



TMC5240

## 36V 2A<sub>RMS</sub>+のスマート集積ステッピング・モータ・ドライバおよびコントローラ

### 概要

**TMC5240** は、スマート高性能ステッピング・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イス (SPI、UART) および広範な診断機能を備えています。このデバイスは、ジャークを最適化した柔軟なランプ・ジェネレータで、256 マイクロステップ、組み込みインデクサ、全機能内蔵 36V、3.0A<sub>MAX</sub> H ブリッジおよび電力散逸のない内蔵電流検出 (ICS) 機能をベースとし、業界最先端のステッピング・モータ・ドライバの自動ターゲット・ポジショニングを可能にします。

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

高集積、高エネルギー効率、小フォーム・ファクタという特長を備えているため、コスト効率の高いソリューション向けの小型でスケーラブルなシステムを実現します。このフル機能ソリューションにより、学習曲線を最小限に抑えると共に、最高クラスの性能を可能にできます。

H ブリッジ電界効果トランジスタ (FET) は、インピーダンスが非常に低いため、駆動効率が高く、発生する熱を最小限に抑えます。代表的な合計  $R_{ON}$  (ハイ・サイド + ロー・サイド) は  $0.23\Omega$  です。

H ブリッジごとの最大実効電流は、4 層 PCB を想定すると、室温で  $I_{RMS} = 2.1\text{A}_{RMS}$  です。

H ブリッジごとの最大出力電流は  $I_{MAX} = 5.0\text{A}_{MAX}$  で、過電流保護 (OCP) によって制限されます。

この電流は、熱に関する考慮事項の制限を受けるため、実際の最大実効電流は、PCB グランド・プレーン、ヒートシンク、換気などのアプリケーションの熱特性に依存します。

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

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

TMC5240 は、豊富な診断機能、および短絡保護/OCP、サーマル・シャットダウン、低電圧ロックアウト (UVLO) などの保護機能を備えています。

サーマル・シャットダウン・イベントと UVLO イベントの際にには、ドライバはディスエーブルされます。

更に、TMC5240 には、ドライバ温度の測定、モータ温度の推定、1つの外部アナログ入力の測定を行う機能が備わっています。

TMC5240 は、いずれも露出パッドを備えた、小型の TQFN32 (5mm × 5mm) パッケージまたは熱的に最適化された TSSOP38 (9.7mm × 4.4mm) パッケージで提供されます。

### アプリケーション

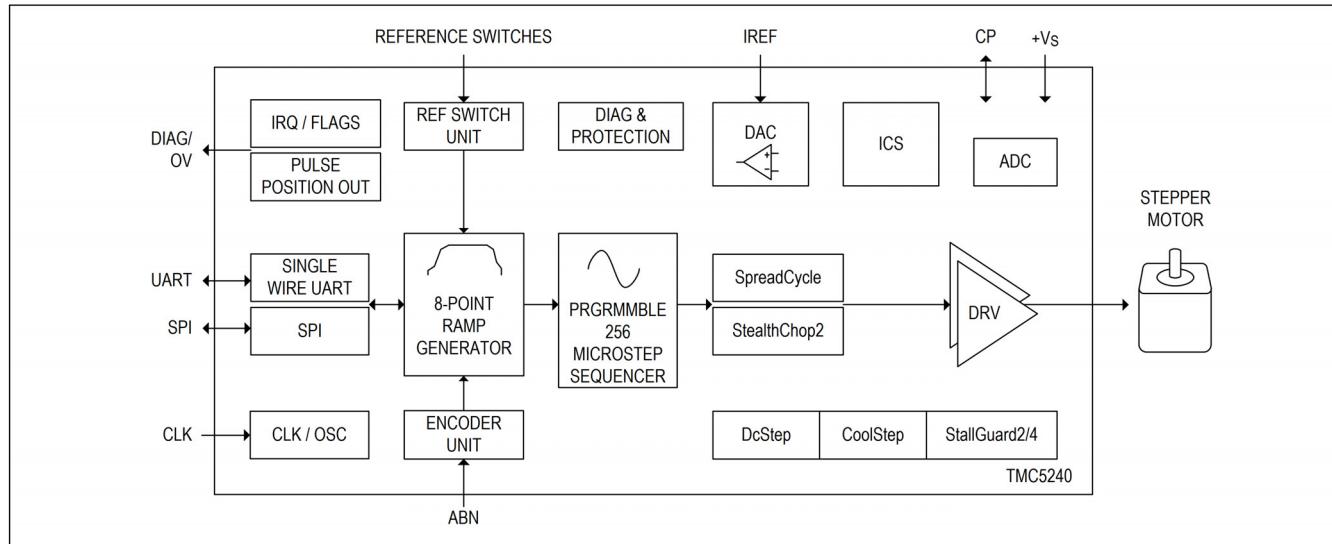
- テキスタイル、ミシン、編み機
- ラボ・オートメーションおよびファクトリ・オートメーション
- 3D プリンタ、ID プリンタ/カード・プリンタ
- リキッド・ハンドリング、医療用アプリケーション
- オフィス・オートメーションおよびペーパー・ハンドリング
- POS、マッサージ・チェア
- ATM、現金回収機、紙幣識別機、現金自動支払機
- CCTV、セキュリティ
- ポンプおよびバルブ制御
- ヘリオスタッフやアンテナのポジショニング

### 利点および特長

- 電圧範囲 : 4.5V~36V DC
- 低  $R_{DS(ON)}$  (ハイ・サイド + ロー・サイド) : 230mΩ (代表値) ( $T_A = 25^\circ\text{C}$ )
- H ブリッジあたりの電流定格 (25°C での代表値) :
  - $I_{MAX} = 5.0\text{A}$  (ブリッジ・ピーク電流)
  - $I_{RMS} = 2.1\text{A}_{RMS}$  (3A の正弦波ピーク)
- 全機能内蔵無損失電流検出
- ジャークを最小限に抑えるための 8 点モーション・コントローラ
- SPI および単線式 UART
- エンコーダ・インターフェイスおよび 2 個のリファレンス・スイッチ入力
- フルステップあたり 256 マイクロステップの最大分解能
- 柔軟な波形テーブルと位相シフトによりモータに適合
- StealthChop2 サイレント・モータ動作
- SpreadCycle 高ダイナミック・モータ制御チョッパー
- ジャークのない StealthChop2 と SpreadCycle の組み合わせ
- StallGuard2 および StallGuard4 によるセンサー不要のモータ負荷検出
- 最大 75% のエネルギー節減を実現する CoolStep 電流制御
- 受動ブレーキおよびフリーホイール・モード
- モータ相の温度の推定
- チップ温度の測定
- 汎用アナログ入力
- すべてを網羅する保護および診断
- 過電圧保護出力
- コンパクトな 5mm × 5mm TQFN32 パッケージまたは 9.7mm × 4.4mm TSSOP38 パッケージ

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
利点および特長	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
TQFN32 5mm × 5mm	10
TSSOP38 9.7mm × 4.4mm EP	10
電気的特性	10
ピン配置	15
TMC5240 TQFN のピン配置	15
TMC5240 TSSOP のピン配置	15
端子説明	16
機能図	19
TMC5240	19
詳細	20
動作原理	20
フル機能モーション・コントローラおよびドライバ	20
主要コンセプト	21
制御インターフェイス	21
内蔵 8 点モーション・コントローラ	21
自動停止パワー・ダウン	21
StealthChop2 および SpreadCycle ドライバ	22
StallGuard2/4 — 機械的負荷検出	22
CoolStep — 負荷適応型電流制御	22
エンコーダ・インターフェイス	23
SPI インターフェイス	23
SPI データグラム構造	23
読み出し／書込みの選択 (WRITE_notREAD)	23
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	24
データ・アライメント	24
SPI 信号	24
SPI のタイミング	25
UART 単線式インターフェイス	25
データグラム構造	25
書込みアクセス	25
読み出しアクセス	26
CRC の計算	27
CRC 計算の C コード例	27

## 目次（続き）

UART 信号	27
複数ノードのアドレス指定	28
StealthChop2	28
自動調整	29
StealthChop2 のオプション	29
StealthChop2 電流レギュレータ	31
下限電流値	33
速度ベースのスケーリング	34
StealthChop2 と SpreadCycle の結合	35
StealthChop2 でのフラグ	37
開放負荷フラグ	37
モータの状態を示す PWM_SCALE_SUM	37
フリーホイーリングおよび受動ブレーキング	37
StealthChop2 を制御するパラメータ	38
SpreadCycle と一般的なチョッパー	39
SpreadCycle チョッパー	41
一般的な定オフ時間チョッパー	43
内蔵電流検出機能	44
モータ電流の設定	45
フルスケール電流レンジの設定	45
速度ベースのモード制御	46
ランプ・ジェネレータ	49
現実の単位への変換	49
動作プロファイル	50
ランプ・モード	50
8点ランプ	52
始動速度および停止速度	52
速度モード	55
ランプの早期終了	55
アプリケーション例：ジョイスティック制御	56
速度スレッショルド	56
リファレンス・スイッチ	57
仮想リファレンス・スイッチ	59
ランプ・ジェネレータの応答時間	59
外部 STEP/DIR ドライバ	59
位置比較機能	60
StallGuard2 の負荷測定	60
StallGuard2 のスレッショルド SGT の調整	61
TCOOLTHRS および THIGH を制限する可変速度	62
高トルク・リップルと共振を伴う小型モータ	63

## 目次（続き）

モータ・コイル抵抗の温度依存性	63
StallGuard2 測定の正確さと再現性	63
StallGuard2 の更新レートおよびフィルタ	63
モータ・ストールの検出	63
StallGuard2 を使用したホーミング	64
StallGuard2 動作の制限	64
StallGuard4 の負荷測定	64
StallGuard4 の調整	66
StallGuard4 の更新レート	66
モータ・ストールの検出	66
StallGuard4 動作の制限	67
CoolStep の負荷適応型電流スケーリング	67
CoolStep 用の設定	67
CoolStep の調整	69
応答時間	69
低速度およびスタンバイ動作	69
診断出力	69
DcStep	70
DcStep のデザインイン	70
DcStep とモーション・コントローラの統合	71
DcStep モードにおけるストール検出	72
DcStep 動作での実際のモータ速度の測定	73
サイン波ルックアップ・テーブル	73
マイクロステップ・テーブル	73
ABN インクリメンタル・エンコーダ・インターフェイス	75
モータの分解能に合うエンコーダ設定	77
リセット、ディスエーブル／停止、パワー・ダウン	77
緊急停止	77
外部リセットおよびスリープ・モード	77
保護およびドライバ診断	78
過電流保護	78
過熱保護およびサーマル・シャットダウン	78
温度測定	78
チップ温度の測定	79
モータ温度の測定	79
過電圧保護と OV ピン	79
短絡保護（GND への短絡および VS への短絡）	80
開放負荷診断	80
低電圧ロックアウト保護	80

## 目次（続き）

ESD 保護	81
外部アナログ入力 AIN のモニタリング	81
クロック発振器とクロック入力	81
内部クロックの使用	81
外部クロックの使用	81
全般的なレジスタ・マッピングおよびレジスタ情報	81
レジスタ・マップ	83
TMC5240	83
レジスタの詳細	90
標準アプリケーション回路	156
標準的なアプリケーション回路	156
大モータ電流	156
ドライバ保護およびEMI 回路	156
型番	158
改訂履歴	158

## 図一覧

図 1. ブロック図 .....	19
図 2. 代表的な外部コンポーネントを使用したブロック図 .....	20
図 3. 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 4. SPI タイミング図 .....	25
図 5. UART のデイジーチェーン例 .....	28
図 6. StealthChop2 の自動調整手順 .....	30
図 7. StealthChop2 : PWM_REG の設定は良好 .....	31
図 8. StealthChop2 : AT#2 時の PWM_REG の設定が過小 .....	32
図 9. 正しく決定された PWM_GRAD(_AUTO)および PWM_OFS(_AUTO) .....	32
図 10. 過小な PWM_GRAD 設定の例 .....	33
図 11. 速度ベースの PWM スケーリング (pwm_yscale = 0) .....	35
図 12. SpreadCycle への切り替えオプションのための TPWMTHRS .....	36
図 13. 代表的なチョッパー減衰フェーズ .....	40
図 14. 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 .....	42
図 15. 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 .....	43
図 16. 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 .....	44
図 17. 速度依存モードの選択 .....	47
図 18. 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 .....	51
図 19. 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 .....	52
図 20. 距離が短いために VMAX に到達できない 8 点ランプ .....	53
図 21. 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 .....	53
図 22. 距離が短いために V1 に達しないプロファイル .....	54
図 23. 距離が短いために TVMAX を維持できないプロファイル .....	54
図 24. オンザフライの目標位置変更があった場合の 8 点ランプの例 .....	55
図 25. ランプ・ジェネレータの速度依存モータ制御 .....	57
図 26. リファレンス・スイッチの使用 (例) .....	58
図 27. 仮想停止スイッチと制限の図解 .....	59
図 28. StallGuard2 の動作原理 .....	61
図 29. 例 : サンプル・モータを用いた SGT の最適設定値と StallGuard2 の読み出し値 .....	63
図 30. StallGuard4 の動作モード .....	65
図 31. モータ電流を負荷に適合させる CoolStep .....	68
図 32. 診断出力構成のオプション .....	70
図 33. DcStep により拡大されるアプリケーション動作領域 .....	71
図 34. 過負荷状態時の DcStep の速度プロファイル .....	72
図 35. LUT プログラミングの例 .....	74
図 36. OFFSET_SIN90 を使用したコサイン波のシフト .....	75
図 37. インクリメンタル・エンコーダの ABN 信号の概要 .....	76
図 38. ブレーキ・チョッパー回路の例 .....	80
図 39. 標準的なアプリケーション回路 .....	156

## 図一覧 (続き)

図 40. 簡単な ESD 対策 .....	157
図 41. モータ出力保護の強化 .....	158

## 表一覧

表 1. SPI データグラム構造 .....	23
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 .....	24
表 4. UART 書込みアクセス・データグラム構造 .....	25
表 5. UART 読出しアクセス要求データグラムの構造 .....	26
表 6. UART 読出しアクセス返答データグラムの構造 .....	26
表 7. TMC5240 UART のインターフェイス信号 .....	27
表 8. 最大 255 ノードをアドレス指定する UART 例 .....	28
表 9. StealthChop2 自動調整 AT#1 および AT#2 の制約事項と要件 .....	29
表 10. StealthChop2 用の PWM 周波数の選択 (太字を推奨) .....	31
表 11. StealthChop2 を制御するパラメータ .....	38
表 12. SpreadCycle および一般的な定オフ時間チョッパーを制御するパラメータ .....	40
表 13. SpreadCycle モードのパラメータ .....	42
表 14. 帰還抵抗の推奨値 .....	44
表 15. モータ電流を制御するパラメータ .....	45
表 16. I <sub>FS</sub> フルスケール・ピーク・レンジの設定値 ( $R_{REF} = 12k\Omega$ の場合の例) .....	46
表 17. DRV_CONF のビット 1 および 0 の設定と様々な $R_{REF}$ に基づく、フルスケール実効値電流 I <sub>FS</sub> (単位: アンペア (A RMS)) .....	46
表 18. 速度ベースのモード制御パラメータ .....	48
表 19. ランプ・ジェネレータのパラメータと単位の関係 .....	49
表 20. X_COMPARE_REPEAT オプションと周期的なパルス動作 .....	60
表 21. StallGuard2 関連パラメータ .....	61
表 22. StallGuard4 関連のパラメータ .....	65
表 23. CoolStep の重要パラメータ .....	67
表 24. CoolStep のその他のパラメータとステータス情報 .....	68
表 25. 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 .....	77
表 26. フルスケール電流設定に基づく過電流保護スレッショルド .....	80
表 27. レジスタ・マップの概要 .....	81

**絶対最大定格**

$V_S \sim GND$	-0.3V ~ 41V
$V_{DDIV8} \sim GND$	-0.3V ~ min (2.2, $V_S + 0.3$ )V
$AGND \sim GND$	-0.3V ~ +0.3V
$OUT1A, OUT2A, OUT1B, OUT2B$	-0.3V ~ $V_S + 0.3$ V
$V_{CP} \sim GND$	$V_S - 0.3V \sim \min (44, V_S + 6)$ V
$CPO \sim GND$	$V_S - 0.3V \sim \min (44, V_S + 6)$ V
$CPI \sim GND$	-0.3V ~ min (41, $V_S + 0.3$ )V
$SLEEPN \sim GND$	-0.3V ~ $V_S + 0.3$ V

$IREF, AIN \sim GND$	-0.3V ~ min (2.2, $V_{DDIV8} + 0.3$ )V
$V_{CC\_IO} \sim GND$	-0.3V ~ 5.5V
ロジック入力／出力電圧～GND	-0.3V ~ $V_{CC\_IO} + 0.3$ V
$OV \sim GND$	-0.3V ~ 6V
動作温度範囲	-40°C ~ +125°C
ジャンクション温度	+165°C
保存温度範囲	-65°C ~ +150°C
はんだ処理温度 (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****TQFN32 5mm × 5mm**

Package Code	T3255+5C
Outline Number	<b>21-0140</b>
Land Pattern Number	<b>90-0013</b>
<b>Thermal Resistance, Single-Layer Board:</b>	
Junction to Ambient ( $\theta_{JA}$ )	47°C/W
Junction to Case ( $\theta_{JC}$ )	1.7°C/W
<b>Thermal Resistance, Four-Layer Board:</b>	
Junction to Ambient ( $\theta_{JA}$ )	29°C/W
Junction to Case ( $\theta_{JC}$ )	1.7°C/W

**TSSOP38 9.7mm × 4.4mm EP**

Package Code	U38E+3C
Outline Number	<b>21-0714</b>
Land Pattern Number	<b>90-0435</b>
<b>Thermal Resistance, Four-Layer Board:</b>	
Junction to Ambient ( $\theta_{JA}$ )	25°C/W
Junction to Case ( $\theta_{JC}$ )	1°C/W

最新のパッケージ外形図とランド・パターン（フットプリント）に関しては、[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**

( $V_S = 4.5V \sim 36V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 、代表値は  $T_A = 25^\circ C$  および  $V_S = 24V$  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
<b>POWER SUPPLY</b>						
Supply Voltage Range	$V_S$		4.5	36	V	
Sleep Mode Current Consumption	$I_{VS}$	$V(SLEEPN) = 0$		4	18	$\mu A$

## 電気的特性（続き）

( $V_S = 4.5V \sim 36V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 、代表値は  $T_A = 25^\circ C$  および  $V_S = 24V$  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
Quiescent Current Consumption	$I_{VS}$	$V(SLEEPN) = 1, V(DRV\_ENN) = 1$		3.5	5	mA
1.8V Regulator Output Voltage	$V_{VDD}$	$V_S = 4.5V$		1.8		V
$V_{DD}$ Current Limit	$I_{V18LIM}$		20			mA
Charge Pump Voltage	$V_{CP}$			$V_S + 2.7$		V
Logic I/O Supply Voltage Range	$V_{CC\_IO}$		2.2	5.5		V
Sleep Mode Current Consumption	$I_{VCC\_IO}$	$V(SLEEPN) = 0$	5	10		$\mu A$
Quiescent Current Consumption	$I_{VCC\_IO}$	$V(SLEEPN) = 1$	35	60		$\mu A$
<b>LOGIC LEVEL INPUTS-OUTPUTS</b>						
Input Voltage Level - High	$V_{IH}$		$0.7 \times V_{CC\_IO}$			V
Input Voltage Level - Low	$V_{IL}$		$0.3 \times V_{CC\_IO}$			V
Input Hysteresis	$V_{HYS}$		$0.15 \times V_{CC\_IO}$			V
Internal Pullup/Pulldown Resistance	$R_{PULL}$	to GND or to $V_{CC\_IO}$	60	100	140	$k\Omega$
Input Leakage	$I_{nLeak}$	Inputs without pullup/pulldown resistance	-1		+1	$\mu A$
Output Logic-Low Voltage	$V_{OL}$	$I_{LOAD} = 5mA$			0.4	V
Push-Pull Output Logic-High Voltage	$V_{OH}$	$I_{LOAD} = 5mA$	$V_{CC\_IO} - 400mV$			
Open-Drain Output Logic High Leakage Current	$I_{OH}$	$V(PIN) = 5.5V$	-1		+1	$\mu A$
SLEEPN Voltage Level High	$V_{IH\_SLEEPN}$		0.9			V
SLEEPN Voltage Level Low	$V_{IL\_SLEEPN}$				0.6	V
SLEEPN Pulldown Input Resistance	$R_{PD\_SLEEPN}$		0.8	1.5		$M\Omega$
<b>OUTPUT SPECIFICATIONS</b>						
Output ON-Resistance Low Side	$R_{ON\_LS}$	Full-scale bits = 10	0.11	0.2		$\Omega$
		Full-scale bits = 01	0.15	0.28		
Output ON-Resistance Low Side	$R_{ON\_LS}$	Full-scale bits = 00	0.28	0.54		$\Omega$
Output ON-Resistance High Side	$R_{ON\_HS}$		0.12	0.22		$\Omega$
Output Leakage	$I_{LEAK}$		-5		+5	$\mu A$

## 電気的特性（続き）

( $V_S = 4.5V \sim 36V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 、代表値は  $T_A = 25^\circ C$  および  $V_S = 24V$  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
Output Slew Rate	SR	Slew-rate bits = 00		100		V/μs
		Slew-rate bits = 01		200		
		Slew-rate bits = 10		400		
		Slew-rate bits = 11		800		
<b>PROTECTION CIRCUITS</b>						
Overcurrent Protection Threshold	OCP	Full-scale bits = 10	5.0			A
		Full-scale bits = 01	3.33			
		Full-scale bits = 00	1.67			
Overcurrent Protection Blanking Time	T <sub>OCP</sub>		0.9	1.5	2.3	μs
UVLO Threshold on $V_S$	UVLO	$V_S$ falling	3.75	3.9	4.05	V
UVLO Threshold on $V_S$ Hysteresis	UVLOHYS			0.12		V
UVLO Threshold on $V_{CC\_IO}$	UVLO	$V_{CC\_IO}$ falling	0.9	1.5	1.95	
$V_{CC\_IO}$ UVLO Hysteresis	UVLOVCCH			100		mV
Thermal Protection Threshold Temperature	TSD			165		°C
Thermal Protection Temperature Hysteresis				20		°C
<b>CURRENT REGULATION</b>						
IREF Pin Resistor Range	R <sub>REF</sub>		12	60		kΩ
IREF Output Voltage	V <sub>REF</sub>		0.882	0.9	0.918	V
Full-Scale Current Constant	KIFS	IFS = 1A		11.75		A x kΩ
Full-Scale Current Constant	KIFS	IFS = 2A		24		A x kΩ
Full-Scale Current Constant	KIFS	IFS = 3A		36		A x kΩ
Regulation Accuracy	DITRIP1	Output current from 7% to 100% FS, $R_{REF} = 12k\Omega$	-5		+5	%
<b>FUNCTIONAL TIMINGS</b>						
SLEEP Time	t <sub>SLEEP</sub>	SLEEPN = 0 to OUT_ three state		50		μs
Wake-Up Time from Sleep	T <sub>WAKE</sub>	SLEEPN = 1 to normal operation		2.5		ms
Enable Time	TEN	Time from DRV_ENN pin falling edge to driver on		1.5		μs
Disable Time	TEN	Time from DRV_ENN pin rising edge to driver off		6		μs

## 電気的特性（続き）

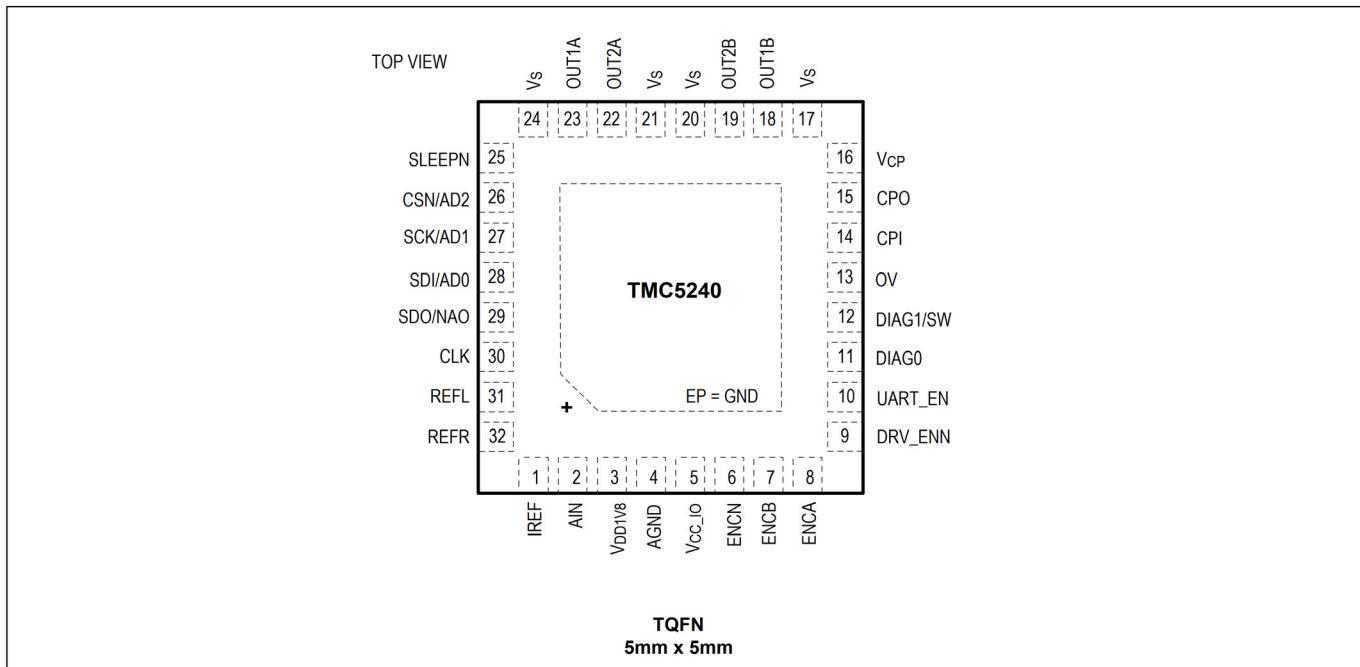
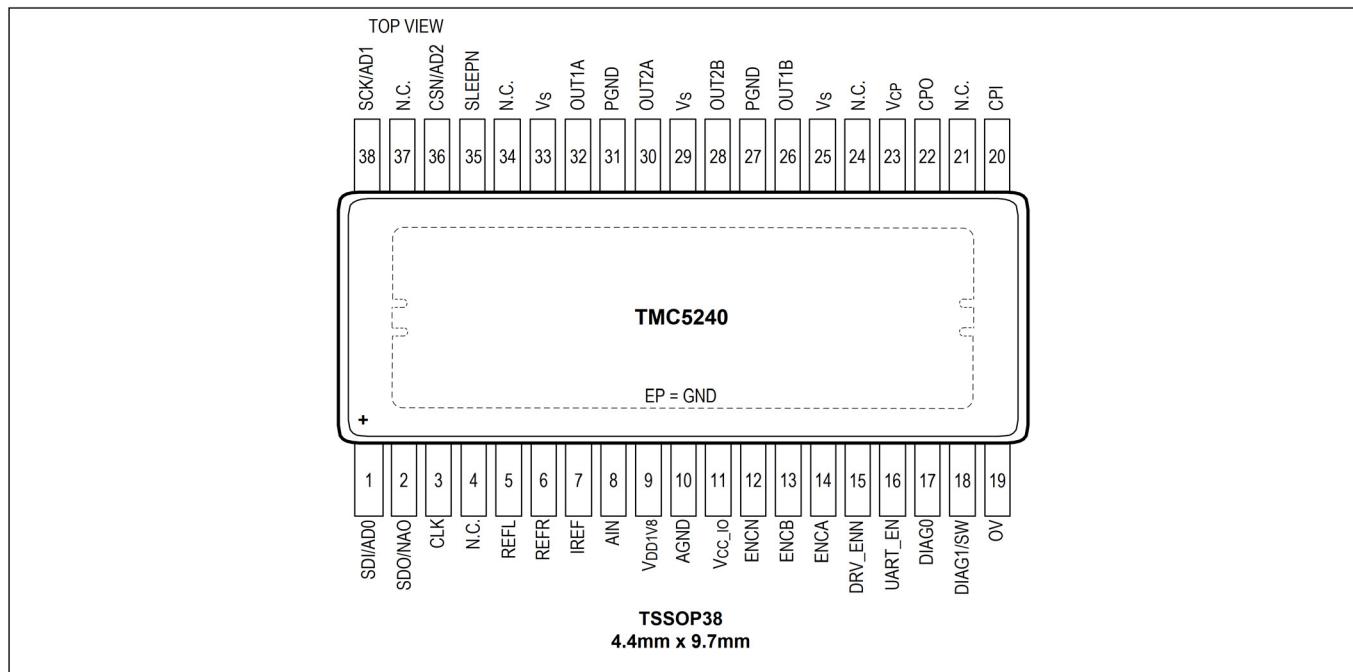
( $V_S = 4.5V \sim 36V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 、代表値は  $T_A = 25^\circ C$  および  $V_S = 24V$  を前提としています。制限値は  $T_A = +25^\circ C$  で 100% テストされています。動作温度範囲および関連する電源電圧範囲における制限値は、設計および特性評価により裏付けられています。)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
<b>CLOCK</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		
External Clock Timeout Detection in Cycles of Internal $f_{CLKOSC}$			12	16		
External Clock Detection Lower Frequency Threshold	$f_{CLKLO}$		4			MHz
<b>SPI TIMINGS</b>						
SCK Valid Before or After Change of CSN	$t_{CC}$		$T_{SCLK}$			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	$t_{DO}$	$V_{CC\_IO} = 3.3V$	27	40		ns
SDI, SCK, and CSN Filter Delay Time	$t_{FILT}$	Rising and falling edge	10			ns
<b>ENCODER TIMING</b>						
Encoder Counting Frequency	$f_{CNT}$		$< 2/3 f_{CLK}$			
A/B/N Input Low Time	$t_{ABNL}$		$3t_{CLK} + 20$			ns
A/B/N Input High Time	$t_{ABNH}$		$3t_{CLK} + 20$			ns
A/B/N Spike Filtering Time	$t_{FILTABN}$	Rising and falling edge	$3t_{CLK}$			
<b>ADC/Analog Input/Temperature</b>						
ADC Resolution		12 bit + sign	13			Bit
Analog Input Voltage Range	$V_{AIN}$		0	1.25		V
Analog Input Leakage	$I_{AIN,leak}$		-1	+1		uA

## 電気的特性（続き）

( $V_S = 4.5V \sim 36V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 、代表値は  $T_A = 25^\circ C$  および  $V_S = 24V$  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
Analog Input Frequency	$f_{AIN}$	Assuming undersampling at AIN is accepted, the AIN input frequency needs to be lower than the given max value for a meaningful ADC conversion for a single ADC channel.			70	kHz
Driver Temperature Accuracy	$T_{DRIVER}$			$\pm 10$		°C
Supply Voltage Measurement Accuracy			-5		+5	%
ADC Sample Rate	$f_{SAMPLE, ADC}$			$f_{CLK} / 2048$		

**ピン配置****TMC5240 TQFN のピン配置****TMC5240 TSSOP のピン配置**

## 端子説明

ピン		名称	説明	リファレンス電源	タイプ
TQFN32	TSSOP38				
4	10	AGND	アナログ・グランド。グランド・プレーンに接続します。		GND
-	27, 31	PGND	電源グランド。グランド・プレーンに接続します。		GND
17, 20, 21, 24	25, 29, 33	V <sub>s</sub>	モータ電源電圧。GND プレーン/露出パッドへ最短のループを形成してピン近くでのフィルタによる除去機能を提供します。		Supply
3	9	V <sub>DD1V8</sub>	内蔵 1.8V レギュレータの出力。最高性能を発揮させるには、AGND との間のピン近くに 2.2μF 以上のセラミック・コンデンサを接続します。		Supply
16	23	V <sub>CP</sub>	チャージ・ポンプ電圧。1.0μF のコンデンサを用いて V <sub>s</sub> に接続します。  誘導性ピークを避けるために、コンデンサの正側端子を V <sub>s</sub> ピンの近くに接続します。		Analog Output
5	11	V <sub>CC_IO</sub>	回路の IO レベルを定義するために外部電源から供給されるデジタル IO 電源電圧。出力ピンに適切な電圧レベルを設定するためには、V <sub>CC_IO</sub> と V <sub>s</sub> の間に 2.2μF のセラミック・コンデンサを接続します。	V <sub>CC_IO</sub>	Analog Input
15	22	CPO	チャージ・ポンプ・コンデンサ出力。		Analog Output
14	20	CPI	チャージ・ポンプ・コンデンサ入力。22nF、50V のコンデンサを用いて CPI に接続します。		Analog Output
30	3	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。内蔵のクロックフェイズ・オーバー回路が外部クロック信号が失われた場合に保護します。	V <sub>CC_IO</sub>	Digital Input
31	5	REFL	内蔵ランプ・ジェネレータ用の左側リファレンス入力。	V <sub>CC_IO</sub>	Digital Input
32	6	REFR	内蔵ランプ・ジェネレータ用の右側リファレンス入力。	V <sub>CC_IO</sub>	Digital Input
26	36	CSN/AD2	SPI チップ・セレクト入力 (負でアクティブ) (UART_EN = 0) または UART モードのアドレス入力 2 (+4) (UART_EN = 1)。	V <sub>CC_IO</sub>	Digital Input (pull up)
27	38	SCK/AD1	SPI シリアル・クロック入力 (UART_EN = 0) または UART モードのアドレス入力 1 (+2) (UART_EN = 1)。	V <sub>CC_IO</sub>	Digital Input (pull up)
28	1	SDI/AD0	SPI データ入力 (UART_EN = 0) または UART モードのアドレス入力 0 (+1) (UART_EN = 1)。	V <sub>CC_IO</sub>	Digital Input (pull up)
29	2	SDO/NAO	SPI データ出力 (スリーステート) (UART_EN = 0) または UART モードの次のアドレス出力 (NAO) (UART_EN = 1)。	V <sub>CC_IO</sub>	Digital Output
1	7	IREF	電流スケーリング用のアナログ・リファレンス電流。 外付け抵抗を介して GND に接続します。	V <sub>CC_IO</sub>	Analog Input
10	16	UART_EN	インターフェイス選択ピン。  ローに接続された場合、SPI インターフェイスがイネーブルされます。  ハイに接続された場合、UART インターフェイスがイネーブルされます。  内蔵プルダウン抵抗。	V <sub>CC_IO</sub>	Digital Input (pull down)
7	13	ENCB	エンコーダの B チャンネル入力。	V <sub>CC_IO</sub>	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
TQFN32	TSSOP38				
8	14	ENCA	エンコーダの A チャンネル入力。	V <sub>CC_IO</sub>	Digital Input (pull up)
6	12	ENCN	エンコーダの N チャンネル入力。	V <sub>CC_IO</sub>	Digital Input (pull up)
9	15	DRV_ENN	イネーブル入力。このピンがハイ・レベルに駆動されると、電力段のスイッチがオフ（すべてのモータ出力がフロート状態）になります。	V <sub>CC_IO</sub>	Digital Input (pull up)
11	17	DIAG0	診断出力 DIAG0。  内蔵モーション・コントローラから外部ドライバへの割込みまたは STEP 出力。  オープン・ドレイン・モードでは外部プルアップ抵抗を使用します。  システムがリセット状態の場合は、このピンがアクティブにローになり、リセット状態であることを外部コントローラに通知します。	V <sub>CC_IO</sub>	Digital Output
12	18	DIAG1/SW	診断出力 DIAG1。  内蔵モーション・コントローラから外部ドライバへの位置比較または DIR 出力。  オープン・ドレイン・モードでは外部プルアップ抵抗を使用します。  UART モードでは単線式 I/O となります。	V <sub>CC_IO</sub>	Digital IO
25	35	SLEEPN	ロー・アクティブのパワー・ダウン入力／リセット入力。  デバイスをスリープ・モードにするには、連続的なロー・レベルを印加します。  SLEEPN はプルダウンを内蔵しています。  使用しない場合は、V <sub>S</sub> または V <sub>CC_IO</sub> （これは高電圧ピンです）に接続します。  IC がスリープ・モード／リセットから回復した後は、再度使用する前に再設定を行う必要があります。スリープ・モードの間、レジスタの内容は保存されません。  IC の再設定の間、DRV_ENN を用いてブリッジ・ドライバをディスエーブルしたままにすることを推奨します。  モータ速度が高い場合には決して使用しないでください。	V <sub>S</sub>	Analog Input (pull down)
19	28	OUT2B	モータ・コイル B の出力 2。	V <sub>S</sub>	Analog output
18	26	OUT1B	モータ・コイル B の出力 1。	V <sub>S</sub>	Analog output
22	30	OUT2A	モータ・コイル A の出力 2。	V <sub>S</sub>	Analog output
23	32	OUT1A	モータ・コイル A の出力 1。	V <sub>S</sub>	Analog output

## 端子説明（続き）

ピン		名称	説明	リファレンス電源	タイプ
TQFN32	TSSOP38				
EP	EP	GND	露出ダイ・パッド 露出ダイ・パッドは GND プレーンに接続します。GND プレーンとの間には熱伝導のためにできるだけ多くのビアを使用します。電力段および内部回路には GND ピンとして機能します。		GND
-	4, 21, 24, 34, 37	N.C.	内部接続なし。このピンは開放のままにするか、冷却効率を向上するために GND に接続します。		N.C.
13	19	OV	プログラマブルなスレッショルド電圧を備えた過電圧インジケータ出力（オープン・ドレイン）。電源電圧を制限するために、外部 MOSFET を負荷抵抗と共に接続します。外部プルアップ抵抗が必要です。ADC により $f_{CLK}/2048$ のサイクルで更新されます。	$V_{CC\_IO}$	Digital Output (open drain)
2	8	AIN	汎用アナログ入力。内部 ADC により、 $f_{CLK}/2048$ のサイクルで測定されます。 入力範囲は 0~1.25V です。 値は SPI/UART を介して使用可能です。	$V_{CC\_IO}$	Analog Input

## 機能図

## TMC5240

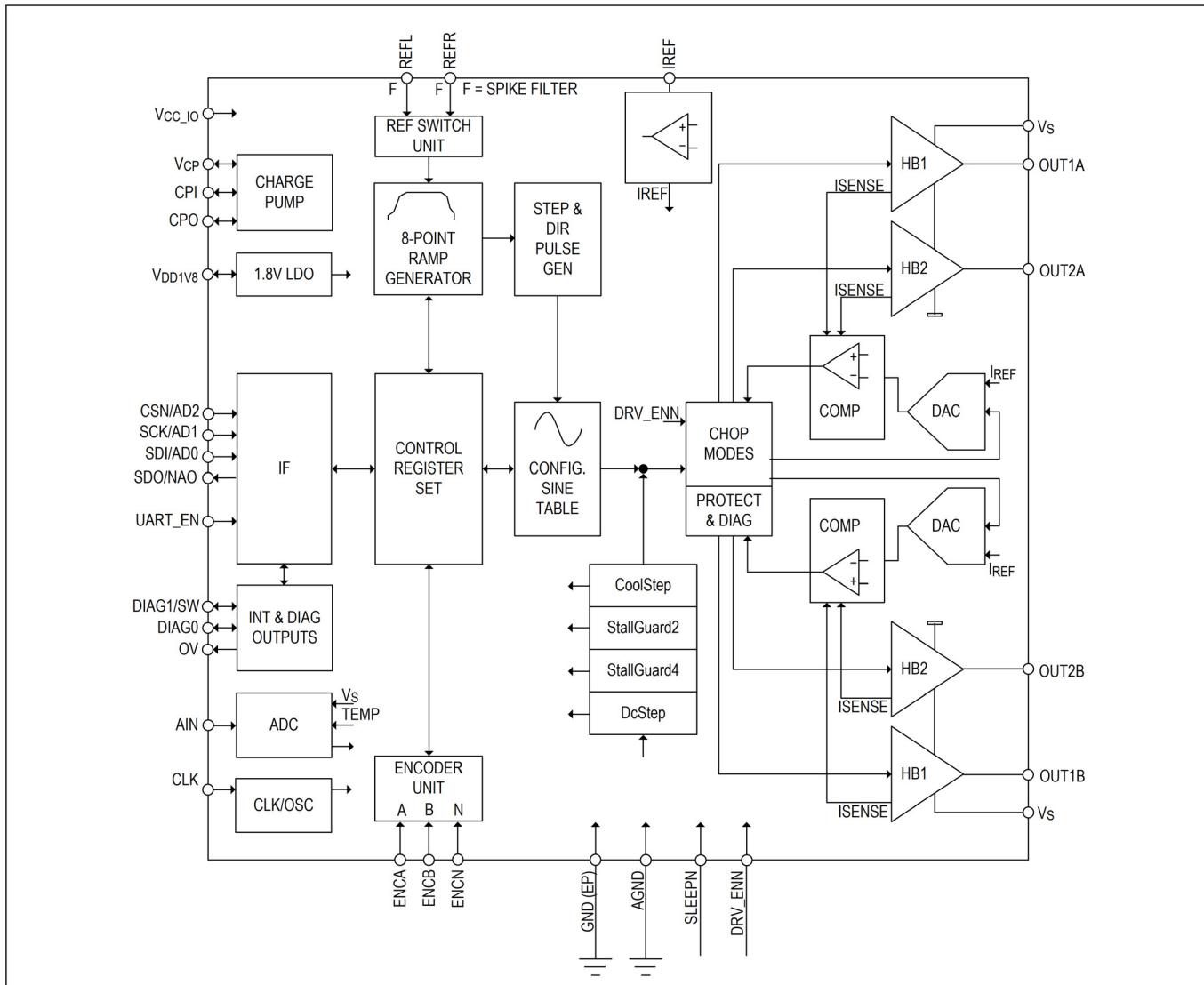


図 1. ブロック図

## 詳細

## 動作原理

## フル機能モーション・コントローラおよびドライバ

TMC5240 モーション・コントローラおよびドライバ・チップは、CPU とステッピング・モータの間のインターフェイスとなるインテリジェントな電源コンポーネントです。すべてのステッピング・モータ・ロジックが TMC5240 に完全に内蔵されています。モータを制御するソフトウェアは不要です。単に目標ポジションを指定するだけです。TMC5240 は、ドライバおよびコントローラのシステムオンチップ集積により可能となっている、数多くのユニークな強化機能を備えています。TMC5240 の 8 点ランプ・ジェネレータは、StealthChop、CoolStep、StallGuard、DcStep、SpreadCycle を自動で使用し、あらゆるモータ動作を最適化します。

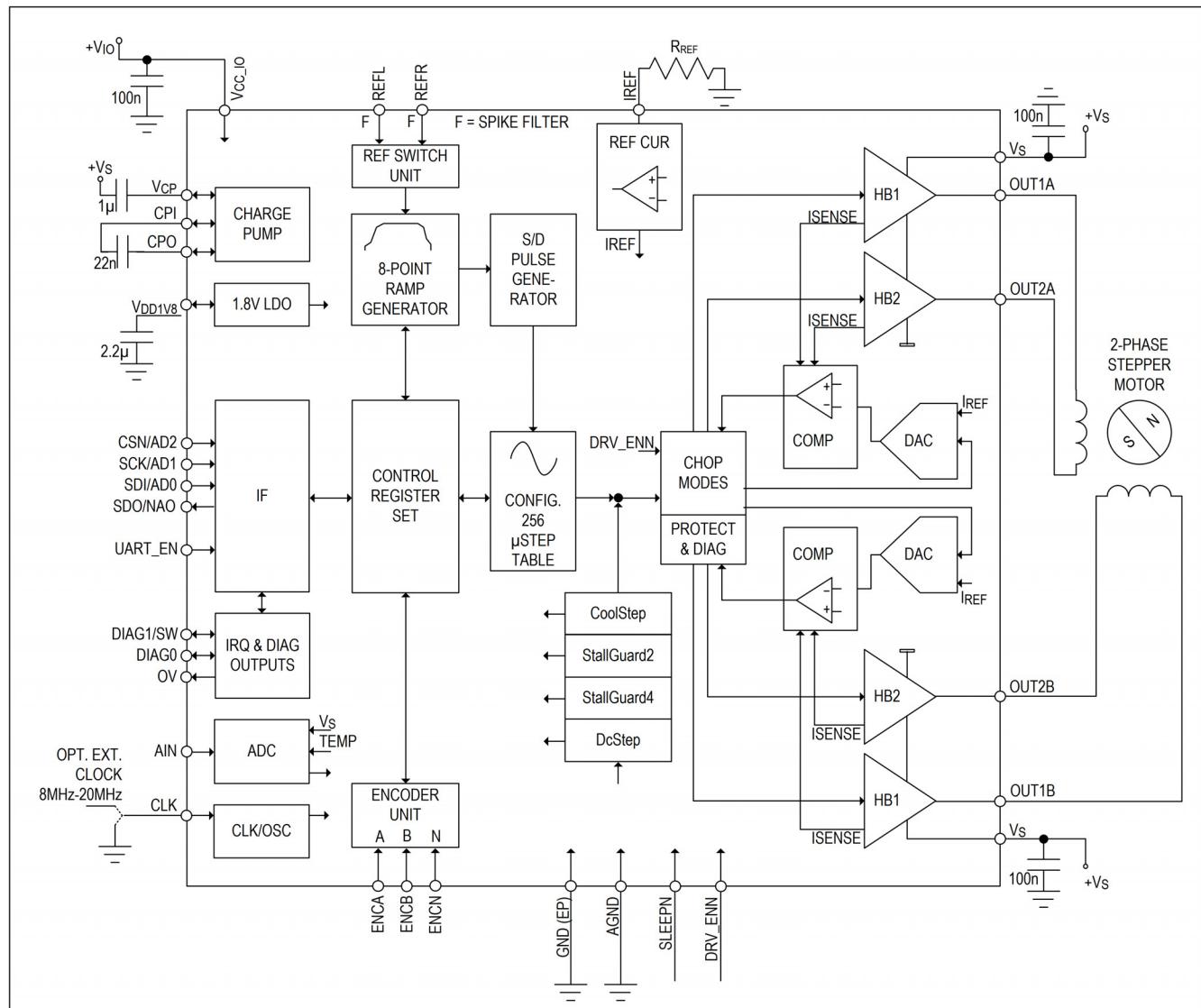


図 2. 代表的な外部コンポーネントを使用したブロック図

## 主要コンセプト

TMC5240 は、ADI-Trinamic 製品独自の高度な機能を各種備えています。これらの機能は、多くのステッピング・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。

<i>StealthChop2</i>	モータの無音動作および無音停止を実現する、無騒音高精度チョッパー・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 を小停止モータ電流まで拡張します。
<i>SpreadCycle</i>	動的動作を最大限に高める高精度なサイクルごとの電流制御。
<i>StallGuard2</i>	SpreadCycle 用のセンサーレス・ストール検出および機械的負荷測定。
<i>StallGuard4</i>	StealthChop 用のセンサーレス・ストール検出および機械的負荷測定。
<i>CoolStep</i>	モータやドライバの効率を最大にして、温度上昇を最小限に抑えるようモータ電流を適応させるために、StallGuard 測定を使用します。

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

## 制御インターフェイス

TMC5240 は、SPI インターフェイスおよび UART ベースの単線式インターフェイスの両方に対応し、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イスの組み合わせの選択は、UART\_EN 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イス選択に応じて GND または V<sub>CC\_IO</sub> にハードワイヤ接続できます。

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

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

## 内蔵 8 点モーション・コントローラ

内蔵の 32 ビット・モーション・コントローラは、モータ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、加速度値および減速度値と最大運動速度で構成されます。始動速度と停止速度が、速度スレッショ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合わせてサポートされるため、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、即時対応が可能です。

## 利点

- 柔軟なランプ・プログラミング
- 加減速に対しモータ・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
- 停止およびストール状態への即時反応

## 自動停止パワー・ダウ

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

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

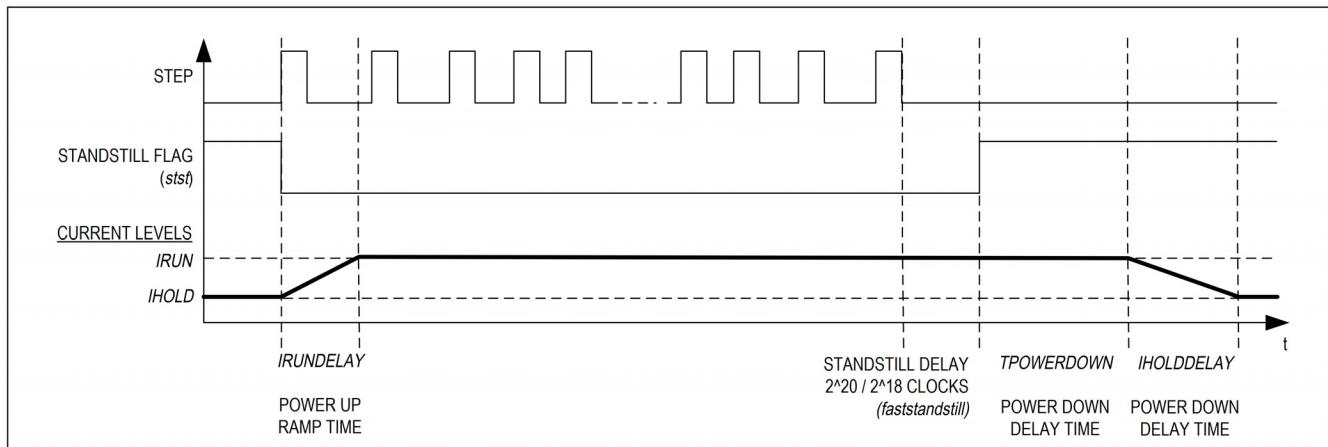


図 3. 停止時およびランプアップ時の自動モータ電流制御

### StealthChop2 および SpreadCycle ドライバ

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

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

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

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

SpreadCycle はサイクルごとの高度なチョッパー・モードです。広い速度範囲および負荷範囲にわたり、円滑な動作と良好な共振減衰を可能にします。SpreadCycle チョッパー方式は、高速減衰サイクルを自動的に統合・調整するため、確実にゼロ交差性能を実現します。

### 利点

- 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

### StallGuard2/4 – 機械的負荷検出

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、負荷がモータをストールさせる下限未満の場合における、ストール検出やその他の用途 (CoolStep 負荷適応型電流低減など) で用いることができます。

この機能でより多くの情報が得られるため、センサーレス・ホーミングや駆動機能の診断などの機能が可能になります。StallGuard2 は SpreadCycle チョッパーと組み合わせられますが、StallGuard4 を StealthChop2 と組み合わせるには別の原理を用います。

### CoolStep – 負荷適応型電流制御

CoolStep は、最適な電流でモータを駆動します。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 では最適な電流でモータを駆動するため、トルク・リザーブが ca (基本動定格荷重) 50% の標準的な動作と比較して、モータ効率が向上します。

## 利点

- 最高のエネルギー効率、消費電力は最大 75% 低減
- モータの発熱が減少
- 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・インターフェイス

TMC5240 は、外部インクリメンタル・エンコーダ用にエンコーダ・インターフェイスを備えています。エンコーダは、モーション・コントローラのホーミングに（リファレンス・スイッチの代替として）使用したり、エンコーダ・ポジションとランプ・ジェネレータ・ポジションの整合性チェックをオンザフライで行うために使用したりできます。プログラマブル・プリスケーラを用いると、エンコーダの分解能をモータの分解能に合わせることができます。32 ビットのエンコーダ・カウンタを備えています。

## SPI インターフェイス

### SPI データグラム構造

TMC5240 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 の倍数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の間中、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 バイトのデータで構成されます。これにより、レジスタ・セットとの間で直接、32 ビットのデータ・ワード通信を行うことができます。各レジスタは、使用するデータ・ビットが 32 ビット未満であっても、32 データ・ビットを通じてアクセスできます。

簡単化のため、各レジスタは 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は 0 です。
- 書込みアクセスでは、アドレス・バイトの最上位ビットは 1 です。

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

表 1. SPI データグラム構造

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

### 読出し／書込みの選択 (WRITE\_notREAD)

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

読み出しアクセス要求のデータグラムはダミー書き込みデータを使用します。読み出しだけは、その後の読み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転送されます。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。

TMC5240との間でデータの読み書きが行われる場合は必ず、返されるMSBにはSPIのステータスが含まれます。SPI\_STATUSは8個のうちのいくつかの選択されたステータス・ビットです。

例：

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

表2. SPI読み出し/書き込みフローの例

ACTION	DATA SENT TO TMC5240	DATA RECEIVED FROM TMC5240
read XACTUAL	0x2100000000	0xSS & unused data*
read VACTUAL	0x2100000000	0xSS & XACTUAL
write VMAX = 0x00ABCDEF	0xA700ABCDEF	0xSS & XACTUAL
write VMAX = 0x00123456	0xA700123456	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ピングされ、次のSPI転送で使用できます。

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

ビット	名称	コメント
7	status_stop_r	RAMP_STAT[1] – 1: 停止右側スイッチのステータスを指示 (モーション・コントローラ専用)
6	status_stop_l	RAMP_STAT[0] – 1: 停止左側スイッチのステータスを指示 (モーション・コントローラ専用)
5	position_reached	RAMP_STAT[9] – 1: 目標ポジションに到達したことを指示 (モーション・コントローラ専用)
4	velocity_reached	RAMP_STAT[8] – 1: 目標速度に到達したことを指示 (モーション・コントローラ専用)
3	standstill	DRV_STATUS[31] – 1: モータ停止を指示
2	sg2	DRV_STATUS[24] – 1: StallGuard フラグがアクティブであることを指示
1	driver_error	GSTAT[1] – 1: ドライバ・エラーを指示 (GSTATの読み出しでクリア)
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)

## 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

TMC5240のSPIバスには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の立ち上がりエッジでSDIからのデータをラッピングし、立ち下がりエッジの後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。TMC5240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、複数チップをデイジーチェーン接続できます。

全バス・トランザクション中、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ると、CSNの立ち上がりエッジの前に受信された最後の40ビットのみが、コマンドとして認識されます。

### SPIのタイミング

SPIの最大周波数は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。SPIバス・トランザクションのタイミング・パラメータを図に示します。タイミング値はECの表に記載されています。

SPIインターフェイスはSPI MODE 3を使用します。

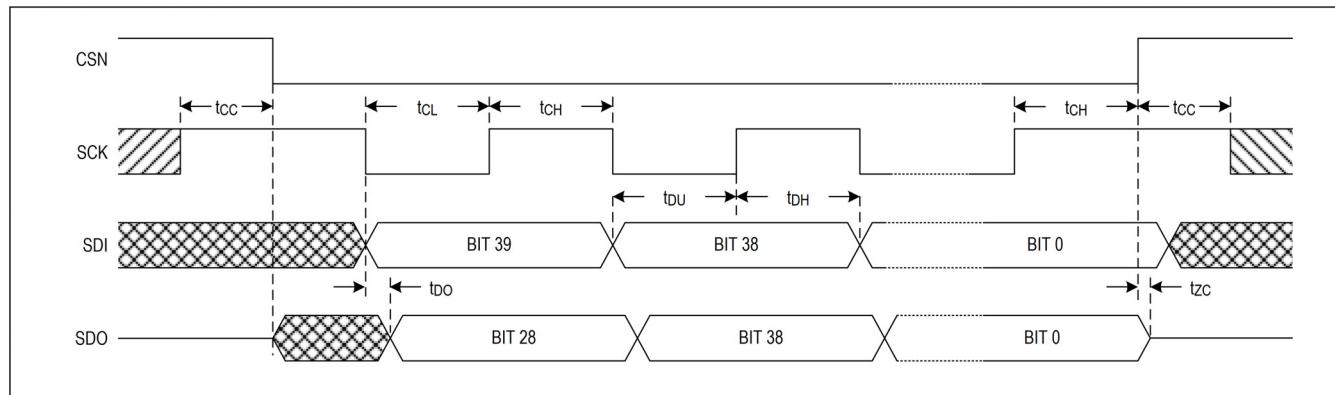


図 4. SPI タイミング図

### UART 単線式インターフェイス

UART 単線式インターフェイスでは、任意のマイクロコントローラ UART を用いて TMC5240 を制御できます。これは、RS485に基づくインターフェイスと同様に、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、長いインターフェイス距離（2つのPCB間のケーブルを介する場合など）を接続できます。自動ボーラート検出機能を備えているため、このインターフェイスは容易に導入可能です。

### データグラム構造

#### 書き込みアクセス

表 4. UART 書込みアクセス・データグラム構造

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

同期ニブルが TMC5240 との間の伝送の前に置かれ、最初の伝送バイトに組み込まれた後にアドレス指定バイトが置かれます。伝送ごとに、内部ボーラート分周器とUARTホストクロックの同期が可能です。実際のボーラートは適応型であり、内部クロック周波数の変化は補償されます。そのため、ボーラートは有効な範囲内で自由に選択できます。伝送される各バイトは、開始ビット（DIAG1/SWのロジック0、ロー・レベル）で始まり、停止ビット（DIAG1/SWのロジック1、ハイ・レベル）で終了します。ビット時間は、開始ビットの始まり（1から0への遷移）から同期フレームの終了（ビット2およびビット3の1から0への遷移）までの時間を計測して計算されます。すべてのデータは送信バイト単位です。32ビット・データ・ワードは最上位バイト・ファーストで伝送されます。

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

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

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

レジスタ・アドレスのビット 7 は読み出し（0）アクセスまたは書き込み（1）アクセスを識別します。例：アドレス 0x10 は書き込みアクセスでは 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 ホストは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、タイムアウト（12 ビット分の時間）の原因となります。この間はデータ・ラインがアイドル状態になる必要があります。CRC の誤りなど、その他のエラーも同様に処理されます。これにより、エラー状態の後の伝送を安全に再同期できます。このメカニズムにより、ボー・レートが直前の値の 15% 未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。

受け入れられた書き込みデータグラムごとに、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リソリッジされます。データグラム・カウンタを読み出すことで、UART ホストは初期化シーケンスまたは単一書き込みアクセスが成功したかどうかをチェックできます。読み出しアクセスではカウンタは変更されません。

## 読み出しアクセス

表 5. UART 読出しアクセス要求データグラムの構造

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

読み出しアクセス要求データグラムの構造は、書き込みアクセスのデータグラム構造と同じですが、使用するユーザ・ビットの数が少なくなっています。その機能は、UART ノードのアドレス指定と、読み出しアクセスに必要なレジスタ・アドレスの伝送です。TMC5240 は、UART ホストが読み出し要求のために用いるボー・レートと同じレートで応答します。

ホストからノードへのクリーンなバス遷移を確保するために、TMC5240 は読み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てその時間が経過した後に最初の返答バイトを送り、次いで読み出し要求を送ります。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 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み出しアクセス時に、アドレス指定されていないノードが伝送エラーを検出する可能性があります。

表 6. UART 読出しアクセス返答データグラムの構造

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

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

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

## CRC の計算

8 ビット CRC 多項式を用いて、読み出しおよび書き込み両方のアクセスをチェックします。これにより、最大 8 個のシングルビット・エラーを検出できます。初期値がゼロの CRC8-ATM 多項式を、同期バイトとアドレス指定バイトを含む LSB～MSB に適用します。同期ニブルは常に正しいと仮定されています。TMC5240 は、固有のノード・アドレスを含む、正しく伝送されたデータグラムに対してのみ、応答します。書き込みアクセス・データグラムが正しく受信されるごとに、データグラム・カウンタが増加します。

$$CRC = x^8 + x^2 + x^1 + x^0$$

シリアル計算例

**CRC = (CRC << 1) OR (CRC.7 XOR CRC.1 XOR CRC.0 XOR [new incoming bit])**

## CRC 計算の C コード例

```
void swuart_calcCRC(UCHAR* datagram, UCHAR datagramLength)
{
    int i,j;
    UCHAR* crc = datagram + (datagramLength-1); // メッセージの最後のバイトにある CRC
    UCHAR currentByte;
    *crc = 0;
    for (i = 0; i<(datagramLength-1); i++) {           // メッセージの全バイトに対し実行
        currentByte = datagram[i];                      // アレイから送信されるバイトを取得
        for (j = 0; j<8; j++) {
            if ((*crc >> 7) ^ (currentByte&0x01)) // CRC に基づく XOR 操作の結果を更新
            {
                *crc = (*crc << 1) ^ 0x07;
            }
            else
            {
                *crc = (*crc << 1);
            }
            currentByte = currentByte >> 1;
        } // CRC ビット対象
    } // メッセージ・バイト対象
}
```

## UART 信号

TMC5240 の UART インターフェイスは 5 種類の信号で構成されます。UART モードでは、各ノードは、その固有アドレスを持つデータグラムを正しく受信したかどうかを確認するために、単線ピン DIAG1/SW を連続的にチェックします。ピンはこの期間、入力に切り替わります。前述のように、このピンは同期ニブルに基づくポー・レートに対応します。読み出しアクセスの場合、DIAG1/SW の出力ドライバをオンに切り替え、同じポー・レートで応答を送信します。

表 7. TMC5240 UART のインターフェイス信号

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

### 複数ノードのアドレス指定

1~8 個の TMC5240 を单一の UART バス・インターフェイスを用いてホストがアドレス指定する場合、単純なハードウェア・アドレス選択を使用できます。個々の UART ノード・アドレスの設定は、UART アドレスピン (SDI、SCK、CSN) を V<sub>CC</sub>\_IO および GND に接続することで行います。

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

- すべてのアドレス・ピンおよび最初の TMC5240 の SDI/AD0 を GND に接続します。
- 最初の TMC5240 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 TMC5240 をその固有のノード・アドレスにプログラムします。**注**：ノードがそのノード・アドレスで初期化された後は、次のノードのアドレス[0]ピン (SDI/AD0) に接続されているその SDO/NAO 出力をロジック 0 にプログラムして、次のノードをすべての後続のノードと区別できるようにする必要があります。
- これで、2 番目のノードがアクセス可能となり、固有のノード・アドレスを持つことができます。これ以降のノードも固有ノード・アドレスに続けてプログラムできます。

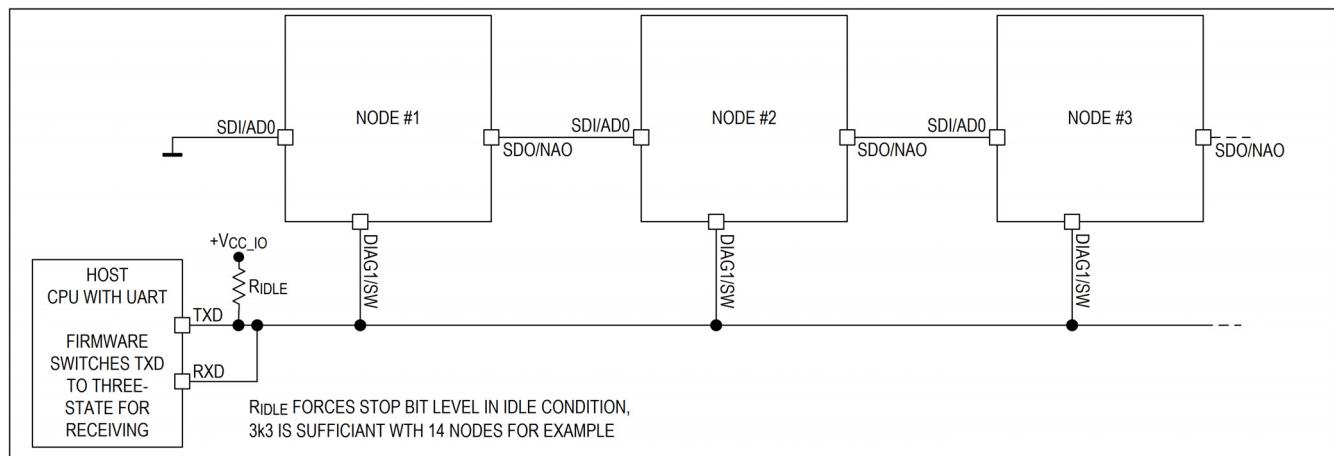


図 5. UART のデイジーチェーン例

表 8. 最大 255 ノードをアドレス指定する UART 例

PHASE	NODE #1	NODE #2	NODE #3
Addressing phase 1	address 0, NAO is high	address 1	address 1
Addressing phase 2	program to address 254 & set NAO low	address 0, NAO is high	address 1
Addressing phase 3	address 254	program to address 253 & set NAO low	address 0
Addressing phase 4	address 254	address 253	program to address 252 & set NAO low
Addressing phase x	continue procedure		

### 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、モータは絶対的に無雑音です。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。オプションの設定では、特別な場合の設定を調整することや、自動適応アルゴリズムの初期値を設定することが可能ですが、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、お使いの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が、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。このように、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・ダウントできます。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。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)。次の図のフローチャートに、この調整プロシージャを示します。

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

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>ピン Vs 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>大量のバック EMF が生じ、最大稼働電流に達する可能性のある速度にモータを駆動。条件 :           <ul style="list-style-type: none"> <li><math>1.5 \times PWM\_OFS\_AUTO \times (IRUN + 1)/32 &lt; PWM\_SCALE\_SUM &lt; 4 \times PWM\_OFS\_AUTO \times (IRUN + 1)/32</math></li> <li><math>PWM\_SCALE\_SUM &lt; 255</math></li> </ul> </li> </ul> <p>ヒント : 代表的な範囲は 60RPM~300RPM です。</p>	$\pm 1$ の変更には 8 個のフルステップが必要です。 PWM_GRAD_AUTO の最適値が 50 以下の代表的なモータの場合、デフォルト値 0 から開始するには最大 400 フルステップが必要です。

### ヒント :

自動調整用の最適条件を決定するには、評価用ボードを用います。

初期調整パラメータを求めるには、ファームウェアの初期化用の 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 :

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、評価用ボードを用いて最適な調整条件をチェックしてください。モータ・タイプごとに定める PWM\_OFS および PWM\_GRAD の設定値には、初期値を用いることを推奨します。

GLOBALSCALER や Vs 電圧を変更すると、自動調整プロセスの結果は無効になります。モータ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た場合は必ず、自動調整は変更された条件に適応します。

### StealthChop2 のオプション

モータ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 電圧がスケーリングされます。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、そのバック EMF (例えばその速度に正比例)、電源電圧の実際のレベルなど、いくつかの要素が影響します。2 つのモードの PWM レギュレーションが使用できます。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を提供します。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には自動モードを推奨します。

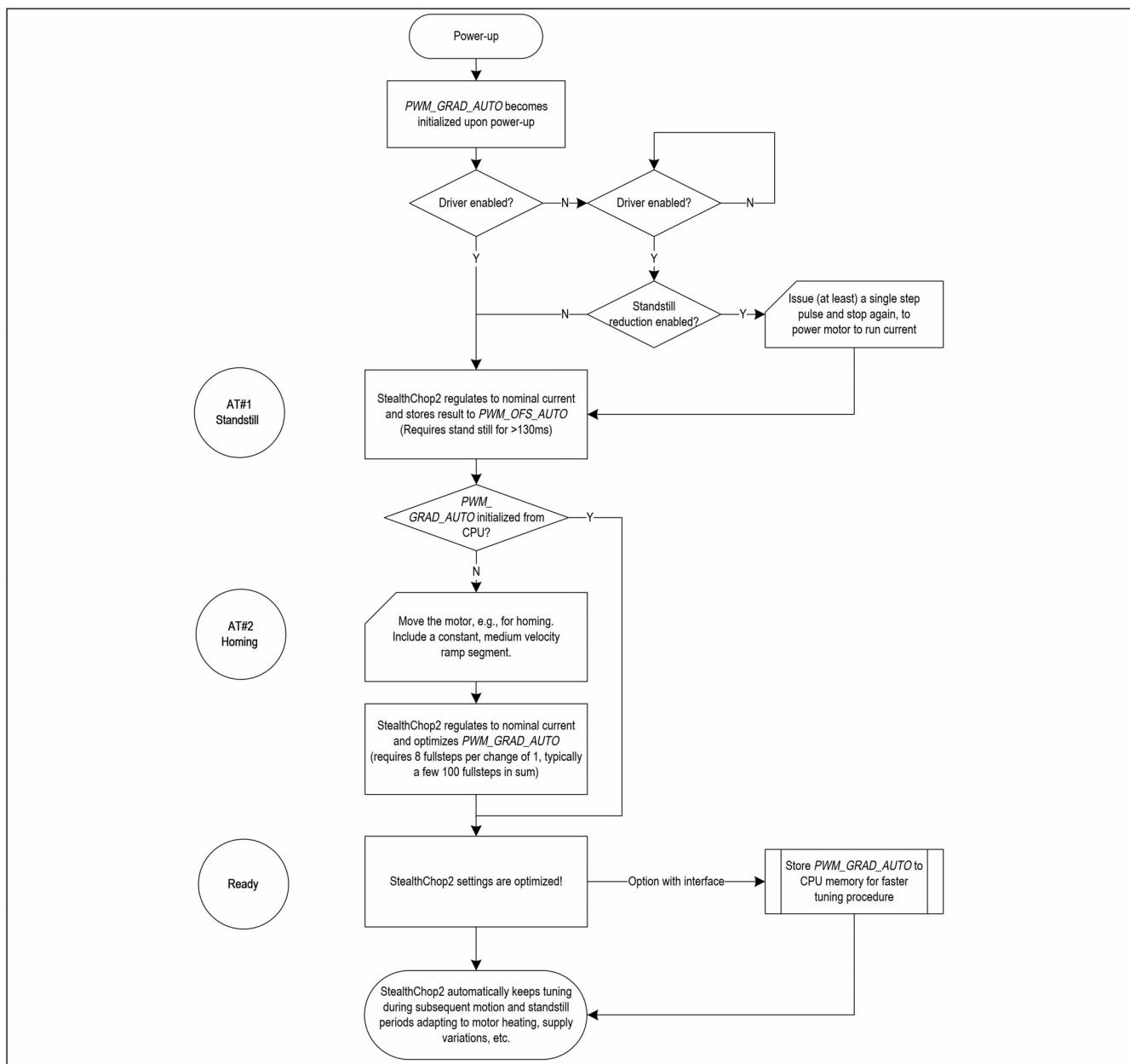


図 6. StealthChop2 の自動調整手順

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応答するためには、自動調整モードで動作させてください。

非自動モード (*pwm\_yscale = 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、インターフェイ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源の周波数に合わせることができます。ほとんどのアプリケーションでは、20kHz~50kHz の範囲での設定が適しています。これにより、低電流リップルおよび高速時の良好な性能と動的な消費電力とのバランスをとることができます。

表 10. StealthChop2 用の PWM 周波数の選択 (太字を推奨)

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

### 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\_yscale = 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ー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、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<sub>REF</sub> 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くてはなりません。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 で高加速度にするには、*PWM\_REG* の設定を大きくする必要があります。ホーミングの速度と加速度を注意深く選択することで、多くの場合、レギュレーション勾配は最小設定 (*PWM\_REG = 1*) にすれば十分です。*PWM\_REG* の設定は、必要な最高速の加速や減速ランプについて最適化する必要があります (次の2つの図を比較してください)。

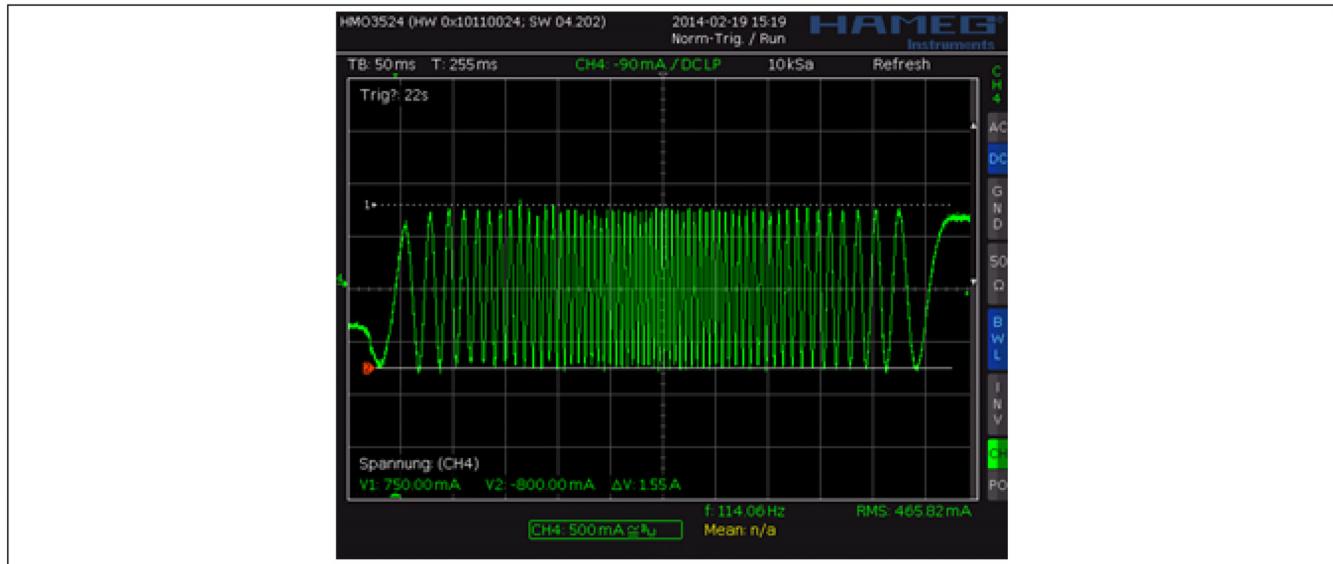


図 7. StealthChop2 : *PWM\_REG* の設定は良好

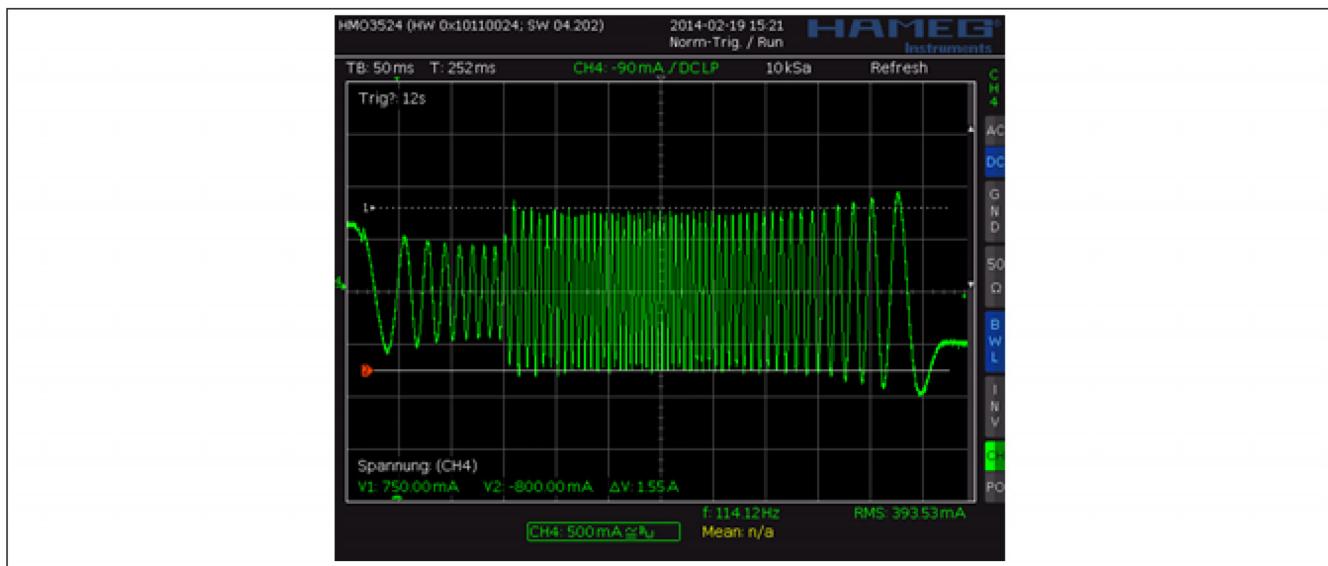


図 8. StealthChop2 : AT#2 時の PWM\_REG の設定が過小

AT#2 フェーズおよび完了した自動調整プロシージャでの PWM\_REG 設定値（または PWM\_OFS および PWM\_GRAD に対する非自動設定値）の質は、次の図に示すように、加速フェーズ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。

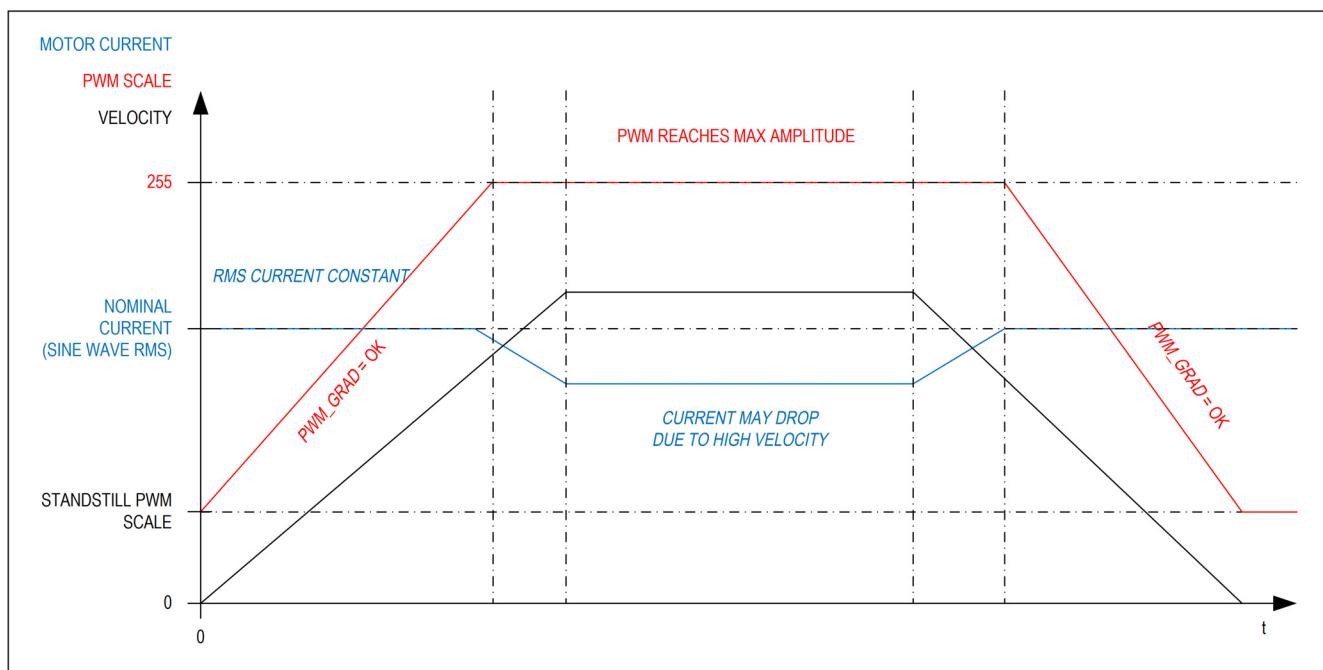


図 9. 正しく決定された PWM\_GRAD(\_AUTO)および PWM\_OFS(\_AUTO)

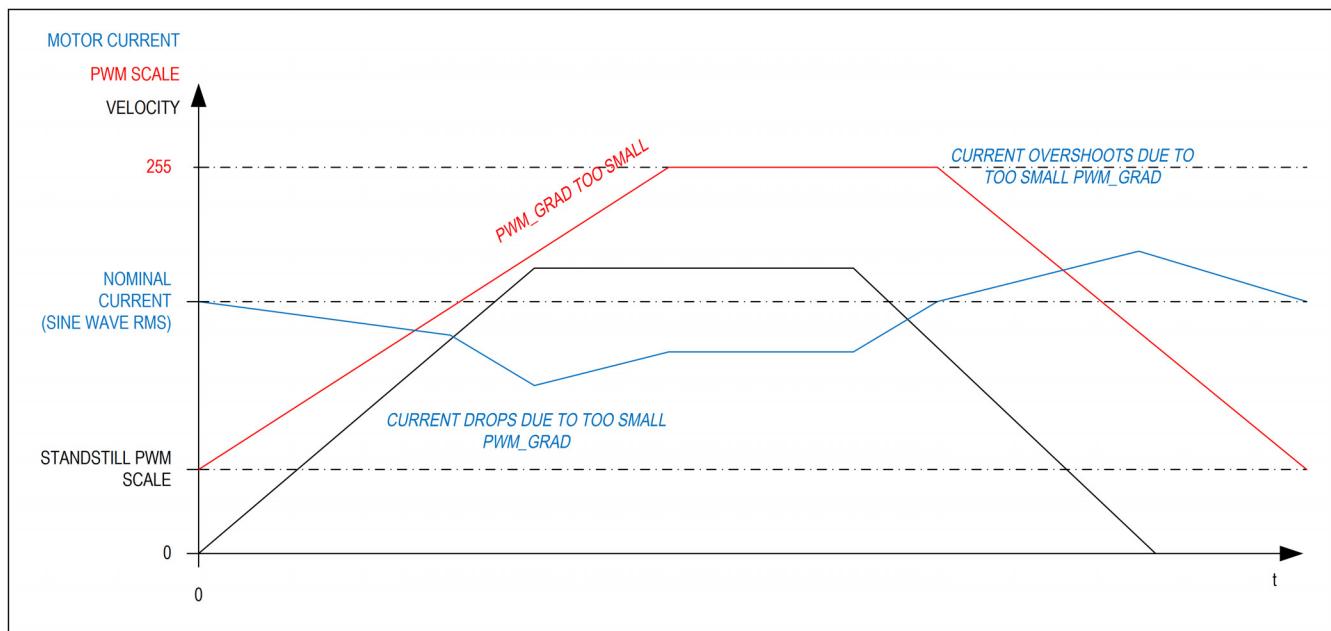


図 10. 過小な PWM\_GRAD 設定の例

## 下限電流値

`pwm_meas_sd_enable` の設定に応じて、StealthChop2 の電流レギュレータ原理ではモータ電流レギュレーションに下限値を設けます。チョッパーがオンのフェーズ時 (`pwm_meas_sd_enable = 0`) にコイル電流を測定する際、コイル電流のレギュレーションが可能な最小チョッパー・デューティサイクルは、TBL とチョッパー周波数設定値によって設定されるプ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最小コイル電流は、電源電圧およびチョッパー周波数と共に増加します。プ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（停止パワー・ダウンのため）は、`PWM_OFS_AUTO` および `PWM_GRAD_AUTO` に基づき自動で行われます。これらは非自動電流スケーリングでの、`PWM_OFS` および `PWM_GRAD` に対応するものです。フリーホイーリング・オプションを使えば、モータ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。

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

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

$I_{LowerLimit}$  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。この下限値では必要な設定に達するのに不十分な場合は、`pwm_meas_sd_enable = 1` と設定されていることを確認してください。

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

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

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

これは、関連する設定をすべて考慮すると、自動調整のモータ目標電流値は  $225mA$  以上としなくてはならないことを意味します。この下限電流値は、`GLOBALSCALER` を通じたモータ電流の変更の場合にもあてはまります。

## 注意：

自動調整の場合、下記のコイル電流下限値が適用されます。

*IRUN* ≥ 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 LOWERLIMIT* : 自動調整用の *pwm\_meas\_sd\_enable* (レジスタ *PWM\_CONF[22]*) のビットの設定に応じて、コイル電流の下限値が適用されます。自動調整の AT#1 フェーズでのモータ電流はこの下限値を上回っていることが必要です。*I 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 *I HOLD* 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。

電流下限値を克服するには、*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

速度ベースのスケーリングは、クロック・サイクル単位で測定した 1 ステップおきの時間に基づき (例えば *TSTEP* に基づき)、StealthChop2 振幅をスケーリングします。このコンセプトは、基本的に、電流測定を必要としません。レギュレーション・ループが不要であるためです。純粋な速度ベースのスケーリングは、*pwm\_autoscale* を 0 に設定した場合のみ、プログラミングを通じて可能です。基本的な考え方とは、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り、そのため  $I = U/R$  の基本公式に基づき、目標電流を生じさせるには一定の電圧振幅が必要です。つまり、コイルの抵抗  $R$  と、PWM 値に比例する電源電圧  $U$  から、電流  $I$  が求められます。*PWM\_OFS* の初期値は次式で計算できます。

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

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

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

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

モータ速度の増加時には、モータのバック EMF 電圧が増加します。バック EMF 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。TMC5240 は、これを補償するために、2 つめの速度依存係数 (*PWM\_GRAD*) を備えています。このモードにおける全体的な実効 PWM 振幅 (*PWM\_SCALE\_SUM*) は、次式のように、マイクロステップ周波数に依存して自動的に計算されます。

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

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

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

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

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

*C<sub>BEMF</sub>* は、モータのバック EMF 定数で、単位は V/ラジアン/秒です。

*MSPR* は、1/256 マイクロステップ分解能を基準とする 1 回転あたりのマイクロステップ数です。例えば、1.8° のモータの場合、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。

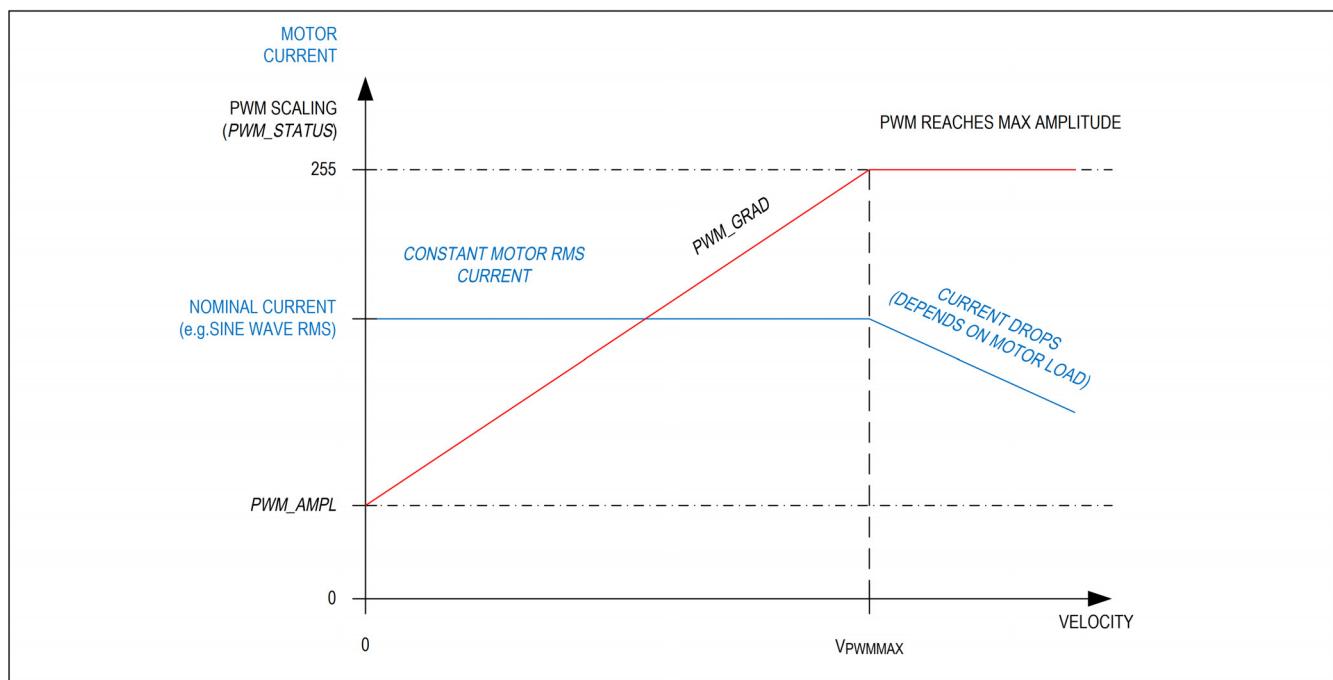


図 11. 速度ベースの PWM スケーリング (pwm\_yscale = 0)

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。

#### モータのバック EMF 定数の解釈 :

バック EMF 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、バック EMF 定数  $C_{BEMF}$  の数値は、トルク定数の数値と同じ値です。例えば、トルク定数が 1Nm/A のモータの  $C_{BEMF}$  は 1V/rad/s です。そのようなモータを 1rps (1rps = 每秒 1 回転 = 6.28 rad/s) で回転させると、発生するバック EMF の電圧は 6.28V となります。したがって、バック EMF 定数は次のように計算できます。

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

$I_{COILNOM}$  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。

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

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

#### StealthChop2 と SpreadCycle の結合

高速度運動が必要なアプリケーションでは、SpreadCycle の方が、高速レンジにおいてより安定な動作をもたらす場合があります。無騒音動作と最大の動的性能を組み合わせるため、TMC5240 では、速度スレッショルドに基づき StealthChop2 と SpreadCycle を組み合わせることができます。この場合、StealthChop2 は低速時のみ有効です。

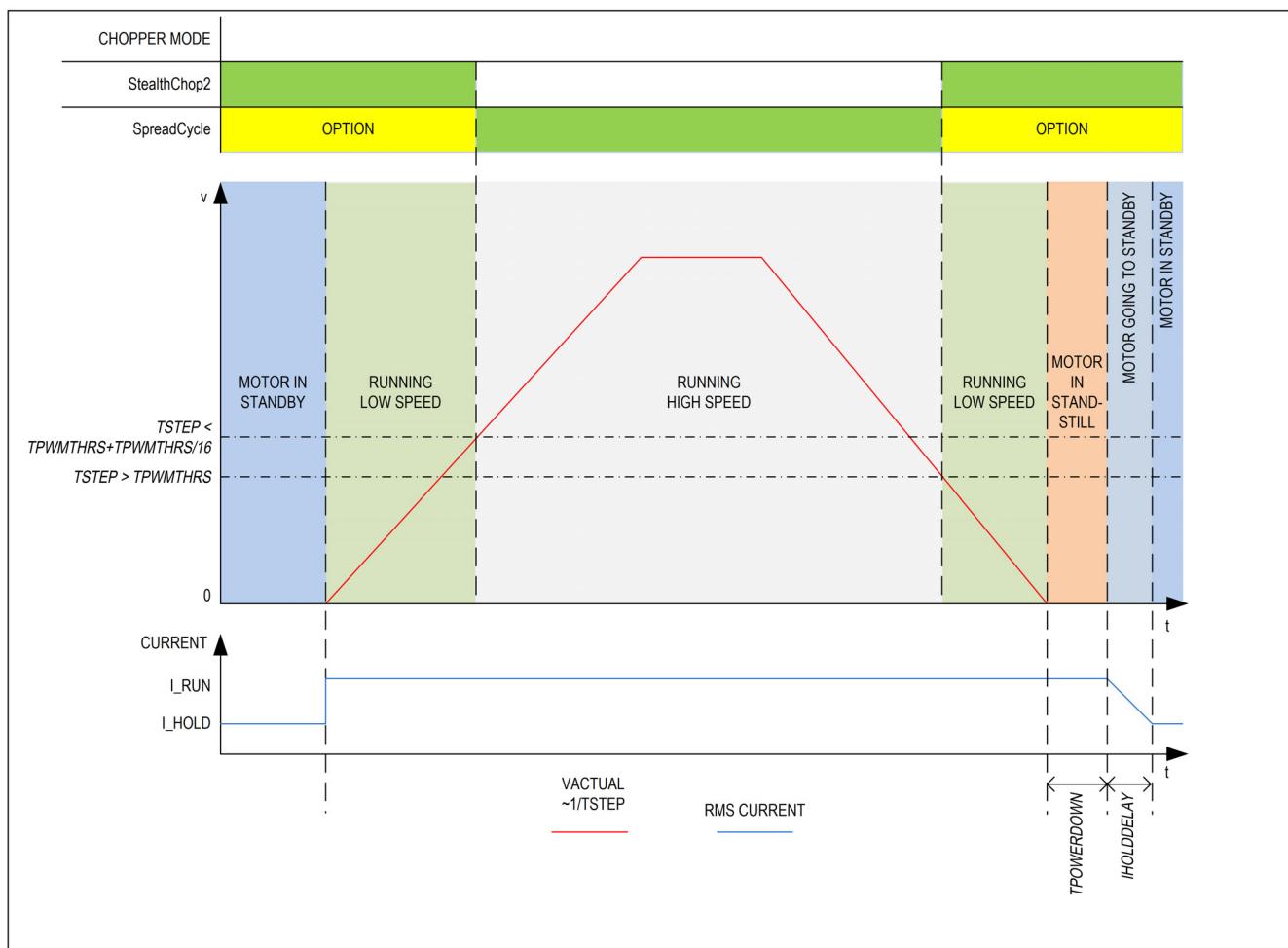


図 12. SpreadCycle への切り替えオプションのための TPWMTHRS

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

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

#### SpreadCycle へのジャークのない切り替え：

ジャークが発生するのは高速で切り替えを行う場合です。それは、モータのバック EMF (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速度が速い場合、このジャークが高い強度で発生します。ジャークが大きい場合、一時的な過電流状態 (モータのコイル抵抗に依存) が生じる場合もあります。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。そのため、SpreadCycle と StealthChop2 の間でドライバを切り替える場合には、ジャークを考慮してください。TPWMTHRS で制御された自動切り替えでは、ドライバは StallGuard4 を用いて位相シフトを定めることにより自動的にジャークを除去できます。これは、速度が切り替えスレッショルド未満に戻るまで、SpreadCycle に対し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てはなりません。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。このように、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、チャッパーが別のモードに切り替わる位相の間に、速度レッショ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になるか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つ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 チャ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、モータ停止といったバック EMF 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 の間にわたり大きく増加します。低速時、バック EMF は電源電圧に比べてごくわずかにすぎないため、短絡検出がトリガされるおそれはありません。

ドライバ段を、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ョルドを 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において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 チャッパーを用いてオンデマンド開放負荷テストを行ってください。この方法が最も安全な結果をもたらします。StealthChop2 では、PWM\_SCALE\_SUM をチェックするとコイル抵抗が適切かどうか確認できます。

### モータの状態を示す PWM\_SCALE\_SUM

自動スケーリングで PWM\_SCALE\_SUM を読み出すことで、モータ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維持できません。

### フリーホイー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 をゼロに設定し、FREEWHEEL 設定を用いて目的のオプションを選択することでイネーブルできます。目的のオプションがイネーブルされるのは、TPOWERDOWN および IHOLDDELAY で指定された時間が経過した後です。短時間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。フリーホイーリング・オプションを用いると、フリーホイー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。しかし、受動ブレーキングでは、連続的なトルクが加えられている場合、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

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

表 11. StealthChop2 を制御するパラメータ

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

表 11. StealthChop2 を制御するパラメータ（続き）

PWM_OFS	ユーザ定義の PWM 振幅（オフセット）。PWM_OFFSET_AUTO の自動調整に対し、速度ベースのスケーリングおよび初期化の値を与えます。 デフォルト = 0x1D	0 ... 255	PWM_OFS = 0 に設定すると、電流設定に基づく線形電流スケーリング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、速度ベースのスケーリングおよび初期化の値を与えます。 デフォルト = 0	0 ... 255	
PWM_SCALE_SUM	実際の設定値によって決定される実際の PWM スケーリング。この値は、PWM_GRAD/OFS_AUTO 値用の 8 ビットよりも高精度（10 ビット）で示されます。 デフォルト = 0	0 ... 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。このフリーホイーリング・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、両コイルの短絡オプションにより受動ブレーキングが実現できます。 デフォルト = 0	0 1 2 3	通常動作 フリーイーリング コイルが LS ドライバを介して短絡 コイルが HS ドライバを介して短絡
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 スケーリング補正のリード・バック。調整時にはほぼ 0 に安定化します。 デフォルト = 0	-255 ... 255	（読み出し専用）SpreadCycle での動作時、スケーリング値はフリーズされます
PWM_GRAD_AUTO PWM_OFS_AUTO	自動調整のモニタリングと PWM_OFS および PWM_GRAD の初期値の決定が可能です。 デフォルト = 0	0 ... 255	（読み出し専用）
TOFF	モータ・ドライバの全般的なイネーブル。実際の値は StealthChop2 には影響しません。 デフォルト = 0	0 1 ... 15	ドライバ・オフ ドライバ・イネーブル
TBL	コンパレータのブランки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。より高い容量性負荷には 3 が必要となる場合もあります。設定値を下げるとき、StealthChop2 はより小さいコイル電流値となるよう安定化します。 デフォルト = 2	0 1 2 3	16 t <sub>CLK</sub> 24 t <sub>CLK</sub> 36 t <sub>CLK</sub> 54 t <sub>CLK</sub>

## SpreadCycle と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、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、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。次の図に、様々なチョッパー・フェーズを示します。

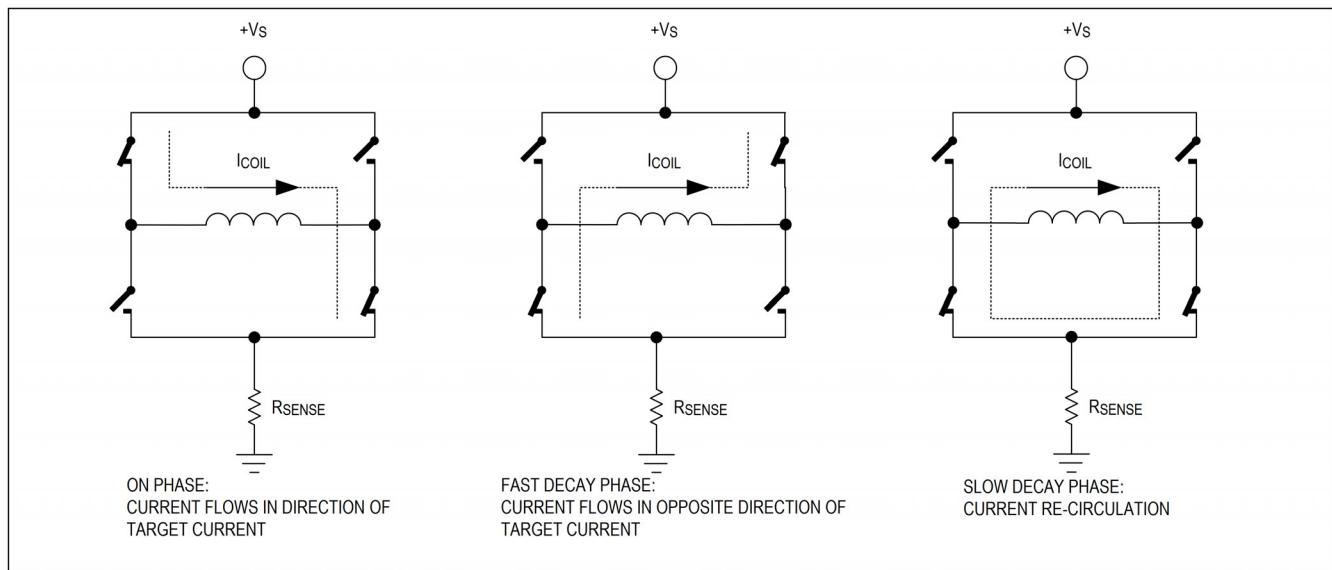


図 13. 代表的なチョッパー減衰フェーズ

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、モータの電気的な損失や電流リップルを低減するうえで重要です。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一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  $R_{DS(ON)}$  ベースの電流測定でスパイクが生じます。この期間（通常 1~2 マイクロ秒）、電流は測定できません。ブランки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。SpreadCycle と呼ばれる新しい高性能チョッパー・アルゴリズムと、定評のある定オフ時間チョッパー・モードです。定オフ時間モードは、オン、高速減衰、低速減衰の 3 つのフェーズを順番に繰り返します。SpreadCycle モードは、オン、低速減衰、高速減衰、第二低速減衰の 4 つのフェーズを順番に繰り返します。

チョッパー一周波数は、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。周波数が低すぎると、可聴ノイズを発生する可能性があります。周波数を上げると、モータの電流リップルは減りますが、周波数が高すぎると磁気損失が生じる場合があります。また、ドライバの消費電力も、周波数の増加によって上昇します。動的消費電力の原因となるスイッチング勾配の影響が高まるためです。そのため、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。チョッパー一周波数は、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 12. SpreadCycle および一般的な定オフ時間チョッパーを制御するパラメータ

パラメータ	説明	設定	コメント
TOFF	低速減衰時間（オフ時間）を設定します。この設定により、最大チョッパー一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、このパラメータは用いませんが、モータをイネーブルするために必要です。StealthChop2 で動作させる場合のみ、任意の設定が可能です。	1...15	オフ時間の設定値 $N_{CLK} = 24 + 32 \times TOFF$ (1 に設定すると 24 クロックの最小ブランкиング時間で動作します)
	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		

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

TBL	コンパレータのブランкиング時間を選択します。この時間は、スイッチング・イベントとリンク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、容量性負荷が高い場合には、設定値を2または3にする必要があります。  デフォルト = 2	0	16 $t_{CLK}$ 制限事項：この設定を使用できるのは、8MHz 以下の外部クロック発振器を組み合わせる場合のみです
		1	24 $t_{CLK}$ 制限事項：使用できるのは、内部クロックで用いる場合、または13MHz 以下の外部クロックを使用する場合です。
		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

### SpreadCycle チョッパー

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

各チョッパー・サイクルは、オン・フェーズ、低速減衰フェーズ、高速減衰フェーズ、第二低速減衰フェーズで構成されます。2 つの低速減衰フェーズおよびチョッパーあたり 2 つのブランки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、モータやドライバの低消費電力を実現するのに重要です。

低速減衰時間  $TOFF$  の開始値の計算例：

- 目標チョッパー周波数 : 25kHz
  - $TOFF = 1 / 25\text{kHz} \times 50 / 100 \times 1 / 2 = 10\mu\text{s}$
  - 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、次のようにになります： $TOFF = (t_{OFF} \times f_{CLK} - 12) / 32$
- クロックが 12MHz の場合、 $TOFF = 3.4$  という結果が得られるため、 $TOFF$  を 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、 $TOFF$  を 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 に設定すると共に短めの TBL 設定を用いると、良い結果が得られることがあります。

ヒステリシスの開始設定値になると、ドライバは最小限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、チョッパーは、電流を立上がりおよび立下がりのどちらの目標電流に対しても正確に電流を安定化できます。モータ・コイルに電流リップルを導入するために必要な時間は、チョッパー周波数の減少ももたらします。そのため、ヒステリシスの設定値を高くすると、チョッパー周波数は低下します。モータのインダクタンスにより、チョッパーがモータ電流の変化に追随する能力が制限されます。また、オン・フェーズの時間と高速減衰の時間は、ブランкиング時間より長くなくてはなりません。ブランки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値が小さすぎる場合、サイン波の波形をゼロ遷移付近でチェックすると、両半波の間に小さな棚形状が見られます。中程度の速度（毎秒 100 フルステップ～400 フルステップ）では、ヒステリシスの設定値が低すぎるとモータのハミングや振動が増加する原因となります。ヒステリシスの設定値が高すぎる場合は、チョッパー周波数が低下し、チョッパー・ノイズが増加しますが、波形には何ら好影響を及ぼしません。

いくつかの実験を行うとわかるように、設定はモータにはまったく依存しません。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ー・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ки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。これを実現するのが難しい場合は、オフ時間の設定値を減らすことができます。

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

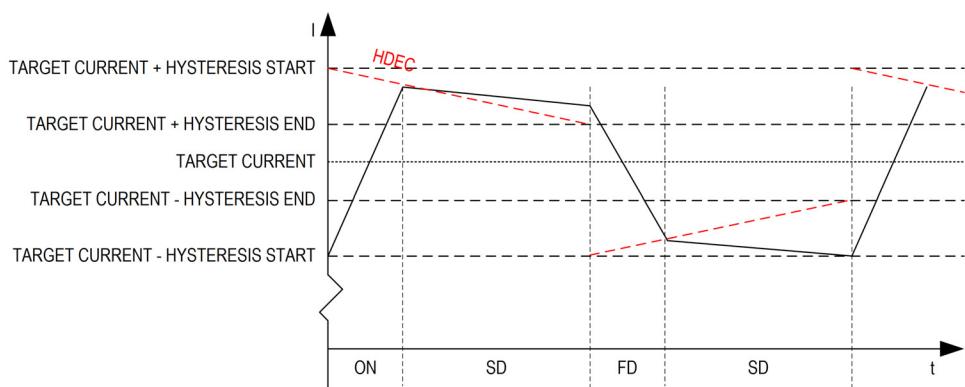


図 14.1 チョッパー・サイクルの間のコイル電流を示す、SpreadCycle チョッパーの概略図

表 13. SpreadCycle モードのパラメータ

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

HSTRT = 0かつHEND = 0の場合でも、TMC5240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。

例：

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

HEND = 6 (実効終了値を  $6 - 3 = 3$  に設定)

HSTRT = 0 (最小ヒステリシスを設定。1に設定されるので、 $3 + 1 = 4$ )

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

HEND = 0 (実効終了値を-3に設定)

HSTRT = 6 (ヒステリシス終了に対する実行開始値を+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正弦波の立下がり勾配に追随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ため、長すぎてはいけません。これを調整するには、オシロ・スコープを用いるか、様々な速度でモータ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。

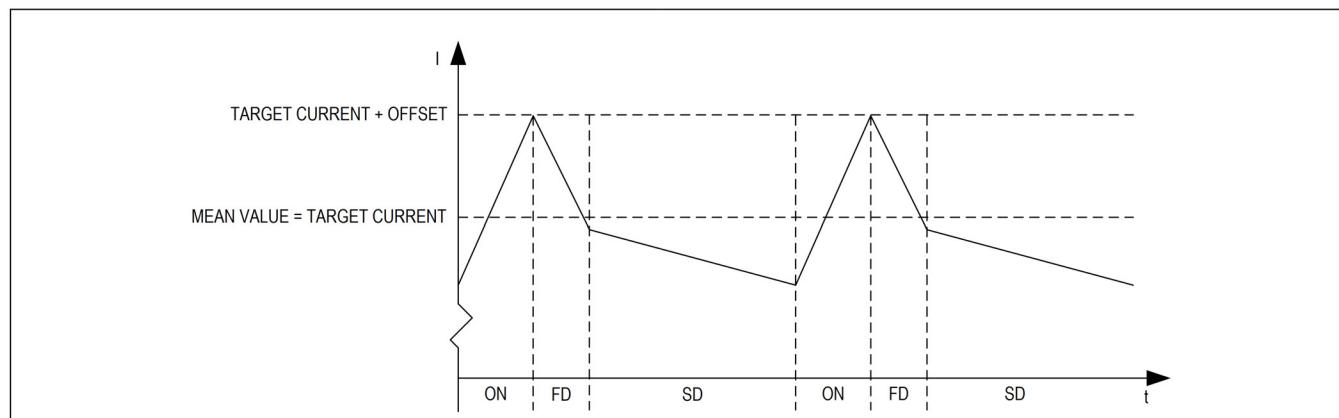


図 15. コイル電流を表すオフセットを持つ一般的な定オフ時間チョッパー

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

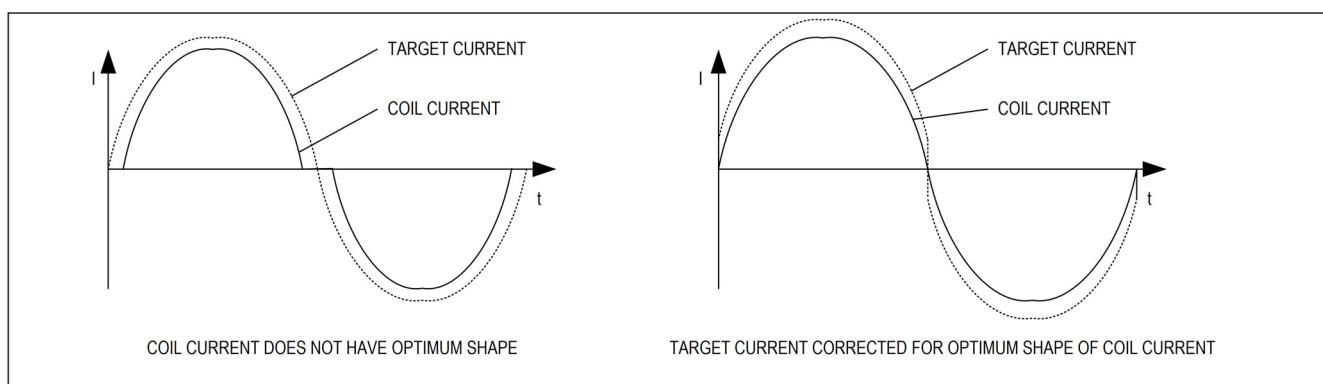


図 16. 一般的なチョッパーのゼロ交差とサイン波オフセットを使用した補正

表 14. 帰還抵抗の推奨値

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

## 内蔵電流検出機能

TMC5240 には、電力散逸のない電流検出機能が内蔵されています (ICS)。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。そのため、外付け検出抵抗に基づく主流のアプリケーションと比較して、ICS は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

TMC5240 では、モータの相電流を設定できます。次の表に示すパラメータは、電流スケーリングや電流のランプ・アップおよびランプ・ダウンに適応させることができます。

表 15. モータ電流を制御するパラメータ

パラメータ	説明	設定	コメント
IRUN	モータ動作中の電流スケーリング。内蔵の正弦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...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。 IRUN と比べ IHOLD では 16 より小さい値も可能です。  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、モータの TZEROWAIT 後のパワー・ダウンのためのクロック・サイクル数を、 $2^{18}$ クロックのインクリメントで制御します。0 = 即時パワー・ダウン、1~15 : $2^{18}$ 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 $4 \times 15 \times 2^{18}$ 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 1...15	IHOLD への即時パワー・ダウン  $1 \times 2^{18} \dots 15 \times 2^{18}$ クロックごとに電流をデクリメント
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・トルクを確立するには、短時間でのパワー・アップが重要ですが、わずかな遅延時間を設けることで、可聴ノイズを低減し、また、電源電流の不連続を防止できます。  デフォルト = 4	0 1...15	IRUN への即時パワー・アップ  電流インクリメント・ステップあたりの遅延は IRUNDELAY $\times$ 512 クロックの倍数

## フルスケール電流レンジの設定

フルスケール電流  $I_{FS}$  は、ピーク電流の設定値です。

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

精度 1% の標準的な低電力抵抗で十分です。

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

これは、できる限り最高の電流制御分解能を活用するために必要です。

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

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

比例定数  $K_{IFS}$  は、選択したフルスケール・レンジ設定 (DRV\_CONF レジスタのビット 1 および 0) に依存します。外付け抵抗  $R_{REF}$  の範囲は  $12\text{k}\Omega \sim 60\text{k}\Omega$  です。

$$I_{FS} = K_{IFS}(KV) / R_{REF}(\text{k}\Omega)$$

表 16.  $I_{FS}$  フルスケール・ピーク・レンジの設定値 ( $R_{REF} = 12\text{k}\Omega$  の場合の例)

REGISTER CONFIG	$K_{IFS}$ (A x kΩ)	MAX. FS SETTING (PEAK)	TYPICAL $R_{DS(ON)}$ (HS + LS)	NOTES
DRV_CONF bits 1..0				
11	36	3A	0.23Ω	Optimized efficiency and extended operating range up to 3A <sub>FS</sub> .
10	36	3A	0.23Ω	Optimized efficiency and extended operating range up to 3A <sub>FS</sub> .
01	24	2A	0.27Ω	Reduced operating range up to 2A <sub>FS</sub> . When high accuracy at lower current is required.
00 (default)	11.75	1A	0.40Ω	Reduced operating range up to 1A <sub>FS</sub> . When high accuracy at low current is required.

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

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

$R_{REF}$ (kΩ)	MAX FULL SCALE CURRENT (A RMS) BASED ON DRV_CONF BITS 1..0 SETTING AND $K_{IFS}$ (A x kΩ)			
	DRV_CONF BITS 1..0 = 11	DRV_CONF BITS 1..0 = 10	DRV_CONF BITS 1..0 = 01	DRV_CONF BITS 1..0 = 00
	$K_{IFS} = 36$	$K_{IFS} = 36$	$K_{IFS} = 24$	$K_{IFS} = 11.75$
12	2,12	2,12	1,41	0,69
15	1,70	1,70	1,13	0,55
18	1,41	1,41	0,94	0,46
22	1,16	1,16	0,77	0,38
27	0,94	0,94	0,63	0,31
33	0,77	0,77	0,51	0,25
39	0,65	0,65	0,44	0,21
47	0,54	0,54	0,36	0,18
56	0,45	0,45	0,30	0,15

### 速度ベースのモード制御

TMC5240 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。

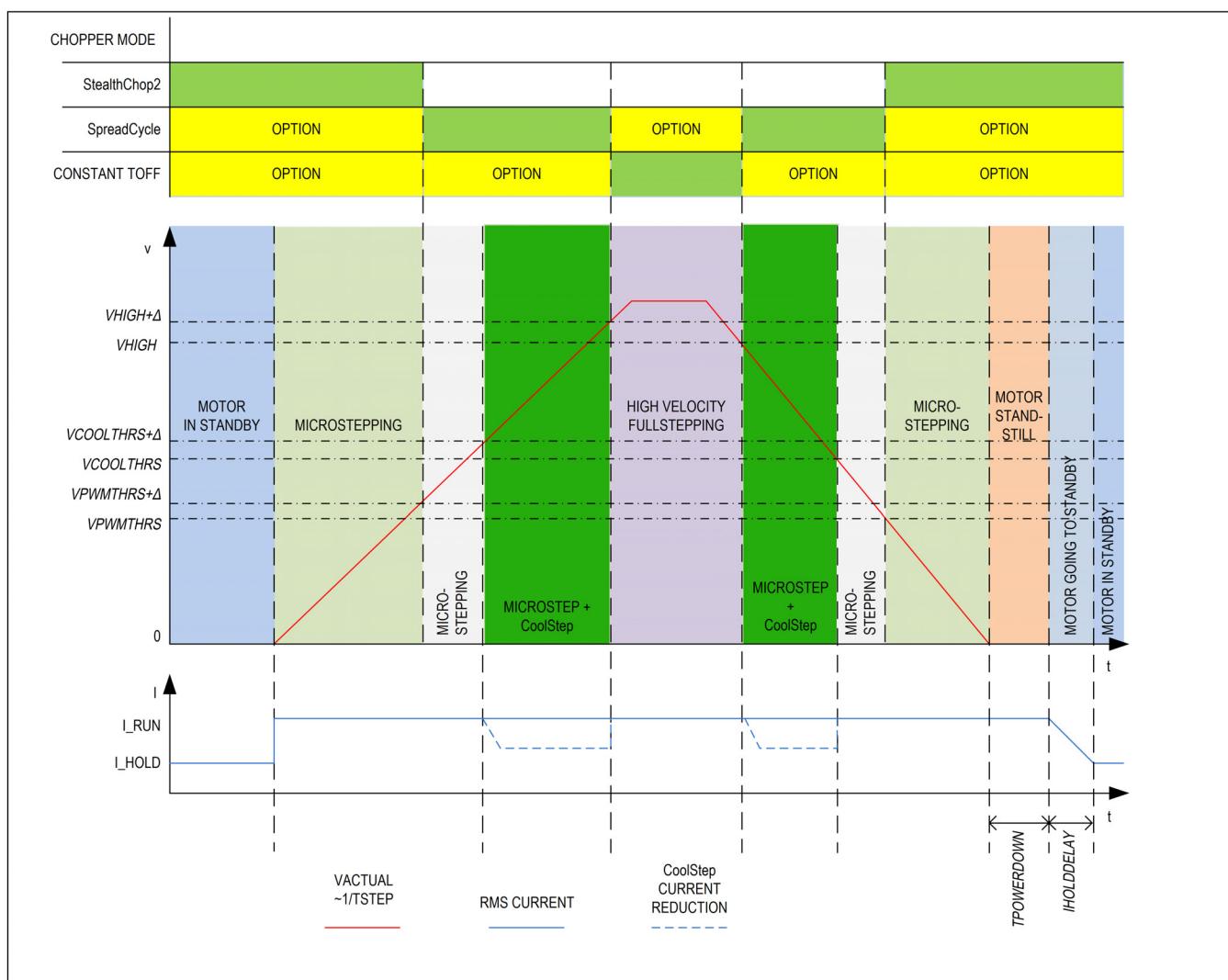


図 17. 速度依存モードの選択

上図に、使用可能なすべてのスレッショ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、スレッショルドをそれに合わせて変化させる必要はありません。スレッショルドは、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、これらのスレッショ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、スレッショ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。

自動速度スレッショ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、ユーザの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、一度パラメータ設定すれば、ソフトウェアを介して活性化したり非活性化したりする必要はありません。

表 18. 速度ベースのモード制御パラメータ

パラメータ	説明	設定	コメント
stst	各動作モードでモータ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 220 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) 後にモータ電流がパワー・ダウンするまでの遅延時間です。時間範囲は、約 0~4 秒 ( $f_{CLK} = 16\text{MHz}$ の場合) です。0 に設定すると遅延なし、1 に設定すると 1 クロック・サイクルの遅延です。それ以上は、 $2^{18}$ クロック・サイクルの離散的なステップでインクリメントします。  デフォルト：0xA	0...255	$2^{18} * t_{CLK}$ の倍数で表した時間
TSTEP	ステップ入力周波数から引き出される 2 つの 1/256 マイクロステップ間の実際の測定時間 (1/ $f_{CLK}$ 単位)。オーバーフローまたは停止時の測定値は、 $(2^{20}) - 1$ です。  デフォルト／リセット：0	0...1048575	ステータス・レジスタ、読み出し専用。 $t_{CLK}$ 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...1048575	StealthChop2 の動作上限速度スレッショ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 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との使用で有効化されます (設定されている場合)</li> </ul> デフォルト：0	0...1048575	CoolStep と StallGuard2 の動作下限速度スレッショ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...1048575	CoolStep と StallGuard2 およびオプションの高速ステップ・モードを用いた動作の上限スレッショルドを制御するための設定値
small_hysteresis	TSTEP (下限速度スレッショルド) と (TSTEP $\times$ 15/16) - 1 または (TSTEP $\times$ 31/32) - 1 (上限速度スレッショルド) 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
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° 位置でのマイクロステップ・テーブルからの電流値を用います。  デフォルト：0	0	フルステップへの切り替わりなし
		1	高速時にフルステップ

表 18. 速度ベースのモード制御パラメータ（続き）

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 1	チョッパー・モードの変化なし 高速時は一般的な定 Toff チョッパー
en_pwm_mode	StealthChop2 電圧 PWM イネーブル・フラグ（速度スレッシュルドに依存）。停止時にのみオフ状態からオン状態に切り替わります。 デフォルト : 0	0 1	StealthChop2 不使用 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、目標位置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。TMC5240 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。これは、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、ジャイロ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

TMC5240 は、すべての内部動作に対し、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<sub>CLK</sub> を基準にします。安定性と再現性を最高のものにするため、外部水晶発振器をタイム・ベースとして用いるか、マイクロコントローラからクロック信号を供給することを推奨します。

v[TMC5240] および a[TMC5240] は、TMC5240 の内部単位です。これらは、TMC5240 の速度／加速度レジスタに書込む必要のある値です。計算ツールは製品のウェブサイトおよび評価用ツールから入手できます。

表 19. ランプ・ジェネレータのパラメータと単位の関係

PARAMETER / SYMBOL	UNIT	DESCRIPTION
f <sub>CLK</sub>	[Hz]	Clock frequency of the TMC5240
s	[s]	Second
US	microstep	
FS	fullstep	
USC - microstep count	-	microstep resolution in number of microsteps (i.e. the number of microsteps between two fullsteps – normally 256)
FSC - fullstep count	-	motor fullsteps per rotation, e.g., 200
μstep velocity v[Hz]	microsteps / s	$v[\text{Hz}] = v[\text{TMC5240}] * (f_{\text{CLK}}[\text{Hz}] / 2^{24})$
μstep acceleration a[Hz/s]	microsteps / s <sup>2</sup>	$a[\text{Hz/s}] = a[\text{TMC5240}] * f_{\text{CLK}}[\text{Hz}]^2 / 2^{42}$
rotations per second v[rps]	rotations / s	$v[\text{rps}] = v[\text{microsteps/s}] / \text{USC} / \text{FSC}$
rps acceleration a[rps/s <sup>2</sup> ]	rotations / s <sup>2</sup>	$a[\text{rps/s}^2] = a[\text{microsteps/s}^2] / \text{USC} / \text{FSC}$
ramp steps[microsteps] = rs	microsteps	$rs = (v[\text{TMC5240}])^2 / a[\text{TMC5240}] / 2^8$ microsteps during linear acceleration ramp (assuming acceleration from 0 to v)

表 19. ランプ・ジェネレータのパラメータと単位の関係（続き）

TSTEP, Txxx_THRS	-	$\begin{aligned} \text{TSTEP} &= f_{\text{CLK}} / f_{256\text{STEP}} = f_{\text{CLK}} / (f_{\text{STEP}} * 256 / \text{USC}) \\ &= 2^{24} / (\text{VACTUAL} * 256 / \text{USC}) \end{aligned}$ <p>The time reference for velocity thresholds is referred to the actual 1/256 microstep frequency (<math>f_{256\text{STEP}}</math>) of the step input respectively velocity <math>v[\text{Hz}]</math>.</p>
Ramp generator update rate	[Hz]	$f_{\text{UPDATE}} = f_{\text{CLK}} / 512$ <p>VACTUAL updates with this frequency.</p>

まれに、加速度の上限値がアプリケーションに制限を課すことがあります。クロック周波数を下げて動作させる場合や、モータ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 のオプション *interp* を 1 にセットし、*MRES* を %0001 に設定すると、同じ速度設定に対しモータ速度を 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

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

加速と減速の 3 種類のセットは自由に組み合わせられます。遷移速度 *V1* と *V2* を用いることで、3 つの加速および減速設定値の間での速度に依存した切り替えが可能です。標準的な高速アプリケーションでは、速度が高くなるほどモータのトルクが減少するため、加速度値と減速度値は小さくなります。システム内のフリクションを考慮すると、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。このようにして、時間が重要なアプリケーションでのモータの動作速度は最大化されます。

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

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

注：

始動速度は、不要な場合ゼロに設定できます。

停止速度は、不要な場合、低い値（1000～10）に設定できます。背景：*TSTOP = 0* とすると、設定されたマイクロステップ目標位置に正確に到達できない可能性があります。

*VSTOP* は常に *VSTART* 以上に設定されるよう、注意してください。これにより、短い動作でも目標位置で正しく終了できます。

ジャーキー除去を無効化するには *TVMAX* をゼロに設定します。

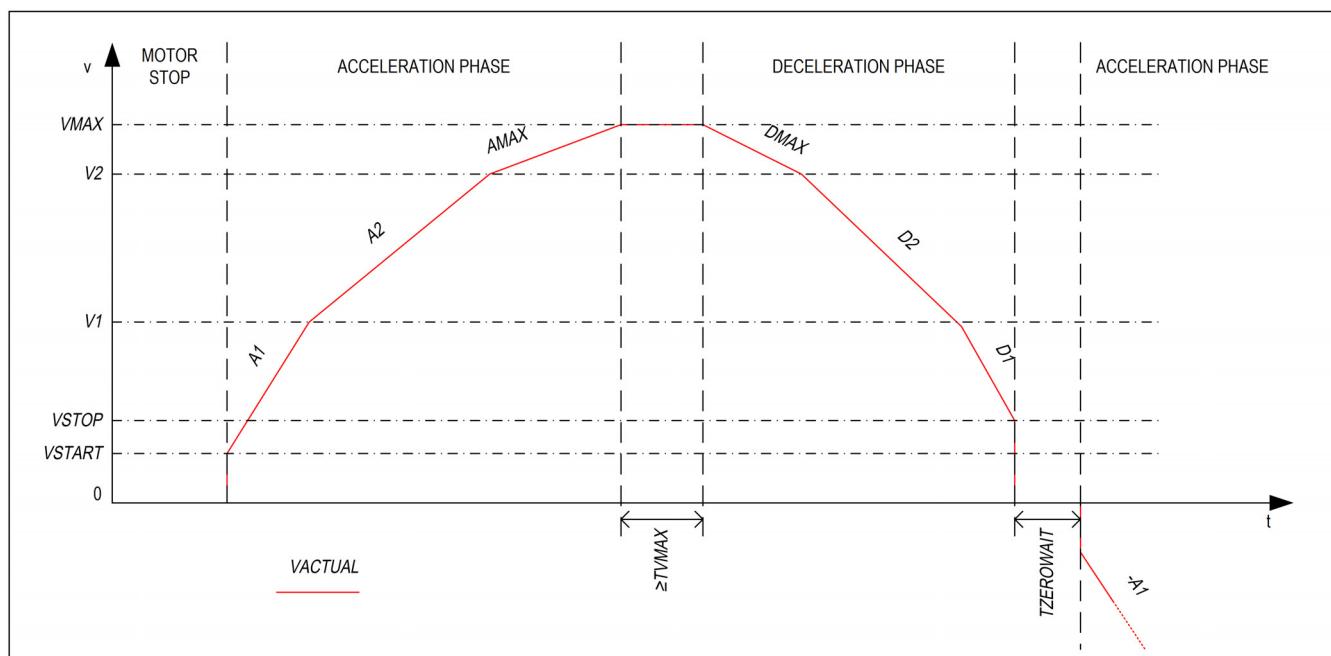
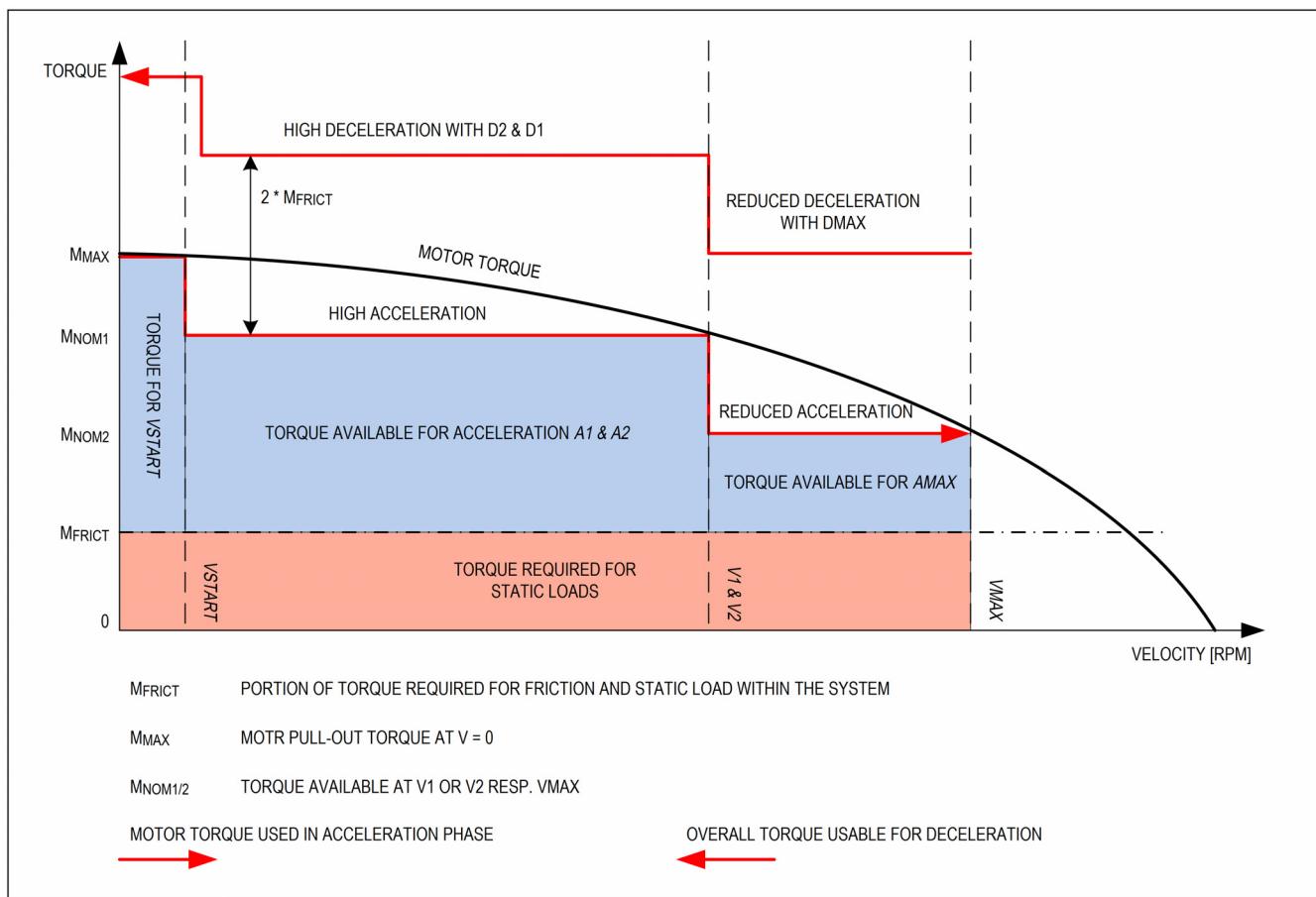


図 18. 2 回目の動作が負の方向に向かうランプ・ジェネレータ速度パターン



## 8 点ランプ

### 始動速度および停止速度

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

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

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

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

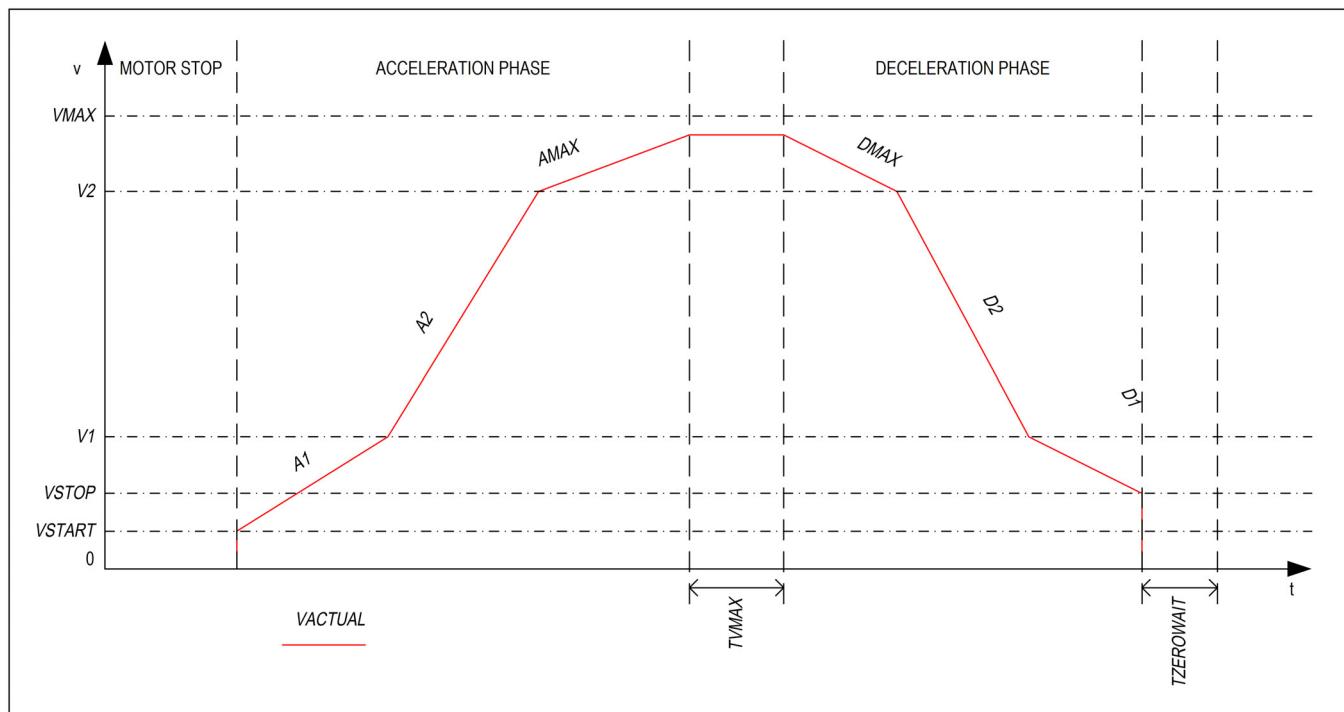


図 20. 距離が短いために VMAX に到達できない 8 点ランプ

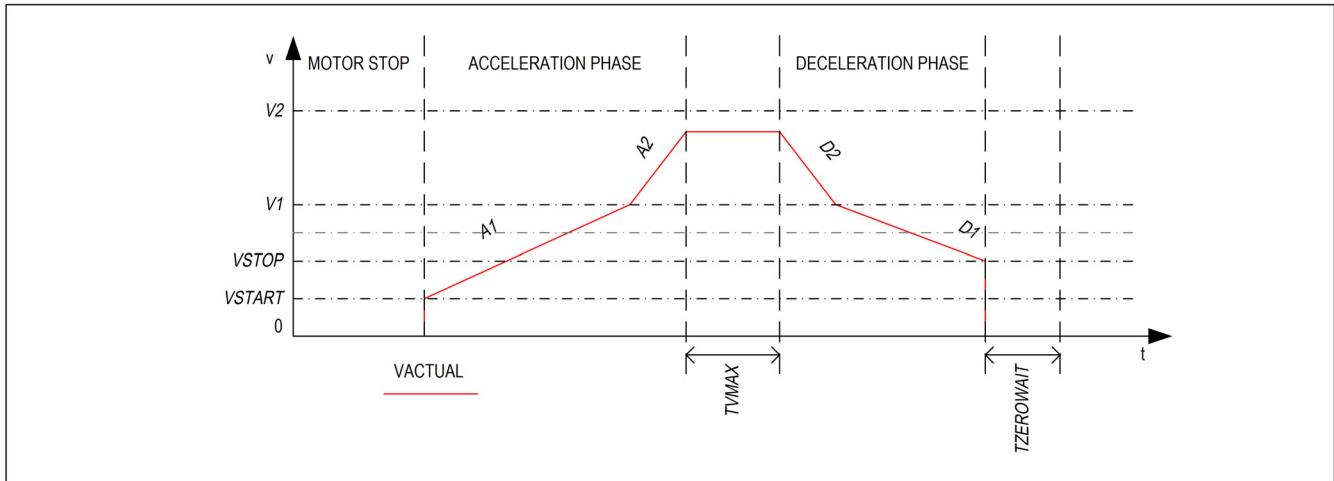


図 21. 距離が短いために V2 に達せず AMAX フェーズと DMAX フェーズがないプロファイル

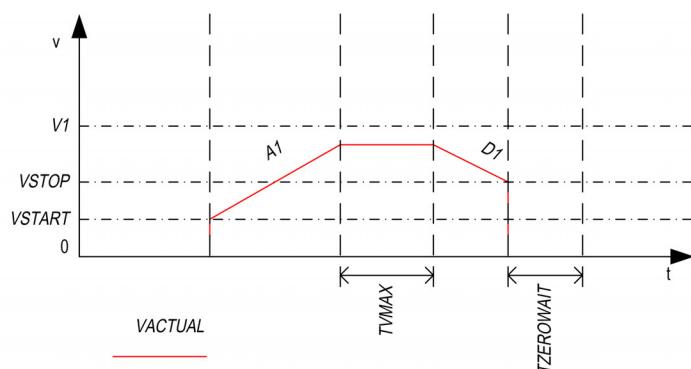


図 22. 距離が短いために V1 に達しないプロファイル

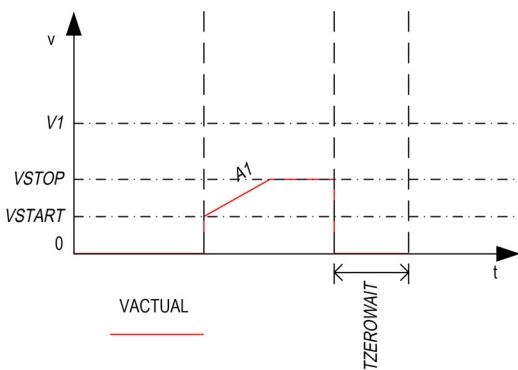


図 23. 距離が短いために TVMAX を維持できないプロファイル

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

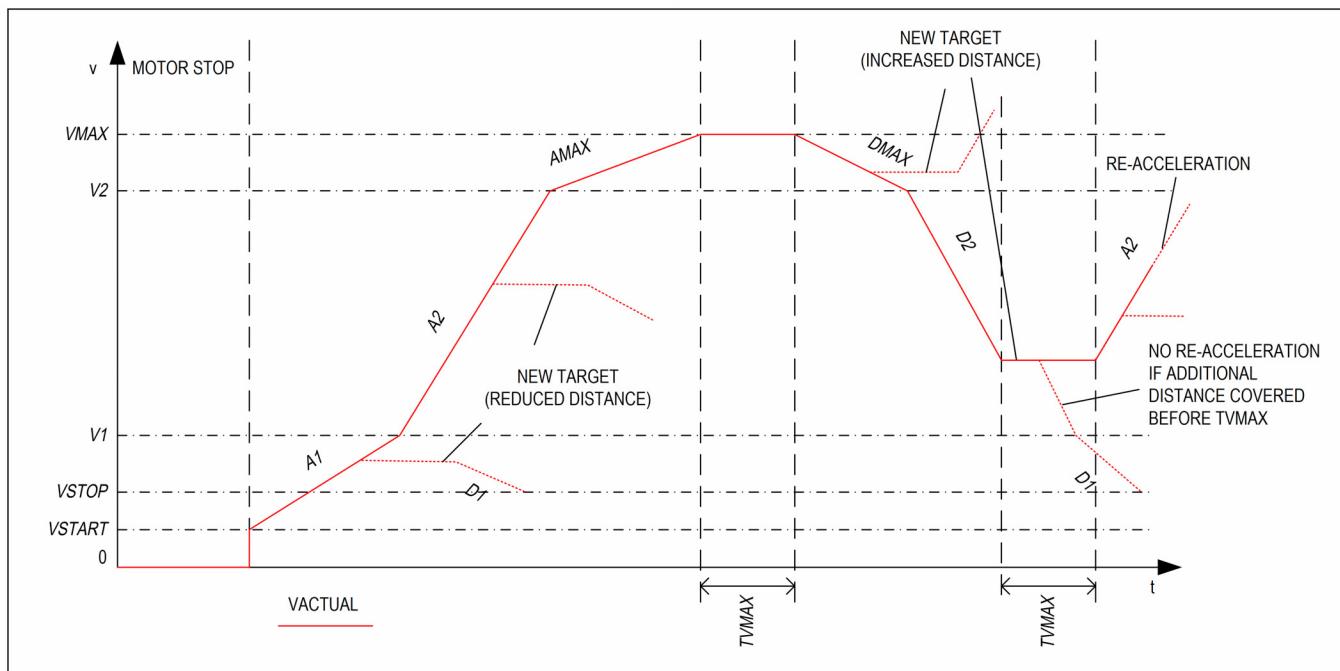


図 24. オンザフライの目標位置変更があった場合の 8 点ランプの例

## 速度モード

速度モードの動作では、異なる加速度設定および減速度設定を用いないことで、簡単に使用できるようにしています。速度モードでは  $AMAX$  および  $VMAX$  のみが関係します。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。

モータを停止まで減速するためには、 $VMAX$  をゼロにセットすれば十分です。フラグ  $vzero$  はモータの停止を通知します。フラグ  $velocity_reached$  は、目標速度に達したことを常に通知します。

## ランプの早期終了

システムとインタラクティブである必要がある場合、アプリケーションによっては、目標位置に達する前に速度ゼロにランプ・ダウンすることで動作を終了させる必要があります。

加速設定値を用いて動作を終了す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。 $VMAX = 0$  に設定し、 $AMAX$  を必要な減速度値に設定します。これにより、線形ランプを使用してモータが停止されます。
2. ポジショニング・モードでの停止には、 $VSTART = 0$  および  $VMAX = 0$  に設定します。この場合、 $VSTOP$  は使用しません。ドライバは  $AMAX$ 、 $A1$ 、 $A2$  ( $V1$  および  $V2$  で定義) を用いて速度ゼロまで減速します。
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の時間を設定します。ドライバは減速を行い、最終的に停止に至ります。a)またはb)のオプションを用いて  $TZEROWAIT$  時間中に動作を停止するには、実際の速度をポーリングします。
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。位置を比較し ( $X\_ACTUAL$  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、余すところのない有効な動作範囲を確保します。

ジョイスティック制御を実現するには：

1. 動作方向を制御し、動作制限値を設定するために、ポジショニング・モードを使用します。これらの制限値は、仮想停止スイッチとして使用できます（例えば *VIRTUAL\_STOP\_L* および *VIRTUAL\_STOP\_R*）。
2. ジョイスティックの入力に応じていつでも、*VSTART* から *VMAX* 最大値までの範囲に変更します。*VSTART* = 0 の場合、*IHOLD* および *IRUN* でそれぞれ定義された実際の電流スケーリングを考慮する *CS\_ACTUAL* を設定することで、*CoolStepMAX* = 0 で動作を止めるともできます。モーション・コントローラは、*V1* および *V2* によって決定される *A1*、*A2*、*AMAX* を用いて、ランプ・アップおよびランプ・ダウント速度を適応させます。
3. 加速度設定値を変更しない場合には、*XTARGET* を書き換える必要はなく、*VMAX* だけを変更します。
4. *DMAX*、*D1*、*D2*、*VSTOP* を使用できるのは、ランプ・コントローラが目標位置に達したためにスロー・ダウンする場合、または、目標位置が別の方向の点に変更された場合のみです。

### 速度スレッショルド

ランプ・ジェネレータは、実際の速度 *VACTUAL* と合わせていくつかの速度スレッショルドを提供します。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、スレッショ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、設定値 *THIGH* と *TCOOLTHRS* で決めることができます。*TSTEP* は、これらのスレッショ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、スレッショルドの設定値より *1/16* または *1/32* だけ高くなります。StealthChop のスレッショルド *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ョルドは、2 つのマイクロステップ間の時間 *TSTEP* と組み合わされます。

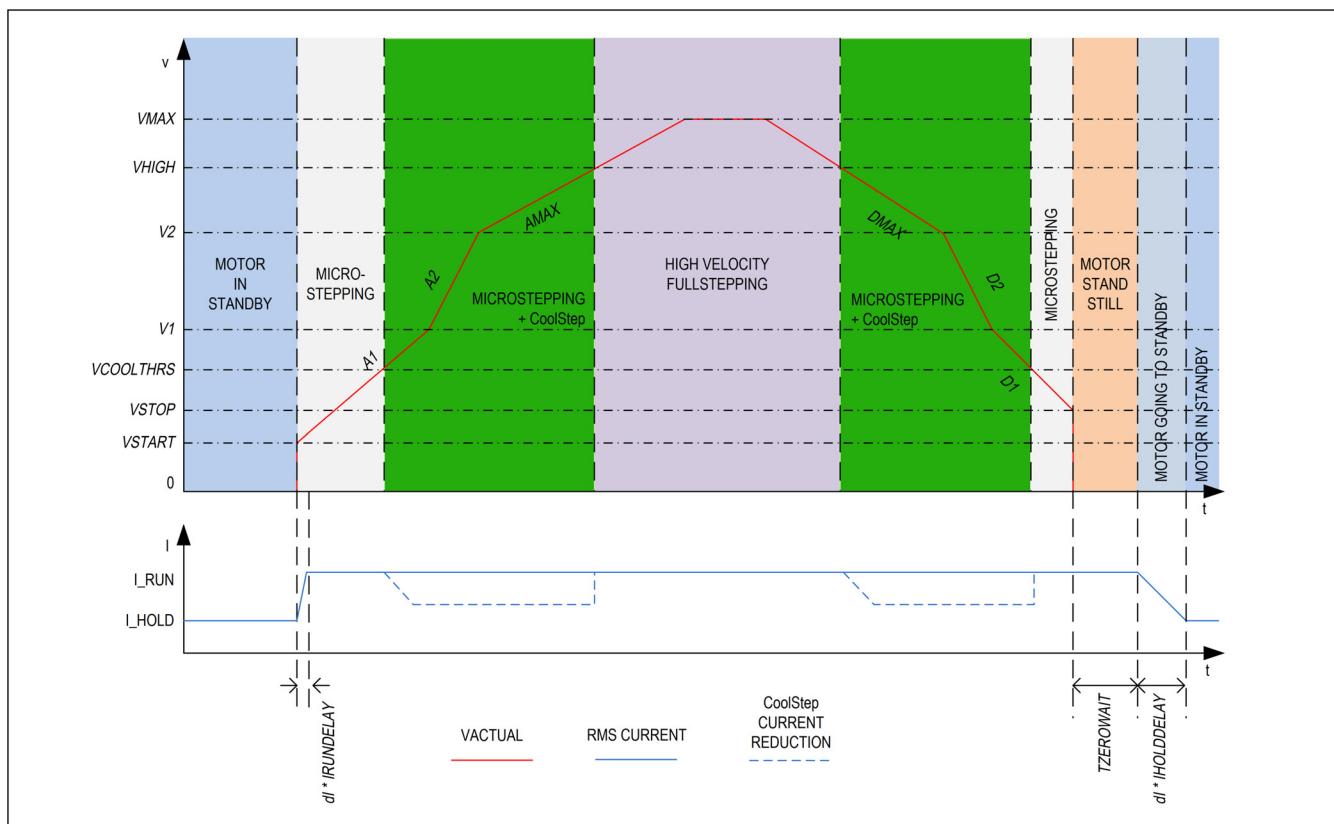


図 25. ランプ・ジェネレータの速度依存モータ制御

### 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位置を設定する必要があります。

リファレンス位置は、StallGuard2、StallGuard4、またはリファレンス・スイッチによって検出できる機械的な停止を用いて見出すことができます。

線形駆動の場合、機械的な動作範囲を超えることはできません。これは、左リファレンス・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、異常な状態に対しても確保できます。そのため、ランプ・ジェネレータは、*SW\_MODE* レジスタで設定されたい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は次の 2 通りがあります。

- スイッチに達した場合に急激に停止できます。これは緊急時や StallGuard2 に基づくホーミングの場合に有用です。
- または、ソフト停止機能（ビット *en\_softstop* = 1）を用いて、減速設定（*DMAX*、*V2*、*D2*、*V1*、*D1*）によりモータ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位置 *XACTUAL* を保持レジスタ *XLATCH* にラッチすると、リファレンス・スイッチの位置の正確なスナップショットが得られます。

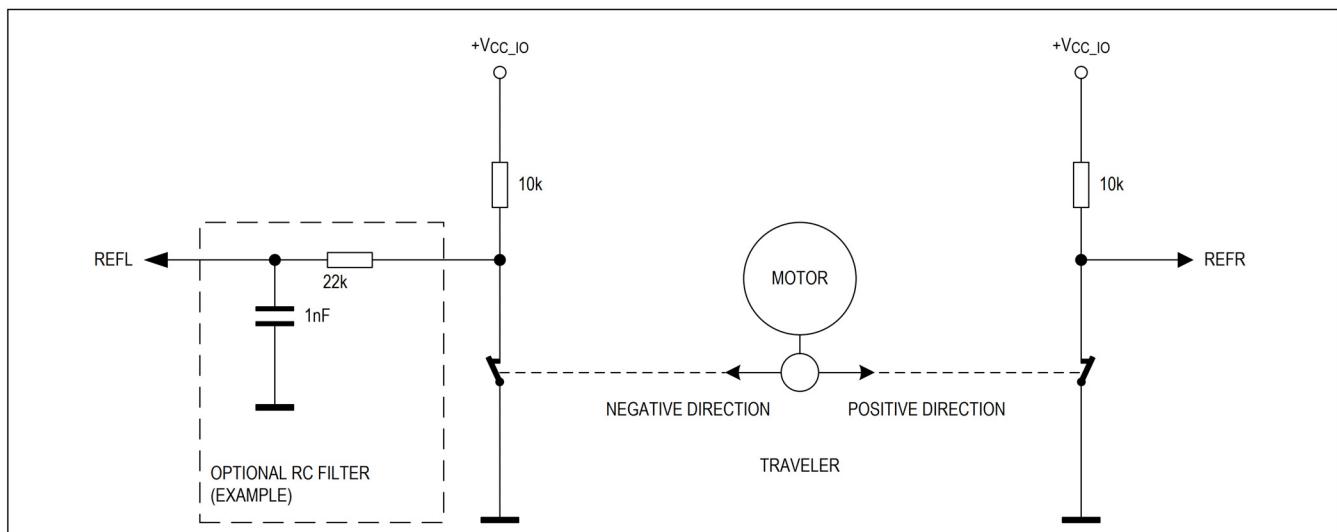


図 26. リファレンス・スイッチの使用（例）

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

- 機械式スイッチ、
- フォト・インターラプタ、または
- ホール・センサー。

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

ケーブルが長い場合、TMC5240 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、誤った書き込みによりロジック・レベル入力を破壊してしまうおそれを低減できますが、アプリケーションに関しては考慮すべき一定の遅延も加わります。

#### ホーミング・プローシージャの実行：

1. ホーミング・スイッチから距離を置くなどして、ホーミング・スイッチが押されることのないようしてください。
2. 目的のスイッチ・イベントでの位置ラッチを活性化し、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 ベースのホーミングでは、ハード・ストップ (*en\_softstop* = 0) を用いる必要があります。
3. 動作ランプをスイッチの方向に起動します。（左スイッチはより負の位置に移動させ、右スイッチはより正の位置に移動させます）。位置ランピング・コマンドを用いることで、この動作をタイムアウトできます。
4. スイッチに達すると直ちに、位置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、ラッチ位置と実際の位置の差を計算します。StallGuard2 ベースのホーミング時、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位置レジスタに書込みます。これでホーミングが終了です。位置 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 をホーミングに用いた場合は、*RAMP\_STAT* に読み出しがアクセスを行うことで、StallGuard2 の停止イベント *event\_stop\_sg* がクリアされ、モータは停止条件から解放されます。

#### 3つ目のスイッチによるホーミング：

アプリケーションによっては、機械式のリミット・スイッチとは独立に動作する、追加のホーム・スイッチを用います。TMC5240 のエンコーダ機能は、位置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、立上がりイベント時または立下がりイベント時あるいはその両方でスナップショットできます。また、この機能は、割込み出力も可能です。

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込み出力をトリガできます（割込みが *DIAG0* で示された場合に、*ENC\_STATUS* の *n\_event* を読み出すことでチェック）。

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. スイッチがトリガされたら、XLATCH を読み出します。これはスイッチ・イベントが生じた位置を示します。
4. スイッチ・イベントの検出後、モータを停止させ、実際の位置から XLATCH を差し引きます (必要なステップの詳細は、上述のホーミング・プロセスで説明しています)。

### 仮想リファレンス・スイッチ

TMC5240 は、リファレンス・スイッチが 1 つしかない、あるいはまったくないアプリケーションに対応するため、仮想リファレンス・スイッチをサポートし (StallGuard ホーミング)、物理的な移動範囲を安全に制限しています。仮想停止スイッチが活性化するのは、実際のモータ位置 (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、仮想停止は、エンコーダ位置 (X\_ENC) のモニタ用に切り替えることもできます。これを選択するには、virtual\_stop\_enc を設定します。

仮想停止の値 (VIRTUAL\_STOP\_R, VIRTUAL\_STOP\_L) は、符号付き 32 ビット移動範囲のオーバーフロー／アンダーフローに対して十分な距離を保って設定し、ソフト減速を用いた場合にモータが減速できるようにします。

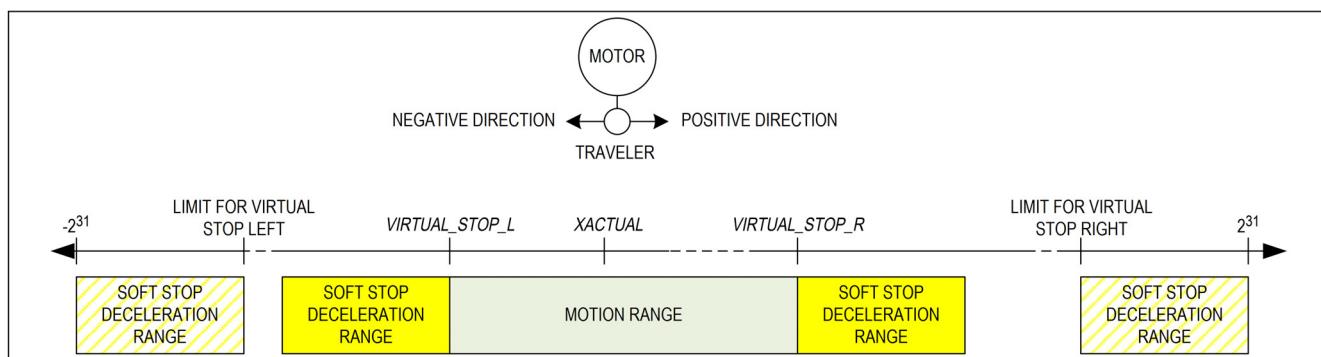


図 27. 仮想停止スイッチと制限の図解

### ランプ・ジェネレータの応答時間

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

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

ただし、低移動速度や低加速度設定時、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ため、これは目標位置に達するまでに大きな遅延が生じる原因となります。ランプを短時間で終了させるには、VSTOP を最小 100~1000 の範囲に設定します (100 の場合は約 10ms 未満、1000 の場合は約 1ms 未満になります)。

### 外部 STEP/DIR ドライバ

TMC5240 は、内部ランプ・ジェネレータを用いることで、ADI-Trinamic の TMC262C、TMC2160、または TMC2240 などの外部 STEP/DIR ドライバを制御し、強力なステッピング・モータ・アプリケーションを実現できます。この構成では、通常内部ドライバは使用しませんが、2 個のモータを同期動作させる場合などに、外部ドライバに追加して用いることもできます。

DIAG0 出力と DIAG1\_SW 出力は、diag0\_nint\_step および diag1\_nposcomp\_dir の GCONF フラグをセットすることで、STEP と DIR 出力に對してイネーブルされます。DcStep 機能や自動モータ電流制御機能などのその他の内部ドライバ機能は、このモードでは使用できません。外部ドライバから TMC5240 への帰還がないためです。

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

外部 STEP 信号には次の 2 つのオプションが選択できます。

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

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

### 位置比較機能

位置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、*XACTUAL* = *X\_COMPARE* の場合に常に、アクティブ・ハイのレベルを出力します。そのため、パルスの長さは、*XACTUAL* が *X\_COMPARE* に一致している時間、つまり、実際の速度における 1 マイクロステップの長さに対応します。

繰り返し機能を用いれば、周期的な比較パルスをトリガできます。必要な周期 (マイクロステップの距離) をプログラムするには、*X\_COMPARE\_REPEAT* を用います (最大  $2^{24} - 1$  ステップ)。

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

値	説明
0	繰り返しなし
>1	最初の比較位置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 : $2 \sim 2^{24} - 1$ マイクロステップ

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

### StallGuard2 の負荷測定

異なるモータ制御方式を適合させるために、TMC5240 には 2 つの基本チョッパー・モードに対応する、2 種類の StallGuard センサ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 動作に対し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、負荷がモータをストールさせる下限未満の場合における、ストール検出やその他の用途 (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、StallGuard の値 (SG\_RESULT) は減少します。ストールを正しく検出するには調整が必要です。モータが過負荷／ストール状態となった場合に SG\_RESULT が 0 (あるいはほぼ 0) に達するように、StallGuard のスレッショルド (SGTHRS) を設定します。

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

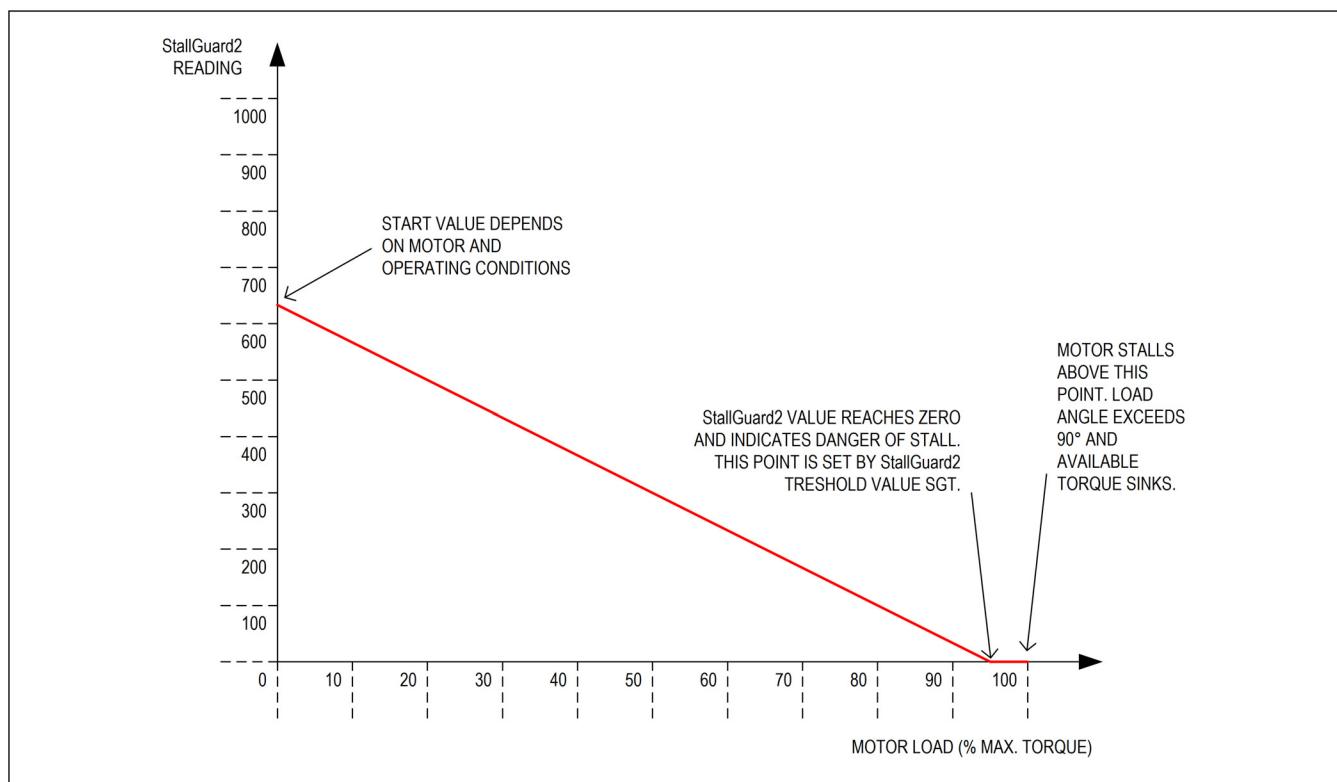


図 28. StallGuard2 の動作原理

表 21. StallGuard2 関連パラメータ

パラメータ	説明	設定	コメント
SGT	この符号付きの値は、ストール検出用の StallGuard2 スレッショルド・レベルを制御し、読み出し用に最適な測定範囲を設定します。値を小さくすると感度が高くなります。ゼロはほとんどのモータで使用できる開始値です。値を大きくすると StallGuard2 の感度は低下し、ストールを示すために必要なトルクが大きくなります。	0	中立値
		+1...+63	低感度
		-1...-64	高感度
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
ステータス・ワード	説明	範囲	コメント
SG_RESULT	これは StallGuard2 の結果です。読み出し値が大きいと機械的負荷は小さいことを示します。読み出し値が小さいと負荷が高く、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 の読み出し値が約 0~100 となるよう、SGT 設定値を調整します。	0...1023	0 : 最大負荷 低値 : 高負荷 高値 : 低負荷

### StallGuard2 のスレッショルド SGT の調整

StallGuard2 値 SG\_RESULT は、モータ固有の特性と、アプリケーション固有の負荷、速度、電源電圧、電流レベルに対する要求による影響を受けます。そのため、StallGuard2 のスレッショルド 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ーラクティブに調整を行うことです。

**StallGuard SGT を調整するための初期手順 :**

- モータ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*は符号付きの値です。そのため、負または正の値が可能です。
- ここで、*sg\_stop*をイネーブルし、ストール時にいつでもモータ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*（書き込み後クリア）をクリアして、モータを再始動します。
- モータ・ストール直前の負荷増大時に*SG\_RESULT*が0と約100の間にあり、負荷がない場合に*SG\_RESULT*が100以上増加すれば、最適設定になっています。ほとんどの場合、*SGT*は特定の動作速度または動作範囲に合うよう調整できます。設定が特定の範囲（目的の速度の80%～120%など）、かつ、極端なモータ条件下（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 の自動設定を可能にするオプション手順 :**

*SGT*設定の背景にある基本的な考え方とは、StallGuard測定をモータ内の抵抗性負荷に対して補償する、というものです。停止時あるいは極めて低い速度時には、抵抗性損失がモータのエネルギー・バランスの主要因になります。機械的なパワーがゼロまたはほぼゼロであるためです。したがって、*SGT*は、速度がほぼゼロの場合の最適値に設定できます。このアルゴリズムは、環境条件やモータ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。モータは十分なバックEMFを発生しないためです。そのため、機械的負荷は結果に大きな違いを与えません。
- sfilt*をスイッチ・オンします。ここで、*SGT*を0から*SG\_RESULT*が増加を始める値まで増加させます。*SGT*が大きいと、*SG\_RESULT*は最大値まで増加します。再び、*SG\_RESULT*がゼロにとどまる最大値まで減少させます。これで、*SGT*が可能な限り高感度の値に設定されます。速度が上がるにつれ*SG\_RESULT*が増加するのを観察すると、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、StallGuard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ョ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、モータのバックEMFが電源電圧に近付き、速度を更に増加させた場合にモータ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*に影響するため、より厳密なレギュレーションを行うことでより正確な値が得られます。*SG\_RESULT*の測定は高分解能であり、以降のセクションで説明するように精度を高めるための方法がいくつかあります。

**TCOOLTHRS および THIGH を制限する可変速度**

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

多くのアプリケーションでは、ほとんどの時間で单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ョルドを用います。決定した動作点の範囲外、例えば、*TCOOLTHRS*をマッチング値に設定する場合のセンサーレス・ホーミング・プロシージャに先立つ加速フェーズ時では、ストール検出はディスエーブルされます。上限値は*THIGH*で指定されます。

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

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

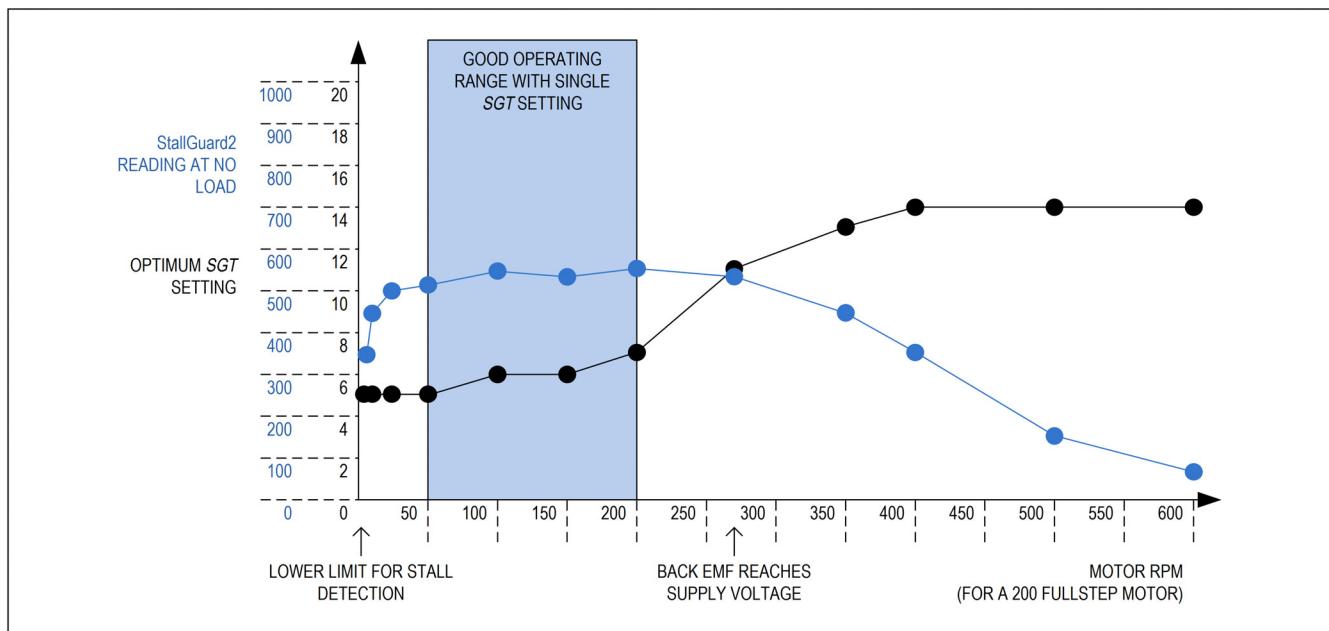


図 29. 例：サンプル・モータを用いた SGT の最適設定値と StallGuard2 の読み出し値

## 高トルク・リップルと共に振れを伴う小型モータ

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、温度上昇に比例して *SG\_RESULT* を直線的に減少させることができます。

## 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。

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

## 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所以、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の方がフルステップごとに更新を行うより重要な場合があります。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る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 相の磁石のミスマッチ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、StallGuard を用いたセンサーレス・ホーミングで最善の結果を得ようとする場合は、フィルタを無効化してください。

## モータ・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・スレッショ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・スレッショルドは、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。*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。負荷のない場合と最大負荷がある場合の *SG\_RESULT* 値をチェックします。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 の読み出しが得られるように *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、最小の読み出しが示されます。ステップ・ロスの後、モータは振動し、*SG\_RESULT* の読み出しが増加します。

### **StallGuard2 を使用したホーミング**

線形駆動のホーミングでは、モータをハード・ストップの方向に移動させることができます。StallGuard2 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 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 の sg\_stop をセット)。ストールが検出されると、ランプ・ジェネレータは動作を停止し、VACTUAL をゼロにしてモータを停止させます。ストップ条件は、DRV\_STATUS のフラグ StallGuard 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 をディスエーブルできます。あるいは、RAMP\_STAT を読み出すことで、モータを再度イネーブルできます。動作パラメータが変更されていない場合、RAMP\_STAT の event\_stop\_sg フラグの「読み出し後クリア」機能は、TZEROWAIT の時間の経過後、モータを再始動させます。

### **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、生成されるバック EMF は低く、測定は不安定で環境条件（温度など）に依存するようになります。前述の自動調整プロシージャによってこれを補償できます。その他の条件でも、SGT の設定が極端な値になったり、測定値 SG\_RESULT のモータ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、モータのバック EMF が電源電圧に達することで特徴づけられます。

### **StallGuard4 の負荷測定**

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

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

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

StallGuard4 では、モータ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、次の図に示すように、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付くにつれ、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。

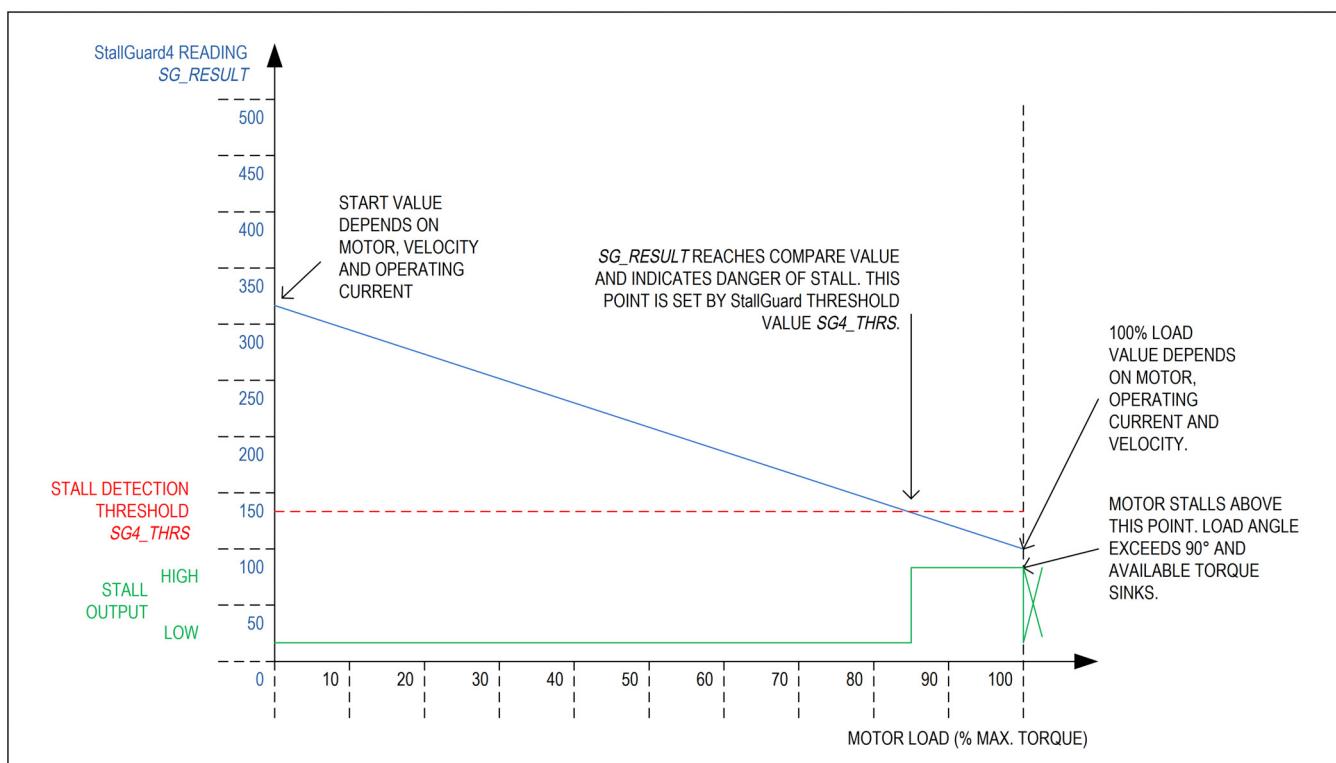


図 30. StallGuard4 の動作モード

表 22. StallGuard4 関連のパラメータ

パラメータ	説明	設定	コメント
SG4_THRS	この値はストール検出のための StallGuard4 スレッショルドを制御します。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。	0...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。
ステータス・ワード	説明	範囲	コメント
SG4_RESULT	これは StallGuard4 の結果です。読み出し値が大きいと機械的負荷は小さいことを示します。読み出し値が小さいと負荷が高く、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 VCOOLTHR のような速度スレッショルドとは独立に生成されます。結果は SG4_IND_x の計測値から計算され、StallGuard2 より高い精度と同程度の範囲を実現するために 1 ビットが加えられます	0...510	低値：最大負荷 高値：低／無負荷
SG4_IND_3 SG4_IND_2 SG4_IND_1 SG4_IND_0	モータの 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...255	低値：最大負荷 高値：低／無負荷

表 22. 帰還抵抗の推奨値（続き）

<i>sg4_filt_en</i>	0 : フィルタを使用する動作、 <i>SG4_RESULT</i> フルステップごとに更新 1 : フィルタリング動作、 <i>SG4_IND_0...3</i> を使用可能、 <i>SG4_RESULT</i> は最後の4つの <i>SG4_IND_x</i> 測定値の平均を示します	0 1	0 : フィルタ・オフ 1 : フィルタリング動作、 <i>SG4_IND</i> の値を使用可能
<i>sg_angle_offset</i>	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 <i>SG4_RESULT</i> を用いることで、StealthChop2 と SpreadCycle の間の最適化されたスイッチングを可能にします。位相オフセットは保存され、StealthChop2 にスイッチング・バックする際に再度減算されます。	0 1	0 : 角度補正なし 1 : StealthChop2 と SpreadCycle のスイッチングを最適化

## StallGuard4 の調整

StallGuard4 値の *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 のスレッショルド *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。

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

- モータをアプリケーションの通常の動作速度で動作させ、*SG4\_RESULT* をモニタします。
- モータ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）。
- ここで、DIAG 出力を介して StallGuard 出力信号をモニタ（動作の下限速度に合うよう *TCOOLTHRS* も設定）し、パルスが各出力に表れるとモータを停止させます。ストール時には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、*SG4\_THRS* を増加します。
- ストールが確実に検出され、ストールの瞬間に DIAG にパルスが発生したら、最適な設定が実現されています。ほとんどの場合、*SG4\_THRS* は特定の動作速度または動作範囲に合うよう調整できます。設定が特定の範囲（目的の速度の 80%～120%など）、かつ、極端なモータ条件下（適用可能な最低および最高温度）でも確実に機能することを確認してください。

*SG4\_RESULT* が *SG4\_THRS* 未満になると、StallGuard は DIAG にパルスを送ります。これが生じるのは、StealthChop2 モードの場合と、*TCOOLTHRS*  $\geq$  *TSTEP*  $>$  *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单一パルスに反応することが必要です。StallGuard が良好な結果をもたらすことのできる下限速度スレッショルドに一致するよう、*TCOOLTHRS* を設定します。

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

## StallGuard4 の更新レート

StallGuard4 の測定値 *SG4\_RESULT* は、モータ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所以、この方法によりストールを十分確実に検出できます。

StallGuard4 には測定用に次の 2 つのオプションがあります。

- sg4\_filt\_en* = 0 : 単一測定、各フルステップ後に更新、1 フルステップごとに有効。この測定により、コイル電圧がゼロに伝送するたびに *SG4\_RESULT* が完全に更新されるため、負荷の変動に対して最速の応答が可能になります。そのため、固い障害物がある場合のストール検出に最適です。
- sg4\_filt\_en* = 1 : このモードでは、次に示す 4 つの個別の信号が生成されます。コサイン波（コイル A）の立下がり 0 遷移時の *SG4\_IND\_0*、コサイン波の立上がり 0 遷移時の *SG4\_IND\_1*、サイン波（コイル B）の立下がり 0 遷移時の *SG4\_IND\_2*、サイン波の立上がり 0 遷移時の *SG4\_IND\_3* です。*SG4\_RESULT* の実際の値は、4 つのすべての測定値の平均値で、フルステップごとに更新されます。そのため、各フルステップが全体の結果に及ぼす影響は 25%だけです。このモードは、柔らかい障害物がある場合の検出、あるいは、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・モータ阻害）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・スレッショ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、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・スレッショルドは、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。より精緻に行われた評価も、固定スレッショ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が除外されます。

### 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、生成されるバック EMF は低く、測定は不安定で環境条件（温度など）に依存するようになります。その他の条件でも、測定値 *SG4\_RESULT* のモータ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、モータのバック EMF が電源電圧を超えることで特徴づけられます。

### CoolStep の負荷適応型電流スケーリング

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

### CoolStep 用の設定

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

表 23. CoolStep の重要なパラメータ

パラメータ	説明	範囲	コメント
SEMIN	下限スレッショルドを設定する 4 ビットの符号なし整数。 <i>SG_RESULT</i> がこのスレッショルドを下回る（負荷が大きいことを示す）場合、CoolStep は両方のコイルへの電流を増加します。4 ビットの <i>SEMIN</i> の値が 32 倍され、10 ビットの <i>SG_RESULT</i> の値の範囲の下半分を占めます。（このパラメータの名前は、CoolStep の以前の名称である smartEnergy に由来するものです。）	0	CoolStep をディスエーブル
		1...15	スレッショルドは <i>SEMIN</i> × 32
SEMAX	上限スレッショルドを制御する 4 ビットの符号なし整数。このスレッショルド以上の値の <i>SG_RESULT</i> 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。上限スレッショルドは、( <i>SEMIN</i> + <i>SEMAX</i> + 1) × 32 です。	0...15	スレッショルドは ( <i>SEMIN</i> + <i>SEMAX</i> + 1) × 32

次に示す図は、CoolStep の動作範囲を示すものです。

- 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、*SG\_RESULT* が (*SEMIN* + *SEMAX* + 1) × 32 を上回り、電流は減少します。

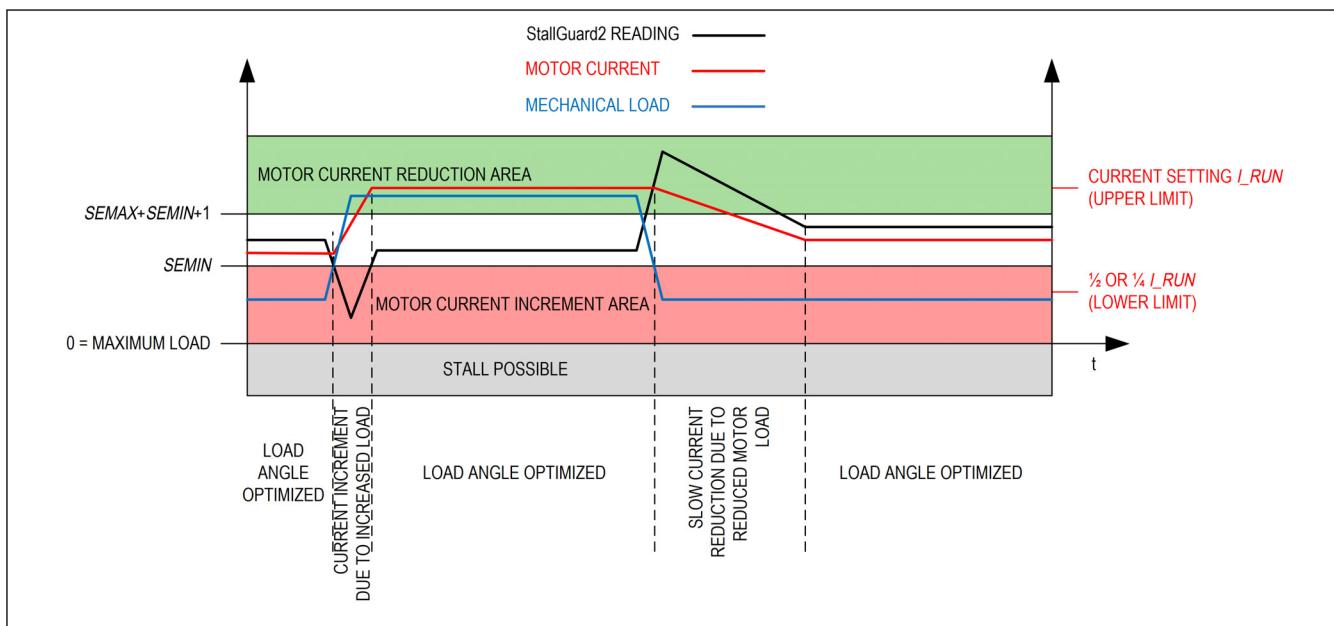


図 31. モータ電流を負荷に適合させる CoolStep

表 24. CoolStep のその他のパラメータとステータス情報

パラメータ	説明	範囲	コメント
SEUP	電流のインクリメント・ステップを設定します。SEMINで設定された下限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。	0...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 / StallGuard4 読出し値の数を設定します。	0...3	デクリメントあたりの StallGuard2 測定値の数：32、8、2、1
SEIMIN	IRUN 電流設定値をスケーリングして、CoolStep 動作の下限モータ電流制限を設定します。	0	0 : IRUN の 1/2
	StealthChop2 を使用する場合、特に 25%まで低減する必要がある場合は、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。	1	1 : IRUN の 1/4
TCOOLTHRS	CoolStep に切り替えるための下限速度スレッショルド。この速度を下回ると CoolStep はディスエーブルされます。StallGuard2 が安定な結果を与える速度範囲の下限に合わせてください。	1... $2^{20}$ - 1	スレッショルド値を 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ョルド値。この速度を超えると CoolStep はディスエーブルされます。StallGuard2 / StallGuard4 が安定した結果を与える速度範囲に合わせてください。	1... $2^{20}$ - 1	また、フルステップへの切り替えなど、その他の機能も制御します。
ステータス・ワード	説明	範囲	コメント
CS_ACTUAL	このステータス値は、CoolStep によって制御される実際のモータ電流スケールを示します。値の上限は IRUN の値で、下限は SEMIN で指定される IRUN の一部です。	0...31	1/32、2/32、... 32/32

### CoolStep の調整

CoolStep を SpreadCycle と併用して調整する前に、まず StallGuard2 のスレッショルド・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では、コイル電流の増加を減少よりもはるかに速く行うことができます。下限スレッショ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、より深刻なイベントであるためです。応答が遅すぎる場合、モータはストールする可能性があります。反対に、上限スレッショルドに達したことへの応答が遅くても、電力を節減する機会を逸する以上の深刻なリスクはありません。

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

### 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。

推奨：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ョルドを設定することです。加速および減速は通常、短時間で行われるものなので、フル・モータ電流が必要であるとはいえる、時間が短いため全体の消費電力への影響はわずかなものにすぎません。

### 低速度およびスタンバイ動作

CoolStep は、停止時および非常に低 RPM 時のモータ負荷は測定できないため、ランプ・ジェネレータには下限速度スレッショルドが設けられています。これには、アプリケーション固有のデフォルト値を設定する必要があります。このスレッショルド未満では、*IRUN* または *IHOLD* を介した通常の電流設定が有効です。上限スレッショルドは *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。

どちらのスレッショルドも 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

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

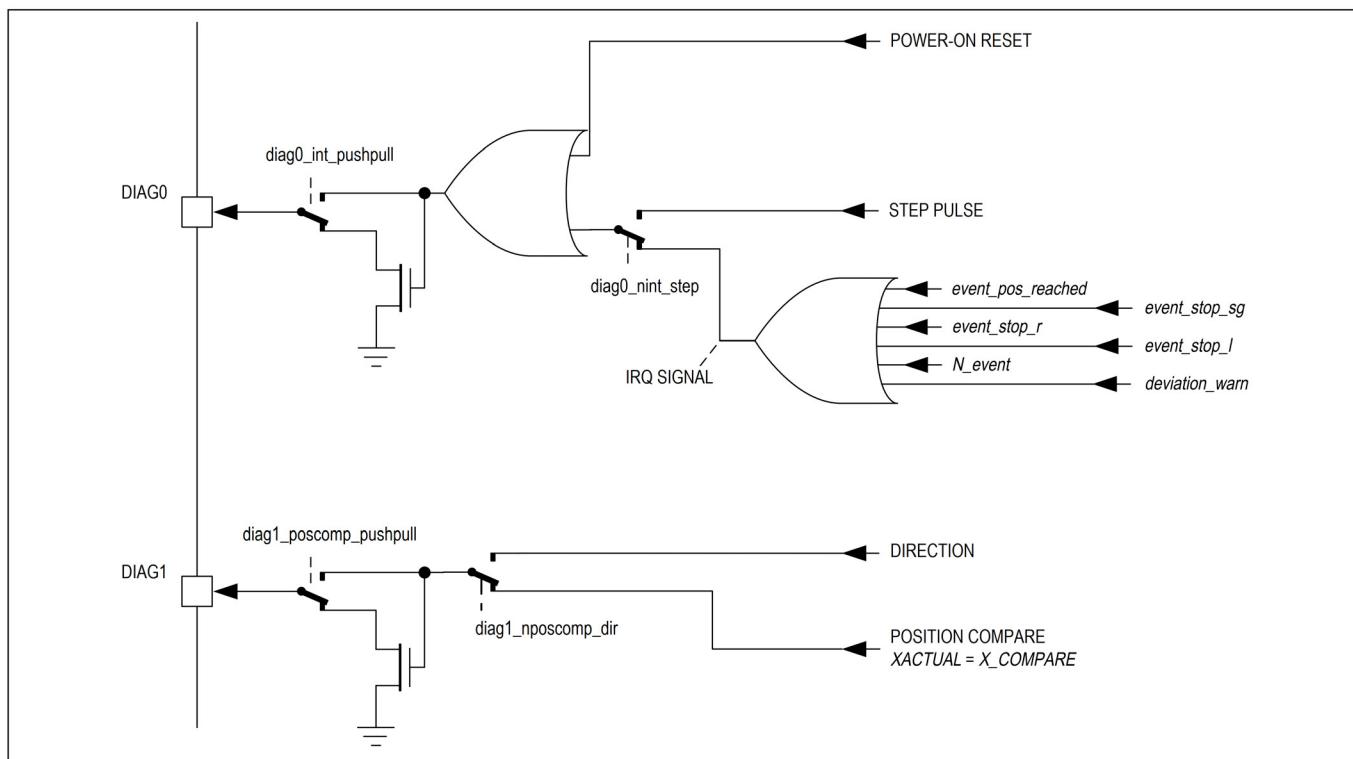


図 32. 診断出力構成のオプション

## 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、負荷に対応できる限り、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がなく、重負荷をできるだけ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。

## 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、最大 50% のトルクの安全マージンが必要です。DcStep では、利用可能な最大モータ・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は、保持トルク（モータが実現できる最大トルク）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。

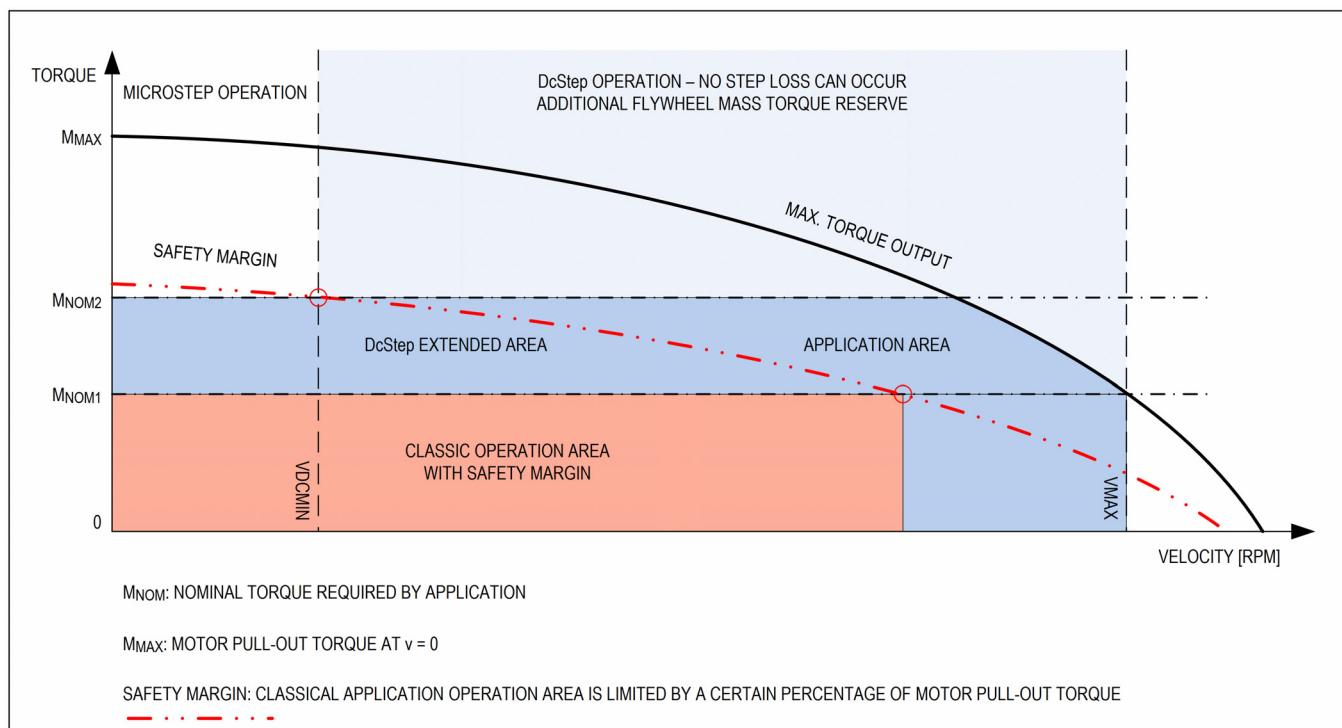


図 33. DcStep により拡大されるアプリケーション動作領域

### 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 TOFF 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。これには、最小動作速度  $VDCMIN$  を設定する必要があります。 $VDCMIN$  は、DcStep がモータ動作の高信頼性検知が可能な最小動作速度に設定されます。 $VDCMIN$  未満の速度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す。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  $VDCMIN$  未満になった場合に、StallGuard2 によって処理されます。

**注意：**DcStep では、サイン波の位相極性が、768~255 の MSCNT 範囲内で正、256~767 の範囲内では負となることが必要です。コサイン極性は、0~511 では正、512~1023 では負である必要があります。1 だけ位相がシフトすると 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。デフォルトの表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。

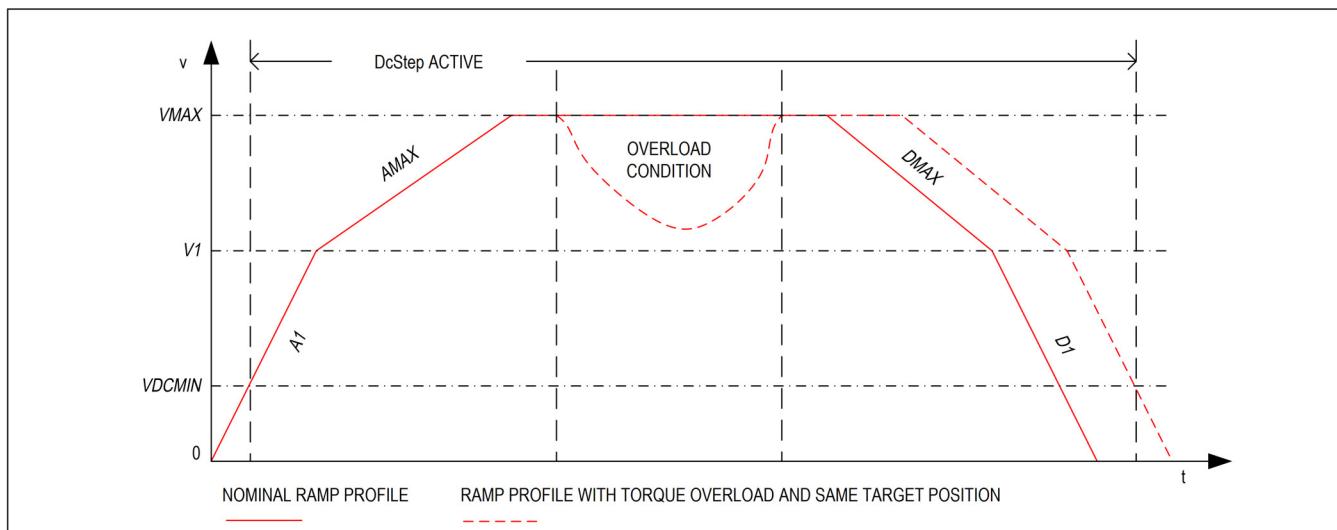


図 34. 過負荷状態時の DcStep の速度プロファイル

### 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安全に検知できないようなモータ依存の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、あるいはステップを失う可能性があります。ステップ・ロスを確実に検出してモータの再始動を回避するために、ストール時に停止 (stop-on-stall) 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、ストール時に停止 (stop-on-stall) 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  $0^\circ$ ~ $90^\circ$  の負荷角 ( $0$ ~ $255$  に対応) のみが生じるので、DcStep の負荷値の範囲は、 $0$ ~ $255$  に制限されます。特定の状況では、負荷角が一時的に  $90^\circ$ ~ $180^\circ$  の範囲に逸脱する場合があります。これは安定的でないとはいえ、読み出し値は最大 511 になります。StallGuard2 をイネーブルするには、*VDCMIN* よりわずかに高い速度または最大 *VMAX* に応じて、*TCOOLTHRS* も設定します。

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

パラメータ	説明	範囲	コメント
<i>vhighfs</i> & <i>vhighchm</i>	<i>CHOPCONF</i> におけるこれらのチョッパー設定フラグは、DcStep 動作用に設定されている必要があります。 <i>VDCMIN</i> を超過すると直ちに、チョッパーは <i>T<sub>OFF</sub></i> が一定のチョッパーに切り替わり、フルステップ動作をするようになります。	0/1	DcStep には 1 を設定
<i>TOFF</i>	DcStep は、多くの場合、 <i>CHOPCONF</i> のオフ時間値が大きい方が多大なメリットが得られます。2 を超える設定値を推奨します。	2..15	8~15 の設定値は、DcStep の動作に 8 を設定する場合と異なる点はありません。
<i>VDCMIN</i>	これは、内部ランプ・ジェネレータを使用する場合の DcStep 動作の下限スレッショルドです。このスレッショルド未満では、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 <i>VDCMIN</i> で動作します。 <i>DC_TIME</i> の設定値と共に調整してください。  StealthChop を活性化しても DcStep はディスエーブルされます。	0..2 <sup>22</sup>	0 : DcStep をディスエーブル DcStep 動作のためには下限速度に設定します。

DC_TIME	<p>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。値を大きくすると、より大きなトルクとより高い速度が可能となり、値を小さくすると、VDCMIN で設定された下限まで動作速度を下げることができます。</p> <p>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。</p>	0..1023	この設定に対する下限値は、クロック・サイクル単位の $t_{BLANK}$ (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	<p>この設定は、DcStep モードでのストール検出を制御します。感度を高めにはこの値を増加します。</p> <p>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 <code>sg_stop</code> 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。</p>	0..255	$DC\_TIME / 16$ よりわずかに大きな値に設定します

### DcStep 動作での実際のモータ速度の測定

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備えて、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 の速度を測定することは、位置カウンタ XACTUAL に基づいて可能です。

そのため、位置カウンタの 2 つのスナップショットを既知の時間差で取得します。

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

例：

クロック周波数が 16.0MHz の場合、速度値に 0.954s の測定遅延が直接生じ、実際の DcStep 速度の 1/100 に 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

TMC5240 は、マイクロステップの電流波形を保存するためにプログラマブルなルックアップ・テーブルを備えています。デフォルトでは、このテーブルはサイン波で事前プログラムされており、ほとんどのステッピング・モータにとって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ピングを著しく改善することができます。ユーザに対する利点は次のとおりです。

- マイクロステッピング - 低コスト・モータで著しく改善
- モータ - 滑らか、かつ静かに動作
- トルク - 機械的な共振が低減することでトルクが向上
- 低周波数のモータ - 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

### マイクロステップ・テーブル

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

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

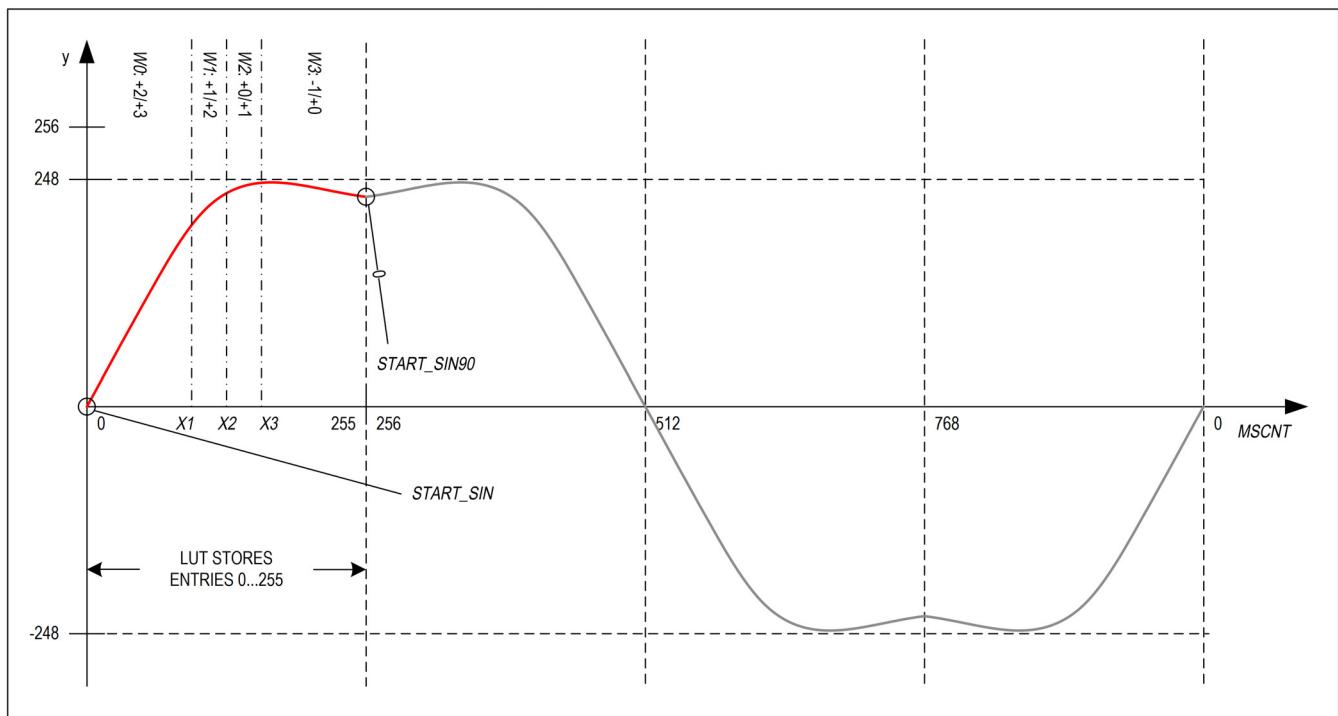


図 35. LUT プログラミングの例

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、インクリメンタル・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：

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 でもかまいません）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つめのモータ・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。

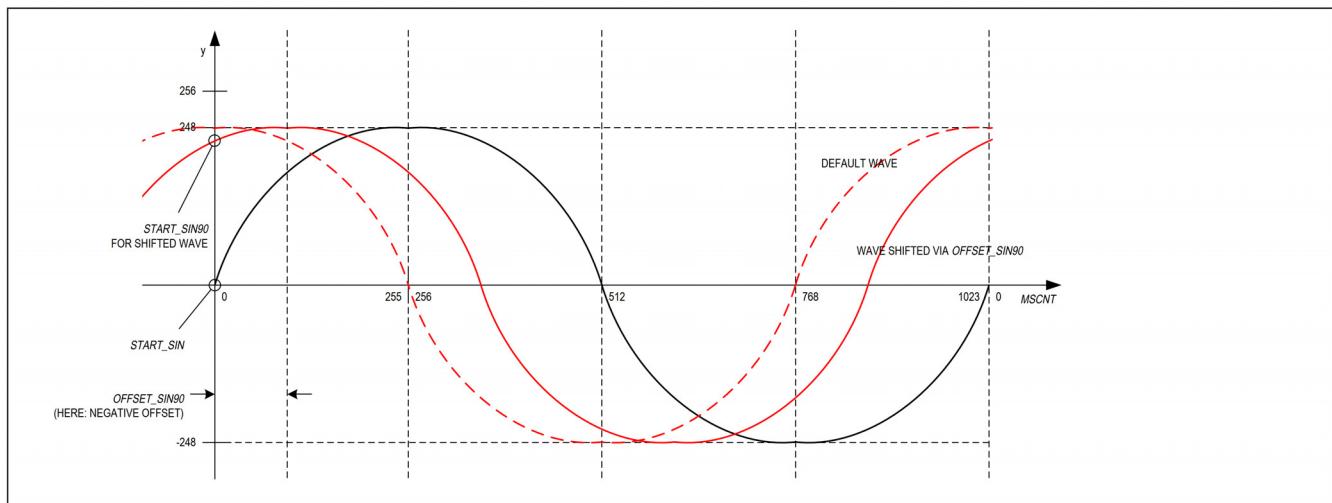


図 36. OFFSET\_SIN90 を使用したコサイン波のシフト

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

`MSLUT[0] = %10101010101010101011010101010100 = 0xAAAAB554`

`MSLUT[1] = %010010101001010101010010101010 = 0x4A9554AA`

`MSLUT[2] = %001001000100100100101001001001 = 0x24492929`

`MSLUT[3] = %0001000000100000100001000100010 = 0x10104222`

`MSLUT[4] = %111110111111111111111111111111 = 0xFBFFFFFF`

`MSLUT[5] = %1011010110110110111011101111101 = 0xB5BB777D`

`MSLUT[6] = %010010010010010101010101010110 = 0x49295556`

`MSLUT[7] = %0000000001000000100001000100010 = 0x00404222`

`MSLUTSEL = 0xFFFF8056:`

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

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

`MSLUTSTART = 0x00F70000:`

`START_SIN_0 = 0, START_SIN90 = 247`

モータの位相シフトを最適化するには、StealthChop2 の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。-17 または+17 までは `START_SIN90 = 246` を用います。`START_SIN` は常に 0 です。

## ABN インクリメンタル・エンコーダ・インターフェイス

TMC5240 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・インターフェイスを備えています。エンコーダは、デジタル・インクリメンタル直交信号（通常 A および B と名付けます）およびインデックス信号（通常ヌルには N、ゼロには Z、インデックスには I が用いられます）を通じて位置を示します。

## N 信号

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

エンコーダがラッチされたイベントのチェック

- 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・カウントを示します。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読み出します。GCONF レジスタのビット *diag0\_nint\_step* を用いて、割込みラインの DIAG0 ピンを設定する必要があります。

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

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

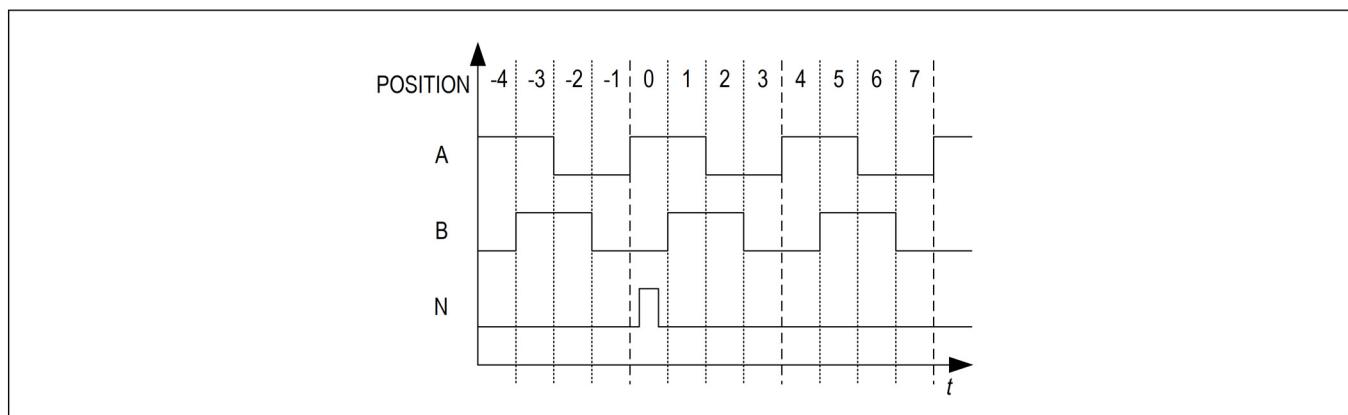


図 37. インクリメンタル・エンコーダの ABN 信号の概要

## エンコーダ・カウンタ X\_ENC

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

### レジスタ ENC\_STATUS

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

### エンコーダ定数 ENC\_CONST

エンコーダ定数（またはエンコーダ・ファクタ）*ENC\_CONST* は、インクリメンタル・エンコーダの直交信号 AB の極性がそれぞれ変化したときに、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、固定数表示により、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・ファクタ :  $ENC\_CONST = 0x0001.0x0000 = FACTOR.FRACTION$
- -1.0 のエンコーダ・ファクタ :  $ENC\_CONST = 0xFFFF.0x0000$ 。これは  $0x00010000$  の 2 の補数です。これは、 $(2^{16} - (FACTOR + 1)).(2^{16} - FRACTION)$  に等しくなります。
- 10 進数モードのエンコーダ・ファクタ 25.6 :  $00025.6000 = 0x0019.0x1770 = FACTOR.DECIMALS$  (DECIMALS = 小数点以下 4 術)
- 10 進数モードのエンコーダ・ファクタ -25.6 :  $(2^{16} - (25 + 1)).(10000 - 6000) = (2^{16} - 26).(4000) = 0xFFE6.0x0FA0$
- 負のエンコーダ定数は、次式を用いて計算できます :  $(2^{16} - (FACTOR + 1)).(10000 - DECIMALS)$

### モータの分解能に合うエンコーダ設定

モータ・パラメータに対するエンコーダの設定例：

1. USC = 256 マイクロステップ
2. FSC = 200 フルステップ・モータ
3. ファクタ =  $FSC \times USC / \text{エンコーダ分解能}$

表 25. 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

ENCODER RESOLUTION	REQUIRED ENCODER FACTOR	COMMENT
200	256	
360	$142.2222$ $= 9320675.5555 / 2^{16}$ $= 1422222.2222 / 10000$	No exact match possible!
500	$102.4$ $= 6710886.4 / 2^{16}$ $= 1024000 / 10000$	Exact match with decimal setting
1000	51.2	Exact match with decimal setting
1024	50	
4000	12.8	Exact match with decimal setting
4096	12.5	
16384	3.125	

例：

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

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

### リセット、ディスエーブル／停止、パワー・ダウント

#### 緊急停止

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

#### 外部リセットおよびスリープ・モード

リセットおよびスリープ・モードは SLEEPN ピンで制御されます。

持続時間が  $30\mu\text{s}$  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)。

$30\mu\text{s}$  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、あるいはスリープ状態およびリセット状態からの解放後、レジスタは再設定する必要があります。

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

モータ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、VsまたはVcc\_IO（これは高電圧ピンです）に接続します。

## 保護およびドライバ診断

TMC5240のドライバでは、GNDへの短絡保護や低電圧検出など、フルセットの診断機能および保護機能が利用できます。開放負荷状態の検出では、モータ・コイルの接続が切離されているかどうかをテストできます。詳細については [DRV\\_STATUS](#) レジスタの表を参照してください。

ステータス・フラグの他、TMC5240では、チップ温度の測定および読み出しや、モータ位相用の巻線温度へのフィードバックが可能です。

システムの信頼性と回路全体の保護能力を向上するため、TMC5240には過電圧コンパレータとトリガ出力 OV が内蔵され、電源電圧の過剰な上昇があった場合に外部スイッチを制御します。

## 過電流保護

過電流保護 (OCP) は、レール（電源電圧とグランド）への短絡と出力 (OUT1A、OUT2A、OUT1B、OUT2B) 間の短絡からデバイスを保護します。

OCP スレッショルドは、選択したフルスケール電流範囲に依存します。また、それぞれのスレッショルド値については [電気的特性](#) の表を参照してください。

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

デグリッチ時間（ブランкиング時間）より長い間にわたって出力電流が OCP スレッショルドを上回る場合、OCP イベントが検出されます。

OCP イベントが検出されると、Hブリッジが直ちにディスエーブルされます。

短絡保護が 3 回行われると、フォルト・フラグ (DRV\_STATUS レジスタの s2ga、s2gb、s2vsb、s2vsa) がセットされブリッジが連続的にディスエーブルされます。

デバイスは依然として活性で、設定やステータスの読み出しが可能です。

電源ブリッジを再度イネーブルするには、DRV\_ENN ピンをローにして再度ハイにする必要があります。

別のオプションは、CHOPCONF の TOFF を 0 にして電源ブリッジをディスエーブルし、TOFF > 0 にしてブリッジを再度イネーブルすることです。

## 過熱保護およびサーマル・シャットダウン

TMC5240 は過熱保護機能を内蔵しています。

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

更に、TMC5240 は、ADC ベースで設定可能な過熱事前警告レベルをサポートしています。これは、パラメータ OVERTEMPPREWARNING\_VTH を用いてレジスタ OTW\_OV\_VTH で設定できます。ADC がチップの平均温度を検出するのに対し、ドライバ段はこれよりはるかに高い温度となっている可能性があります。このことは、TMC5240 がサーマル・シャットダウン状態になる可能性があり、事前警告は低温に設定されている場合でもアサートされないことがある、ということを示すにすぎません。

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

## 温度測定

TMC5240 は内部チップ温度およびモータ温度を測定する機能を備えています。

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

### チップ温度の測定

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

°C 単位の最終温度は次の式で計算できます。

$$\text{ADC\_TEMP} = 7.7 \times \text{TEMP} + 2038$$

$$\text{TEMP} [^{\circ}\text{C}] = \frac{\text{ADC\_TEMP} - 2038}{7.7}$$

### 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、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* 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。

### 過電圧保護と OV ピン

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

この電圧は、ドライバ出力段で電源レールに帰還されます。

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

ドライバや接続回路を保護するために、TMC5240 では、過電圧の検出および保護のための機構を備えています。

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

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

電源電圧は内蔵 ADC で常にモニタされています。

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

電源電圧の実際の ADC 値は、レジスタ *ADC\_VSUPPLY\_AIN* のパラメータ *ADC\_VSUPPLY* で読み出せます。

ADC の値を *V<sub>S</sub>* に変換したり、あるいはその逆を行ったりするには、次の式を用います。

$$V_S = \text{ADC\_VSUPPLY} \times 9.732\text{mV}$$

OV 出力ピンは、過電圧モニタの実際の状態を示します。

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

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

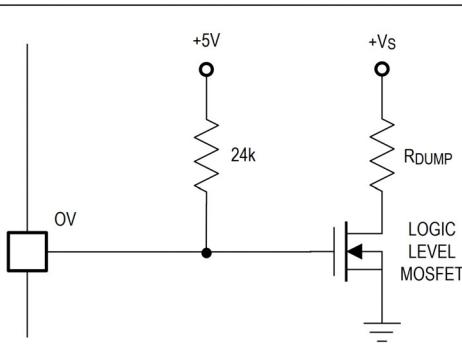


図 38. ブレーキ・チョッパー回路の例

### 短絡保護 (GNDへの短絡および VSへの短絡)

TMC5240 の電力段は、ハイサイド MOSFET を流れる電流を追加測定することで、回路の短絡状態に対し保護されます。大半の短絡状態は、システムのグランドに接続された導電素子に接触した場合などにおけるモータ・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、基本的に短絡を回避する必要があります。

フルスケール電流設定に応じて、様々な過電流保護スレッショルドでローサイドの短絡保護がトリガします。

表 26. フルスケール電流設定に基づく過電流保護スレッショルド

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

### 開放負荷診断

断線したケーブルは、コネクタが確実に差し込まれていない場合などのように、よく見られるシステム故障の原因です。TMC5240 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。したがって、低電圧条件、高モータ速度設定、あるいは短絡および過熱条件も、開放負荷フラグをトリガする可能性があります。モータ停止時には、コイルの電流がゼロであるため、開放負荷を検知できません。

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

### 低電圧ロックアウト保護

TMC5240 は、Vs、V<sub>CC<sub>IO</sub></sub>、チャージ・ポンプに対する UVLO 保護機能を備えています。

V<sub>s</sub> の UVLO 条件は 4.05V (最大値) 未満でトリガされます。

V<sub>CC<sub>IO</sub></sub> の UVLO 条件は 1.95V (最大値) 未満でトリガされます。

チャージ・ポンプの UVLO 条件は、コンデンサ値が不適切であることなどによるチャージ・ポンプのエラー条件が生じた場合にトリガされます。

V<sub>s</sub> の UVLO 条件は、レジスタ *GSTAT* のフラグ *vm\_uvlo* で読み出せます。このフラグは「書き込んでクリア」フラグです。これをクリアするにはアクティブに 1 に設定する必要があります。

V<sub>CC<sub>IO</sub></sub> の UVLO 時には、IC との通信は不可能で、ドライバはディスエーブルされます。DIAG0 ピンはアクティブ・ロー (オープン・ドレン) 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。

TMC5240 のモータ位相出力ピンは、正電圧電源 (V<sub>S</sub> ピン) に 1 $\mu$ 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。

## 外部アナログ入力 AIN のモニタリング

TMC5240 は、外部アナログ入力 AIN を備えており、内部 ADC で連続的にサンプリングされます。

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

ADC の値を V<sub>AIN</sub> に変換したり、あるいはその逆を行ったりするには、次の式を用います。

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

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

## クロック発振器とクロック入力

### 内部クロックの使用

内部クロック発振器を使用するには、CLK 入力ピンを IC 付近で直接 GND に接続します。

内部クロックは、12.5MHz (代表値) の周波数で動作しています。

### 外部クロックの使用

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

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

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

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

外部クロック入力は、外部クロックが CLK ピンに供給されると直ちにイネーブルされます。

レジスタ *IOIN* の読み出しひつ *ext\_clk* は、現在そのクロック源を使用しているかをフィードバックします (1 = 外部クロック)。

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

## 全般的なレジスタ・マッピングおよびレジスタ情報

このセクションでは、レジスタ・マップに関する全般的な情報を示します。

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

– すべてのレジスタは、特に指定のない限り、パワー・アップ時に 0 にリセットされます。

– 書込みアクセス時にはアドレス Addr に 0x80 を追加してください。

表 27. レジスタ・マップの概要

レジスタ	説明
全般的な設定レジスタ	これらのレジスタは以下の内容を含みます <ul style="list-style-type: none"> <li>グローバル設定</li> <li>グローバル・ステータス・フラグ</li> <li>インターフェイス設定</li> <li>I/O 信号設定</li> </ul>

表 27. レジスタ・マップの概要（続き）

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

## レジスタ・マップ

## TMC5240

ADDRESS	NAME	MSB							LSB			
<b>General Configuration Registers</b>												
0x00	<a href="#">GCONF[31:24]</a>	-	-	-	-	-	-	-	-			
	<a href="#">GCONF[23:16]</a>	-	-	-	length_step_pulse[3:0]				direct_mode			
	<a href="#">GCONF[15:8]</a>	stop_enable	small_hysteresis	diag1_poscomp_pushtpull	diag0_int_pushpull	-	-	-	diag1_np_poscomp_dir			
	<a href="#">GCONF[7:0]</a>	diag0_nint_step	-	-	shaft	-	en_pwm_mode	fast_standstill	-			
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_cp	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>	-	-	-	-	SILICON_RV[2:0]						
	<a href="#">IOIN[15:8]</a>	ADC_ER	EXT_CLK	EXT_RESETS_DET	OUTPUT	COMP_B1_B2	COMP_A1_A2	COMP_B	COMP_A			
	<a href="#">IOIN[7:0]</a>	reserved	UART_EN	ENCN	DRV_EN_N	ENCA	ENCB	REFR	REFL			
0x05	<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]										
0x06	<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]										
0x0A	<a href="#">DRV_CONF[31:24]</a>	-	-	-	-	-	-	-	-			
	<a href="#">DRV_CONF[23:16]</a>	-	-	-	-	-	-	-	-			

ADDRESS	NAME	MSB							LSB
	<a href="#">DRV_CONF[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">DRV_CONF[7:0]</a>	-	-	SLOPE_CONTROL[1:0]	-	-	-	CURRENT_RANGE[1:0]	
0x0B	<a href="#">GLOBAL SCALER[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">GLOBAL SCALER[23:16]</a>	-	-	-	-	-	-	-	-
	<a href="#">GLOBAL SCALER[15:8]</a>	-	-	-	-	-	-	-	-
	<a href="#">GLOBAL SCALER[7:0]</a>							GLOBALSCALER[7:0]	
<b>Velocity Dependent Configuration Registers</b>									
0x10	<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]	
0x11	<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]	
0x12	<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]	
0x13	<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]	
0x14	<a href="#">TCOOLTHRS[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">TCOOLTHRS[23:16]</a>	-	-	-	-	-	-	TCOOLTHRS[19:16]	
	<a href="#">TCOOLTHRS[15:8]</a>							TCOOLTHRS[15:8]	
	<a href="#">TCOOLTHRS[7:0]</a>							TCOOLTHRS[7:0]	
0x15	<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>									
0x20	<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]
0x21	<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]	

ADDRESS	NAME	MSB							LSB
0x22	<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]			
0x23	<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]			
0x24	<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]			
0x25	<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]			
0x26	<a href="#">AMAX[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">AMAX[23:16]</a>	-	-	-	-	-	-	AMAX[17:16]	
	<a href="#">AMAX[15:8]</a>					AMAX[15:8]			
	<a href="#">AMAX[7:0]</a>					AMAX[7:0]			
0x27	<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]			
0x28	<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]			
0x29	<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]			
0x2A	<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]			
0x2B	<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]			
0x2C	<a href="#">TZEROWAIT[31:24]</a>	-	-	-	-	-	-	-	-
	<a href="#">TZEROWAIT[23:16]</a>	-	-	-	-	-	-	-	-

ADDRESS	NAME	MSB							LSB				
	<a href="#">TZEROWAIT[15:8]</a>	TZEROWAIT[15:8]											
	<a href="#">TZEROWAIT[7:0]</a>	TZEROWAIT[7:0]											
0x2D	<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]											
0x2E	<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]											
0x2F	<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]											
0x30	<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]											
<b>Ramp Generator Driver Feature Control Registers</b>													
0x33	<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]											
0x34	<a href="#">SW_MODE[31:24]</a>	-	-	-	-	-	-	-	-				
	<a href="#">SW_MODE[23:16]</a>	-	-	-	-	-	-	-	-				
	<a href="#">SW_MODE[15:8]</a>	-	virtual_st op_enc	en_virtua l_stop_r	en_virtua l_stop_l	en_softst op	sg_stop	en_latch _encoder	latch_r_i nactive				
	<a href="#">SW_MODE[7:0]</a>	latch_r_a ctive	latch_l_i nactive	latch_l_a ctive	swap_lr	pol_stop _r	pol_stop _l	stop_r_e nable	stop_l_e nable				
0x35	<a href="#">RAMP_STAT[31:24]</a>	-	-	-	-	-	-	-	-				
	<a href="#">RAMP_STAT[23:16]</a>	-	-	-	-	-	-	-	-				
	<a href="#">RAMP_STAT[15:8]</a>	status_v irtual_st o_p_r	status_v irtual_st o_p_l	status_s g	second_ move	t_zerowa it_active	vzero	position_ reached	velocity_ reached				
	<a href="#">RAMP_STAT[7:0]</a>	event_po s_reache d	event_st op_sg	event_st op_r	event_st op_l	status_la tch_r	status_la tch_l	status_st op_r	status_st op_l				
0x36	<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]											
<b>Encoder Registers</b>													
0x38	<a href="#">ENCMODE[31:24]</a>	-	-	-	-	-	-	-	-				
	<a href="#">ENCMODE[23:16]</a>	-	-	-	-	-	-	-	-				

ADDRESS	NAME	MSB							LSB				
	<a href="#">ENCMODE[15:8]</a>	-	-	-	-	-	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				
0x39	<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]											
0x3A	<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]											
0x3B	<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				
0x3C	<a href="#">ENC_LATCH[31:24]</a>	ENC_LATCH[31:24]											
	<a href="#">ENC_LATCH[23:16]</a>	ENC_LATCH[23:16]											
	<a href="#">ENC_LATCH[15:8]</a>	ENC_LATCH[15:8]											
	<a href="#">ENC_LATCH[7:0]</a>	ENC_LATCH[7:0]											
0x3D	<a href="#">ENC_DEVIATION[31:24]</a> 1	-	-	-	-	-	-	-	-				
	<a href="#">ENC_DEVIATION[23:16]</a> 1	-	-	-	-	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]											
0x3E	<a href="#">VIRTUAL_STOP_L[31:24]</a> 4	VIRTUAL_STOP_L[31:24]											
	<a href="#">VIRTUAL_STOP_L[23:16]</a> 6	VIRTUAL_STOP_L[23:16]											
	<a href="#">VIRTUAL_STOP_L[15:8]</a> 1	VIRTUAL_STOP_L[15:8]											
	<a href="#">VIRTUAL_STOP_L[7:0]</a>	VIRTUAL_STOP_L[7:0]											
0x3F	<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>ADC Registers</b>													
0x50	<a href="#">ADC_VSUPPLY_AIN[3:24]</a>	-	-	-	ADC_AIN[12:8]								
	<a href="#">ADC_VSUPPLY_AIN[2:16]</a>	ADC_AIN[7:0]											

ADDRESS	NAME	MSB							LSB
	<a href="#">ADC_VSUPPLY_AIN[1:8]</a>	-	-	-	ADC_VSUPPLY[12:8]				
	<a href="#">ADC_VSUPPLY_AIN[7:0]</a>	ADC_VSUPPLY[7:0]							
0x51	<a href="#">ADC_TEMP[31:24]</a>	-	-	-	RESERVED[12:8]				
	<a href="#">ADC_TEMP[23:16]</a>	RESERVED[7:0]							
	<a href="#">ADC_TEMP[15:8]</a>	-	-	-	ADC_TEMP[12:8]				
	<a href="#">ADC_TEMP[7:0]</a>	ADC_TEMP[7:0]							
0x52	<a href="#">OTW_OV_VTH[31:24]</a>	-	-	-	OVERTEMPPREWARNING_VTH[12:8]				
	<a href="#">OTW_OV_VTH[23:16]</a>	OVERTEMPPREWARNING_VTH[7:0]							
	<a href="#">OTW_OV_VTH[15:8]</a>	-	-	-	OVERVOLTAGE_VTH[12:8]				
	<a href="#">OTW_OV_VTH[7:0]</a>	OVERVOLTAGE_VTH[7:0]							
<b>Motor Driver Registers</b>									
0x60	<a href="#">MSLUT_0[31:24]</a>	MSLUT_0[31:24]							
	<a href="#">MSLUT_0[23:16]</a>	MSLUT_0[23:16]							
	<a href="#">MSLUT_0[15:8]</a>	MSLUT_0[15:8]							
	<a href="#">MSLUT_0[7:0]</a>	MSLUT_0[7:0]							
0x61	<a href="#">MSLUT_1[31:24]</a>	MSLUT_1[31:24]							
	<a href="#">MSLUT_1[23:16]</a>	MSLUT_1[23:16]							
	<a href="#">MSLUT_1[15:8]</a>	MSLUT_1[15:8]							
	<a href="#">MSLUT_1[7:0]</a>	MSLUT_1[7:0]							
0x62	<a href="#">MSLUT_2[31:24]</a>	MSLUT_2[31:24]							
	<a href="#">MSLUT_2[23:16]</a>	MSLUT_2[23:16]							
	<a href="#">MSLUT_2[15:8]</a>	MSLUT_2[15:8]							
	<a href="#">MSLUT_2[7:0]</a>	MSLUT_2[7:0]							
0x63	<a href="#">MSLUT_3[31:24]</a>	MSLUT_3[31:24]							
	<a href="#">MSLUT_3[23:16]</a>	MSLUT_3[23:16]							
	<a href="#">MSLUT_3[15:8]</a>	MSLUT_3[15:8]							
	<a href="#">MSLUT_3[7:0]</a>	MSLUT_3[7:0]							
0x64	<a href="#">MSLUT_4[31:24]</a>	MSLUT_4[31:24]							
	<a href="#">MSLUT_4[23:16]</a>	MSLUT_4[23:16]							
	<a href="#">MSLUT_4[15:8]</a>	MSLUT_4[15:8]							
	<a href="#">MSLUT_4[7:0]</a>	MSLUT_4[7:0]							
0x65	<a href="#">MSLUT_5[31:24]</a>	MSLUT_5[31:24]							
	<a href="#">MSLUT_5[23:16]</a>	MSLUT_5[23:16]							
	<a href="#">MSLUT_5[15:8]</a>	MSLUT_5[15:8]							
	<a href="#">MSLUT_5[7:0]</a>	MSLUT_5[7:0]							
0x66	<a href="#">MSLUT_6[31:24]</a>	MSLUT_6[31:24]							
	<a href="#">MSLUT_6[23:16]</a>	MSLUT_6[23:16]							
	<a href="#">MSLUT_6[15:8]</a>	MSLUT_6[15:8]							
	<a href="#">MSLUT_6[7:0]</a>	MSLUT_6[7:0]							
0x67	<a href="#">MSLUT_7[31:24]</a>	MSLUT_7[31:24]							
	<a href="#">MSLUT_7[23:16]</a>	MSLUT_7[23:16]							

ADDRESS	NAME	MSB							LSB						
	<a href="#">MSLUT_7[15:8]</a>	MSLUT_7[15:8]													
	<a href="#">MSLUT_7[7:0]</a>	MSLUT_7[7:0]													
0x68	<a href="#">MSLUTSEL[31:24]</a>	X3[7:0]													
	<a href="#">MSLUTSEL[23:16]</a>	X2[7:0]													
	<a href="#">MSLUTSEL[15:8]</a>	X1[7:0]													
	<a href="#">MSLUTSEL[7:0]</a>	W3[1:0]		W2[1:0]		W1[1:0]		W0[1:0]							
0x69	<a href="#">MSLUTSTART[31:24]</a>	OFFSET_SIN90[7:0]													
	<a href="#">MSLUTSTART[23:16]</a>	START_SIN90[7:0]													
	<a href="#">MSLUTSTART[15:8]</a>	-	-	-	-	-	-	-	-						
	<a href="#">MSLUTSTART[7:0]</a>	START_SIN[7:0]													
0x6A	<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]													
0x6B	<a href="#">MSCURACT[31:24]</a>	-	-	-	-	-	-	-	CUR_A[8]						
	<a href="#">MSCURACT[23:16]</a>	CUR_A[7:0]													
	<a href="#">MSCURACT[15:8]</a>	-	-	-	-	-	-	-	CUR_B[8]						
	<a href="#">MSCURACT[7:0]</a>	CUR_B[7:0]													
0x6C	<a href="#">CHOPCONF[31:24]</a>	diss2vs	diss2g	reserved	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]								
0x6D	<a href="#">COOLCONF[31:24]</a>	-	-	-	-	-	-	-	sfilt						
	<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]									
0x6E	<a href="#">DCCTRL[31:24]</a>	-	-	-	-	-	-	-	-						
	<a href="#">DCCTRL[23:16]</a>	DC_SG[7:0]													
	<a href="#">DCCTRL[15:8]</a>	-	-	-	-	-	-	DC_TIME[9:8]							
	<a href="#">DCCTRL[7:0]</a>	DC_TIME[7:0]													
0x6F	<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	s2vsd	-	-	SG_RESULT[9:8]							
	<a href="#">DRV_STATUS[7:0]</a>	SG_RESULT[7:0]													
0x70	<a href="#">PWMCONF[31:24]</a>	PWM_LIM[3:0]					PWM_REG[3:0]								
	<a href="#">PWMCONF[23:16]</a>	pwm_dis	pwm_me	as_sd_e	enable	FREEWHEEL[1:0]	pwm_au	pwm_au	PWM_FREQ[1:0]						
	<a href="#">PWMCONF[15:8]</a>	PWM_GRAD[7:0]													

ADDRESS	NAME	MSB							LSB		
	<a href="#">PWMCONF[7:0]</a>	PWM_OFS[7:0]									
0x71	<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]									
0x72	<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]									
0x74	<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_filt_en		
	<a href="#">SG4_THRS[7:0]</a>	SG4_THRS[7:0]									
0x75	<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]									
0x76	<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]									

## レジスタの詳細

### GCONF (0x0)

グローバル設定フラグ

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	-	-	-	-	-	-	-	-
<b>Reset</b>	-	-	-	-	-	-	-	-
<b>Access Type</b>	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	-	-	-	length_step_pulse[3:0]				direct_mode
<b>Reset</b>	-	-	-	0x0				0x0
<b>Access Type</b>	-	-	-	Write, Read				Write, Read

BIT	15	14	13	12	11	10	9	8
Field	stop_enable	small_hysteresis	diag1_poscomp_pushpull	diag0_int_pushpull	—	—	—	diag1_nposcomp_dir
Reset	0x0	0x0	0x0	0x0	—	—	—	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	—	—	—	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	diag0_nint_step	—	—	shaft	—	en_pwm_mode	fast_standstill	—
Reset	0x0	—	—	0x0	—	0x0	0x0	—
Access Type	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
length_step_pulse	20:17	cDriverのみ : length_step_pulse = 0 : STEP出力はステップごとに切り替わる ; length_step_pulse = 1~15 : クロック・サイクル数で表したSTEPピンのハイ時間	
direct_mode	16	シリアル・インターフェイ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 via serial interface: Register XTARGET (0x2D) specifies signed coil A current (bits 8..0) and coil B current (bits 24..16). In this mode, the current is scaled by IHOLD setting. Velocity based current regulation of StealthChop2 is not available in this mode. The automatic StealthChop2 current regulation will work only for low stepper motor velocities.
stop_enable	15	モータ・ハード・ストップ機能を有効化。	0x0: Normal operation 0x1: Emergency stop: ENCA stops the sequencer when tied high (no steps become executed by the sequencer, motor goes to standstill state).
small_hysteresis	14		0x0: Hysteresis for step frequency comparison is 1/16 0x1: Hysteresis for step frequency comparison is 1/32
diag1_poscomp_pushpull	13	DIAG1の出力タイプ設定。	0x0: DIAG1 is open collector output (active low) 0x1: Enable DIAG1 push pull output (active high)
diag0_int_pushpull	12	DIAG0の出力タイプ設定。	0x0: DIAG0_SW is open collector output (active low) 0x1: Enable DIAG0_SW push pull output (active high)
diag1_nposcomp_dir	8	UARTを使用しない場合のDIAG1の出力設定。	0x0: DIAG1 outputs position compare signal 0x1: Enable DIAG1 as DIR output for external STEP/DIR driver
diag0_nint_step	7	DIAG0の出力設定。	0x0: DIAG0 outputs interrupt signal 0x1: Enable DIAG0 as STEP output (half frequency, dual edge triggered or frequency when length_step_pulse != 0) for external STEP/DIR driver
shaft	4	モータの方向／方向符号の変更。	0x0: Default motor direction 0x1: Inverse motor direction

ビット・フィールド	ビット	説明	デコード
en_pwm_mode	2	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 IHOLD=nominal IRUN current, only.
fast_standstill	1	停止検出までのステップの実行タイムアウト	0x0: Normal time: 2^20 clocks 0x1: Short time: 2^18 clocks

**GSTAT (0x1)**

グローバル・ステータス・フラグ

(各フラグをクリアするには「1」のビットを再書き込みします)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	vm_uvlo	register_reset	uv_cp	drv_err	reset
<b>Reset</b>	—	—	—	0x1	0x1	0x1	0x0	0x1
<b>Access Type</b>	—	—	—	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	1: 最後のリセット後にVs低電圧が発生しました。（ヒント：最初のブートアップ後はアクティブになります。デバイスが動作しているときにフォルトを検出するにはブートアップ後にフラグをクリアします）	
register_reset	3	ヒント：最初のブートアップ後はアクティブになります。デバイスが動作しているときにレジスタ・マップのリセットを検出するにはブートアップ後にフラグをクリアします	0x0: normal operation 0x1: Indicates that the registermap has been reset. All registers have been cleared to reset values.
uv_cp	2	チャージ・ポンプ低電圧条件フラグ。（ヒント：最初のブートアップ後はアクティブになります。デバイスが動作しているときにフォルトを検出するにはブートアップ後にフラグをクリアします）	0x0: normal operation 0x1: Indicates an undervoltage on the charge pump. The driver is disabled during undervoltage. This flag is latched for information.

ビット・フィールド	ビット	説明	デコード
drv_err	1	ドライバ・エラー・フラグ	0x0: normal operation 0x1: 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 (ヒント: 最初のブートアップ後はアクティブになります。デバイスが動作しているときにリセットされたことを検出するにはブートアップ後にフラグをクリアします)	0x0: normal operation 0x1: Indicates that the IC has been reset.

### IFCNT (0x2)

インターフェイス伝送カウンタ。

このレジスタは、UARTインターフェイ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。読み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。

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	IFCNT[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
IFCNT	7:0	インターフェイス伝送カウンタ。このレジスタは、UARTインターフェイ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。読み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
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	SENDDELAY[3:0]			
<b>Reset</b>	—	—	—	—	0x0			
<b>Access Type</b>	—	—	—	—	Write, Read			
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	NODEADDR[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
SENDDELAY	11:8	SWUARTのノード設定	0x0: 8 bit times (not allowed with multiple nodes) 0x2: 3 × 8 bit times 0x4: 5 × 8 bit times 0x6: 7 × 8 bit times 0x8: 9 × 8 bit times 0xA: 11 × 8 bit times 0xC: 13 × 8 bit times 0xE: 15 × 8 bit times
NODEADDR	7:0	<b>NODEADDR :</b> これらの8ビットは、UARTインターフェイス用のデバイスのアドレスを設定します。アドレスはSDI、SCK、CSNIによる定義に従つて1ずつ7までインクリメントします。 CSN、SCK、SDI 000 : +0 001 : +1 010 : +2 011 : +3 100 : +4 101 : +5 110 : +6 111 : +7 範囲 : 0~254 (254より先へはインクリメントしません)	

## I0IN (0x4)

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

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	VERSION[7:0]							
<b>Reset</b>								
<b>Access Type</b>	Read Only							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	SILICON_RV[2:0]		
<b>Reset</b>	—	—	—	—	—	0x0		
<b>Access Type</b>	—	—	—	—	—	Read Only		
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	ADC_ERR	EXT_CLK	EXT_RES_DET	OUTPUT	COMP_B1_B2	COMP_A1_A2	COMP_B	COMP_A
<b>Reset</b>	0x0	0x0	0x0	0x1	0x0	0x0	0x0	0x0
<b>Access Type</b>	Read Only	Read Only	Read Only	Write, Read	Read Only	Read Only	Read Only	Read Only
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	reserved	UART_EN	ENCN	DRV_ENN	ENCA	ENCB	REFR	REFL
<b>Reset</b>		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

ビット・フィールド	ビット	説明
VERSION	31:24	0x40 = ICの最初のバージョン 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。
SILICON_RV	18:16	シリコン・リビジョン番号
ADC_ERR	15	1 : ADCが正しく動作していないことを示しています。ADC機能は使用しないでください。
EXT_CLK	14	0 : クロック信号 (12.5MHz) を生成するために内部発振器が用いられます。 1 : クロック信号を生成するために外部発振器が用いられます。
EXT_RES_DET	13	1 : REFとGNDの間に外付け抵抗が接続されています 0 : 外付け抵抗は検出されていません
OUTPUT	12	UART_EN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ICのチェーン・アドレス指定を行うことです。注意 : 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。
COMP_B1_B2	11	COMP_B1_B2 (StallGuard4コンパレータB、ICテスト用)
COMP_A1_A2	10	COMP_A1_A2 (StallGuard4コンパレータA、ICテスト用)
COMP_B	9	COMP_B (チョッパー・コンパレータB、ICテスト用)
COMP_A	8	COMP_A (チョッパー・コンパレータA、ICテスト用)
reserved	7	
UART_EN	6	1 = UARTインターフェイスがイネーブル
ENCN	5	Nチャンネルの状態
DRV_ENN	4	ドライバのディスエーブル/イネーブル状態。
ENCA	3	Aチャンネルの状態
ENCB	2	Bチャンネルの状態
REFR	1	
REFL	0	

**X\_COMPARE (0x5)**

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	X_COMPARE[31:24]							
<b>Reset</b>	0xFFFFFFFF							
<b>Access Type</b>	Write, Read							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	X_COMPARE[23:16]							
<b>Reset</b>	0xFFFFFFFF							
<b>Access Type</b>	Write, Read							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	X_COMPARE[15:8]							
<b>Reset</b>	0xFFFFFFFF							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	X_COMPARE[7:0]							
<b>Reset</b>	0xFFFFFFFF							
<b>Access Type</b>	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・コントローラ位置ストローブ用の位置比較レジスタ。 X_COMPAREは絶対的な位置です。 位置パルスは出力SWP_DIAG1で使用できます。</p> <p>XACTUAL = X_COMPARE: 出力信号PP（位置パルス）がハイになります。 位置が不一致の場合はロー状態に戻ります。</p> <p>X_COMPARE_REPEATが1より大きい場合、X_COMPAREは、<u>periodic</u>位置ストローブ・トリガ出力の位置リファレンスです。</p>

**X\_COMPARE\_REPEAT (0x6)**

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	-	-	-	-	-	-	-	-
<b>Reset</b>	-	-	-	-	-	-	-	-
<b>Access Type</b>	-	-	-	-	-	-	-	-

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

ピット・フィールド	ピット	説明
X_COMPARE_REPEAT	23:0	<p>このレジスタは、マイクロステップ単位の相対距離を（MRESの設定に基づき）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、位置比較パルスが発生します。</p> <p>それによって、X_COMPAREレジスタは、正または負の方向に向かって行われたX_COMPARE_REPEATステップのモジューロ計算を行うためのベース位置を定義します。</p> <p>X_COMPAREの位置に達すると、X_COMPAREはX_COMPARE_REPEATだけインクリメント／デクリメントされます。</p>

### DRV\_CONF (0xA)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	-							
<b>Reset</b>	-							
<b>Access Type</b>	-							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	-							
<b>Reset</b>	-							
<b>Access Type</b>	-							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	-							
<b>Reset</b>	-							
<b>Access Type</b>	-							

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	SLOPE_CONTROL[1:0]		—	—	CURRENT_RANGE[1:0]	
<b>Reset</b>	—	—	0x0		—	—	0x0	
<b>Access Type</b>	—	—	Write, Read		—	—	Write, Read	

ビット・フィールド	ビット	説明	デコード
SLOPE_CONTROL	5:4	勾配制御設定	0x0: 100V/μs 0x1: 200V/μs 0x2: 400V/μs 0x3: 800V/μs
CURRENT_RANGE	1:0	この設定により、モータ電流範囲に合わせたドライバのRDSon電流検出機能の基本的な調整ができます。最高の電流精度を実現するには、最小のフィッティング範囲を選択してください。値はピーク電流の設定値です。	0x0: 1A 0x1: 2A 0x2: 3A 0x3: 3A

### GLOBAL SCALER (0xB)

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

ピット・フィールド	ピット	説明
GLOBALSCALER	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乗じられます。この値を選択してから、他の設定値を調整してください。この値はチョッパーのヒステリシスにも影響するためです。この値は、モータ電流を微調整することだけを目的としています。</p> <p>0 : フルスケール（または256を書き込み）      1 ... 31 : 動作には許容されません      32 ... 255 : 最大電流の32/256~255/256</p> <p>ヒント：最高の結果を得るには、128より大きな値にすることを推奨します。</p>

### IHOLD\_IRUN (0x10)

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					

ピット・フィールド	ピット	説明
IRUNDELAY	27:24	<p>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。</p> <p>0 : 即時パワー・アップ      1~15 : 電流インクリメント・ステップあたりの遅延はIRUNDELAY * 512クロックの倍数</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・ジャーキーを回避できます。</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2^18クロックの倍数</p>
IRUN	12:8	<p>モータ動作電流 (0=1/32、…、31=32/32)</p> <p>ヒント：最高のマイクロステップ性能を得るには、設定値の範囲を16~31にします。</p>

ビット・フィールド	ビット	説明
IHOLD	4:0	停止時電流 (0=1/32、…、31=32/32) StealthChop2モードと組み合わせた場合、IHOLD = 0と設定することで、モータ停止のためにフリーホールドまたはコイル短絡回路を選択できます。

**TPOWERDOWN (0x11)**

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

ビット・フィールド	ビット	説明
TPOWERDOWN	7:0	<p>TPOWERDOWNは、モータの停止 (stst) 後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。</p> <p>注意 : StealthChop2のPWM_OFFSET_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。</p> <p>リセット後のデフォルト = 10  <math>0 \cdots (2^8 - 1) \times 2^{18} t_{CLK}</math></p>

**TSTEP (0x12)**

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—

BIT	23	22	21	20	19	18	17	16	
<b>Field</b>	—	—	—	—	TSTEP[19:16]				
<b>Reset</b>	—	—	—	—	0x0				
<b>Access Type</b>	—	—	—	—	Read Only				
BIT	15	14	13	12	11	10	9	8	
<b>Field</b>	TSTEP[15:8]								
<b>Reset</b>	0x0								
<b>Access Type</b>	Read Only								
BIT	7	6	5	4	3	2	1	0	
<b>Field</b>	TSTEP[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
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に関するスレッショルドはすべて、比較値の1/16のヒステリシスを使用してクロックまたはステップ周波数のジッタを補償します。フラグsmall_hysteresisはこのヒステリシスをより小さな値である1/32に変更します。</p> <p>(Txxx × 15/16) - 1または (Txxx × 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に対して測定されるTSTEPは、(2<sup>24</sup> / V) ≤ TSTEP ≤ 2<sup>24</sup> / V - 1の範囲になります。</p> <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、各マイクロステップの速度を示します。これは、安定していない可能性があるため、モータ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 (0x13)

BIT	31	30	29	28	27	26	25	24	
<b>Field</b>	—	—	—	—	—	—	—	—	
<b>Reset</b>	—	—	—	—	—	—	—	—	
<b>Access Type</b>	—	—	—	—	—	—	—	—	
BIT	23	22	21	20	19	18	17	16	
<b>Field</b>	—	—	—	—	TPWMTHRS[19:16]				
<b>Reset</b>	—	—	—	—	0x0				
<b>Access Type</b>	—	—	—	—	Write, Read				

BIT	15	14	13	12	11	10	9	8
<b>Field</b>	TPWMTHRS[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TPWMTHRS[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
TPWMTHRS	19:0	<p>これは、StealthChop2の電圧PWMモードでの上限速度です。  <math>TSTEP \geq TPWMTHRS</math></p> <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>

### TCOOLTHRS (0x14)

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

ビット・フィールド	ピット	説明
TCOOLTHRS	19:0	<p>これは、スマート・エネルギーCoolStepおよびStallGuard機能をスイッチ・オンするための低いスレッショルド速度です。 (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 <code>sg_stop</code> 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イネーブルされます。DcStep以外のモードでは、速度がこのスレッショルド未満になると再びディスエーブルされます。</p> <p><math>TCOOLTHRS \geq TSTEP \geq THIGH</math> :</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><math>TCOOLTHRS \geq TSTEP</math></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) がイネーブルされます (設定されている場合)</li> </ul>

### THIGH (0x15)

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スイッチングを可能にします。（符号なし）</p> <p>ストール検出機能は、THIGHスレッショルドを通過する場合は必ず2~3電気的周期分スイッチ・オフされ、スイッチング・モードの影響を補償します。</p> <p><i>TSTEP ≤ THIGH :</i></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>• 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>• 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>

### RAMP MODE (0x20)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	—	—	RAMPMODE[1:0]	
<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
RAMPMODE	1:0	モーション・コントローラのランピング・モード	<p>0x0: Positioning mode (using all A, D and V parameters)</p> <p>0x1: Velocity mode to positive VMAX (using AMAX acceleration)</p> <p>0x2: Velocity mode to negative VMAX (using AMAX acceleration)</p> <p>0x3: Hold mode (velocity remains unchanged, unless stop event occurs)</p>

**XACTUAL (0x21)**

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	XACTUAL[31:24]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	XACTUAL[23:16]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	XACTUAL[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	XACTUAL[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
XACTUAL	31:0	実際のモータ位置（符号付き） ヒント：駆動のホーミングを行う場合、通常必要なのはこの値が変更されることだけです。ポジショニング・モードでは、このレジスタの内容を変更すると、動作が開始されます。

**VACTUAL (0x22)**

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

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VACTUAL[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
VACTUAL	23:0	<p>ランプ・ジェネレータによる実際のモータ速度（符号付き）</p> <p>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。</p> <p><math>+/- (2^{23}) - 1</math> [μsteps / t]</p>

### VSTART (0x23)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	VSTART[17:16]	
<b>Reset</b>	—	—	—	—	—	—	0x0	
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	VSTART[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	VSTART[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
VSTART	17:0	<p>モータの始動速度（符号なし）</p> <p>一般的な使用では、<math>VSTOP \geq VSTART</math>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。</p> <p><math>0..(2^{18}) - 1</math> [μsteps / t]</p>

## A1 (0x24)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	A1[17:16]	
<b>Reset</b>	—	—	—	—	—	—	0x0	
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	A1[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	A1[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
A1	17:0	VSTARTからV1への最初の加速 (符号なし) 0..(2 <sup>18</sup> ) - 1 [μsteps / ta <sup>2</sup> ]

## V1 (0x25)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	V1[19:16]			
<b>Reset</b>	—	—	—	—	0x0			
<b>Access Type</b>	—	—	—	—	Write, Read			
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	V1[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	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ョルド速度（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 0..(2 <sup>20</sup> ) - 1 [μsteps / t]

### AMAX (0x26)

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

ビット・フィールド	ビット	説明
AMAX	17:0	V1からVMAXへの2回目の加速（符号なし） これは、速度モードの加速度値および減速度値です。 0..(2 <sup>18</sup> ) - 1 [μsteps / ta <sup>2</sup> ]

## VMAX (0x27)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	VMAX[22:16]						
<b>Reset</b>	—	0x0						
<b>Access Type</b>	—	Write, Read						
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	VMAX[15:8]							—
<b>Reset</b>	0x0							—
<b>Access Type</b>	Write, Read							—
BIT	7	6	5	4	3	2	1	0
<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	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 $VMAX \geq VSTART$ となるようにしてください）（符号なし） これは速度モードでの目標速度です。動作中いつでも変更できます。 0..(2 <sup>23</sup> ) - 512 [usteps / t]

## DMAX (0x28)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	DMAX[17:16]	
<b>Reset</b>	—	—	—	—	—	—	0x0	
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	

BIT	15	14	13	12	11	10	9	8
<b>Field</b>	DMAX[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	DMAX[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
DMAX	17:0	VMAXとV1の間の減速（符号なし） 0..(2 <sup>18</sup> ) - 1 [μsteps / ta <sup>2</sup> ]

### TVMAX (0x29)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	TVMAX[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	TVMAX[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	TVMAX[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	TVMAX[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
TVMAX	15:0	<p>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。</p> <p>0 : 一定速度フェーズでの最小時間設定を無効化します &gt;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</p> <p>0..(2<sup>16</sup>) - 1 * 512 t<sub>CLK</sub></p> <p><b>注 :</b> ポジション・モード時と停止時に、VMAXの設定後、このレジスタを設定します。ランプ・モードに戻す場合、TVMAXの遅延がトリガされないよう、速度モード時にはTVMAX = 0に設定します。</p>

### D1 (0x2A)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	-	-	-	-	-	-	-	-
<b>Reset</b>	-	-	-	-	-	-	-	-
<b>Access Type</b>	-	-	-	-	-	-	-	-
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	-	-	-	-	-	-	D1[17:16]	
<b>Reset</b>	-	-	-	-	-	-	0xA	
<b>Access Type</b>	-	-	-	-	-	-	Write, Read	
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	D1[15:8]							
<b>Reset</b>	0xA							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	D1[7:0]							
<b>Reset</b>	0xA							
<b>Access Type</b>	Write, Read							

ビット・フィールド	ビット	説明
D1	17:0	<p>V1からVSTOPへの減速（符号なし）</p> <p><b>注意 :</b> ポジショニング・モードでは、V1 = 0であっても、この値を0には設定しないでください。</p> <p>1..(2<sup>16</sup>) - 1 [μsteps / ta<sup>2</sup>] リセット後のデフォルト = 10</p>

## VSTOP (0x2B)

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

ビット・フィールド	ビット	説明
VSTOP	17:0	<p>モータ停止速度 (符号なし)</p> <p>ヒント : 短距離でのポジショニングを可能にするにはVSTOP <math>\geq</math> VSTARTとなるよう設定します。</p> <p>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。</p> <p>1...(2<sup>18</sup>) - 1 [μsteps / t] リセット後のデフォルト = 10</p>

## TZEROWAIT (0x2C)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—

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

ビット・フィールド	ビット	説明
TZEROWAIT	15:0	速度ゼロへのランプ・ダウン後、次の動作または方向反転を開始できるまでの待機時間を定義します。時間範囲は、約0~2秒です。  この設定により、例えばVSTOPから-VSTARTへといった、過剰な加速を避けることができます。  0..(2^16) - 1 × 512 t <sub>CLK</sub>

### XTARGET (0x2D)

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

ビット・フィールド	ビット	説明
XTARGET	31:0	<p>ランプ・モード用の目標位置（符号付き）<i>RAMPMODE</i> = 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位置を書き込みます。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><b>ヒント：</b> 位置はラップ・アラウンド可能です。そのため、<i>XTARGET</i>の値は、符号なしの数値として処理することもできます。</p> <p><b>ヒント：</b> 可能な最大変位は<math>+/-((2^{31}) - 1)</math>です。</p> <p>ヒント：動作時に<i>V1</i>、<i>D1</i>または<i>DMAX</i>を増加する場合、必要に応じて、その後の<i>XTARGET</i>を書き換え、2回目の加速フェーズをトリガします。</p> <p><math>-2^{31}.. +(2^{31}) - 1</math></p>

## V2 (0x2E)

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	<p>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</p> <p>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</p> <p>0..(2<sup>20</sup>) - 1 [μsteps / t]</p>

## A2 (0x2F)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	A2[17:16]	
<b>Reset</b>	—	—	—	—	—	—	0x0	
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	A2[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	A2[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
A2	17:0	V1からV2への加速 (符号なし) 0..(2^18) - 1 [μsteps / ta^2]

## D2 (0x30)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	D2[17:16]	
<b>Reset</b>	—	—	—	—	—	—	0xA	
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	D2[15:8]							
<b>Reset</b>	0xA							
<b>Access Type</b>	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	D2[7:0]							
Reset	0xA							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
D2	17:0	<p><i>V2からV1への減速 (符号なし)</i></p> <p>注意: 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。</p> <p>1..(2<sup>18</sup>) - 1 [μsteps / t<sub>a</sub><sup>2</sup>] リセット後のデフォルト = 10</p>

### VDCMIN (0x33)

dcStep の開始速度

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

ビット・フィールド	ビット	説明
VDCMIN	22:8	<p>自動整流DcStepは、VDCMINを超える速度でイネーブルされます（符号なし）</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も最小ステップ速度として用いられます。ステップ・ロスを検出するには、ストール時に停止する機能（sg_stop）を活性化します。</p> <p>0 : ディスエーブル、DcStepがオフ  <math> VACT  \geq VDCMIN \geq 256</math> :           <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>ヒント : DcStepを動作させるには、DCCTRLパラメータも設定します。            (値および比較のためにはビット22~8のみが用いられます)</p>
reserved	7:0	常に0を読み出します

## SW\_MODE (0x34)

スイッチ・モード設定

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

ビット・フィールド	ピット	説明	デコード
virtual_stop_enc	14	仮想停止のソース (VIRTUAL_STOP_LおよびVIRTUAL_STOP_R) 0 : 仮想停止はランプ・ジェネレータ位置XACTUALと関連 1 : 仮想停止はエンコーダ位置X_ENCと関連	
en_virtual_stop_r	13	1 : アクティブ・ライトの仮想ストップ条件時に自動モータ停止を有効化	
en_virtual_stop_l	12	1 : アクティブ・レフトの仮想ストップ条件時に自動モータ停止を有効化	
en_softstop	11	0 : ハード・ストップ 1 : ソフト・ストップ ソフト・ストップ・モードでは、常にDMAX、V1、D1、V2、D2、VSTOP、TZEROWAITの減速ランプ設定を用いてモータを停止させます。停止が発生するのは、速度の符号がリファレンス・スイッチ位置（負の速度の場合はREFL および VIRTUAL_STOP_L、正の速度の場合はREFR および VIRTUAL_STOP_R）に合致し、かつ、それぞれのスイッチ停止機能が有効化されている場合です。  ハード・ストップも、モータが解放される前にTZEROWAITを用います。  注意：ソフト・ストップは、StallGuard2と組み合わせて用いないでください。高速度でのStealthChop動作にはソフト・ストップを用いてください。この場合、ハード・ストップは重度の過電流の原因となるため、避ける必要があります。	0x0: Hard stop 0x1: Soft stop
sg_stop	10	StallGuard2による停止をイネーブル (DcStepモードでの使用可)。停止イベント後のモータの解放はできません。速度スレッショルドにはTCOOLTHRをプログラムします。  ヒント：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が安定した結果を供給できる値をモータ速度が超えるまで待ってください。この速度スレッショルドは、TCOOLTHR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	1 : リファレンス切り替えイベント時にエンコーダの位置をENC_LATCHにラッチします。	
latch_r_inactive	8	1 : 右リファレンス・スイッチ入力REFRの非活性化ゴーイング・エッジで、XLATCHへの位置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。	
latch_r_active	7	1 : 右リファレンス・スイッチ入力REFRの活性化ゴーイング・エッジで、XLATCHへの位置のラッチを活性化します。 ヒント：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。	

ビット・フィールド	ビット	説明	デコード
latch_l_inactive	6	1 : 左リファレンス・スイッチ入力REFLの非活性化ゴーイング・エッジで、XLATCHへの位置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。	
latch_l_active	5	1 : 左リファレンス・スイッチ入力REFLの活性化ゴーイング・エッジで、XLATCHへの位置のラッチを活性化します。 ヒント: 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。	
swap_lr	4	1 : 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	
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 0 = 非反転、ハイ・アクティブ : REFRのレベルがハイになるとモータが停止します 1 = 反転、ロー・アクティブ : REFRのレベルがローになるとモータが停止します	
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 0 = 非反転、ハイ・アクティブ : REFLのレベルがハイになるとモータが停止します 1 = 反転、ロー・アクティブ : REFLのレベルがローになるとモータが停止します	
stop_r_enable	1	1 : 右リファレンス・スイッチ入力がアクティブとなっている間の自動モータ停止を有効化します ヒント: 停止スイッチが解放されるとモータは再始動します。	
stop_l_enable	0	1 : 左リファレンス・スイッチ入力がアクティブとなっている間の自動モータ停止を有効化します ヒント: 停止スイッチが解放されるとモータは再始動します。	

### RAMP\_STAT (0x35)

ランプのステータスおよびスイッチ・イベントのステータス

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	status_virtu al_stop_r	status_virtu al_stop_l	status_sg	second_mo ve	t_zerowait_ active	vzero	position_rea ched	velocity_rea ched
<b>Reset</b>	0x1	0x1	0x0	0x0	0x0	0x0	0x0	0x0
<b>Access Type</b>	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_r eached	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)
status_sg	13	1 : CoolStep ドライバまたはDcStepユニットからの入力をアクティブなStallGuard2が受けていることを示す信号 (イネーブルされている場合)。  ヒント: このフラグをポーリングする場合、ストール・イベントが検出されない可能性があります - ストール・イベントを確実に検出できるよう <code>sg_stop</code> 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示す信号 (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 <code>TZEROWAIT</code> がアクティブであることを示す信号。この時間、モータは停止状態です。
vzero	10	1 : 実際の速度が0であることを示す信号。
position_reached	9	1 : 目標位置に達していることを示す信号。 このフラグは、 <code>XACTUAL</code> が <code>XTARGET</code> 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示す信号。 このフラグは、 <code>VACTUAL</code> が <code>VMAX</code> と一致している場合にセットされます。
event_pos_reached	7	1 : 目標位置に達したことを示す信号 ( <code>position_reached</code> 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、 <code>interrupt output</code> 信号とOR接続されます。
event_stop_sg	6	1 : アクティブなStallGuard2停止イベントを示す信号。 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、動作を再開します。 (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、 <code>interrupt output</code> 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ストップにより右停止条件がアクティブ。 停止条件と割込み条件は、 <code>RAMP_MODE</code> 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、取り除くことができます。 <code>soft_stop</code> 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、 <code>interrupt output</code> 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ストップにより左停止条件がアクティブ。 停止条件と割込み条件は、 <code>RAMP_MODE</code> 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、取り除くことができます。 <code>soft_stop</code> 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、 <code>interrupt output</code> 信号とOR接続されます。
status_latch_r	3	1 : 右ラッチ・レディ ( <code>SW_MODE</code> の <code>latch_r_active</code> または <code>latch_r_inactive</code> の設定を用いて位置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)

ピット・フィールド	ピット	説明
status_latch_l	2	1 : 左ラッチ・レディ ( <i>SW_MODE</i> の <i>latch_l_active</i> または <i>latch_l_inactive</i> の設定を用いて位置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 (0x36)

ランプ・ジェネレータのラッチ位置

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

ピット・フィールド	ピット	説明
XLATCH	31:0	ランプ・ジェネレータのラッチ位置。プログラマブルなスイッチ・イベント時に <i>XACTUAL</i> をラッチします ( <i>SW_MODE</i> を参照)。 ヒント: エンコーダ位置は、整合性チェックができるよう、 <i>XLATCH</i> と共に <i>ENC_LATCH</i> にラッチされる場合もあります。

### ENCMODE (0x38)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—

BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	enc_sel_decimal	latch_x_act	clr_enc_x
<b>Reset</b>	—	—	—	—	—	0x0	0x0	0x0
<b>Access Type</b>	—	—	—	—	—	Write, Read	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	pos_neg_edge[1:0]		clr_once	clr_cont	ignore_AB	pol_N	pol_B	pol_A
<b>Reset</b>	0x0			0x0	0x0	0x0	0x0	0x0
<b>Access Type</b>	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
enc_sel_decimal	10	エンコーダ・プリスケーラのモード選択	0x0: Encoder prescaler divisor binary mode: Counts <i>ENC_CONST(fractional part)</i> /65536 0x1: Encoder prescaler divisor decimal mode: Counts in <i>ENC_CONST(fractional part)</i> /10000
latch_x_act	9	位置ラッチ設定	0x0: disabled 0x1: Also latch <i>XACTUAL</i> position together with <i>X_ENC</i> . Allows latching the ramp generator position upon an N channel event as selected by <i>pos_edge</i> and <i>neg_edge</i> .
clr_enc_x	8	エンコーダ・ラッチ設定	0x0: Upon N event, <i>X_ENC</i> becomes latched to <i>ENC_LATCH</i> only 0x1: Latch and additionally clear encoder counter <i>X_ENC</i> 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 going N event 0x2: N channel is valid upon inactive going N event 0x3: N channel is valid upon active going and inactive going N event
clr_once	5	位置ラッチ設定	0x0: disabled 0x1: Latch or latch and clear <i>X_ENC</i> on the next N event following the write access
clr_cont	4	位置ラッチ設定	0x0: disabled 0x1: Always latch or latch and clear <i>X_ENC</i> upon an N event (once per revolution, it is recommended to combine this setting with edge sensitive N event)
ignore_AB	3	Nイベント設定	0x0: An N event occurs only when polarities given by <i>pol_N</i> , <i>pol_A</i> and <i>pol_B</i> 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

ビット・フィールド	ビット	説明	デコード
pol_B	1	Nチャンネル・イベントに必要なBの極性	0x0: neg 0x1: pos
pol_A	0	Nチャンネル・イベントに必要なAの極性	0x0: neg 0x1: pos

**X\_ENC (0x39)**

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

ビット・フィールド	ビット	説明
X_ENC	31:0	実際のエンコーダ位置 (符号付き)

**ENC\_CONST (0x3A)**

BIT	31	30	29	28	27	26	25	24
Field	ENC_CONST[31:24]							
Reset	0x10000							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	ENC_CONST[23:16]							
Reset	0x10000							
Access Type	Write, Read							

BIT	15	14	13	12	11	10	9	8
<b>Field</b>	ENC_CONST[15:8]							
<b>Reset</b>	0x10000							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ENC_CONST[7:0]							
<b>Reset</b>	0x10000							
<b>Access Type</b>	Write, Read							

ピット・フィールド	ピット	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><math>X\_ENC</math>で加算される量は以下のとおりです  <math>+/-ENC\_CONST / (2^{16} \times X\_ENC)</math> (2進数)          または  <math>+/-ENC\_CONST / (10^4 \times X\_ENC)</math> (10進数)</p> <p><i>ENCMODE</i>の<math>enc\_sel\_decimal</math>ビットにより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：  <math>\pm [\mu\text{steps}/2^{16}]</math>  <math>\pm(0 \dots 32767.999847)</math>          10進数：  <math>\pm(0.0 \dots 32767.9999)</math>          リセット後のデフォルト = 1.0 (= 65536)</p>

## ENC\_STATUS (0x3B)

エンコーダのステータス情報

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	—	—	—	—	—	—	deviation_warn	n_event
<b>Reset</b>	—	—	—	—	—	—	0x0	0x0
<b>Access Type</b>	—	—	—	—	—	—	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		0x0: no warning 0x1: deviation_warn cannot be cleared while a warning still persists. Set <i>ENC_DEVIATION</i> to zero to disable.
n_event	0		0x0: no event 0x1: Event detected. To clear the status bit, write with a 1 bit at the corresponding position.

### ENC\_LATCH (0x3C)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	ENC_LATCH[31:24]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Read Only							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	ENC_LATCH[23:16]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Read Only							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	ENC_LATCH[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Read Only							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	ENC_LATCH[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
ENC_LATCH	31:0	Nイベント時にラッチされたエンコーダ位置X_ENC

## ENC\_DEVIATION (0x3D)

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

ピット・フィールド	ピット	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 これにより、ENC_STATUS.deviation_warnフラグがセットされます 0 = この機能をオフ。

## VIRTUAL\_STOP\_L (0x3E)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	VIRTUAL_STOP_L[31:24]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	VIRTUAL_STOP_L[23:16]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	VIRTUAL_STOP_L[15:8]							
<b>Reset</b>	0x0							
<b>Access Type</b>	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	エンコーダまたはランプの位置に基づく仮想停止スイッチ。停止は、符号を含めた比較に基づき行われます $virtual\_stop\_enc = 1$ : $X\_ENC \leq VIRTUAL\_STOP\_L$ $virtual\_stop\_enc = 0$ : $XACTUAL \leq VIRTUAL\_STOP\_L$ $-2^{31} \dots$ $+(2^{31}) - 1$

### VIRTUAL\_STOP\_R (0x3F)

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	エンコーダに基づく仮想停止スイッチ。停止は、符号を含めた比較に基づき行われます $virtual\_stop\_enc = 1$ : $X\_ENC \geq VIRTUAL\_STOP\_R$ $virtual\_stop\_enc = 0$ : $XACTUAL \geq VIRTUAL\_STOP\_R$ $-2^{31} \dots$ $+(2^{31}) - 1$

## ADC\_VSUPPLY\_AIN (0x50)

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

ピット・フィールド	ピット	説明
ADC_AIN	28:16	AINピンの電圧値（整数）。 更新レート = 2048クロックごと $V_{AIN} = ADC\_AIN * 305.2\mu V$
ADC_VSUPPLY	12:0	$V_s$ の電圧の実際の値（ローパス・フィルタでフィルタリング） 更新レート : 2048クロックごと $V_s = ADC\_VSUPPLY * 9.732mV$

## ADC\_TEMP (0x51)

BIT	31	30	29	28	27	26	25	24
Field	—	—	—	RESERVED[12:8]				
Reset	—	—	—					
Access Type	—	—	—	Read Only				
BIT	23	22	21	20	19	18	17	16
Field	RESERVED[7:0]							
Reset								
Access Type	Read Only							

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

ビット・フィールド	ビット	説明
RESERVED	28:16	
ADC_TEMP	12:0	実際の温度（ローパス・フィルタでフィルタリング） 更新レート：2048クロックごと。 TEMP[ °C] = $\frac{\text{ADC\_TEMP} - 2038}{7.7}$

### OTW\_OV\_VTH (0x52)

BIT	31	30	29	28	27	26	25	24	
Field	—	—	—	OVERTEMPPREWARNING_VTH[12:8]					
Reset	—	—	—	0xB92					
Access Type	—	—	—	Write, Read					
BIT	23	22	21	20	19	18	17	16	
Field	OVERTEMPPREWARNING_VTH[7:0]								
Reset	0xB92								
Access Type	Write, Read								
BIT	15	14	13	12	11	10	9	8	
Field	—	—	—	OVERVOLTAGE_VTH[12:8]					
Reset	—	—	—	0xF25					
Access Type	—	—	—	Write, Read					
BIT	7	6	5	4	3	2	1	0	
Field	OVERVOLTAGE_VTH[7:0]								
Reset	0xF25								
Access Type	Write, Read								

ビット・フィールド	ビット	説明
OVERTEMPPREWARNING_VTH	28:16	過熱警告スレッショルド・レジスタ： ADC_TEMP >= OVERTEMPPREWARNING_VTH 過熱警告がトリガされます (リセット：0xB92が120°Cに相当します)

ピット・フィールド	ピット	説明
OVERVOLTAGE_VTH	12:0	出力OV用の過電圧スレッショルド。デフォルト：38V、36VはADC入力の1.125Vに相当します。

### MSLUT\_0 (0x60)

マイクロステップ・テーブルの項目 0～31

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_0[31:24]							
Reset	0xAAAAAB554							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_0[23:16]							
Reset	0xAAAAAB554							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_0[15:8]							
Reset	0xAAAAAB554							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_0[7:0]							
Reset	0xAAAAAB554							
Access Type	Write, Read							

ピット・フィールド	ピット	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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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_S/NとSTART_S/N90のMSCNTの位置0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00  ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト = サイン波テーブル</p>

### MSLUT\_1 (0x61)

マイクロステップ・テーブルの項目 32～63

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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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の位置0に対して保存されています。</p> <p>ofs31, ofs30, ..., ofs01, ofs00  ...</p> <p>ofs255, ofs254, ..., ofs225, ofs224</p> <p>リセット後のデフォルト = サイン波テーブル</p>

## MSLUT\_2 (0x62)

マイクロステップ・テーブルの項目 64~95

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

BIT	23	22	21	20	19	18	17	16
<b>Field</b>	MSLUT_2[23:16]							
<b>Reset</b>	0x24492929							
<b>Access Type</b>	Write, Read							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	MSLUT_2[15:8]							
<b>Reset</b>	0x24492929							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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の位置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\_3 (0x63)

マイクロステップ・テーブルの項目 96～127

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	MSLUT_3[31:24]							
<b>Reset</b>	0x10104222							
<b>Access Type</b>	Write, Read							
BIT	23	22	21	20	19	18	17	16
<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
<b>Field</b>	MSLUT_3[15:8]							
<b>Reset</b>	0x10104222							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MSLUT_3[7:0]							
<b>Reset</b>	0x10104222							
<b>Access Type</b>	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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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の位置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\_4 (0x64)

マイクロステップ・テーブルの項目 128~159

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

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MSLUT_4[7:0]							
<b>Reset</b>	0xFBFFFFFF							
<b>Access Type</b>	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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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の位置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\_5 (0x65)

マイクロステップ・テーブルの項目 160～191

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

ビット・フィールド	ピット	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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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の位置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\_6 (0x66)

マイクロステップ・テーブルの項目 192~223

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	MSLUT_6[31:24]							
<b>Reset</b>	0x49295556							
<b>Access Type</b>	Write, Read							
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	MSLUT_6[23:16]							
<b>Reset</b>	0x49295556							
<b>Access Type</b>	Write, Read							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	MSLUT_6[15:8]							
<b>Reset</b>	0x49295556							
<b>Access Type</b>	Write, Read							
BIT	7	6	5	4	3	2	1	0
<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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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の位置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\_7 (0x67)

マイクロステップ・テーブルの項目 224～255

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_7[31:24]							
Reset	0x404222							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_7[23:16]							
Reset	0x404222							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_7[15:8]							
Reset	0x404222							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_7[7:0]							
Reset	0x404222							
Access Type	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項目xと項目x+1の差を示します。</p> <p>0: <math>W = \%00: -1</math>  <math>\%01: +0</math>  <math>\%10: +1</math>  <math>\%11: +2</math></p> <p>1: <math>W = \%00: +0</math>  <math>\%01: +1</math>  <math>\%10: +2</math>  <math>\%11: +3</math></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の位置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>

### MSLUTSEL (0x68)

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	0x1		0x1		0x1		0x2	
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1の開始</p> <p>サイン波のルックアップ・テーブルは、個々のステップ幅制御項目 <math>Wx</math> 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、<math>X1</math>、<math>X2</math>、<math>X3</math>で選択できます。</p> <p>セグメント0は0～<math>X1-1</math>の範囲です。      セグメント1は<math>X1</math>～<math>X2-1</math>の範囲です。      セグメント2は<math>X2</math>～<math>X3-1</math>の範囲です。      セグメント3は<math>X3</math>～255の範囲です。</p> <p>定義さ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。  <math>0 &lt; X1 &lt; X2 &lt; X3</math></p>
X2	23:16	<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、個々のステップ幅制御項目 <math>Wx</math> 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、<math>X1</math>、<math>X2</math>、<math>X3</math>で選択できます。</p> <p>セグメント0は0～<math>X1-1</math>の範囲です。      セグメント1は<math>X1</math>～<math>X2-1</math>の範囲です。      セグメント2は<math>X2</math>～<math>X3-1</math>の範囲です。      セグメント3は<math>X3</math>～255の範囲です。</p> <p>定義さ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。  <math>0 &lt; X1 &lt; X2 &lt; X3</math></p>
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、個々のステップ幅制御項目 <math>Wx</math> 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、<math>X1</math>、<math>X2</math>、<math>X3</math>で選択できます。</p> <p>セグメント0は0～<math>X1-1</math>の範囲です。      セグメント1は<math>X1</math>～<math>X2-1</math>の範囲です。      セグメント2は<math>X2</math>～<math>X3-1</math>の範囲です。      セグメント3は<math>X3</math>～255の範囲です。</p> <p>定義さ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。  <math>0 &lt; X1 &lt; X2 &lt; X3</math></p>
W3	7:6	<p><math>ofs(X3) \sim ofs(255)</math>でのLUT幅の選択</p> <p>幅制御ビットのコーディング <math>W0 \sim W3</math> :</p> <ul style="list-style-type: none"> <li>%00 : MSLUTの項目が0、1の場合の選択 : -1、+0</li> <li>%01 : MSLUTの項目が0、1の場合の選択 : +0、+1</li> <li>%10 : MSLUTの項目が0、1の場合の選択 : +1、+2</li> <li>%11 : MSLUTの項目が0、1の場合の選択 : +2、+3</li> </ul>
W2	5:4	<p><math>ofs(X2) \sim ofs(X3-1)</math>でのLUT幅の選択</p> <p>幅制御ビットのコーディング <math>W0 \sim W3</math> :</p> <ul style="list-style-type: none"> <li>%00 : MSLUTの項目が0、1の場合の選択 : -1、+0</li> <li>%01 : MSLUTの項目が0、1の場合の選択 : +0、+1</li> <li>%10 : MSLUTの項目が0、1の場合の選択 : +1、+2</li> <li>%11 : MSLUTの項目が0、1の場合の選択 : +2、+3</li> </ul>

ビット・フィールド	ビット	説明
W1	3:2	$ofs(X1) \sim ofs(X2-1)$ でのLUT幅の選択 幅制御ビットのコーディング $W0 \sim W3$ ： %00 : MSLUTの項目が0、1の場合の選択 : -1, +0 %01 : MSLUTの項目が0、1の場合の選択 : +0, +1 %10 : MSLUTの項目が0、1の場合の選択 : +1, +2 %11 : MSLUTの項目が0、1の場合の選択 : +2, +3
W0	1:0	$ofs00 \sim ofs(X1-1)$ でのLUT幅の選択 Width control bit coding $W0 \dots W3$ ： %00: MSLUTの項目が0、1の場合の選択 : -1, +0 %01: MSLUTの項目が0、1の場合の選択 : +0, +1 %10: MSLUTの項目が0、1の場合の選択 : +1, +2 %11: MSLUTの項目が0、1の場合の選択 : +2, +3

### MSLUTSTART (0x69)

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

BIT	31	30	29	28	27	26	25	24
Field	OFFSET_SIN90[7:0]							
Reset	0x0							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	START_SIN90[7:0]							
Reset	0d247							
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	0x0							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
OFFSET_SIN90	31:24	コサイン波に対する符号付きオフセット (+/-127マイクロステップ)。 $MSCNT = 0$ の位置でマイクロステップ波形テーブルに適合するよう、 $START_SIN90$ を修正します。
START_SIN90	23:16	$START_SIN90$ は、 $MSCNT = 0$ (テーブル位置 $256 + OFFSET_SIN90$ ) の位置でのコサイン波マイクロステップ・テーブル項目に絶対値を与えます。
START_SIN	7:0	$START_SIN$ は、マイクロステップ・テーブル項目0に絶対値を与えます。

## MSCNT (0x6A)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	MSCNT[9:8]	
<b>Reset</b>	—	—	—	—	—	—	0x0	
<b>Access Type</b>	—	—	—	—	—	—	Read Only	
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	MSCNT[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
MSCNT	9:0	マイクロステップ・カウンタ。実際の位置をCUR_Aのマイクロステップ・テーブルに示します。CUR_Bは256のオフセットを使用します（2相モータ）。 ヒント：MSLUTSTARTまたはMSLUTおよびMSLUTSELを再初期化する前に、MSCNTがゼロとなる位置に移動させます。

## MSCURACT (0x6B)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	CUR_A[8]
<b>Reset</b>	—	—	—	—	—	—	—	0xF7
<b>Access Type</b>	—	—	—	—	—	—	—	Read Only
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	CUR_A[7:0]							
<b>Reset</b>	0xF7							
<b>Access Type</b>	Read Only							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	—	CUR_B[8]
<b>Reset</b>	—	—	—	—	—	—	—	0x0
<b>Access Type</b>	—	—	—	—	—	—	—	Read Only

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

ビット・フィールド	ビット	説明
CUR_A	24:16	MSLUTから読み出した、モータ位相A（コサイン波）の実際のマイクロステップ電流（電流によるスケーリングなし）。
CUR_B	8:0	MSLUTから読み出した、モータ位相B（サイン波）の実際のマイクロステップ電流（電流によるスケーリングなし）。

### CHOPCONF (0x6C)

BIT	31	30	29	28	27	26	25	24
Field	diss2vs	diss2g	reserved	interp	MRES[3:0]			
Reset	0x0	0x0	0x0	0x1	0x0			
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read			
BIT	23	22	21	20	19	18	17	16
Field	TPFD[3:0]				vhighchm	vhighfs	-	TBL[1]
Reset	0x4				-	-	-	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	-	0x2		
Access Type	Write, Read	Write, Read	-	Write, Read	Write, Read	Write, Read		
BIT	7	6	5	4	3	2	1	0
Field	HEND_OFFSET[0]	HSTRT_TFD210[2:0]			TOFF[3:0]			
Reset	0x2	0x5			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 VS protection is on 0x1: Short to VS 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
reserved	29	予約済み、使用しないでください	
interp	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

ビット・フィールド	ピット	説明	デコード
MRES	27:24	<p>マイクロステップの分解能の選択</p> <p>%0000 : 元来の256マイクロステップ設定。通常内部モーション・コントローラではこの設定を使用します。</p> <p>%0001～%1000 : 128、64、32、16、8、4、2、FULLSTEP 低下したマイクロステップ分解能。 この分解能は、1/4 サイン波あたりのマイクロステップ項目の数を示します。 低下した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 = <math>2^{MRES}</math> [microsteps]</p>	
TPFD	23:20	<p>受動高速減衰時間</p> <p>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。</p> <p>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。</p> <p><math>N_{CLK} = 128 \times TPFD</math></p> <p>%0000 : 無効 %0001 ... %1111: 1 ... 15</p>	
vhighchm	19	<p>高速チョッパー・モード</p> <p>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。</p>	
vhighfs	18	<p>高速フルステップ選択</p> <p>このビットは、VHIGHを超えた場合にフルステップへの切り替わりをイネーブルします。切り替わりは45°の位置でのみ生じます。フルステップの目標電流は、45°位置でのマイクロステップ・テーブルからの電流値を用います。</p>	
TBL	16:15	<p>TBLブランкиング時間設定。</p> <p>コンパレータのブランкиング時間をクロック・サイクル数単位で設定します。</p> <p>ヒント：ほとんどのアプリケーションには24クロックまたは36クロックを推奨します。</p> <p>TBL = 0x0の場合の制限：8MHz以下の外部クロック発振器を組み合わせる場合のみ使用できます</p> <p>TBL = 0x1の場合の制限：内部クロックまたは13MHz以下の外部クロックを適用する場合に使用できます。</p>	0x0: 16 clocks 0x1: 24 clocks 0x2: 36 clocks 0x3: 48 clocks

ビット・フィールド	ピット	説明	デコード
chm	14	チョッパー・モードの選択。 これが有効になるのは、 <i>en_pwm_mode</i> が0に設定されている場合か <i>TSTEP</i> < <i>TPWMTHRS</i> の場合です。	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	chm = 1の場合の高速減衰モード	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 = ヒステリシスの低い値 %0000~%1111： ヒステリシスは-3、-2、-1、0、1、…、12 (この設定値の1/512が電流設定値に加算されます) これはヒステリシス・チョッパー用に用いられるヒステリシス値です。  chm = 1の場合： OFFSET = サイン波のオフセット %0000~%1111： オフセットは-3、-2、-1、0、1、…、12 これは、サイン波のオフセットで、この値の1/512が各サイン波項目の絶対値に加算されます。	
HSTRRT_TFD210	6:4	<b>chm = 0の場合：</b> HENDに加算されるヒステリシス開始値 HSTRRT  %000~%111： ヒステリシスの低い値HENDに1、2、…、8が加算されます (この設定値の1/512が電流設定値に加算されます)  注意：実効HEND + HSTRRT ≤ 16 ヒント：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  <b>chm = 1の場合：</b> 高速減衰時間設定値TFD [2..0] 高速減衰時間設定値 (MSB : fd3)： %0000~%1111： N <sub>CLK</sub> = 32 × TFDの高速減衰時間設定値TFD (%0000 : 低速減衰のみ)	
TOFF	3:0	オフ時間TOFFおよびドライバのイネーブル  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	

## COOLCONF (0x6D)

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

ビット・フィールド	ビット	説明	デコード
sfilt	24	StallGuard2およびStallGuard4のフィルタをイネーブル	0x0: Standard mode, high time resolution for StallGuard 0x1: Filtered mode, StallGuard signal updated for each four fullsteps only to compensate for motor pole tolerances
sgt	22:16	StallGuard2のスレッショルド値  この符号付きの値は、ストール出力用の StallGuard2 レベルを制御し、読み出し用に最適な測定範囲を設定します。値を小さくすると感度が高くなります。ゼロはほとんどのモータで使用できる開始値です。 -64～+63 : 値を大きくすると StallGuard2 の感度は低下し、ストールを示すために必要なトルクが大きくなります。	
seimin	15	スマート電流制御用の最小電流	0x0: 1/2 of current setting (I/RUN) 0x1: 1/4 of current setting (I/RUN)
sedn	14:13	電流のダウン・ステップ速度  %00 : 32個のStallGuard2値ごとに1ステップずつ減少します。 %01 : 8個のStallGuard2値ごとに1ステップずつ減少します。 %10 : 2個のStallGuard2値ごとに1ステップずつ減少します。 %11 : 1個のStallGuard2値ごとに1ステップずつ減少します。	

ビット・フィールド	ビット	説明	デコード
semax	11:8	スマート電流制御用のStallGuard2ヒステリシス値  StallGuard2の結果が $(SEMIN + SEMAX + 1) \times 32$ 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測定値ごとの電流インクリメント・ステップ %00 ... %11: 1, 2, 4, 8	
semin	3:0	スマート電流制御用の最小StallGuard2値およびスマート電流イネーブル  StallGuard2の結果が $SEMIN \times 32$ 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 (0x6E)

DcStep (DC) 自動整流設定レジスタ (DCEN ピンまたは VDCMIN でイネーブル)

ヒント: 高マイクロステップ分解能または補間操作を用いると、DcStep の供給する StallGuard 信号が向上します。vhighfs が活性化されている場合、VHIGH を超えている場合でも DC\_SG を使用できます。最高の結果を得るには、vhighchm もセットします。

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	DC_SG[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	DC_TIME[9:8]	
<b>Reset</b>	—	—	—	—	—	—	0x0	
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	
BIT	7	6	5	4	3	2	1	0
<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( $DC\_SG * 16/f_{CLK}$ )  $DC\_TIME/16$ 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( $DC\_TIME * 1/f_{CLK}$ )。実効ブランкиング時間TBLよりわずかに大きい値に設定します。

### DRV\_STATUS (0x6F)

BIT	31	30	29	28	27	26	25	24
Field	stst	olb	ola	s2gb	s2ga	otpw	ot	stallguard
Reset								
Access Type	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only	Read Only
BIT	23	22	21	20	19	18	17	16
Field	—	—	—	CS_ACTUAL[4:0]				
Reset	—	—	—					
Access Type	—	—	—	Read Only				
BIT	15	14	13	12	11	10	9	8
Field	fsactive	stealth	s2vsb	s2vsd	—	—	SG_RESULT[9:8]	
Reset					—	—		
Access Type	Read Only	Read Only	Read Only	Read Only	—	—	Read Only	
BIT	7	6	5	4	3	2	1	0
Field	SG_RESULT[7:0]							
Reset								
Access Type	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・パルスから $2^{20}$ クロック後に生じます。	
olb	30	位相Bの開放負荷インジケータ  Hint: This is just an informative flag. The driver takes no action upon it. False detection may occur in fast motion and standstill. Check during slow motion, only.	0x0: normal operation 0x1: Open load detected on phase B. Hint: This is just an informative flag. The driver takes no action upon 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  Hint: This is just an informative flag. The driver takes no action upon it. False detection may occur in fast motion and standstill. Check during slow motion, only.	0x0: normal operation 0x1: Open load detected on phase A. Hint: This is just an informative flag. The driver takes no action upon it. False detection may occur in fast motion and standstill. Check during slow motion, only.

ピット・フィールド	ピット	説明	デコード
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 by the DRV_ENN input.
s2ga	27	位相Aのグランドへの短絡インジケータ	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 by the DRV_ENN input.
otpw	26	過熱事前警告フラグ	0x0: normal operation 0x1: Overtemperature pre-warning threshold is exceeded. The overtemperature pre-warning 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 <i>otpw</i> is also cleared due to cooling down of the IC. The overtemperature flag is common for both bridges.
stallguard	24	StallGuard2/StallGuard4のステータス	0x0: normal operation 0x1: Motor stall detected by StallGuard2 (in SpreadCycle operation) resp. by StallGuard4 (in StealthChop2 operation) 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 実際の電流制御スケーリング、COOLCONFレジスタの設定により制御されるスマート・エネルギー電流スケーリングのモニタリング用、または自動電流スケーリング機能のモニタリング用	
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 fullstep as defined by chopper mode settings and velocity thresholds
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 by the ENN input.
s2vsb	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 by the ENN input.

ビット・フィールド	ビット	説明	デコード
SG_RESULT	9:0	<p>StallGuard2の結果またはStallGuard4の結果（実際のチョッパー・モードに依存）、または、SpreadCycleを使用しモータ温度検出のために停止しているコイルAのPWMオン時間</p> <p>機械的負荷測定：</p> <p>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。StallGuard2では、値が0の場合負荷が最大であることを示します。最適なSGT設定値を用いた場合、これはモータ・ストールのインジケータとなります。ストール検出では、SG_RESULTと0を比較してストールの検出を行います。SG_RESULTは、これをプログラマブルな上限値および下限値と比較することで、CoolStep動作の基盤として使用されます。StealthChop2モードでは使用できません。</p> <p>StallGuard2は、マイクロステップ動作またはDcStepで最もよく機能します。</p> <p>SpreadCycleモード時の温度測定：</p> <p>停止時には、StallGuard2の結果は取得できません。代わりに、SG_RESULTは、モータ・コイルAのチョッパー・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位置に特定の電流設定値で動作させると、チョッパー・オン時間を読み出すことでモータ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、チョッパー・オン時間も増加します。StallGuard4 の詳細については、<a href="#">SG4_RESULT</a>を参照してください。</p>	

### PWMCONF (0x70)

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	PWM_LIM[3:0]						PWM_REG[3:0]	
<b>Reset</b>	0xC						0x4	
<b>Access Type</b>	Write, Read						Write, Read	
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	pwm_dis_re_g_stst	pwm_meas_sd_enable	FREEWHEEL[1:0]		pwm_autograd	pwm_autoscale	PWM_FREQ[1:0]	
<b>Reset</b>	0x0	0x0	0x0		0x1	0x1	0x0	
<b>Access Type</b>	Write, Read	Write, Read	Write, Read		Write, Read	Write, Read	Write, Read	
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	PWM_GRAD[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Write, Read							

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PWM_OFS[7:0]							
<b>Reset</b>	0x1D							
<b>Access Type</b>	Write, Read							

ビット・フィールド	ビット	説明	デコード
PWM_LIM	31:28	<p>スイッチ・オン時のPWM自動スケール振幅制限</p> <p>SpreadCycleからStealthChop2にスイッチ・バックする場合のPWM_SCALE_AUTOの制限値この値は、スイッチ・バック時の自動電流制御のビット7~4に対する上限値を定義します。StealthChop2へのモード切り替え時の電流ジャーカーを低減するよう設定できます。</p> <p>PWM_GRADまたはPWM_GRAD_AUTOオフセットは制限しません。</p> <p>(デフォルト = 12)</p>	
PWM_REG	27:24	<p>レギュレーション・ループの勾配</p> <p><i>pwm_autoscale</i> = 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振幅変化です。 (1~15) :</p> <p>1 : 0.5インクリメント (最遅レギュレーション)</p> <p>2 : 1インクリメント</p> <p>3 : 1.5インクリメント</p> <p>4 : 2インクリメント (リセット後のデフォルト)</p> <p>...</p> <p>8 : 4インクリメント</p> <p>...</p> <p>15 : 7.5インクリメント (最速レギュレーション)</p>	
pwm_dis_reg_stst	23	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。このオプションは、停止時のレギュレーション・ノイズを抑制します。	
pwm_meas_sd_enable	22	デフォルト = 0 ; 1 : モータ電流を測定するためにロー・サイドで低速減衰フェーズを用いて、電流下限値を低減します。	
FREEWHEEL	21:20	<p>様々な停止モードを可能にします</p> <p>モータ電流設定値がゼロ (<i>I_HOLD</i> = 0) の場合の停止オプション。</p> <p>%00 : 通常動作</p> <p>%01 : フリーホイーリング</p> <p>%10 : LS ドライバを用いたコイル短絡</p> <p>%11 : HS ドライバを用いたコイル短絡</p>	

ビット・フィールド	ピット	説明	デコード
pwm_autograd	19	PWM自動勾配調整	<p>0x0: Fixed value for <math>PWM\_GRAD</math> (<math>PWM\_GRAD\_AUTO = PWM\_GRAD</math>)</p> <p>0x1: Automatic tuning (only with <math>pwm\_autoscale = 1</math>) (<i>Reset default</i>) <math>PWM\_GRAD\_AUTO</math> is initialized with <math>PWM\_GRAD</math> while <math>pwm\_autograd=0</math> and becomes optimized automatically during motion.</p> <p><u>Preconditions</u></p> <ol style="list-style-type: none"> <li>1. <math>PWM\_OFS\_AUTO</math> has been automatically initialized. This requires standstill at <math>IRUN</math> for &gt;130ms in order to a) detect standstill b) wait &gt; 128 chopper cycles at <math>IRUN</math> and c) regulate <math>PWM\_OFS\_AUTO</math> so that <math>-1 &lt; PWM\_SCALE\_AUTO &lt; 1</math></li> <li>2. Motor running and <math>1.5 \times PWM\_OFS\_AUTO \times (IRUN+1)/32 &lt; PWM\_SCALE\_SUM &lt; 4 \times PWM\_OFS\_AUTO \times (IRUN+1)/32</math> and <math>PWM\_SCALE\_SUM &lt; 255</math>.</li> </ol> <p><u>Time required for tuning <math>PWM\_GRAD\_AUTO</math></u> About 8 fullsteps per change of <math>\pm 1</math>. Also enables use of reduced chopper frequency for tuning <math>PWM\_OFS\_AUTO</math>.</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 <math>IRUN</math> and <math>IHOLD</math> have no influence! The resulting PWM amplitude (limited to 0...255) is: <math>PWM\_OFS * ((CS\_ACTUAL+1) / 32) + PWM\_GRAD * 256 / TSTEP</math></p> <p>0x1: Enable automatic current control (<i>Reset default</i>)</p>
PWM_FREQ	17:16	PWM周波数選択： %00 : $f_{PWM} = 2/1024 f_{CLK}$ (リセット後のデフォルト) %01 : $f_{PWM} = 2/683 f_{CLK}$ %10 : $f_{PWM} = 2/512 f_{CLK}$ %11 : $f_{PWM} = 2/410 f_{CLK}$	
PWM_GRAD	15:8	<p>PWM振幅の速度依存勾配： <math>PWM\_GRAD \times 256 / TSTEP</math></p> <p>この値は、速度に依存したモータ・バックEMFを補償するために <math>PWM\_OFS</math> に加算されます。</p> <p>自動スケーリングの初期値として <math>PWM\_GRAD</math> を用いること、自動調整プロセスを加速できます。これを行うには、<math>pwm\_autoscale = 0</math> とし、<math>PWM\_GRAD</math> 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、<math>pwm\_autoscale = 1</math> に設定します。終了後は StealthChop2 をイネーブルします。</p> <p>ヒント： 初期調整の後、必要な初期値は <math>PWM\_GRAD\_AUTO</math> で読み出せます。</p>	

ビット・フィールド	ピット	説明	デコード
PWM_OFS	7:0	<p>停止時のモータのフル電流 (<math>CS\_ACTUAL = 31</math>) に関するユーザ定義によるPWM振幅オフセット (0~255)。 (リセット後のデフォルト = 30)</p> <p>自動スケーリングの初期値として <math>PWM\_OFS</math> 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、<math>pwm\_autoscale = 0</math> とし、<math>PWM\_OFS</math> 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、<math>pwm\_autoscale = 1</math> に設定します。終了後は StealthChop2 をイネーブルします。</p> <p><math>PWM\_OFS = 0</math> 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ーリング・ダウンできなくなります。この設定は特定の条件下でのみ用いてください。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です。これにより、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ます。</p> <p><math>PWM\_OFS &gt; 0</math> 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 (<math>IHOLD\_IRUN</math> 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 (0x71)

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

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	$PWM\_SCALE\_AUTO[8]$
<b>Reset</b>	—	—	—	—	—	—	—	0x0
<b>Access Type</b>	—	—	—	—	—	—	—	Read Only
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	PWM_SCALE_AUTO[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Read Only							
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	$PWM\_SCALE\_SUM[9:8]$	
<b>Reset</b>	—	—	—	—	—	—	0x0	
<b>Access Type</b>	—	—	—	—	—	—	Read Only	

BIT	7	6	5	4	3	2	1	0
<b>Field</b>	PWM_SCALE_SUM[7:0]							
<b>Reset</b>	0x0							
<b>Access Type</b>	Read Only							

ピット・フィールド	ピット	説明
PWM_SCALE_AUTO	24:16	
PWM_SCALE_SUM	9:0	ピット: 9~0 : [0~1023] PWM_SCALE_SUM : 実際のPWMデューティ・サイクル。この値は、サイン波テーブルから読み出されるCUR_AとCUR_Bの値をスケーリングするために用いられます。1023 : 最大デューティ・サイクル。この値は、より高精度のデューティ・サイクル読出しのため、2ビット[1,0]に拡張できます。ビット9~2は、他のPWMデューティ・サイクル関連レジスタの8ビット値に対応します。

### PWM\_AUTO (0x72)

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

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

ピット・フィールド	ピット	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 (0x74)**

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
<b>Field</b>	—	—	—	—	—	—	sg_angle_offset	sg4_filt_en
<b>Reset</b>	—	—	—	—	—	—	0x1	0x0
<b>Access Type</b>	—	—	—	—	—	—	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
<b>Field</b>	SG4_THRS[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
sg_angle_offset	9	1 : TPWMTHRSを介した制御によりStealthChop2からSpreadCycleにスイッチングする際の、StallGuard4に基づく自動位相シフト補償
sg4_filt_en	8	1 : SG4フィルタをイネーブル、0 : SG4フィルタをディスエーブル
SG4_THRS	7:0	ストールの検出スレッショルド。StallGuard4値SG4_RESULTがこのスレッショルドと比較されます。 ストールが示されるのは以下の場合です $SG4\_RESULT \leq SG4\_THRS$  SG4_THRSは可能なSG4_RESULTの範囲の半分を占めます

**SG4\_RESULT (0x75)**

BIT	31	30	29	28	27	26	25	24
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
<b>Field</b>	—	—	—	—	—	—	—	—
<b>Reset</b>	—	—	—	—	—	—	—	—
<b>Access Type</b>	—	—	—	—	—	—	—	—

BIT	15	14	13	12	11	10	9	8
Field	—	—	—	—	—	—	SG4_RESULT[9:8]	
Reset	—	—	—	—	—	—	0x0	
Access Type	—	—	—	—	—	—	Read Only	
BIT	7	6	5	4	3	2	1	0
Field	SG4_RESULT[7:0]							
Reset	0x0							
Access Type	Read Only							

ビット・フィールド	ビット	説明
SG4_RESULT	9:0	<p>StallGuard4専用のStallGuard結果。</p> <p>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。</p> <p>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。</p> <p>10ビットにスケー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。</p>

### SG4\_IND (0x76)

BIT	31	30	29	28	27	26	25	24
Field	SG4_IND_3[7:0]							
Reset	0x0							
Access Type	Read Only							
BIT	23	22	21	20	19	18	17	16
Field	SG4_IND_2[7:0]							
Reset	0x0							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	SG4_IND_1[7:0]							
Reset	0x0							
Access Type	Read Only							
BIT	7	6	5	4	3	2	1	0
Field	SG4_IND_0[7:0]							
Reset	0x0							
Access Type	Read Only							

ビット・フィールド	ビット	説明
SG4_IND_3	31:24	<p>SG4_filt_en = 1の場合 :</p> <p>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。</p>
SG4_IND_2	23:16	<p>SG4_filt_en = 1の場合 :</p> <p>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。</p>

ピット・フィールド	ピット	説明
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。

## 標準アプリケーション回路

### 標準的なアプリケーション回路

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

すべてのフィルタ・コンデンサは、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、べた膜の共通グランド・プレーンを用います。V<sub>DDIV8</sub> フィルタリング・コンデンサは、V<sub>DDIV8</sub> ピンに直接接続します。

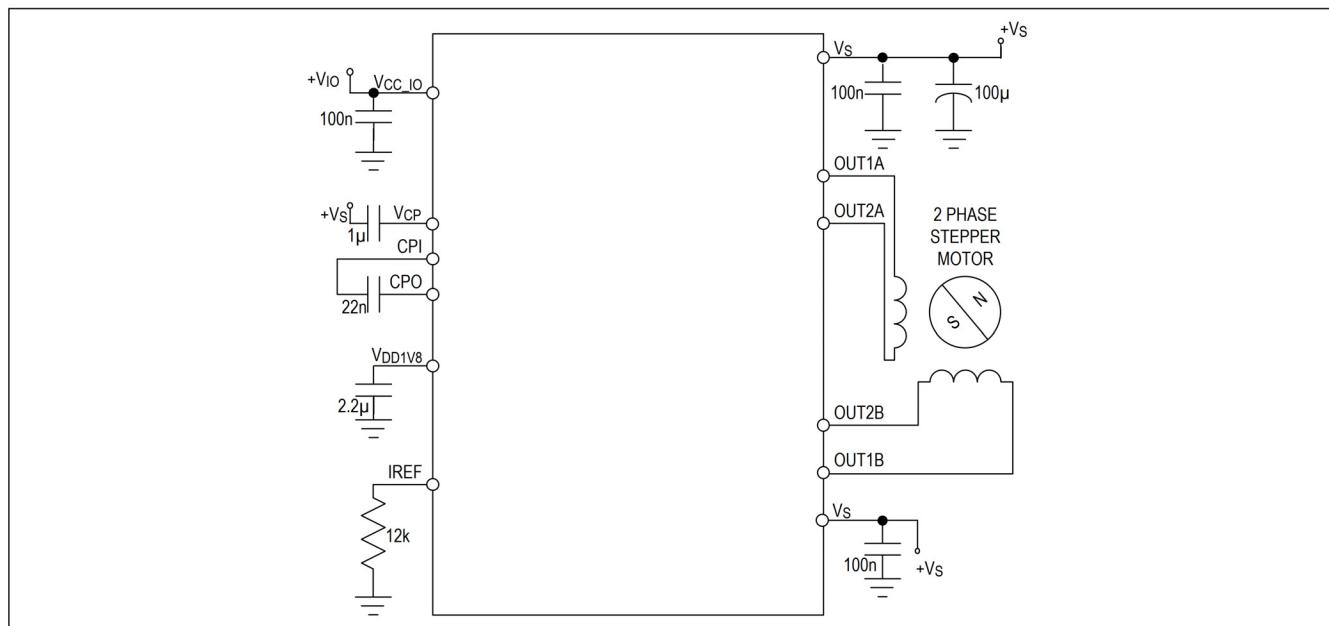


図 39. 標準的なアプリケーション回路

### 大モータ電流

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

おおよその目安として、PCB 設計の熱的性質は、モータ電流が長時間にわたり 1.5A RMS 以上となると重要になる可能性があります。抵抗による消費電力はモータ電流の 2 乗に比例して増加することに注意してください。一方、このことは、モータ電流をわずかに減らすだけでも、大幅に放熱を減らしエネルギーを節約できることを意味します。

### 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・チップ内に ESD 回路があるにもかかわらず、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 にも及ぶ重大な ESD イベントを引き起こす傾向にあります。モータ自体を PCB グラ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）

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

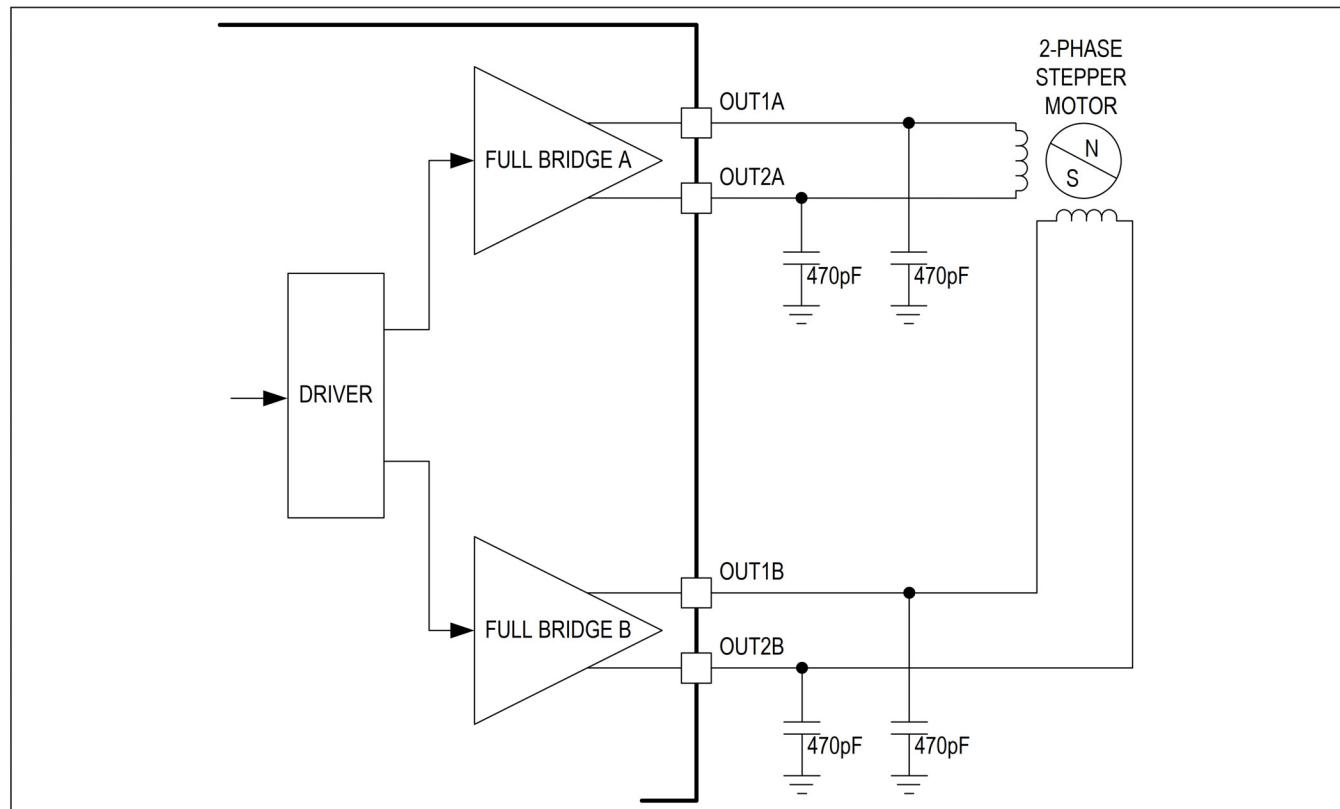


図 40. 簡単な ESD 対策

より精巧な方法は、LC フィルタを用いてドライバ出力をモータ・コネクタとデカッピングすることです。コイル端子間のバリスタ V1 および V2 は、活線接続に起因するコイルの過電圧をなくします。オプションで、バリスタ (V1A, V1B, V2A, V2B) により、すべての出力を ESD 電圧から保護します。バリスタは電源電圧の定格に合わせてください。SMD (表面実装) インダクタはモータ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）

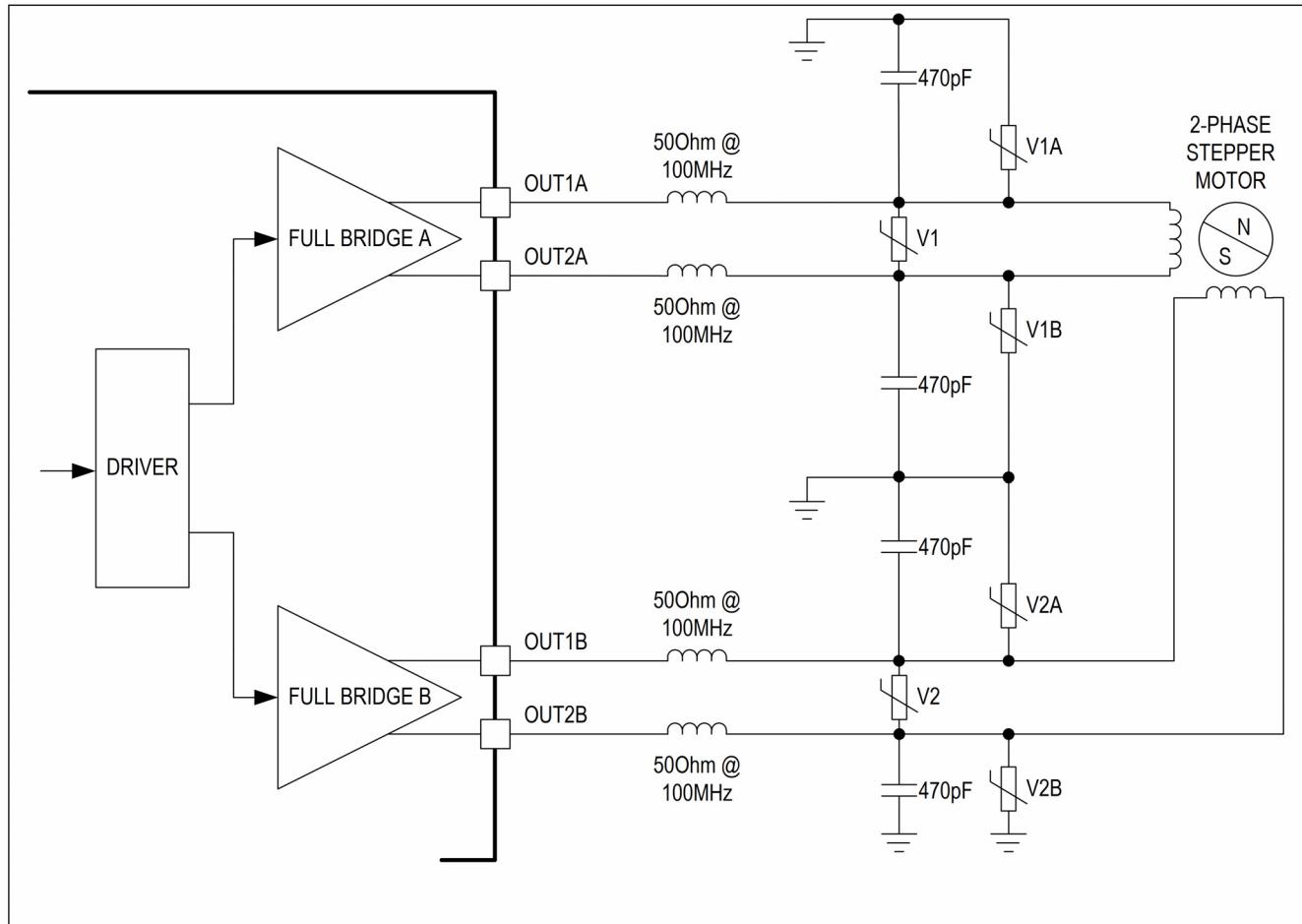


図 41. モータ出力保護の強化

## 型番

PART NUMBER	TEMPERATURE RANGE	PIN-PACKAGE
TMC5240ATJ+	-40°C to +125°C	32 TQFN - 5mm x 5mm
TMC5240ATJ+T	-40°C to +125°C	32 TQFN - 5mm x 5mm
TMC5240AUU+	-40°C to +125°C	38 TSSOP-EP 4.4mm x 9.7mm
TMC5240AUU+T	-40°C to +125°C	38 TSSOP-EP 4.4mm x 9.7mm

+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=デーブ&リール。

## 改訂履歴

版数	改訂日	説明	改訂ページ
0	12/22	市場投入のためのリリース	-