-	-	-	-	-	-	RAMPMODE[1:0]	
Reset	-	-	-	-	-	-	0x0	
Access Type	-	-	-	-	-	-	Write, Read	

ビットフィールド	ビット	説明	デコード
RAMPMODE	1:0	モーション・コントローラのランピング・モード	0x0: Positioning mode (using all A, D, and V parameters) 0x1: Velocity mode to positive VMAX (using AMAX acceleration) 0x2: Velocity mode to negative VMAX (using AMAX acceleration) 0x3: Hold mode (velocity remains unchanged unless stop event occurs)

**MSLUT\_ADDR (0x8)**

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	7	6	5	4	3	2	1	0
Field	-	-	-	MSLUT_ADDR[4:0]				
Reset	-	-	-	0x0				
Access Type	-	-	-	Write, Read				

ビットフィールド	ビット	説明
MSLUT_ADDR	4:0	MSLUT_DATA (0x9) レジスタでRW操作を行うためにアクセスできるMSLUTレジスタを選択します

**MSLUT\_DATA (0x9)**

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_DATA[31:24]							
Reset	0xAAAAAB54							
Access Type	Write, Read							

BIT	23	22	21	20	19	18	17	16
Field	MSLUT_DATA[23:16]							
Reset	0xAAAAAB554							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_DATA[15:8]							
Reset	0xAAAAAB554							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_DATA[7:0]							
Reset	0xAAAAAB554							
Access Type	Write, Read							

ビットフィールド	ビット	説明
MSLUT_DATA	31:0	MSLUT_ADDRで選択します 0x00 : MSLUT_0 0x01 : MSLUT_1 0x02 : MSLUT_2 0x03 : MSLUT_3 0x04 : MSLUT_4 0x05 : MSLUT_5 0x06 : MSLUT_6 0x07 : MSLUT_7 0x08 : MSLUT_START 0x09 : MSLUT_SEL

**X\_COMPARE (0x10)**

BIT	31	30	29	28	27	26	25	24
Field	X_COMPARE[31:24]							
Reset	0xFFFFFFFF							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	X_COMPARE[23:16]							
Reset	0xFFFFFFFF							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	X_COMPARE[15:8]							
Reset	0xFFFFFFFF							
Access Type	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	X_COMPARE[7:0]							
Reset	0xFFFFFFFF							
Access Type	Write, Read							

ビットフィールド	ビット	説明
X_COMPARE	31:0	<p>モーション・コントローラ・ポジション・ストロープ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。 GCONF (0x0) でイネーブルされている場合、ポジション・パルスは出力DIAG1で使用できます。</p> <p>XACTUAL = X_COMPARE : 出力信号 (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。 X_COMPARE_REPEATが1より大きい場合、X_COMPAREは、periodicポジション・ストロープ・トリガ出力のポジション・リファレンスです。</p>

### X\_COMPARE\_REPEAT (0x11)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	X_COMPARE_REPEAT[23:16]							
Reset	0x0							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	X_COMPARE_REPEAT[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	X_COMPARE_REPEAT[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
X_COMPARE_REPEAT	23:0	<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>そのため、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マイクロステップのモジュール計算を行うためのベース・ポジションを定義します。</p>

## IHOLD\_IRUN (0x12)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	IRUNDELAY[3:0]			
Reset	–	–	–	–	0x4			
Access Type	–	–	–	–	Write, Read			
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	IHOLDDELAY[3:0]			
Reset	–	–	–	–	0x1			
Access Type	–	–	–	–	Write, Read			
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	IRUN[4:0]				
Reset	–	–	–	0b11111				
Access Type	–	–	–	Write, Read				
BIT	7	6	5	4	3	2	1	0
Field	–	–	–	IHOLD[4:0]				
Reset	–	–	–	0b01000				
Access Type	–	–	–	Write, Read				

ビットフィールド	ビット	説明
IRUNDELAY	27:24	<p>動作が始まる前のモータ・パワーアップに必要なクロック・サイクル数を制御します。滑らかな遷移にすることで、パワーアップ時のモータ・ジャークを回避できます。</p> <p>0：即時パワーアップ 1~15：電流増加ステップあたりの遅延はIRUNDELAY × 512クロックの倍数</p>
IHOLDDELAY	19:16	<p>停止が検出され（stst = 1）TPOWERDOWNの時間が経過すると直ちにモーション後のモータ・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・ジャークを回避できます。</p> <p>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<sup>18</sup>クロックの倍数</p>

ビットフィールド	ビット	説明
IRUN	12:8	モータ稼働時電流 0 = 1/32 1 = 2/32 ~ 31 = 32/32  ヒント：最高のマイクロステップ性能を得るには、16~31の設定値を推奨します。
IHOLD	4:0	停止時電流 0 = 1/32 1 = 2/32 ~ 31 = 32/32  StealthChop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を停止できます。

## TPOWERDOWN (0x13)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	7	6	5	4	3	2	1	0
Field	TPOWERDOWN[7:0]							
Reset	0x0A							
Access Type	Write, Read							

ビットフィールド	ビット	説明
TPOWERDOWN	7:0	TPOWERDOWNは、モータの停止 (stst) 後からモータ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：StealthChopのPWM_OFFS_AUTOを自動調整するには、少なくとも2に設定することが必要です。  リセット後のデフォルト = 10  $(0 \sim ((2^8) - 1)) \times 2^{18} t_{CLK}$

## TSTEP (0x14)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	TSTEP[19:16]			
Reset	–	–	–	–	0x0			
Access Type	–	–	–	–	Read Only			
BIT	15	14	13	12	11	10	9	8
Field	TSTEP[15:8]							
Reset	0x0							
Access Type	Read Only							
BIT	7	6	5	4	3	2	1	0
Field	TSTEP[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
TSTEP	19:0	<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（1/f<sub>CLK</sub>単位）。オーバーフローまたは停止時の測定値は、(2<sup>20</sup>) - 1です。</p> <p>TSTEPに関連するスレッシュホールド（T<sub>xxx</sub>）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。small_hysteresisフラグ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>(T<sub>xxx</sub> × 15/16) - 1または (T<sub>xxx</sub> × 31/32) - 1は、各比較値の2番目の比較値として使用されます。</p> <p>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<sub>I</sub>に対して測定されるTSTEPは次の範囲になります。</p> $(2^{24} / V) \leq \text{TSTEP} \leq 2^{24} / V - 1.$ <p>DcStepモードでは、TSTEPはモータの平均速度を示すのではなく、マイクロステップごとの速度を示します。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、実際のモータ速度を表すわけではありません。</p>

## TPWMTHRS (0x15)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	TPWMTHRS[19:16]			
Reset	–	–	–	–	0x0			
Access Type	–	–	–	–	Write, Read			
BIT	15	14	13	12	11	10	9	8
Field	TPWMTHRS[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	TPWMTHRS[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
TPWMTHRS	19:0	これは、StealthChop2の電圧PWMモードでの上限速度です。 TSTEP ≥ TPWMTHRS <ul style="list-style-type: none"> <li>StealthChop PWMモードが有効化されます（設定されている場合）</li> <li>DcStepは無効化されます</li> </ul>

## TCOOLTHRS (0x16)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	TCOOLTHRS[19:16]			
Reset	–	–	–	–	0x0			
Access Type	–	–	–	–	Write, Read			
BIT	15	14	13	12	11	10	9	8
Field	TCOOLTHRS[15:8]							
Reset	0x0							
Access Type	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	TCOOLTHRS[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
TCOOLTHRS	19:0	<p>これは、スマート・エネルギーCoolStepおよびStallGuard2/4機能をスイッチ・オンするための下限スレッシュホールド速度です（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、速度がこのスレッシュホールド未満になると再び無効化されます。</p> <p>TCOOLTHRS ≥ TSTEP ≥ THIGH :</p> <ul style="list-style-type: none"> <li>• CoolStepがイネーブルされます（設定されている場合）</li> </ul> <p>TCOOLTHRS ≥ TSTEP :</p> <ul style="list-style-type: none"> <li>• 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）</li> <li>• 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）</li> </ul>

#### THIGH (0x17)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	THIGH[19:16]			
Reset	–	–	–	–	0x0			
Access Type	–	–	–	–	Write, Read			
BIT	15	14	13	12	11	10	9	8
Field	THIGH[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	THIGH[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
THIGH	19:0	<p>この速度設定は、トルク（符号なし）を最大にするために、異なるチョッパー・モードおよびフル・ステップ動作への速度に依存した切り替えを可能にします。</p> <p>ストール検出機能は、THIGHスレッシュホールドを通過すると必ず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 ≤ THIGH :</p> <ul style="list-style-type: none"> <li>• CoolStepは無効化されます（モータは通常電流スケールで動作）</li> <li>• StealthChop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TFD = 0（低速減衰のみの定オフ時間）の場合にchm = 1に切り替わります</li> <li>• vhighfsが設定されていると、モータはフル・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</li> </ul>

## XACTUAL (0x18)

BIT	31	30	29	28	27	26	25	24
Field	XACTUAL[31:24]							
Reset	0x0							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	XACTUAL[23:16]							
Reset	0x0							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	XACTUAL[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	XACTUAL[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
XACTUAL	31:0	<p>実際のモータ・ポジション（符号付き）</p> <p><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、XACTUALがXTARGETと異なっており、かつ、モータの動作が可能なランプ・パラメータとなっている場合に、動作を開始します。</p> <p>動作をさせないままポジショニング・モードでXACTUALを調整するには、モータが停止していることを確認し、VMAX = 0およびVSTART = 0に設定してから、XACTUALを変更します。あるいは、ホールド・モードに変更してからXACTUALを変更します。</p>

## VACTUAL (0x19)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	VACTUAL[23:16]							
Reset	0x0							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	VACTUAL[15:8]							
Reset	0x0							
Access Type	Read Only							
BIT	7	6	5	4	3	2	1	0
Field	VACTUAL[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
VACTUAL	23:0	ランプ・ジェネレータによる実際のモータ速度（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]

## AACTUAL (0x1A)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	AACTUAL[23:16]							
Reset	0x000000							
Access Type	Read Only							

BIT	15	14	13	12	11	10	9	8
Field	AACTUAL[15:8]							
Reset	0x000000							
Access Type	Read Only							
BIT	7	6	5	4	3	2	1	0
Field	AACTUAL[7:0]							
Reset	0x000000							
Access Type	Read Only							

ビットフィールド	ビット	説明
AACTUAL	23:0	ランプ・ジェネレータが使用する電流加速 $0 \sim (2^{24}) - 1$ [μsteps/ta <sup>2</sup> ]

**VSTART (0x1B)**

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	VSTART[17:16]	
Reset	–	–	–	–	–	–	0x0	
Access Type	–	–	–	–	–	–	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	VSTART[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	VSTART[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
VSTART	17:0	モータの始動速度（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。 ポジション・モードでのみ使用します。  $0 \sim (2^{18}) - 1$ [μsteps/t]

## A1 (0x1C)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	A1[17:16]	
Reset	–	–	–	–	–	–	0x0	
Access Type	–	–	–	–	–	–	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	A1[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	A1[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
A1	17:0	VSTARTからV1への最初の加速（符号なし） $0 \sim (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ]

## V1 (0x1D)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	V1[19:16]			
Reset	–	–	–	–	0x0			
Access Type	–	–	–	–	Write, Read			
BIT	15	14	13	12	11	10	9	8
Field	V1[15:8]							
Reset	0x0							
Access Type	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	V1[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
V1	19:0	最初の加速/減速フェーズのスレッシュホールド速度（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 $0 \sim (2^{20}) - 1$ [μsteps/t]

**A2 (0x1E)**

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	-	A2[17:16]	
Reset	-	-	-	-	-	-	0b000000000000000000	
Access Type	-	-	-	-	-	-	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	A2[15:8]							
Reset	0b000000000000000000							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	A2[7:0]							
Reset	0b000000000000000000							
Access Type	Write, Read							

ビットフィールド	ビット	説明
A2	17:0	V1からV2への加速（符号なし） $0 \sim (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ]

## V2 (0x1F)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	V2[19:16]			
Reset	–	–	–	–	0x0			
Access Type	–	–	–	–	Write, Read			
BIT	15	14	13	12	11	10	9	8
Field	V2[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	V2[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
V2	19:0	AMAX/2およびDMAX/2で加速セグメントを活性化する場合の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 $0 \sim (2^{20}) - 1$ [μsteps/t]

## AMAX (0x20)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	AMAX[17:16]	
Reset	–	–	–	–	–	–	0x0	
Access Type	–	–	–	–	–	–	Write, Read	

<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	AMAX[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	AMAX[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
AMAX	17:0	V2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。 $0 \sim (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ]

**VMAX (0x21)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	–	–	–	–	–	–	–	–
<b>Reset</b>	–	–	–	–	–	–	–	–
<b>Access Type</b>	–	–	–	–	–	–	–	–
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	–	VMAX[22:16]						
<b>Reset</b>	–	0x0						
<b>Access Type</b>	–	Write, Read						
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	VMAX[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	VMAX[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
VMAX	22:0	<p>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）</p> <p>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。</p> <p><math>0 \sim (2^{23}) - 512</math> [μsteps/t]</p>

**DMAX (0x22)**

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	-	DMAX[17:16]	
Reset	-	-	-	-	-	-	0x0	
Access Type	-	-	-	-	-	-	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	DMAX[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	DMAX[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
DMAX	17:0	<p>VMAXからV2への減速（符号なし）</p> <p><math>0 \sim (2^{18}) - 1</math> [μsteps/ta<sup>2</sup>]</p>

**D2 (0x23)**

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-

BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	D2[17:16]	
Reset	–	–	–	–	–	–	0x10	
Access Type	–	–	–	–	–	–	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	D2[15:8]							
Reset	0x10							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	D2[7:0]							
Reset	0x10							
Access Type	Write, Read							

ビットフィールド	ビット	説明
D2	17:0	V2からV1への減速（符号なし） 注：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 $1 \sim (2^{18}) - 1$ [μsteps/ta <sup>2</sup> ] リセット後のデフォルト= 10

**D1 (0x24)**

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	D1[17:16]	
Reset	–	–	–	–	–	–	0d10	
Access Type	–	–	–	–	–	–	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	D1[15:8]							
Reset	0d10							
Access Type	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	D1[7:0]							
Reset	0d10							
Access Type	Write, Read							

ビットフィールド	ビット	説明
D1	17:0	V1からVSTOPへの減速（符号なし）  注：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  $1 \sim (2^{16}) - 1$ [μsteps/ta <sup>2</sup> ]  リセット後のデフォルト= 10

## VSTOP (0x25)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	VSTOP[17:16]	
Reset	–	–	–	–	–	–	0d10	
Access Type	–	–	–	–	–	–	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	VSTOP[15:8]							
Reset	0d10							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	VSTOP[7:0]							
Reset	0d10							
Access Type	Write, Read							

ビットフィールド	ビット	説明
VSTOP	17:0	モータ停止速度（符号なし）  ヒント：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  注意：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。  $1 \sim (2^{16}) - 1$ [μsteps/t]  リセット後のデフォルト= 10

## TVMAX (0x26)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	TVMAX[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	TVMAX[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
TVMAX	15:0	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 $(0 \sim (2^{16} - 1)) \times 512 t_{CLK}$

## TZEROWAIT (0x27)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–

<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	TZEROWAIT[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	TZEROWAIT[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
TZEROWAIT	15:0	<p>速度ゼロへのランブ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。</p> <p>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。この設定は速度モードでは使用しません。</p> <p><math>(0 \sim (2^{16} - 1)) \times 512 t_{CLK}</math></p>

**XTARGET (0x28)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	XTARGET[31:24]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	XTARGET[23:16]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	XTARGET[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	XTARGET[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
XTARGET	31:0	<p>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活性化するために、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。</p> <p>ポジションはラップ・アラウンド（折り返し）が可能です。そのため、XTARGETの値は符号なしの数値として処理できます。</p> <p>可能な最大変位は<math>\pm(2^{31} - 1)</math>です。</p> <p>動作中にV1、V2、D1、D2、またはDMAXを増加させる場合、必要に応じて、その後に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。</p> <p><math>-2^{31} \sim (+2^{31} - 1)</math></p>

## VDCMIN (0x29)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	VDCMIN[14:8]						
Reset	–	0b0000000000000000						
Access Type	–	Write, Read						
BIT	15	14	13	12	11	10	9	8
Field	VDCMIN[7:0]							
Reset	0b0000000000000000							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	Reserved[7:0]							
Reset	0x00							
Access Type	Read Only							

ビットフィールド	ビット	説明
VDCMIN	22:8	<p>DcStepの自動整流機能は速度VDCMIN（符号なし）より高い場合に有効化されます。 DcStepは、内蔵ランプ・ジェネレータを用いる場合にのみ使用できます。DcStepは、ステップおよび方向モードでは使用できません。</p> <p>このモードでは、実際の位置はセンサーレス・モータ整流によって決定され、XACTUALにフィードバックされます。モータが重負荷になった場合、VDCMINは最小ステップ速度としても用いられます。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）を活性化します。</p> <p>0：ディスエーブル、DcStepがオフ  VACT  ≥ VDCMIN ≥ 256：  <ul style="list-style-type: none"> <li>• THIGHの設定値を超えた場合と同じ動作をトリガします</li> <li>• 自動整流DcStepをスイッチ・オンします</li> </ul> </p> <p>DcStepを動作させるには、忘れずにDCCTRLパラメータも設定してください  (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)</p>
Reserved	7:0	常に0を読み出します

## SW\_MODE (0x2A)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	virtual_Step_enc	en_virtual_stop_r	en_virtual_stop_l	en_softstop	sg_stop	en_latch_encoder	latch_r_inactive
Reset	–	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	–	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	latch_r_active	latch_l_inactive	latch_l_active	swap_lr	pol_stop_r	pol_stop_l	stop_r_enable	stop_l_enable
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明	デコード
virtual_Step_enc	14	仮想停止のソース（VIRTUAL_STOP_LおよびVIRTUAL_STOP_R）	0x0: Virtual stop relates to ramp generator position XACTUAL 0x1: Virtual stop relates to encoder position X_ENC

ビットフィールド	ビット	説明	デコード
en_virtual_stop_r	13	右仮想ストップ条件のアクティブ時に自動モータ停止	0x0: Disabled 0x1: Enabled
en_virtual_stop_l	12	左仮想ストップ条件のアクティブ時に自動モータ停止をイネーブル	0x0: Disabled 0x1: Enabled
en_softstop	11	ソフトストップ・モードでは、常にD <sub>MAX</sub> 、V <sub>1</sub> 、D <sub>1</sub> 、V <sub>2</sub> 、D <sub>2</sub> 、V <sub>STOP</sub> 、T <sub>ZEROWAIT</sub> の減速ランプ設定を用いてモータを停止させます。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V <sub>IRTUAL_STOP_L</sub> 、正の速度の場合はREFRおよびV <sub>IRTUAL_STOP_R</sub> ）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、モータが解放される前にT <sub>ZEROWAIT</sub> を用います。  <b>注意事項</b> ：ソフトストップは、StallGuard2/4と組み合わせて用いないでください。 <b>高速度でのStealthChop2動作にはソフトストップを用いてください。</b> 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。	0x0: Hard stop 0x1: Soft stop
sg_stop	10	StallGuard2/4により停止をイネーブルします。ディスエーブルすると停止イベント後にモータを解放します。速度スレッシュホールドにはT <sub>COOLTHRS</sub> をプログラムします。  モータの回転上昇中にはイネーブルしないでください。StallGuard2/4が安定した結果を供給できる一定の値をモータ速度が超えるまで待ってください。この速度スレッシュホールドは、T <sub>COOLTHRS</sub> を用いてプログラムする必要があります。	0x0: Disabled 0x1: Enabled
en_latch_encoder	9	リファレンス切り替えイベント時にエンコーダのポジションをENC_LATCHにラッチします。	0x0: Disabled 0x1: Enabled
latch_r_inactive	8	右リファレンス・スイッチ入力REFRのインアクティブ・エッジで、XLATCHへのポジションのラッチを活性化します。活性化レベルはpol_stop_rで定義されます。	0x0: Disabled 0x1: Enabled
latch_r_active	7	右リファレンス・スイッチ入力REFRのアクティブ・エッジで、XLATCHへの位置のラッチを活性化します。  status_latch_rを読み出すことで誤停止イベントを検出するには、latch_r_activeを活性化します。	0x0: Disabled 0x1: Enabled
latch_l_inactive	6	左リファレンス・スイッチ入力REFLのインアクティブ・エッジで、XLATCHへのポジションのラッチを活性化します。活性化レベルはpol_stop_lで定義されます。	0x0: Disabled 0x1: Enabled
latch_l_active	5	左リファレンス・スイッチ入力REFLのアクティブ・エッジで、XLATCHへの位置のラッチを活性化します。  status_latch_lを読み出すことで誤停止イベントを検出するには、latch_l_activeを活性化します。	0x0: Disabled 0x1: Enabled
swap_lr	4	左リファレンス・スイッチ入力REFLと右リファレンス・スイッチ入力REFRを入れ替えます。	0x0: Default assignments 0x1: REFL and REFR switched

ビットフィールド	ビット	説明	デコード
pol_stop_r	3	右リファレンス・スイッチ入力のアクティブな極性を設定します。	0x0: Not inverted, active high: a high level on REFR stops the motor 0x1: Inverted, active low: a low level on REFR stops the motor
pol_stop_l	2	左リファレンス・スイッチ入力のアクティブな極性を設定します。	0x0: Non-inverted, active high: a high level on REFL stops the motor 0x1: Inverted, active low: a low level on REFL stops the motor
stop_r_enable	1	右リファレンス・スイッチ入力が入力がアクティブとなっている間の自動モータ停止をイネーブルします。  停止スイッチが解放されるとモータは再始動します。	0x0: Disabled 0x1: Enabled
stop_l_enable	0	左リファレンス・スイッチ入力が入力がアクティブとなっている間の自動モータ停止をイネーブルします。  停止スイッチが解放されるとモータは再始動します。	0x0: Disabled 0x1: Enabled

## RAMP\_STAT (0x2B)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	status_virtual_stop_r	status_virtual_stop_l	status_sg	second_move	t_zerowait_active	vzero	position_reached	velocity_reached
Reset	0x1	0x1	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Read Only	Read Only	Read Only	Write 1 to Clear, Read	Read Only	Read Only	Read Only	Read Only
BIT	7	6	5	4	3	2	1	0
Field	event_pos_reached	event_stop_sg	event_stop_r	event_stop_l	status_latch_r	status_latch_l	status_stop_r	status_stop_l
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write 1 to Clear, Read	Write 1 to Clear, Read	Read Only	Read Only	Write 1 to Clear, Read	Write 1 to Clear, Read	Read Only	Read Only

ビットフィールド	ビット	説明
status_virtual_stop_r	15	右仮想リファレンス・スイッチのステータス (1 = アクティブ)
status_virtual_stop_l	14	左仮想リファレンス・スイッチのステータス (1 = アクティブ)

ビットフィールド	ビット	説明
status_sg	13	1: ドライバからのアクティブなStallGuard2/StallGuard4への入力が行われていることを指示します (イネーブルされている場合)。  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を活性化してください。
second_move	12	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。  クリアするには1を書き込みます。
t_zerowait_active	11	1: モータの停止後TZEROWAITがアクティブであることを指示します。この時間、モータは停止状態です。
vzero	10	1: 実際の速度が0であることを指示します。
position_reached	9	1: 目標ポジションに達していることを指示します。 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。
velocity_reached	8	1: 目標速度に達していることを指示します。 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。
event_pos_reached	7	1: 目標ポジションに達したことを指示します (position_reachedがアクティブになります)。  フラグおよび割込み条件をクリアするには1を書き込みます。  このビットは、割込み出力信号とOR接続されます。
event_stop_sg	6	1: アクティブなStallGuard2/4停止イベントを指示します。 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、動作を再開できます。  フラグおよび割込み条件をクリアするには1を書き込みます。  このビットは、割込み出力信号とOR接続されます。
event_stop_r	5	1: 停止スイッチまたは仮想停止により右ストップ条件がアクティブ。 ストップ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すが、モータは動作を続けます。  このビットは、割込み出力信号とOR接続されます。
event_stop_l	4	1: 停止スイッチまたは仮想停止により左ストップ条件がアクティブ。 ストップ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すが、モータは動作を続けます。  このビットは、割込み出力信号とOR接続されます。
status_latch_r	3	1: 右ラッチ・レディ (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)。  クリアするには1を書き込みます。
status_latch_l	2	1: 左ラッチ・レディ (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)。  クリアするには1を書き込みます。
status_stop_r	1	右リファレンス・スイッチのステータス (1=アクティブ)
status_stop_l	0	左リファレンス・スイッチのステータス (1=アクティブ)

## XLATCH (0x2C)

BIT	31	30	29	28	27	26	25	24
Field	XLATCH[31:24]							
Reset	0x0							
Access Type	Read Only							
BIT	23	22	21	20	19	18	17	16
Field	XLATCH[23:16]							
Reset	0x0							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	XLATCH[15:8]							
Reset	0x0							
Access Type	Read Only							
BIT	7	6	5	4	3	2	1	0
Field	XLATCH[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
XLATCH	31:0	ランプ・ジェネレータのラッチ・ポジション。プログラマブルな切り替えイベント時にXACTUALをラッチします ( <a href="#">SW_MODE</a> の説明を参照)。  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_LATCHIにラッチされる場合もあります。

## POSITION\_P\_CTRL (0x2D)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	en_tol_on_pos_reached	–	–	–	–
Reset	–	–	–	0x1	–	–	–	–
Access Type	–	–	–	Write, Read	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	tolerance[7:0]							
Reset	0x0							
Access Type	Write, Read							

<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	–	–	–	–	–	–	P[9:8]	
<b>Reset</b>	–	–	–	–	–	–	0x0	
<b>Access Type</b>	–	–	–	–	–	–	Write, Read	
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	P[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明	デコード
en_tol_on_pos_reached	28	モーション・コントローラが目標ポジションに達した後にtoleranceを有効化します	0x0: The P-controller ignores absolute errors below the tolerance 0x1: The P-controller ignores absolute errors below tolerance only when the pos_reached flag (RAMP_STAT[9]) of the motion controller is set to 1
tolerance	23:16	Pコントローラの許容誤差の設定 tolerance未満の誤差は無視されます	
P	9:0	ポジションプレギュレータの比例パラメータ  P = 0の場合、プレギュレータ機能は無効化されます  P > 0の場合、プレギュレータ機能は有効化されます  UINT10.0、端数はなし	

**X\_ENC (0x2E)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	X_ENC[31:24]							
<b>Reset</b>	0x00000000							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	X_ENC[23:16]							
<b>Reset</b>	0x00000000							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	X_ENC[15:8]							
<b>Reset</b>	0x00000000							
<b>Access Type</b>	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	X_ENC[7:0]							
Reset	0x00000000							
Access Type	Write, Read							

ビットフィールド	ビット	説明
X_ENC	31:0	実際のエンコーダ・ポジション（符号付き）

## ENCMODE (0x2F)

BIT	31	30	29	28	27	26	25	24
Field	–	–	BEMF_FILTER_SEL[1:0]		–	–	–	–
Reset	–	–	0x0		–	–	–	–
Access Type	–	–	Write, Read		–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	BEMF_BLANK_TIME[7:0]							
Reset	0x10							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	qsc_enc_en	bemf_hyst[2:0]			nBEMF_AB_N_SEL	enc_sel_decimal	latch_x_act	clr_enc_x
Reset	0x1	0x0			0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read			Write, Read	Write, Read	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	pos_neg_edge[1:0]		clr_once	clr_cont	ignore_AB	pol_N	pol_B	pol_A
Reset	0x0		0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read		Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

ビットフィールド	ビット	説明	デコード
BEMF_FILTER_SEL	29:28	デコーダの逆起電力信号をフィルタにより除去するためのデジタル・ローパス・フィルタ	0x0: $f_{CTOFF} = f_{CLK}/24600$ 0x1: $f_{CTOFF} = f_{CLK}/12300$ 0x2: $f_{CTOFF} = f_{CLK}/6150$ 0x3: $f_{CTOFF} = f_{CLK}/2870$
BEMF_BLANK_TIME	23:16	ブランキング時間は、 $BEMF\_BLANK\_TIME \times ((2^{14}) - 1) \times t_{CLK}$	
qsc_enc_en	15	自己消費モードの間のエンコーダ・イネーブル設定	0x0: Incremental decoder/back EMF decoder is inactive during quiescence to reduce power consumption 0x1: Incremental decoder/back EMF decoder is active during quiescence (default)

ビットフィールド	ビット	説明	デコード
bemf_hyst	14:12	BEMFエンコーダ・コンパレータのヒステリシスを設定	0x0: 10mV 0x1: 25mV 0x2: 50mV 0x3: 75mV 0x4: 100mV 0x5: 150mV 0x6: 200mV 0x7: 250mV
nBEMF_ABN_SEL	11	エンコーダ・ソース・セレクタ	0x0: Back-EMF full-step encoder 0x1: Incremental encoder interface
enc_sel_decimal	10	エンコーダ・プリスケアラのモード選択	0x0: Encoder prescaler divisor binary mode: Counts in ENC_CONST (fractional part)/65536 0x1: Encoder prescaler divisor decimal mode: Counts in ENC_CONST (fractional part)/10000
latch_x_act	9	ポジション・ラッチの設定	0x0: Disabled 0x1: Also latch XACTUAL position together with X_ENC. Allows latching the ramp generator position upon an N channel event as selected by pos_edge and neg_edge.
clr_enc_x	8	エンコーダ・ラッチの設定	0x0: Upon N event, X_ENC becomes latched to ENC_LATCH only 0x1: Latch and additionally clear encoder counter X_ENC at N event
pos_neg_edge	7:6	Nチャンネル・イベントの感度	0x0: N channel event is active during an active N event level 0x1: N channel is valid upon active N event 0x2: N channel is valid upon inactive N event 0x3: N channel is valid upon active and inactive N event
clr_once	5	ポジション・ラッチの設定	0x0: Disabled 0x1: Latch or latch and clear X_ENC on the next N event following the write access
clr_cont	4	ポジション・ラッチの設定	0x0: Disabled 0x1: Always latch or latch and clear X_ENC upon an N event. (Once per revolution, it is recommended to combine this setting with edge sensitive N event.)
ignore_AB	3	Nイベントの設定	0x0: An N event occurs only when polarities given by pol_N, pol_A, and pol_B match 0x1: Ignore A and B polarity for N channel event
pol_N	2	Nのアクティブな極性を定義	0x0: Low active 0x1: High active
pol_B	1	Nチャンネル・イベントに必要なBの極性	0x0: Negative 0x1: Positive
pol_A	0	Nチャンネル・イベントに必要なAの極性	0x0: Negative 0x1: Positive

## ENC\_CONST (0x30)

BIT	31	30	29	28	27	26	25	24
Field	ENC_CONST[31:24]							
Reset	0x00010000							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	ENC_CONST[23:16]							
Reset	0x00010000							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	ENC_CONST[15:8]							
Reset	0x00010000							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	ENC_CONST[7:0]							
Reset	0x00010000							
Access Type	Write, Read							

ビットフィールド	ビット	説明
ENC_CONST	31:0	<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)</p> <p>10進数： ±(0.0~32767.9999)</p> <p>リセット後のデフォルト = 1.0 (= 65536)</p>

## ENC\_STATUS (0x31)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–

BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	7	6	5	4	3	2	1	0
Field	–	–	–	–	–	–	deviation_warn	n_event
Reset	–	–	–	–	–	–	0x0	0x0
Access Type	–	–	–	–	–	–	Write 1 to Clear, Read	Write 1 to Clear, Read

ビットフィールド	ビット	説明	デコード
deviation_warn	1	エンコーダ偏差警告フラグ	0x0: No warning 0x1: Deviation warning. Cannot be cleared while a warning still persists. Set ENC_DEVIATION to zero to disable.
n_event	0	エンコーダNイベント・フラグ クリアするには1を書き込みます。	0x0: No event 0x1: Event detected.

**ENC\_LATCH (0x32)**

BIT	31	30	29	28	27	26	25	24
Field	ENC_LATCH[31:24]							
Reset	0x0							
Access Type	Read Only							
BIT	23	22	21	20	19	18	17	16
Field	ENC_LATCH[23:16]							
Reset	0x0							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	ENC_LATCH[15:8]							
Reset	0x0							
Access Type	Read Only							

BIT	7	6	5	4	3	2	1	0
Field	ENC_LATCH[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
ENC_LATCH	31:0	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

**ENC\_DEVIATION (0x33)**

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	ENC_DEVIATION[19:16]			
Reset	-	-	-	-	0x0			
Access Type	-	-	-	-	Write, Read			
BIT	15	14	13	12	11	10	9	8
Field	ENC_DEVIATION[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	ENC_DEVIATION[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
ENC_DEVIATION	19:0	偏差警告の原因となる、エンコーダ・カウンタとXACTUALのステップ差の最大値。 BEMFエンコーダ機能の場合、偏差はXACTUALではなく0と比較されます。結果はENC_STATUSレジスタのdeviation_warnフラグに現れます。 0 = この機能をオフ。

**VIRTUAL\_STOP\_L (0x34)**

BIT	31	30	29	28	27	26	25	24
Field	VIRTUAL_STOP_L[31:24]							
Reset	0x0							
Access Type	Write, Read							

BIT	23	22	21	20	19	18	17	16
Field	VIRTUAL_STOP_L[23:16]							
Reset	0x0							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	VIRTUAL_STOP_L[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	VIRTUAL_STOP_L[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
VIRTUAL_STOP_L	31:0	<p>エンコーダまたは内部のポジション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。</p> <p>virtual_stop_enc = 1: X_ENC ≤ VIRTUAL_STOP_L</p> <p>virtual_stop_enc = 0: XACTUAL ≤ VIRTUAL_STOP_L</p> <p>-2<sup>31</sup> ~ (+2<sup>31</sup> - 1)</p>

## VIRTUAL\_STOP\_R (0x35)

BIT	31	30	29	28	27	26	25	24
Field	VIRTUAL_STOP_R[31:24]							
Reset	0x0							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	VIRTUAL_STOP_R[23:16]							
Reset	0x0							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	VIRTUAL_STOP_R[15:8]							
Reset	0x0							
Access Type	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	VIRTUAL_STOP_R[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビットフィールド	ビット	説明
VIRTUAL_STOP_R	31:0	<p>エンコーダまたは内部のポジション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。</p> <p>virtual_stop_enc = 1: X_ENC ≥ VIRTUAL_STOP_R</p> <p>virtual_stop_enc = 0: XACTUAL ≥ VIRTUAL_STOP_R</p> <p>-2<sup>31</sup>~(+2<sup>31</sup> - 1)</p>

## MSCNT (0x36)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	MSCNT[9:8]	
Reset	-	-	-	-	-	-	0b0000000000	
Access Type	-	-	-	-	-	-	Read Only	
BIT	7	6	5	4	3	2	1	0
Field	MSCNT[7:0]							
Reset	0b0000000000							
Access Type	Read Only							

ビットフィールド	ビット	説明
MSCNT	9:0	<p>マイクロステップ・カウンタ。実際のポジションをCUR_Bのマイクロステップ・テーブルに示します。CUR_Aは256のオフセットを使用します（2相モータ）。</p> <p>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。</p>

## MSCURACT (0x37)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	CUR_B[8]
Reset	–	–	–	–	–	–	–	0x0
Access Type	–	–	–	–	–	–	–	Read Only
BIT	23	22	21	20	19	18	17	16
Field	CUR_B[7:0]							
Reset	0x0							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	–	CUR_A[8]
Reset	–	–	–	–	–	–	–	0xF7
Access Type	–	–	–	–	–	–	–	Read Only
BIT	7	6	5	4	3	2	1	0
Field	CUR_A[7:0]							
Reset	0xF7							
Access Type	Read Only							

ビットフィールド	ビット	説明
CUR_B	24:16	MSLUTから読み出した、モータ位相B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）。
CUR_A	8:0	MSLUTから読み出した、モータ位相A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）。

## CHOPCONF (0x38)

BIT	31	30	29	28	27	26	25	24
Field	diss2vs	diss2g	dedge	intpol	MRES[3:0]			
Reset	0x0	0x0	0x0	0x1	0x0			
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	TPFD[3:0]				vhighchm	vhighfs	–	TBL[1]
Reset	0x4				0x0	0x0	–	0b10
Access Type	Write, Read				Write, Read	Write, Read	–	Write, Read
BIT	15	14	13	12	11	10	9	8
Field	TBL[0]	chm	–	disfdcc	fd3	HEND_OFFSET[3:1]		
Reset	0b10	0x0	–	0x0	0x0	0x2		
Access Type	Write, Read	Write, Read	–	Write, Read	Write, Read	Write, Read		

BIT	7	6	5	4	3	2	1	0
Field	HEND_OFF SET[0]	HSTRT_TFD210[2:0]			TOFF[3:0]			
Reset	0x2	0b101			0x0			
Access Type	Write, Read	Write, Read			Write, Read			

ビットフィールド	ビット	説明	デコード
diss2vs	31	電源への短絡からの保護を無効化	0x0: Short to V <sub>S</sub> protection is on 0x1: Short to V <sub>S</sub> protection is disabled
diss2g	30	GNDへの短絡からの保護を無効化	0x0: Short to GND protection is on 0x1: Short to GND protection is disabled
dedge	29	ダブル・エッジ・ステップ・パルスを有効化	0x0: Disabled 0x1: Enable step impulse at each step edge to reduce step frequency requirement
intpol	28	256マイクロステップへの補間	0x0: No interpolation 0x1: The actual microstep resolution (MRES) becomes extrapolated to 256 microsteps for smoothest motor operation (useful for STEP/DIR operation only)
MRES	27:24	マイクロステップ分解能の選択  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 <sup>MRES</sup> [microsteps]	0x0: Native 256 microstep setting. Normally use this setting with the internal motion controller. 0x1: 128 0x2: 64 0x3: 32 0x4: 16 0x5: 8 0x6: 4 0x7: 2, half step 0x8: 1, full step 0x9: Unused 0xA: Unused 0xB: Unused 0xC: Unused 0xD: Unused 0xE: Unused 0xF: Unused
TPFD	23:20	受動高速減衰時間  TPFDを用いることで、モータ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 <sub>CLK</sub> = 128 × TPFD %0000 : 無効 %0001~%1111 : 1~15	
vhighchm	19	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。	

ビットフィールド	ビット	説明	デコード
vhighfs	18	高速フル・ステップの選択  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°のポジションでのみ生じます。フル・ステップの目標電流は、45°ポジション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。	
TBL	16:15	TBL/ブランキング時間選択  %00~%11 : コンパレータのブランキング時間を16、24、36、または54クロックに設定します  ほとんどのアプリケーションには%01または%10を推奨します	0x0: 16 0x1: 24 0x2: 36 0x3: 54
chm	14	チョッパー・モード	0x0: Standard mode (SpreadCycle) 0x1: Constant off time with fast decay time. Fast decay time is also terminated when the negative nominal current is reached. Fast decay is after on time.
disfdcc	12	高速減衰モードの設定 (chm = 1)	0x0: Enables current comparator usage for termination of the fast decay cycle 0x1: Disables current comparator usage for termination of the fast decay cycle
fd3	11	TFD[3]  chm = 1の場合 : 高速減衰時間設定TFDのMSB	
HEND_OFFSET	10:7	chm = 0の場合 : HENDヒステリシスの低値。  %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が各サイン波エントリの絶対値に加算されます。	

ビットフィールド	ビット	説明	デコード
HSTRT_TFD210	6:4	<p>chm = 0の場合：HENDに加算されるヒステリシス開始値HSTRT。 %000~%111：ヒステリシスの低値HENDに1~8を加算します（この設定値の1/512が電流設定値に加算されます） 注：実効HEND + HSTRT ≤ 16。</p> <p>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 (MSB: fd3)：%0000~%1111：NCLK = 32 × TFDの高速減衰時間設定TFD（%0000：低速減衰のみ）</p>	
TOFF	3:0	<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>	

## COOLCONF (0x39)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	sflt
Reset	–	–	–	–	–	–	–	0x0
Access Type	–	–	–	–	–	–	–	Write, Read
BIT	23	22	21	20	19	18	17	16
Field	–	sgt[6:0]						
Reset	–	0x0						
Access Type	–	Write, Read						
BIT	15	14	13	12	11	10	9	8
Field	seimin	sedn[1:0]		–	semax[3:0]			
Reset	0x0	0x0		–	0x0			
Access Type	Write, Read	Write, Read		–	Write, Read			
BIT	7	6	5	4	3	2	1	0
Field	–	seup[1:0]		–	semin[3:0]			
Reset	–	0x0		–	0x0			
Access Type	–	Write, Read		–	Write, Read			

ビットフィールド	ビット	説明	デコード
sfilt	24	StallGuard2フィルタをイネーブ。ル。	0x0: Standard mode, high time resolution for StallGuard2 0x1: Filtered mode, StallGuard2 signal updated for each four full steps only to compensate for motor pole tolerances
sgt	22:16	StallGuard2のスレッシュヨ。ルド値  この符号付きの値は、ストール出力用の StallGuard2のレベルを制御し、読出し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は大きくなります。	
seimin	15	スマート電流制御用の最小電流	0x0: 1/2 of current setting (IRUN) (when used with StealthChop2 requires IRUN ≥ 16) 0x1: 1/4 of current setting (IRUN) (when used with StealthChop2 requires IRUN ≥ 28)
sedn	14:13	電流のダウン・ステップ速度	0x0: For each 32 StallGuard2/4 values, decrease by one 0x1: For each 8 StallGuard2/4 values, decrease by one 0x2: For each 2 StallGuard2/4 values, decrease by one 0x3: For each StallGuard2/4 values, decrease by one
semax	11:8	スマート電流制御用の StallGuard2/4 ヒステリシス値  StallGuard2/4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電流が減少します。 %0000~%1111 : 0~15	
seup	6:5	電流アップ・ステップ幅。StallGuard2/4の測定値ごとの電流インクリメント・ステップ	0x0: 1 0x1: 2 0x2: 4 0x3: 8
semin	3:0	スマート電流制御およびスマート電流イネーブ。ル用の最小の StallGuard2/4 値  StallGuard2/4の結果が SEMIN × 32未満になった場合、モータ負荷角を減少するためにモータ電流が増加します。 %0000 : スマート電流制御 CoolStep をオフ %0001~%1111 : 1~15	

## DCCTRL (0x3A)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-

<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	DC_SG[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	–	–	–	–	–	–	DC_TIME[9:8]	
<b>Reset</b>	–	–	–	–	–	–	0x0	
<b>Access Type</b>	–	–	–	–	–	–	Write, Read	
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	DC_TIME[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
DC_SG	23:16	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。
DC_TIME	9:0	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。

**DRV\_STATUS (0x3B)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	stst	olb	ola	s2gb	s2ga	otpw	ot	stallguard
<b>Reset</b>	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
<b>Access Type</b>	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	–	–	–	CS_ACTUAL[4:0]				
<b>Reset</b>	–	–	–	0x0				
<b>Access Type</b>	–	–	–	Read Only				
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	fsactive	stealth	s2vsb	s2vsa	–	–	SG_RESULT[9:8]	
<b>Reset</b>	0x0	0x0	0x0	0x0	–	–	0x0	
<b>Access Type</b>	Read Only	Read Only	Read Only	Read Only	–	–	Read Only	
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	SG_RESULT[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Read Only							

ビットフィールド	ビット	説明	デコード
stst	31	停止インジケータ  このフラグは、各動作モードでモータが停止していることを示します。これは、最後のステップ・パルスの後、2 <sup>13</sup> クロック・サイクル~2 <sup>20</sup> クロック・サイクルの間の停止タイマー設定に応じて生じます。	0x0: Motor moving 0x1: Motor in standstill
olb	30	位相Bの開放負荷インジケータ	0x0: Normal operation 0x1: Open load detected on phase B  This is only an informative flag. The driver takes no action based on it. False detection may occur in fast motion and standstill. Check during slow motion only.
ola	29	位相Aの開放負荷インジケータ	0x0: Normal operation 0x1: Open load detected on phase A  This is only an informative flag. The driver takes no action based on it. False detection may occur in fast motion and standstill. Check during slow motion only.
s2gb	28	位相Bの地絡インジケータ	0x0: Normal operation 0x1: Short to GND detected on phase B. The driver becomes disabled. The flags stay active until the driver is disabled by software (TOFF = 0 or drv_enn = 1).
s2ga	27	位相Aの地絡インジケータ	0x0: Normal operation 0x1: Short to GND detected on phase A. The driver becomes disabled. The flags stay active until the driver is disabled by software (TOFF = 0 or drv_enn = 1).
otpw	26	過熱事前警告フラグ	0x0: Normal operation 0x1: Overtemperature prewarning threshold is exceeded. The overtemperature prewarning flag is common for both bridges.
ot	25	過熱フラグ	0x0: Normal operation 0x1: Overtemperature limit has been reached. Drivers become disabled until otpw is also cleared due to cooling down of the IC. The overtemperature flag is common for both bridges.
stallguard	24	StallGuard2/4のステータス	0x0: Normal operation 0x1: Motor stall detected (SG_RESULT = 0) or DcStep stall in DcStep mode
CS_ACTUAL	20:16	実際のモータ電流/スマート・エネルギー電流  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リングをモニタするための、あるいは自動電流スケーリングの機能をモニタするための、実際の電流制御スケーリング	
fsactive	15	フル・ステップ・アクティブ・インジケータ	0x0: Microstepping active 0x1: Indicates that the driver has switched to full step as defined by chopper mode settings and velocity thresholds

ビットフィールド	ビット	説明	デコード
stealth	14	StealthChop2インジケータ	0x0: StealthChop2 not active 0x1: Driver operates in StealthChop2 mode
s2vsb	13	位相Bの電源への短絡インジケータ	0x0: No error 0x1: Short to supply detected on phase B. The driver becomes disabled. The flags stay active until the driver is disabled by software (TOFF = 0 or drv_enn = 1).
s2vsa	12	位相Aの電源への短絡インジケータ	0x0: No error 0x1: Short to supply detected on phase A. The driver becomes disabled. The flags stay active until the driver is disabled by software (TOFF = 0 or drv_enn = 1).
SG_RESULT	9:0	<p>モータ温度検出のための、停止状態のコイルAのPWMオン時間に関する StallGuard2/4の結果。</p> <p>機械的負荷測定： StallGuard2/4の結果は、機械的なモータ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。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・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>StallGuard2/4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>温度測定： 停止時には、StallGuard2/4の結果は取得できません。代わりに、SG_RESULTは、モータ・コイルAのチョッパ・オン時間を示します。モータを定められたマイクロステップ・ポジションに向けて特定の電流設定値で動作させると、チョッパ・オン時間を読み出すことでモータ温度を大まかに推定できます。モータ温度が上昇するにつれ、コイル抵抗が増し、チョッパ・オン時間も増加します。</p>	

## PWMCONF (0x3C)

BIT	31	30	29	28	27	26	25	24
Field	PWM_LIM[3:0]				PWM_REG[3:0]			
Reset	0xC				0x4			
Access Type	Write, Read				Write, Read			
BIT	23	22	21	20	19	18	17	16
Field	pwm_dis_re g_stst	pwm_meas _sd_enable	FREEWHEEL[1:0]		pwm_autogr ad	pwm_autos cale	PWM_FREQ[1:0]	
Reset	0b0	0b0	0x0		0x1	0x1	0x0	
Access Type	Write, Read	Write, Read	Write, Read		Write, Read	Write, Read	Write, Read	

BIT	15	14	13	12	11	10	9	8
Field	PWM_GRAD[7:0]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	PWM_OFS[7:0]							
Reset	0x1D							
Access Type	Write, Read							

ビットフィールド	ビット	説明	デコード
PWM_LIM	31:28	スイッチ・オン時のPWM自動スケール振幅制限。  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。 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)	
PWM_REG	27:24	レギュレーション・ループの勾配。  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。	0x0: Do not use 0x1: 0.5 increments (slowest regulation) 0x2: 1 increment 0x3: 1.5 increments 0x4: 2 increments (reset default) 0x5: 2.5 increments 0x6: 3 increments 0x7: 3.5 increments 0x8: 4 increments 0x9: 4.5 increments 0xA: 5 increments 0xB: 5.5 increments 0xC: 6 increments 0xD: 6.5 increments 0xE: 7 increments 0xF: 7.5 increments (fastest regulation)
pwm_dis_reg_sts	23	停止時電流安定化制御	0x0: Current regulation active 0x1: Disable current regulation when motor is in standstill and current is reduced (less than IRUN). This option eliminates any regulation noise during standstill.
pwm_meas_sd_enable	22	低速減衰フェーズのローサイド電流測定制御	0x0: Slow decay low-side measurement disabled (default) 0x1: Uses slow decay phases on low side to measure the motor current to reduce the lower current limit
FREEWHEEL	21:20	停止時のモード制御。モータ電流設定値がゼロ (I_HOLD = 0) の場合の停止オプション。	0x0: Normal operation 0x1: Freewheeling 0x2: Coil shorted using LS drivers 0x3: Coil shorted using HS drivers

ビットフィールド	ビット	説明	デコード
pwm_autograd	19	PWM自動勾配調整	<p>0x0: Fixed value for PWM_GRAD (PWM_GRAD_AUTO = PWM_GRAD)</p> <p>0x1: Automatic tuning (only with pwm_autoscale = 10 (reset default))</p> <p>PWM_GRAD_AUTO is initialized with PWM_GRAD while pwm_autograd = 0 and becomes optimized automatically during motion. Preconditions:</p> <ol style="list-style-type: none"> <li>1. PWM_OFS_AUTO has been automatically initialized. This requires standstill at IRUN for &gt; 130ms to a) detect standstill b) wait &gt; 128 chopper cycles at IRUN and c) regulate PWM_OFS_AUTO so that <math>-1 &lt; \text{PWM\_SCALE\_AUTO} &lt; 1</math></li> <li>2. Motor running and <math>1.5 \times \text{PWM\_OFS\_AUTO} &lt; \text{PWM\_SCALE\_SUM} &lt; 4 \times \text{PWM\_OFS\_AUTO}</math> and <math>\text{PWM\_SCALE\_SUM} &lt; 255</math></li> </ol> <p>Time required for tuning PWM_GRAD_AUTO: Approximately 8 full steps per change of <math>\pm 1</math>.</p> <p>Also enables use of reduced chopper frequency for tuning PWM_OFS_AUTO.</p>
pwm_autoscale	18	PWM自動振幅スケーリング	<p>0x0: User-defined feed-forward PWM amplitude. The current settings IRUN and IHOLD have no influence. The resulting PWM amplitude (limited to 0 to 255) is:  <math>\text{PWM\_OFS} \times ((\text{CS\_ACTUAL} + 1)/32) + \text{PWM\_GRAD} \times 256/\text{TSTEP}</math></p> <p>0x1: Enable automatic current control (reset default)</p>
PWM_FREQ	17:16	PWM周波数の選択	<p>0x0: fPWM = 2/1024 fCLK (reset default)</p> <p>0x1: fPWM = 2/683 fCLK</p> <p>0x2: fPWM = 2/512 fCLK</p> <p>0x3: fPWM = 2/410 fCLK</p>
PWM_GRAD	15:8	<p>PWM振幅の速度依存勾配：  <math>\text{PWM\_GRAD} \times 256/\text{TSTEP}</math>  この値は、速度に依存したモータ逆起電力を補償するためにPWM_AMPL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。その後にのみ、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>初期調整の後、必要な初期値は PWM_GRAD_AUTOで読み出せます。</p>	

ビットフィールド	ビット	説明	デコード
PWM_OFS	7:0	<p>停止時のモータのフル電流（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）。</p> <p>自動スケ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、モータ電流をモータ固有の下限測定スレッショルド未満にスケリング・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、つまり、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のみ、用いてください。それにより、モータ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・スレッショルド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>	

## PWM\_SCALE (0x3D)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	PWM_SCALE_AUTO[8]
Reset	-	-	-	-	-	-	-	0b00000000
Access Type	-	-	-	-	-	-	-	Read Only
BIT	23	22	21	20	19	18	17	16
Field	PWM_SCALE_AUTO[7:0]							
Reset	0b000000000							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	PWM_SCALE_SUM[9:8]	
Reset	-	-	-	-	-	-	0b0000000000	
Access Type	-	-	-	-	-	-	Read Only	

BIT	7	6	5	4	3	2	1	0
Field	PWM_SCALE_SUM[7:0]							
Reset	0b0000000000							
Access Type	Read Only							

ビットフィールド	ビット	説明
PWM_SCALE_AUTO	24:16	自動的に決定される電流スケーリング値
PWM_SCALE_SUM	9:0	[0~1023 : 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、デューティサイクルの読出しの場合、分解能が高くなります。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。

## PWM\_AUTO (0x3E)

BIT	31	30	29	28	27	26	25	24
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
Field	PWM_GRAD_AUTO[7:0]							
Reset	0x00							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	-	-	-	-	-	-	-	-
Reset	-	-	-	-	-	-	-	-
Access Type	-	-	-	-	-	-	-	-
BIT	7	6	5	4	3	2	1	0
Field	PWM_OFS_AUTO[7:0]							
Reset	0x0							
Access Type	Read Only							

ビットフィールド	ビット	説明
PWM_GRAD_AUTO	23:16	自動的に決定される勾配値
PWM_OFS_AUTO	7:0	自動的に決定されるオフセット値

## SG4\_THRS (0x3F)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	SG_ANGLE_OFFSET	SG4_filt_en
Reset	–	–	–	–	–	–	0b1	0b0
Access Type	–	–	–	–	–	–	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	SG4_THRS[7:0]							
Reset	0x00							
Access Type	Write, Read							

ビットフィールド	ビット	説明	デコード
SG_ANGLE_OFFSET	9	TPWMTHRSを介した制御により StealthChop2からSpreadCycleに切り替える際の、StallGuard4に基づく自動位相シフト補償の制御	0x0: Disabled 0x1: Enabled
SG4_filt_en	8	SG4フィルタの制御	0x0: Disable SG4 filter 0x1: Enable SG4 filter
SG4_THRS	7:0	ストールの検出スレッシュホールド。StallGuard4値SG4_RESULTがこのスレッシュホールドと比較されます。ストールは、SG_RESULT ≤ SG4_THRSの場合に示されます。	

## SG4\_RESULT (0x40)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–

<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	–	–	–	–	–	–	–	–
<b>Reset</b>	–	–	–	–	–	–	–	–
<b>Access Type</b>	–	–	–	–	–	–	–	–
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	–	–	–	–	–	–	SG4_RESULT[9:8]	
<b>Reset</b>	–	–	–	–	–	–	0x0	
<b>Access Type</b>	–	–	–	–	–	–	Read Only	
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	SG4_RESULT[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Read Only							

ビットフィールド	ビット	説明
SG4_RESULT	9:0	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。

**SG4\_IND (0x41)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	sg4_ind_3[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	sg4_ind_2[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	sg4_ind_1[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	sg4_ind_0[7:0]							
<b>Reset</b>	0x00							
<b>Access Type</b>	Read Only							

ビットフィールド	ビット	説明
sg4_ind_3	31:24	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。
sg4_ind_2	23:16	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。
sg4_ind_1	15:8	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。
sg4_ind_0	7:0	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。

**MSLUT0 (0x80)**

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_0[31:24]							
Reset	0xAAAAB554							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_0[23:16]							
Reset	0xAAAAB554							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_0[15:8]							
Reset	0xAAAAB554							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_0[7:0]							
Reset	0xAAAAB554							
Access Type	Write, Read							

ビットフィールド	ビット	説明
MSLUT_0	31:0	<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  …  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>

## MSLUT1 (0x81)

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_1[31:24]							
Reset	0x4A9554AA							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_1[23:16]							
Reset	0x4A9554AA							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_1[15:8]							
Reset	0x4A9554AA							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_1[7:0]							
Reset	0x4A9554AA							
Access Type	Write, Read							

ビットフィールド	ビット	説明
MSLUT_1	31:0	<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  …  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>

## MSLUT2 (0x82)

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_2[31:24]							
Reset	0x24492929							
Access Type	Write, Read							

<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	MSLUT_2[23:16]							
<b>Reset</b>	0x24492929							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	MSLUT_2[15:8]							
<b>Reset</b>	0x24492929							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	MSLUT_2[7:0]							
<b>Reset</b>	0x24492929							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
MSLUT_2	31:0	<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  …  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>

**MSLUT3 (0x83)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	MSLUT_3[31:24]							
<b>Reset</b>	0x10104222							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	MSLUT_3[23:16]							
<b>Reset</b>	0x10104222							
<b>Access Type</b>	Write, Read							

BIT	15	14	13	12	11	10	9	8
Field	MSLUT_3[15:8]							
Reset	0x10104222							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_3[7:0]							
Reset	0x10104222							
Access Type	Write, Read							

ビットフィールド	ビット	説明
MSLUT_3	31:0	<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  …  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>

## MSLUT4 (0x84)

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_4[31:24]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_4[23:16]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_4[15:8]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	MSLUT_4[7:0]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							

ビットフィールド	ビット	説明
MSLUT_4	31:0	<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  …  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>

**MSLUT5 (0x85)**

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_5[31:24]							
Reset	0xB5BB777D							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_5[23:16]							
Reset	0xB5BB777D							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_5[15:8]							
Reset	0xB5BB777D							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_5[7:0]							
Reset	0xB5BB777D							
Access Type	Write, Read							

ビットフィールド	ビット	説明
MSLUT_5	31:0	<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 … 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>

**MSLUT6 (0x86)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	MSLUT_6[31:24]							
<b>Reset</b>	0x49295556							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	MSLUT_6[23:16]							
<b>Reset</b>	0x49295556							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	MSLUT_6[15:8]							
<b>Reset</b>	0x49295556							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	MSLUT_6[7:0]							
<b>Reset</b>	0x49295556							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
MSLUT_6	31:0	<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 … 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>

**MSLUT7 (0x87)**

<b>BIT</b>	<b>31</b>	<b>30</b>	<b>29</b>	<b>28</b>	<b>27</b>	<b>26</b>	<b>25</b>	<b>24</b>
<b>Field</b>	MSLUT_7[31:24]							
<b>Reset</b>	0x00404222							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>23</b>	<b>22</b>	<b>21</b>	<b>20</b>	<b>19</b>	<b>18</b>	<b>17</b>	<b>16</b>
<b>Field</b>	MSLUT_7[23:16]							
<b>Reset</b>	0x00404222							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>15</b>	<b>14</b>	<b>13</b>	<b>12</b>	<b>11</b>	<b>10</b>	<b>9</b>	<b>8</b>
<b>Field</b>	MSLUT_7[15:8]							
<b>Reset</b>	0x00404222							
<b>Access Type</b>	Write, Read							
<b>BIT</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
<b>Field</b>	MSLUT_7[7:0]							
<b>Reset</b>	0x00404222							
<b>Access Type</b>	Write, Read							

ビットフィールド	ビット	説明
MSLUT_7	31:0	<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  …  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>

## MSLUT\_START (0x88)

BIT	31	30	29	28	27	26	25	24
Field	OFFSET_SIN90[7:0]							
Reset	0x00							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	START_SIN90[7:0]							
Reset	0xF7							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	7	6	5	4	3	2	1	0
Field	START_SIN[7:0]							
Reset	0x00							
Access Type	Write, Read							

ビットフィールド	ビット	説明
OFFSET_SIN90	31:24	CUR_Bに対する符号付きオフセット (±127マイクロステップ)。ポジション0でサイン波テーブルに一致するよう、START_SIN90を調整します。
START_SIN90	23:16	START_SIN90は、ポジション256でのマイクロステップ・テーブル・エントリに絶対電流を与えます。
START_SIN	7:0	START_SINは、マイクロステップ・テーブル・エントリ0に絶対電流を与えます。

## MSLUT\_SEL (0x89)

BIT	31	30	29	28	27	26	25	24
Field	X3[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	X2[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	X1[7:0]							
Reset	0x80							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	W3[1:0]		W2[1:0]		W1[1:0]		W0[1:0]	
Reset	0b01		0b01		0b01		0b10	
Access Type	Write, Read		Write, Read		Write, Read		Write, Read	

ビットフィールド	ビット	説明
X3	31:24	<p>LUTのセグメント3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W<sub>x</sub>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>定められた応答をするには、各値が以下を満たす必要があります。 0 &lt; X1 &lt; X2 &lt; X3</p>
X2	23:16	<p>LUTのセグメント2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W<sub>x</sub>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>定められた応答をするには、各値が以下を満たす必要があります。 0 &lt; X1 &lt; X2 &lt; X3</p>

ビットフィールド	ビット	説明
X1	15:8	<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W<sub>x</sub>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>定められた応答をするには、各値が以下を満たす必要があります。 0 &lt; X1 &lt; X2 &lt; X3</p>
W3	7:6	<p>ofs(X3)~ofs255でのLUT幅の選択。</p> <p>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</p>
W2	5:4	<p>ofs(X2)~ofs(X3 - 1)でのLUT幅の選択。</p> <p>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</p>
W1	3:2	<p>ofs(X1)~ofs(X2 - 1)でのLUT幅の選択。</p> <p>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</p>
W0	1:0	<p>ofs00~ofs(X1 - 1)でのLUT幅の選択。</p> <p>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</p>

## 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、図 57 に示すように、最低限のセットの追加コンポーネントを使用します。実際に用いる電源電圧に応じて、電源  $V_S$  のフィルタ処理に低 ESR のコンデンサを使用します。コンデンサは、チョッパ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。必要なモータ電圧と電流に応じ、最高性能を発揮するために、10 $\mu$ F のコンデンサをドライバ近くに配置す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（使用する場合）にも依存します。 $V_{CC\_IO}$  は、3.3V LDO 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、できる限り関連 IC ピンの近くに配置してください。すべての GND 接続には、強固な共通 GND 層をします。フィルタリング・コンデンサは  $V_{DD1V8}$  ピンに直接接続します。

**注：**最適な電流レギュレーション性能と適切な電流レベルを実現するには、対称の基板レイアウトを TMC5271 に用いることが必要です。

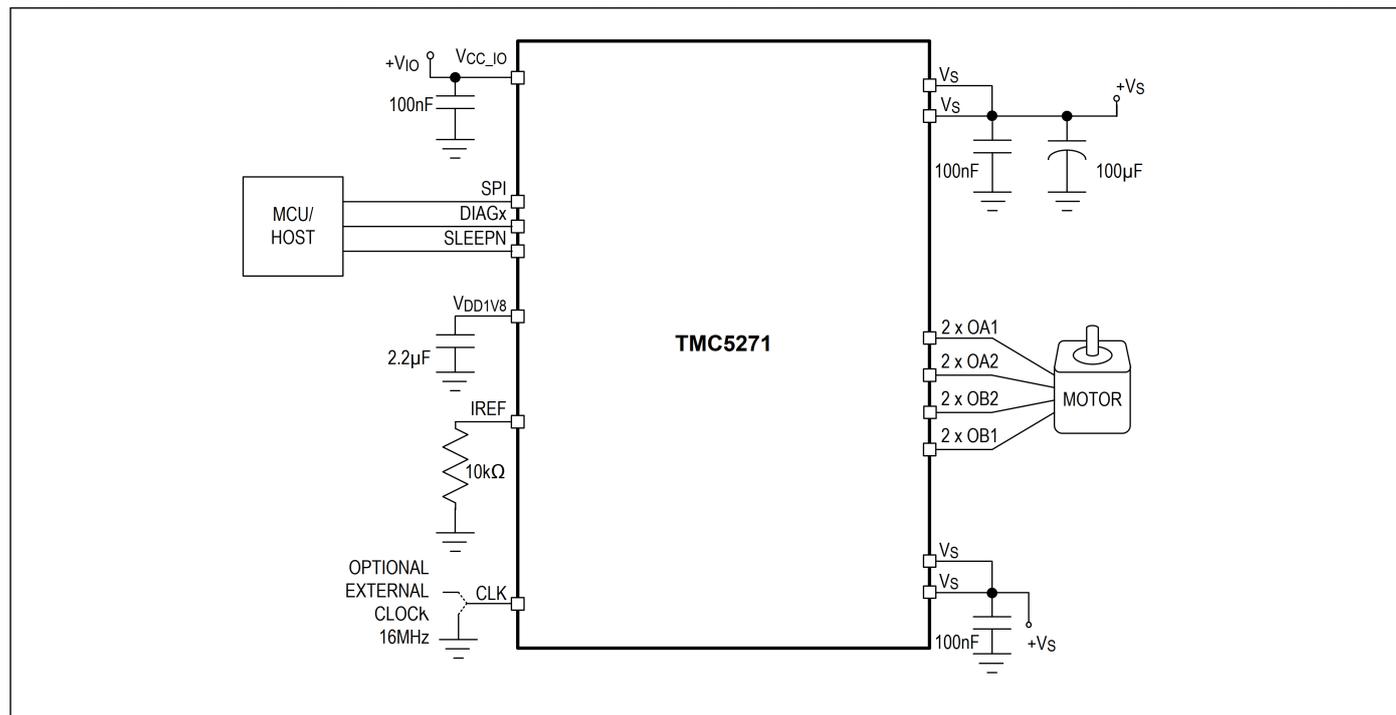


図 57. TMC5271 の標準的なアプリケーション回路

すべての TMC5271 信号は、それぞれの GND をリファレンスとします。すべての GND ピン/ボールは共通のグラウンドに直接接続してください。放熱性を確保するため、PCB の最上層は、大きな GND プレーンに接続して PCB 内の熱を拡散させます。

## 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・チップには ESD 保護回路が内蔵されているにもかかわらず、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・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 の ESD イベントを引き起こす傾向にあります。モータ自体を 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・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の活線挿抜に対して、ある程度までは保護可能です。

## 標準アプリケーション回路 (続き)

TMC5271 は、非常に小型の部品であり、低電圧動作を目標としているため、BOM や基板スペースが不必要に増加することのないよう、外部保護回路の追加使用は慎重に計画する必要があります。

簡単な方法は、図 58 に示すように、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くほど、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の増加の原因となり、それによってドライバの消費電力が大きくなります。そのため電源電圧が増加します。ここに示す値は例です。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、それによって電磁妨害波を減少させます。

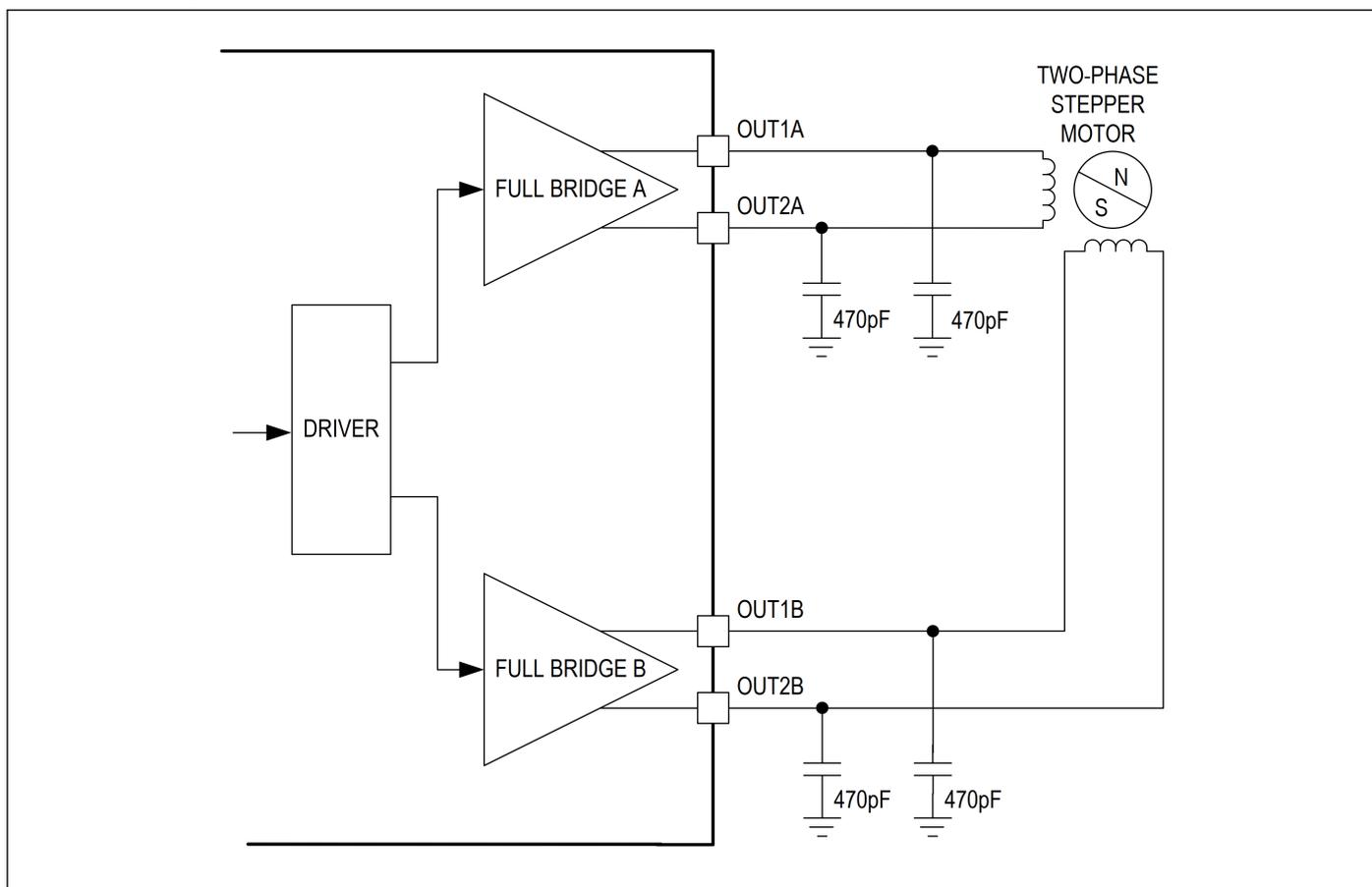


図 58. 簡単な ESD 対策

より精巧な方法を図 59 に示します。この方法では、LC フィルタを用いてドライバ出力をモータ・コネクタとデカップリングします。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。別のオプションは、バリスタ (V1A、V1B、V2A、V2B) により、すべての出力を ESD 電圧から保護することです。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。

## 標準アプリケーション回路 (続き)

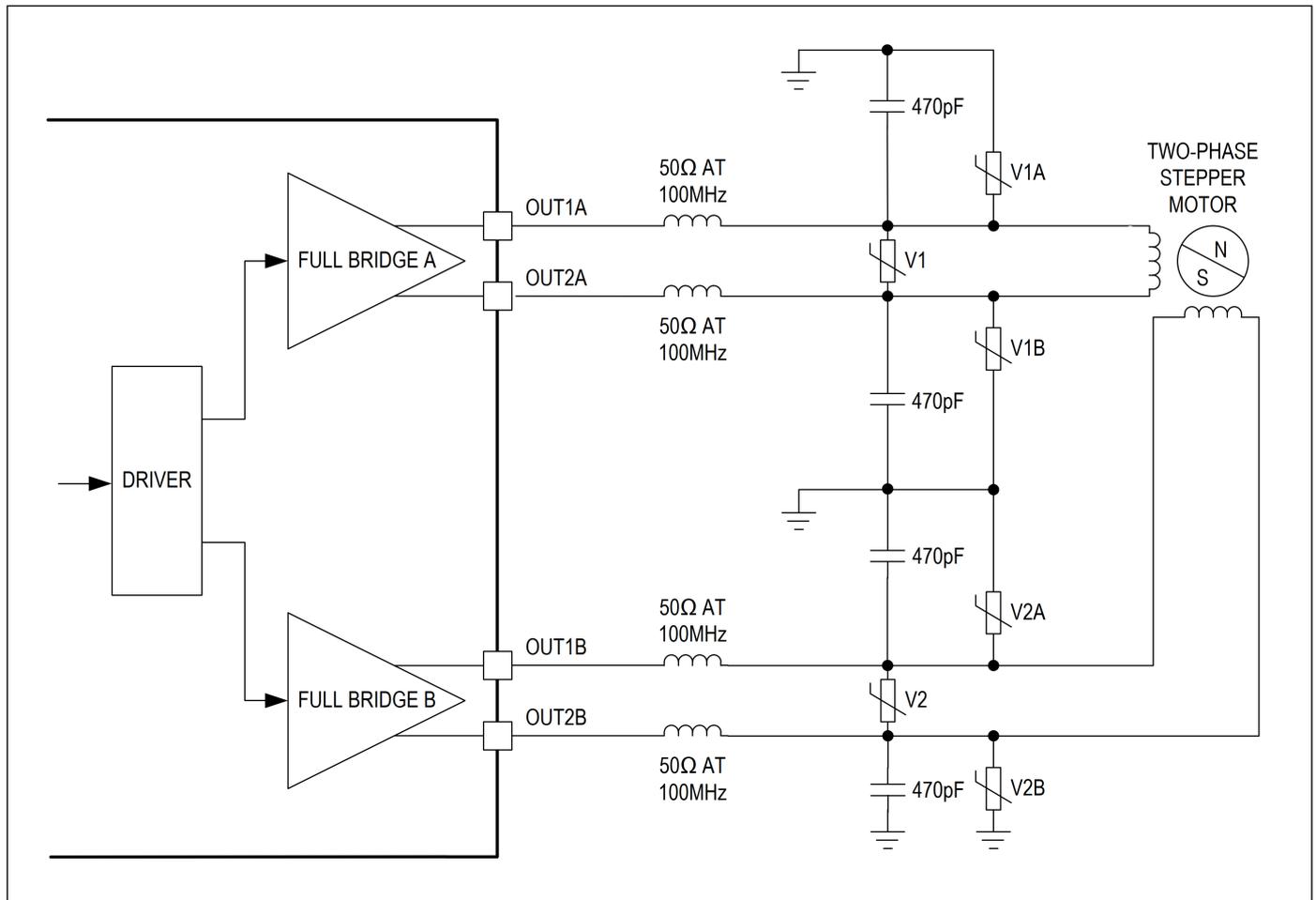


図 59. モータ出力保護の強化

## 型番

PART NUMBER	TEMPERATURE RANGE	PIN-PACKAGE
TMC5271AWX+	-40°C to +125°C	36 WLCSP (2.97mm x 3.13mm)
TMC5271AWX+T	-40°C to +125°C	36 WLCSP (2.97mm x 3.13mm)

+は鉛 (Pb) フリー/RoHS 準拠のパッケージであることを示します。  
T=テープ&リール。

## 改訂履歴

版数	改訂日	説明	改訂ページ
0	6/23	初版発行	-
1	4/24	端子説明 (B4)、ABN インクリメンタル・エンコーダ・インターフェース、低消費電力自己消費モード、CoolStep の調整、モータ温度の測定、図 50、モータの適合化、レジスタ IOIN、レジスタ GCONF を更新。StallGuard2 を使用したホーミングを StallGuard2 および StallGuard4 を使用したホーミングに変更し、StallGuard4 に対する説明も拡張。	16, 89, 93, 79, 94, 99, 106, 120, 115, 77