



TMC2240

36V 2A_{RMS}+の S/D および SPI 付き
スマート集積ステッピング・モータ・ドライバ

概要

TMC2240 は、スマート高性能ステッピング・モータ・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、256 マイクロステップ、組み込みインデクサ、2 つの全機能内蔵 36V、3.0A_{MAX}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・ドライバです。

ADI-Trinamic の 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する一方で、クラス最高の性能を提供します。

H ブリッジ 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 0.23Ω 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 $I_{RMS} = 2.1A_{RMS}$ です。

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

H ブリッジごとの最大フルスケール電流は $I_{FS} = 3.0A$ で、 I_{REF} に接続された外付け抵抗で設定できます。

この電流は、エンベデッド電流駆動レギュレーション回路の最大電流設定値として定義されます。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、スペースと消費電力を大幅に削減できます。

TMC2240 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 イベントの際には、ドライバはデイスエーブルされます。

更に、TMC2240 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での測定を行う機能が備わっています。

TMC2240 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

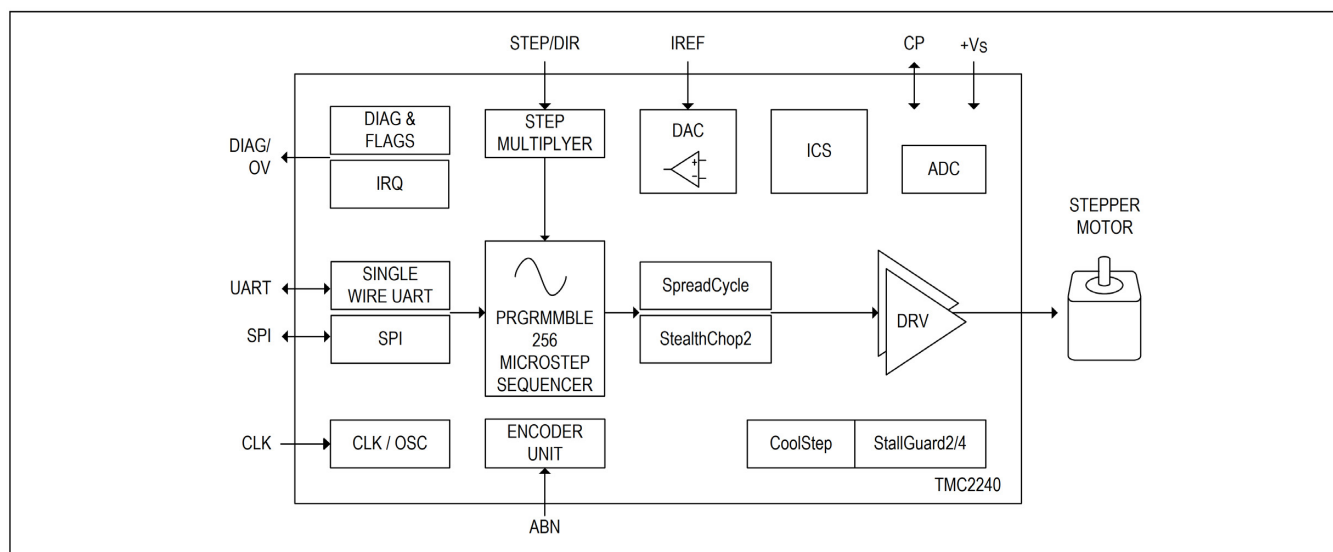
- リキッド・ハンドリング、医療用アプリケーション
- オフィス・オートメーションおよびペーパー・ハンドリング
- POS、マッサージ・チェア
- ATM、現金回収機、紙幣識別機、現金自動支払機
- CCTV、セキュリティ
- ポンプおよびバルブ制御
- ヘリオスタットやアンテナのポジショニング

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、 $T_A = 25^\circ C$)
- H ブリッジあたりの電流定格 (25°C での代表値) :
 - $I_{MAX} = 5.0A$ (ブリッジ・ピーク電流)
 - $I_{RMS} = 2.1A_{RMS}$ (3A のサイン波ピーク)
- 全機能内蔵無損失電流検出
- MicroPlyer ステップ補間機能を備えたステップ/方向 (S/D) インターフェイス
- SPI および単線式 UART
- インクリメンタル・エンコーダ・インターフェイス
- 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
- モータ位相とチップ温度の測定
- 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

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.....	9
パッケージ情報.....	9
TQFN32 5mm × 5mm.....	9
TSSOP38 9.7mm × 4.4mm EP.....	9
電気的特性.....	9
ピン配置.....	14
TMC2240 TQFN のピン配置.....	14
TMC2240 TSSOP のピン配置.....	14
端子説明.....	15
機能図.....	18
TMC2240.....	18
詳細.....	19
動作原理.....	19
シリアル・インターフェイスと診断フィードバック機能を備えたステップおよび方向ドライバ.....	19
主要コンセプト.....	20
制御インターフェイス.....	20
ステップおよび方向インターフェイス.....	20
自動停止パワー・ダウン.....	20
StealthChop2 および SpreadCycle ドライバ.....	21
StallGuard2/4 — 機械的負荷検出.....	21
CoolStep — 負荷適応型電流制御.....	21
エンコーダ・インターフェイス.....	22
SPI インターフェイス.....	22
SPI データグラム構造.....	22
読出し／書込みの選択 (WRITE_notREAD).....	22
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.....	23
データ・アライメント.....	23
SPI 信号.....	23
SPI のタイミング.....	24
UART 単線式インターフェイス.....	24
データグラム構造.....	24
書込みアクセス.....	24
読出しアクセス.....	25
CRC の計算.....	25
CRC 計算の C コード例.....	26

目次（続き）

UART 信号	26
複数ノードのアドレス指定	26
ステップ／方向インターフェイス	27
タイミング	27
分解能の変更	28
MicroPlyer ステップ・インターポレータと静止検出	29
StealthChop2	30
自動調整	30
StealthChop2 のオプション	31
StealthChop2 電流レギュレータ	33
下限電流値	35
速度ベースのスケールリング	36
StealthChop2 と SpreadCycle の結合	37
StealthChop2 でのフラグ	39
開放負荷フラグ	39
モータの状態を示す PWM_SCALE_SUM	39
フリーホイーリングおよび受動ブレーキング	39
StealthChop2 を制御するパラメータ	40
SpreadCycle と一般的なチョッパ	41
SpreadCycle チョッパ	43
一般的な定オフ時間チョッパ	45
内蔵電流検出機能	46
モータ電流の設定	46
フルスケール電流レンジの設定	47
速度ベースのモード制御	48
StallGuard2 の負荷測定	50
StallGuard2 の更新レートおよびフィルタ	52
モータ・ストールの検出	52
StallGuard2 を使用したホーミング	52
StallGuard2 動作の制限	52
StallGuard4 の負荷測定	52
StallGuard4 の調整	54
StallGuard4 の更新レート	54
モータ・ストールの検出	55
StallGuard4 動作の制限	55
CoolStep の負荷適応型電流スケールリング	55
CoolStep 用の設定	55
CoolStep の調整	57
応答時間	57

目次（続き）

低速度およびスタンバイ動作.....	57
診断出力.....	57
サイン波ロックアップ・テーブル.....	59
マイクロステップ・テーブル.....	59
ABN インクリメンタル・エンコーダ・インターフェイス.....	61
モータの分解能に合うエンコーダ設定.....	63
リセット、ディスエーブル/停止、パワー・ダウン.....	63
緊急停止.....	63
外部リセットおよびスリープ・モード.....	63
ポジション・ロスのないステッパ・モータの再起動.....	64
保護およびドライバ診断.....	64
過電流保護.....	65
過熱保護およびサーマル・シャットダウン.....	65
温度測定.....	65
チップ温度の測定.....	65
モータ温度の測定.....	65
過電圧保護と OV ピン.....	66
短絡保護（GND への短絡および VS への短絡）.....	67
開放負荷診断.....	67
低電圧ロックアウト保護.....	67
ESD 保護.....	68
外部アナログ入力 AIN のモニタリング.....	68
クロック発振器とクロック入力.....	68
内部クロックの使用.....	68
外部クロックの使用.....	68
クイック設定ガイド.....	68
電流の設定.....	69
StealthChop2 の設定.....	70
SpreadCycle の設定.....	71
StealthChop2 との組み合わせで CoolStep をイネーブル.....	72
SpreadCycle との組み合わせで CoolStep をイネーブル.....	73
全般的なレジスタ・マッピングおよびレジスタ情報.....	73
レジスタ・マップ.....	75
TMC2240.....	75
レジスタの詳細.....	79
標準アプリケーション回路.....	124
標準的なアプリケーション回路.....	124
大モータ電流.....	124
ドライバ保護および EME 回路.....	124

目次（続き）

型番..... 126

改訂履歴..... 127

図一覧

図 1. ブロック図	18
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	19
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	21
図 4. SPI タイミング図	24
図 5. UART のデジタイズチェーン例	27
図 6. STEP/DIR 信号のタイミング	28
図 7. STEP/DIR 信号用入力フィルタの構造	28
図 8. 立上がり STEP 周波数での MicroPlyer マイクロステップ補間（例：16～256）	30
図 9. StealthChop2 の自動調整手順	32
図 10. StealthChop2：PWM_REG の設定は良好	33
図 11. StealthChop2：AT#2 時の PWM_REG の設定が過小	34
図 12. 正しく決定された PWM_GRAD(_AUTO)および PWM_OFS(_AUTO)	34
図 13. 過小な PWM_GRAD 設定の例	35
図 14. 速度ベースの PWM スケーリング（pwm_autoscale = 0）	37
図 15. SpreadCycle への切り替えオプションのための TPWMTHRS	38
図 16. 代表的なチョッパ減衰フェーズ	42
図 17. 1 チョッパ・サイクルの間のコイル電流を示す、SpreadCycle チョッパの概略図	44
図 18. 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	45
図 19. 一般的なチョッパのゼロ交差とサイン波オフセットを使用した補正	46
図 20. 速度依存モードの選択	49
図 21. StallGuard2 の動作原理	51
図 22. StallGuard4 の動作モード	53
図 23. モータ電流を負荷に適合させる CoolStep	56
図 24. DIAG0 と DIAG1 の出力オプション	58
図 25. コイル B マイクロステップ波の正のゼロ遷移におけるインデックス信号（オープン・ドレイン設定時）	59
図 26. LUT プログラミングの例	60
図 27. OFFSET_SIN90 を使用したコサイン波のシフト	61
図 28. インクリメンタル・エンコーダの ABN 信号の概要	62
図 29. ブレーキ・チョッパ回路の例	67
図 30. 電流設定のためのクイック設定ガイド	69
図 31. StealthChop2 のクイック設定ガイド	70
図 32. SpreadCycle のクイック設定ガイド	71
図 33. StealthChop2 との組み合わせによる CoolStep のクイック設定ガイド	72
図 34. SpreadCycle との組み合わせによる CoolStep のクイック設定ガイド	73
図 35. 標準的なアプリケーション回路	124
図 36. 簡単な ESD 対策	125
図 37. モータ出力保護の強化	126

表一覧

表 1. SPI データグラムの構造	22
表 2. SPI 読出し／書込みフローの例	23
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	23
表 4. UART 書込みアクセス・データグラム構造	24
表 5. UART 読出しアクセス要求データグラムの構造	25
表 6. UART 読出しアクセス返答データグラムの構造	25
表 7. TMC2240 UART のインターフェイス信号	26
表 8. 最大 255 ノードをアドレス指定する UART 例	27
表 9. 位相 A/B コイル電流のフルステップ／ハーフステップ・ルックアップ・テーブル値	29
表 10. StealthChop2 自動調整 AT#1 および AT#2 の制約事項と要件	30
表 11. StealthChop2 用の PWM 周波数の選択（太字を推奨）	32
表 12. StealthChop2 を制御するパラメータ	40
表 13. SpreadCycle および一般的な定オフ時間チョッパを制御するパラメータ	42
表 14. SpreadCycle モードのパラメータ	44
表 15. 定オフ時間チョッパ・モードを制御するパラメータ	46
表 16. モータ電流を制御するパラメータ	46
表 17. I _{FS} フルスケール・ピーク・レンジの設定値（R _{REF} = 12k Ω の場合の例）	47
表 18. 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））	48
表 19. 速度ベースのモード制御パラメータ	50
表 20. StallGuard2 関連のパラメータ	51
表 21. StallGuard4 関連のパラメータ	53
表 22. CoolStep の重要パラメータ	55
表 23. CoolStep のその他のパラメータとステータス情報	56
表 24. 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	63
表 25. ポジション回復の方法	64
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ュホルド	67
表 27. レジスタ・マップの概要	74

絶対最大定格

V _S ~ GND	 -0.3V ~ 41V
V _{DD1V8} ~ GND	 -0.3V ~ min (2.2, V _S + 0.3)V
AGND ~ GND	 -0.3V ~ +0.3V
OUT1A, OUT2A, OUT1B, OUT2B	 -0.3V ~ V _S + 0.3V
V _{CP} ~ GND	 V _S - 0.3V ~ min (44, V _S + 6)V
CPO ~ GND	 V _S - 0.3V ~ min (44, V _S + 6)V
CPI ~ GND	 -0.3V ~ min (41, V _S + 0.3)V
SLEEPN ~ GND	 -0.3V ~ V _S + 0.3V

IREF, AIN ~ GND	 -0.3V ~ min (2.2, V _{DD1V8} + 0.3)V
V _{CC_IO} ~ GND	 -0.3V ~ 5.5V
ロジック入力/出力電圧 ~ GND	 -0.3V ~ V _{CC_IO} + 0.3V
OV ~ 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	21-0140
Land Pattern Number	90-0013
Thermal Resistance, Single-Layer Board:	
Junction to Ambient (θ _{JA})	47°C/W
Junction to Case (θ _{JC})	1.7°C/W
Thermal Resistance, Four-Layer Board:	
Junction to Ambient (θ _{JA})	29°C/W
Junction to Case (θ _{JC})	1.7°C/W

TSSOP38 9.7mm × 4.4mm EP

Package Code	U38E+3C
Outline Number	21-0714
Land Pattern Number	90-0435
Thermal Resistance, Four-Layer Board:	
Junction to Ambient (θ _{JA})	25°C/W
Junction to Case (θ _{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 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/ を参照してください。

電気的特性

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

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER SUPPLY						
Supply Voltage Range	V _S		4.5		36	V
Sleep Mode Current Consumption	I _{VS}	V(SLEEPN) = 0		4	18	μA

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	μA
Quiescent Current Consumption	I_{VCC_IO}	$V(SLEEPN) = 1$		35	60	μA
LOGIC LEVEL INPUTS-OUTPUTS						
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	μA
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	μA
SLEEPN Voltage Level High	$V_{IHSLEEPN}$		0.9			V
SLEEPN Voltage Level Low	$V_{ILSLEEPN}$				0.6	V
SLEEPN Pulldown Input Resistance	$R_{PDSLEEPN}$		0.8	1.5		$M\Omega$
OUTPUT SPECIFICATIONS						
Output ON-Resistance Low Side	R_{ONLS}	Full-scale bits = 10		0.11	0.2	Ω
		Full-scale bits = 01		0.15	0.28	
Output ON-Resistance Low Side	R_{ONLS}	Full-scale bits = 00		0.28	0.54	Ω
Output ON-Resistance High Side	R_{ONHS}			0.12	0.22	Ω
Output Leakage	I_{LEAK}		-5		+5	μA

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		
PROTECTION CIRCUITS						
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 _{OCP}		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 Hysteris	UVLOHYS			0.12		V
UVLO Threshold on V _{CC_IO}	UVLO	V _{CC_IO} falling	0.9	1.5	1.95	
V _{CC_IO} UVLO Hysteresis	UVLOVCCCH			100		mV
Thermal Protection Threshold Temperature	TSD			165		°C
Thermal Protection Temperature Hysteresis				20		°C
CURRENT REGULATION						
IREF Pin Resistor Range	R _{REF}		12		60	kΩ
IREF Output Voltage	V _{REF}		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} = 12kΩ	-5		+5	%
FUNCTIONAL TIMINGS						
SLEEP Time	t _{SLEEP}	SLEEPN = 0 to OUT_ three state			50	μs
Wake-Up Time from Sleep	TWAKE	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
CLOCK						
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
SPI TIMINGS						
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
STEP/DIR TIMINGS						
Step Frequency	f_{STEP}	dedge = 1		$f_{CLK}/8$		
		dedge = 0		$f_{CLK}/4$		
Fullstep Frequency	f_{FS}			$f_{CLK}/512$		
STEP High Time	t_{SH}		$t_{CLK} + 20$			ns
STEP Low Time	t_{SL}		$t_{CLK} + 20$			ns
DIR/STEP to CLK Setup Time	t_{SU}		10			ns
DIR/STEP to CLK Hold Time	t_{SH}		10			ns
DIR to STEP Setup Time	t_{SU}		20			ns

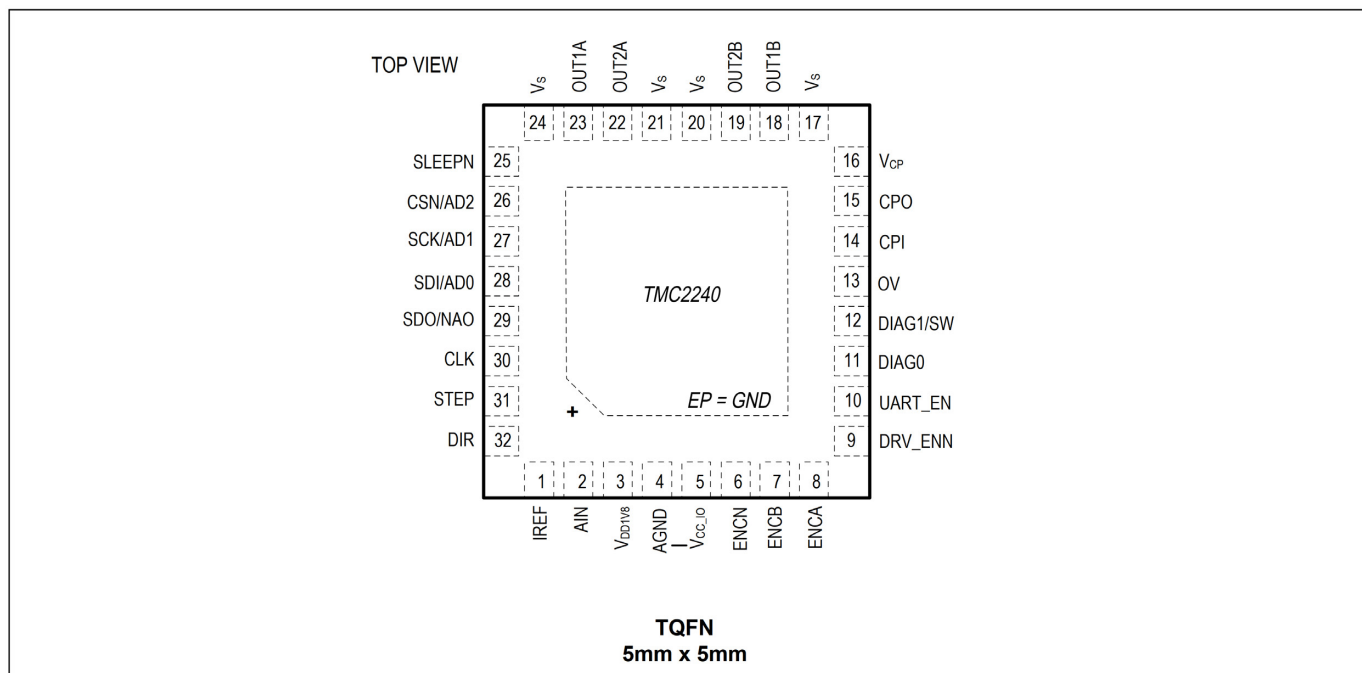
電气的特性（続き）

($V_S = 4.5V \sim 36V$ 、 $R_{REF} = 12k\Omega \sim 24k\Omega$ 、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。)

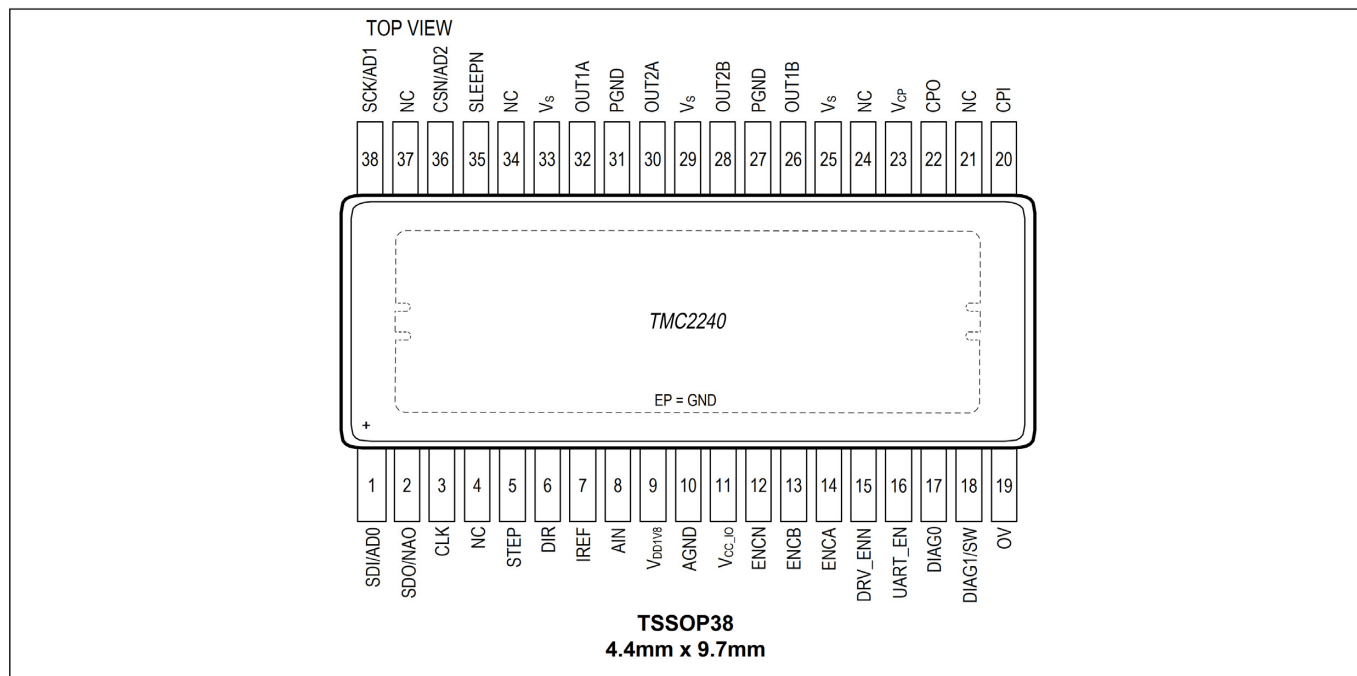
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
DIR to STEP HoldTime	t_H		20			ns
ENCODER TIMING						
Encoder Counting Frequency	f_{CNT}			$< 2/3 f_{CLK}$	f_{CLK}	
A/B/N Input Low Time	t_{ABNL}		$3t_{CLK} + 20$			ns
A/B/N Input High Time	t_{ABNH}		$3t_{CLK} + 20$			ns
A/B/N Spike Filtering Time	$t_{FILTABN}$	Rising and falling edge		$3t_{CLK}$		
ADC/Analog Input/Temperature						
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
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}			± 10		$^\circ C$
Supply Voltage Measurement Accuracy			-5		+5	%
ADC Sample Rate	$f_{SAMPLE, ADC}$			$f_{CLK} / 2048$		

ピン配置

TMC2240 TQFN のピン配置



TMC2240 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・グラウンド。グラウンド・プレーンに接続します。		GND
27, 31		PGND	電源グラウンド。グラウンド・プレーンに接続します。		GND
17, 20, 21, 24	25, 29, 33	V _S	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 _{DD1V8}	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 _{CP}	チャージ・ポンプ電圧。1.0μF のコンデンサを用いて V _S 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 _S ピンの近くに接続します。		Analog Output
5	11	V _{CC_IO}	回路の IO レベルを定義するために外部電源から供給されるデジタル IO 電源電圧。出力ピンに適切な電圧レベルを設定するために必要です。	V _{CC_IO}	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 CPO 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ル・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 _{CC_IO}	Digital Input
31	5	STEP	STEP 入力。	V _{CC_IO}	Digital Input
32	6	DIR	方向入力。	V _{CC_IO}	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 _{CC_IO}	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 _{CC_IO}	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 _{CC_IO}	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 _{CC_IO}	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 _{CC_IO}	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 _{CC_IO}	Digital Input (pull down)
7	13	ENCB	エンコーダの B チャンネル入力。	V _{CC_IO}	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 _{CC_IO}	Digital Input (pull up)
6	12	ENCN	エンコーダの N チャンネル入力。	V _{CC_IO}	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 _{CC_IO}	Digital Input (pull up)
11	17	DIAG0	診断出力 DIAG0。 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 _{CC_IO}	Digital Output
12	18	DIAG1/SW	診断出力 DIAG1。 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 _{CC_IO}	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 はブルダウンを内蔵しています。 使用しない場合は、V _S または V _{CC_IO} （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 _S	Analog Input (pull down)
19	28	OUT2B	モータ・コイル B の出力 2	V _S	Analog Output
18	26	OUT1B	モータ・コイル B の出力 1	V _S	Analog Output
22	30	OUT2A	モータ・コイル A の出力 2	V _S	Analog Output
23	32	OUT1A	モータ・コイル A の出力 1	V _S	Analog Output
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

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				
–	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ュホー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

TMC2240

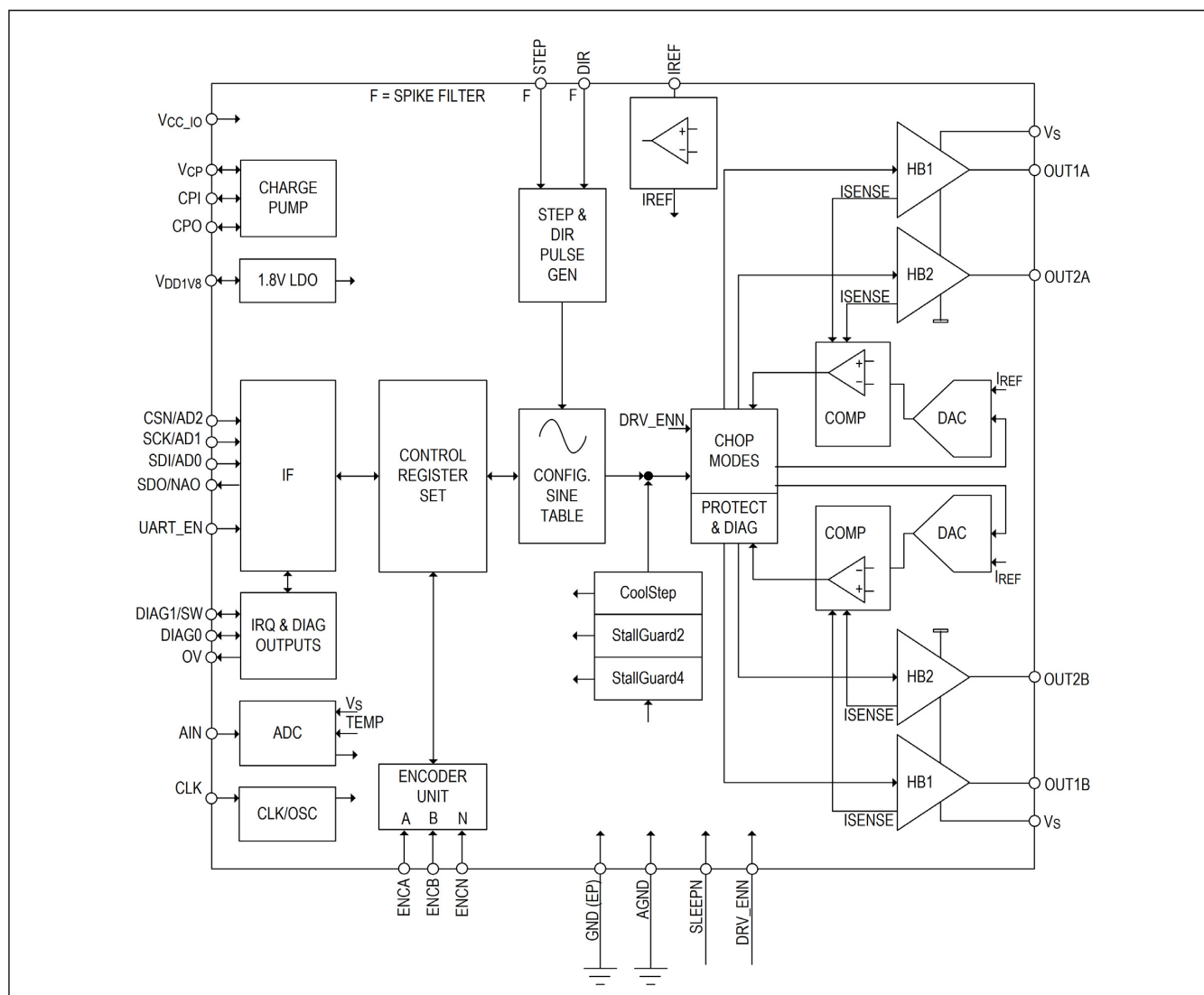


図 1. ブロック図

詳細

動作原理

シリアル・インターフェイスと診断フィードバック機能を備えたステップおよび方向ドライバ

TMC2240 はスマート機能を備えたステップおよび方向方式のステップ・モータ・ドライバで、パラメータ化、モニタリング、および診断のためのシリアル・インターフェイス (SPI、UART) を内蔵しています。

TMC4361A のような外部の高性能モーション・コントローラや CPU が、システム内の追加モータなどの他コンポーネントに同期されたステップ信号と方向信号を生成します。TMC2240 はインテリジェント電流制御に対応しており、そのシリアル・インターフェイスの 1 つを介してモータの状態をフィードバックします。

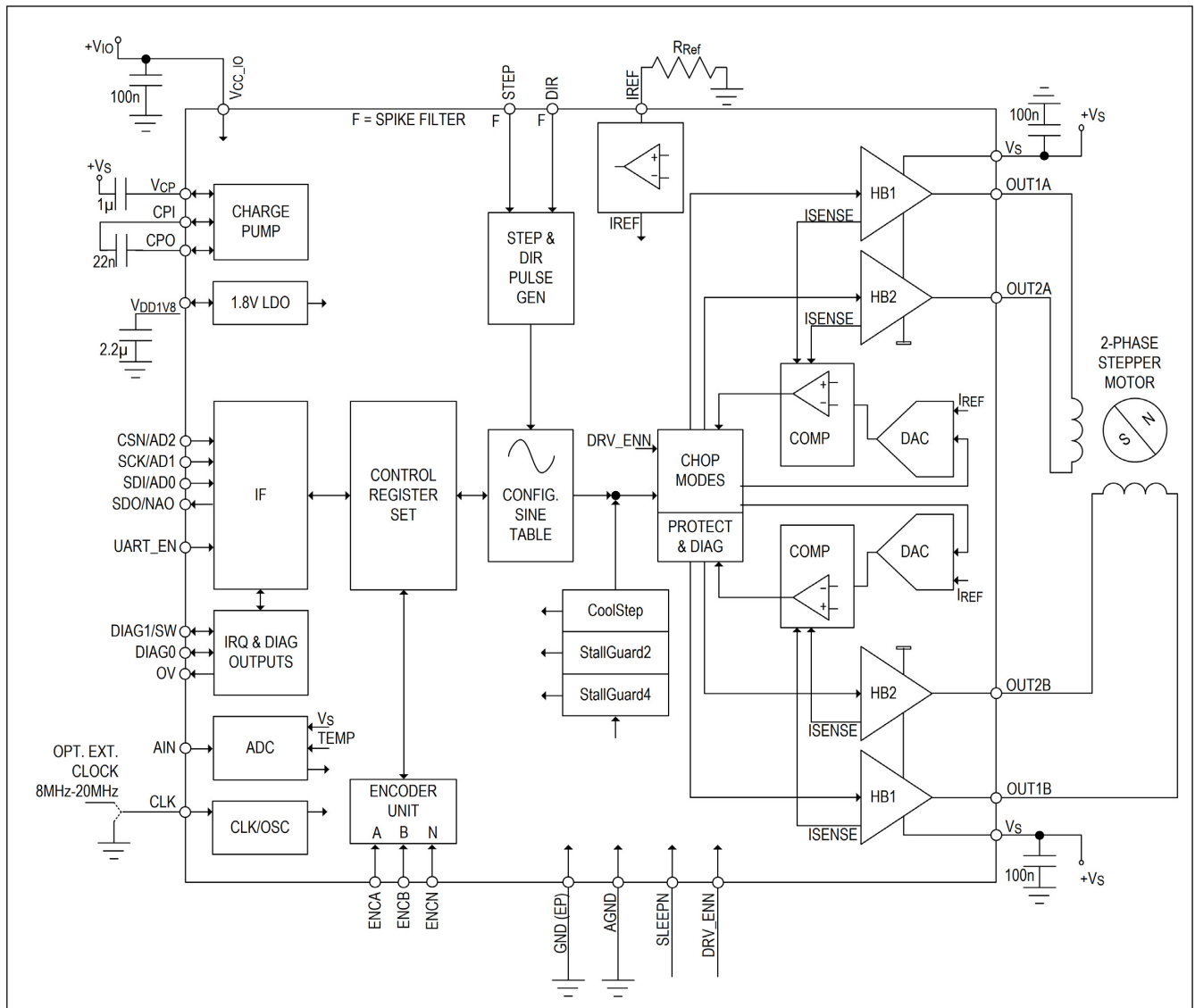


図 2. 代表的な外部コンポーネントを使用したブロック図

主要コンセプト

TMC2240 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停止を実現する、無騒音高精度チョッパー・アルゴリズム。
<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>	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>	センサーレス・ホーミング安全エンド・スイッチとモータ過負荷警告
<i>CoolStep</i>	StallGuard のフィードバックに基づくアクティブ・ピーク電流制御によって最大効率を実現し、モータおよびドライバの温度を抑制
<i>MicroPlyer</i>	低分解能のステップ入力で最大 256 のマイクロステップ動作を実現するマイクロステップ・インターポレータ

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

TMC2240 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_{CC_IO} 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。ボー・レートの設定は不要です。

ステップおよび方向インターフェイス

モータは、ステップおよび方向入力を使用して制御されます。STEP 入力のアクティブ・エッジは、立上がりエッジとするか、立上がりエッジと立下がりエッジ両方とすることができ、これはモード・ビット (*dedge*) により制御されます。両方のエッジを使用する場合は STEP 信号のトグル・レートが半分になり、光絶縁カプラなどの低速インターフェースの制御に有効です。それぞれのアクティブ・エッジにおいては、DIR 入力からサンプリングされたステートが、ステップ動作を順方向にするか逆方向にするかを決定します。各ステップはフルステップまたはマイクロステップとすることができ、フルステップあたりのマイクロステップ数は 2、4、8、16、32、64、128、または 256 です。DIR がロー・ステートのときのステップ・インパルスはマイクロステップ・カウンタ値を加算し、ハイ・ステートのときは減算します。加減算の量はマイクロステップ分解能によって制御されます。内部テーブルはカウンタ値をサイン値とコサイン値に換算し、これらの値がマイクロステッピング用のモータ電流を制御します。

自動停止パワー・ダウン

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

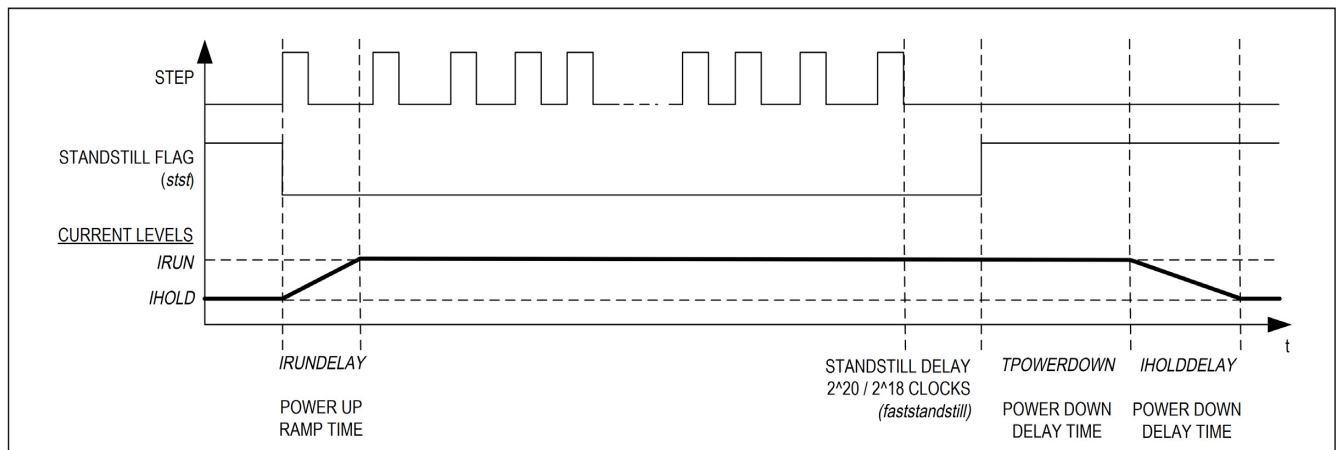


図 3. 停止時およびランプアップ時の自動モータ電流制御

StealthChop2 および SpreadCycle ドライバ

StealthChop2 の原理は電圧チョッパに基づいています。特に、停止時および低速動作時には、ボール・ベアリングで発生するノイズを除き、モータ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・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プ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）約 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

TMC2240 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。このエンコーダは、エンコーダ位置と外部ランプ・ジェネレータ位置間の整合性チェックをオンザフライで行うために使用できます。プログラマブル・プリスケアラを用いると、エンコーダの分解能をモータの分解能に合わせるすることができます。32 ビットのエンコーダ・カウンタを備えています。

SPI インターフェイス

SPI データグラム構造

TMC2240 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、読出しアクセスと書込みアクセスの違いは、読出しアクセスではアドレス指定されたレジスタにはデータを伝送せずにアドレスのみを伝送し、その 32 データ・ビットはダミーであるという点にあります。また、更に続けて読出しまたは書込みアクセスを行うと、その前の読出しサイクルで伝送されたアドレスからのデータ読出しが返されます。

読出しアクセス要求のデータグラムはダミー書込みデータを使用します。読出しデータは、その後の読出しアクセスまたは書込みアクセスで再びコントローラに転送されます。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。

TMC2240 との間でデータの読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) への読出しアクセスの場合、読出しアクセスの直前のアクセス時にアドレス・バイトを 0x21 に設定する必要があります。レジスタ (VACTUAL) への書込みアクセスの場合、アドレス・バイトは 0x80 + 0x22 = 0xA2 に設定する必要があります。読出しアクセスの場合、データ・ビットは任意の値 (-) を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 読出し／書込みフローの例

ACTION	DATA SENT TO TMC2240	DATA RECEIVED FROM TMC2240
read XACTUAL	0x2100000000	0xSS & unused data*
read XACTUAL	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チされ、次の SPI 転送で使用できます。

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

ビット	名称	コメント
7:4	don't care	TMC2240 では使用しません
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 1 のドライバ・エラーを指示 (GSTAT の読出しでクリア)
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

TMC2240 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 の立上がりエッジで SDI からのデータをラッチし、立下がりエッジの後 SDO にデータを送ります。最初に最上位ビットが送られます。TMC2240 でバス・トランザクションを行うには、最低でも 40SCK クロック・サイクルが必要です。

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

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。

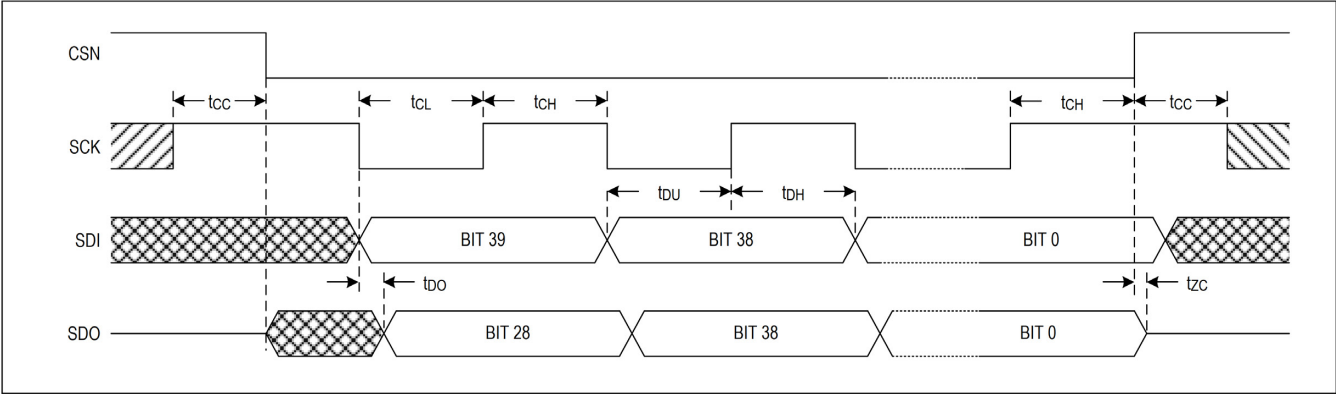


図 4. SPI タイミング図

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 TMC2240 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。自動ポー・レート検出機能を備えているため、このインターフェイスは容易に導入可能です。

データグラム構造

書込みアクセス

表 4. UART 書込みアクセス・データグラム構造

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

同期ニブルが TMC2240 との間の伝送の前に置かれ、最初の伝送バイトに組み込まれた後にアドレス指定バイトが置かれます。伝送ごとに、内部ポー・レート分周器と UART ホスト・クロックの同期が可能です。実際のポー・レートは適応型であり、内部クロック周波数の変化は補償されます。そのため、ポー・レートは有効な範囲内で自由に選択できます。伝送される各バイトは、開始ビット（DIAG1/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_{CLK}/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 は読出し（0）アクセスまたは書込み（1）アクセスを識別します。例：アドレス 0x10 は書込みアクセスでは 0x90 に変更されます。

2 つの連続バイトの開始ビットの間隔が 63 ビット分より長い休止時間になった場合、通信はリセットされます。このタイミングは最後に正しく受信されたデータグラムに基づきます。この場合、バス・アイドル時間である 12 ビット分以上の故障回復時間が経過してから、伝送を再開する必要があります。この方式をとることで、伝送エラーの場合に UART ホスト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ません。

受け入れられた書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ノリッジされます。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込みアクセスが成功したかどうかをチェックできます。読出しアクセスではカウンタは変更されません。

読出しアクセス

表 5. UART 読出しアクセス要求データグラムの構造

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

読出しアクセス要求データグラムの構造は、書込みアクセスのデータグラム構造と同じですが、使用するユーザ・ビットの数が少なくなっています。その機能は、UART ノードのアドレス指定と、読出しアクセスに必要なレジスタ・アドレスの伝送です。TMC2240 は、UART ホストが読出し要求のために用いるボー・レートと同じレートで応答します。

ホストからノードへのクリーンなバス遷移を確保するために、TMC2240 は読出しアクセスに対する返答をすぐには送信せず、プログラマブルな遅延時間を用い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時間設定を用い8ビットの倍数分の時間で設定できます（デフォルトは8ビット分の時間）。複数ノードのシステムでは、すべてのノードに対し SENDDelay を最小でも 2 に設定します。そうしない場合、異なるノードへの読出しアクセス時に、アドレス指定されていないノードが伝送エラーを検出する可能性があります。

表 6. UART 読出しアクセス返答データグラムの構造

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

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

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

CRC の計算

8 ビット CRC 多項式を用いて、読出しおよび書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。TMC2240 は、固有のノード・アドレスを含む、正しく伝送されたデータグラムに対してのみ、応答します。書込みアクセス・データグラムが正しく受信されるごとに、データグラム・カウンタが増加します。

CRC = x⁸ + x² + x¹ + x⁰

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

TMC2240 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ボー・レートに対応します。読出しアクセスの場合、**DIAG1/SW** の出力ドライバをオンに切り替え、同じボー・レートで応答を送信します。

表 7. TMC2240 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 TMC2240 を単一の 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_{CC10} 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 TMC2240 の SDI/AD0 を GND に接続します。
- 最初の TMC2240 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 TMC2240 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。

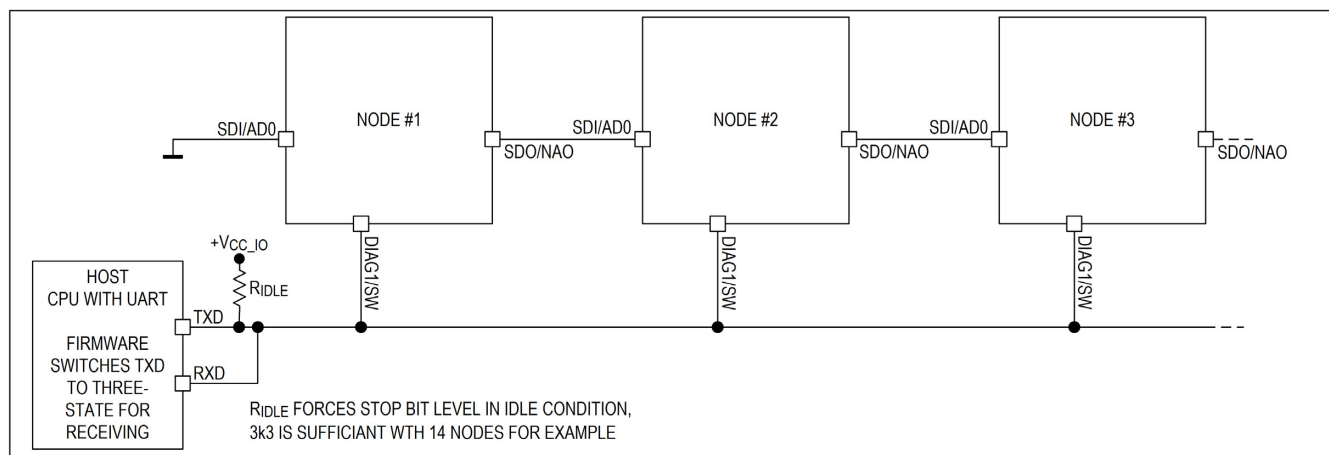


図 5. UART のデジーチェーン例

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		

ステップ/方向インターフェイス

STEP 入力と DIR 入力は、多くの既存モーション・コントローラに使用できるシンプルな標準インターフェイスを提供します。MicroPlyer ステップ・パルス・インターポレータは、もともと粗いステッピング用に設計されたアプリケーションに高分解能マイクロステッピングのスムーズなモータ動作を提供します。

タイミング

下の図に、STEP 信号と DIR 信号のタイミング・パラメータを示します。CHOPCONF レジスタの dedge モード・ビットをセットすると STEP の両方のエッジがアクティブになり、dedge をクリアすると立上がりエッジだけがアクティブになります。STEP と DIR はサンプリングされて、システム・クロックに同期されます。約 10ns の内部アナログ・フィルタは、長い PCB 配線パターンなどによって生じる信号のグリッチを除去します。信号ソースがチップから遠く離れている場合、特に信号をケーブルで伝送する場合は、信号にフィルタをかけるか差動で送信する必要があります。

タイミング・パラメータの仕様については、[電気的特性](#)の表を参照してください。

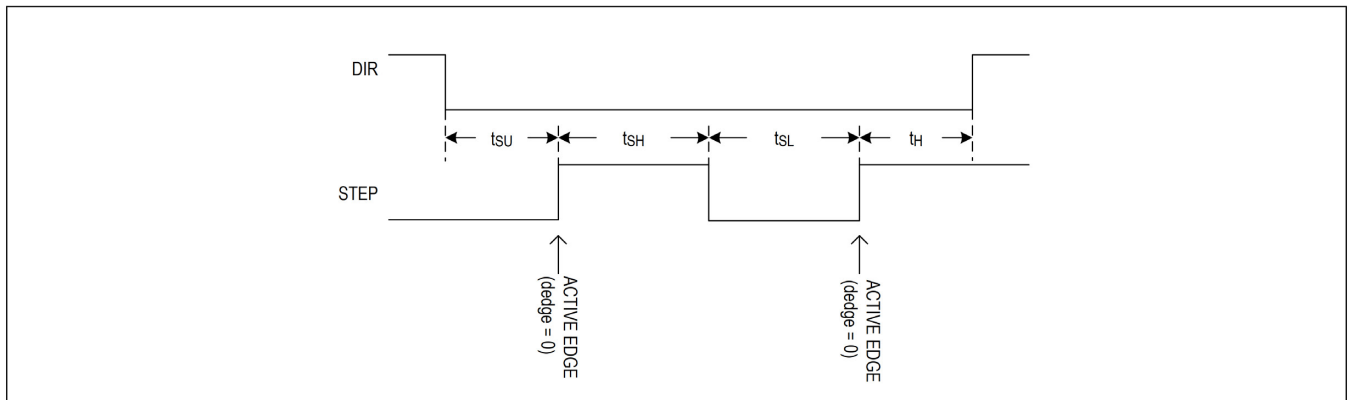


図 6. STEP/DIR 信号のタイミング

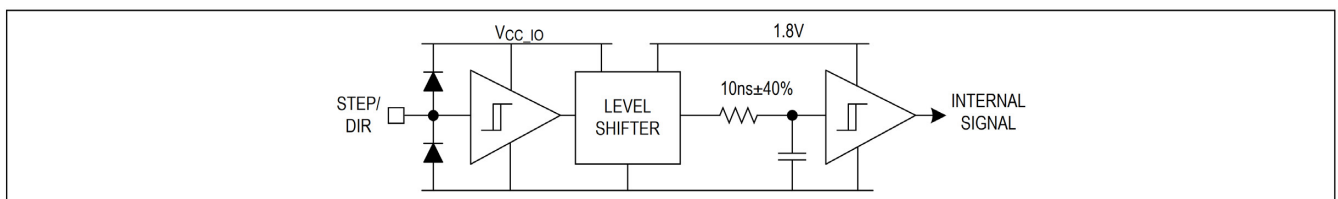


図 7. STEP/DIR 信号用入力フィルタの構造

分解能の変更

マイクロステップ分解能を小さくすると、STEP/DIR インターフェイスのステップ周波数を制限したり、旧型で低性能のドライバとの互換性を確保したりすることができます。1024 個のサイン波エントリを持つ内部マイクロステップ・テーブルは、サイン波のモータ・コイル電流を生成します。これら 1024 個のエントリは、1 回の電気的回転あるいは 4 つのフルステップに相当します。マイクロステップ分解能は、テーブル内で取られるステップ幅を決定します。DIR 入力に応じ、マイクロステップ・カウンタは STEP パルスごとにステップ幅だけ増加 (DIR = 0) または減少 (DIR = 1) します。マイクロステップ分解能はインクリメントと デクリメントを決定します。最大分解能ではシーケンサがステップ・パルスごとに 1 ステップずつ進み、1/2 分解能では 2 ステップずつ進みます。フルステップのインクリメントは最大 256 ステップです。シーケンサは異なるマイクロステップ・レートで、いつでもシームレスに切り替えることのできる特別な機能を備えています。低いマイクロステップ分解能に切り替えるときは目標分解能内で最も近いステップを計算し、その位置における電流ベクトルを読み出します。この動作は、フルステップやハーフステップなどの低分解能では特に重要です。その理由は、ステップ・シーケンス内での何らかの異常は、時計方向と反時計方向に動作するモータの比較時に非対称な動作につながるためです。

例：

フルステップ：テーブル位置 128、384、640、および 896 を通じたサイクル (45°、135°、225°、および 315° の電気的位置、両方のコイルを同じ電流でオン)。各位置でのコイル電流は RMS 値 (0.71 × 振幅) に対応します。ステップ・サイズは 256 (電気的には 90°) です。

ハーフステップ：最初のテーブル位置は 64 (電気的には 22.5°) で、ステップ・サイズは 128 (45° ステップ) です。

1/4 ステップ：最初のテーブル位置は 32 (電気的には 90°/8 = 11.25°) で、ステップ・サイズは 64 (22.5° ステップ) です。

このようにステップは等距離となり、これはどちらの回転方向でも同じです。一部の旧型ドライバも、ステップ・テーブル内でゼロ電流 (テーブル・エントリ 0、0°) とフル電流 (90°) を使用します。この種のステップングは、ドライバおよびモータにおいて実現可能なトルクが小さく消費電力も増加するので、避けられます。

表 9. 位相 A/B コイル電流のフルステップ／ハーフステップ・ルックアップ・テーブル値

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

MicroPlyer ステップ・インターポレータと静止検出

STEP の各アクティブ・エッジにおいて、MicroPlyer は 256x の分解能でマイクロステップを生成します。この機能は、最後のステップ間隔に基づき、ステップ入力の 2 つのステップ・インパルス間の時間を補間します。これにより、シングルスステップ・パルスに対して 2 マイクロステップ（128 マイクロステップから 256 マイクロステップまでの補間）から 256 マイクロステップ（フルステップ入力から 256 マイクロステップ）までが駆動されます。

MicroPlyer 機能は、CHOPCONF レジスタの intpol ビットによって有効にします。

補間された 2 マイクロステップから 256 マイクロステップまでのステップ・レートは、前のステップ周期の時間を測定し、それを最大 256 個の部分に等分することによって決定されます。256 個のマイクロステップが均等に分布している場合、2 つのマイクロステップ間の最大時間は 2^{20} （およそ 100 万システム・クロック・サイクル）に相当します。16MHz のシステム・クロック周波数では、これにより MicroPlyer 動作の最小ステップ入力周波数は 16Hz となります。ステップ・レートが低い場合は、静止イベントを示す STST ビットがセットされます。その周波数では、マイクロステップのレートが $(\text{システム・クロック周波数})/2^{16} \approx 256\text{Hz}$ となります。静止状態が検出されると、ドライバはモータを自動的に保持電流 IHOLD に切り替えます。

注意：MicroPlyer は、安定した STEP 周波数でのみ完全に機能します。STEP 信号のデューティ・サイクルが 50%以外の場合は dedge オプションを使用しないでください。

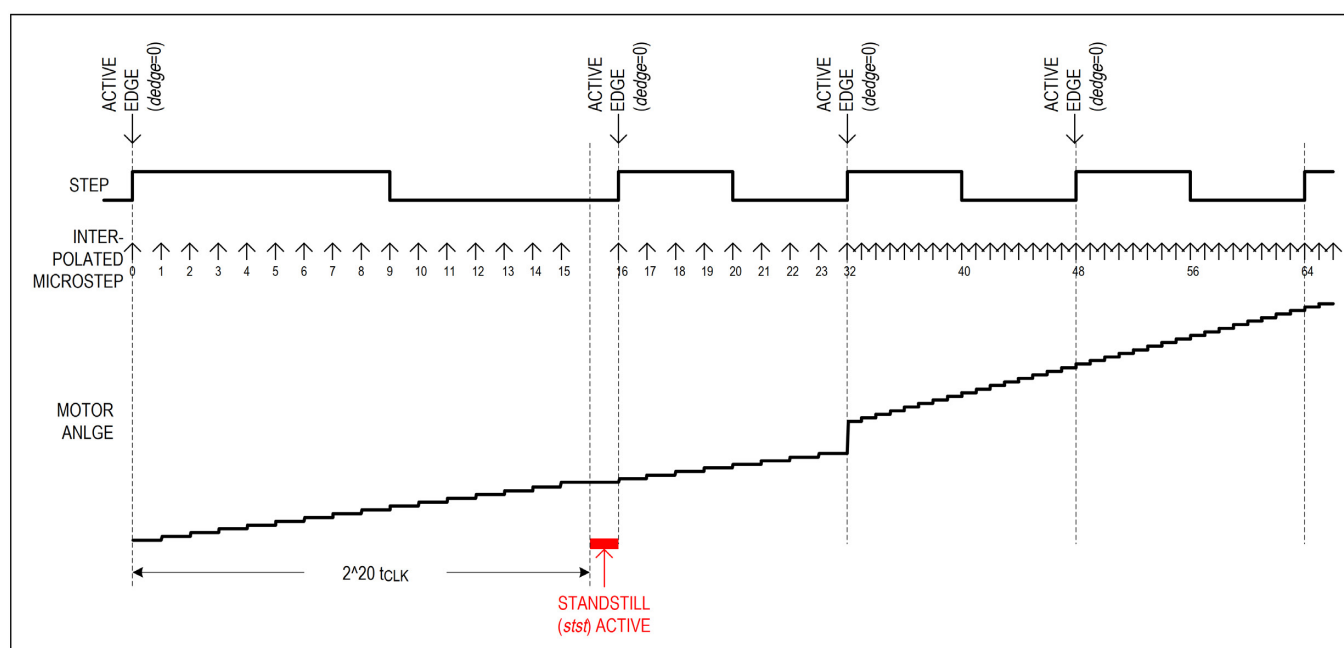


図 8. 立上がり STEP 周波数での MicroPlyer マイクロステップ補間（例：16～256）

この図では、最初の STEP サイクルが静止ビット *stst* を設定するのに十分な長さになっています。このビットは、次の STEP アクティブ・エッジでクリアされます。次に、外部 STEP 周波数が増加します。この高いレートで 1 サイクルが経過した後、MicroPlyer は、補間されたマイクロステップ・レートを高い周波数に合わせて調整します。MicroPlyer が低いレートの最後のサイクルで 16 個のマイクロステップをすべて生成しなかったため、高いレートでの最初のサイクルと 2 番目のサイクルの間でモータ角度がわずかにジャンプします。

StealthChop2

StealthChop2 は、ステッピング・モータ用の極めて静かな動作モードです。これは電圧モード PWM をベースとします。停止時や低速時、モータは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。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。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 10. 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	- 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 (<i>IRUN</i>) と同じ。 - 停止をなくすことが可能な場合は、初期ステップ・パルスが駆動を稼働電流に戻すか、<i>IHOLD</i> を <i>IRUN</i> に設定するよう切り替え。 - ピン <i>V_S</i> は動作レベル。	$\leq 2^{20} + 2 \times 2^{18} t_{CLK}$ 、 $\leq 130\text{ms}$ (内部クロック使用)

表 10. StealthChop2 自動調整 AT#1 および AT#2 の制約事項と要件（続き）

AT#2	PWM_GRAD_AUTO	- 大量のバック EMF が生じ、最大稼働電流に達する可能性のある速度にモータを駆動。条件： $1.5 \times PWM_OFS_AUTO \times (IRUN + 1)/32 < PWM_SCALE_SUM < 4 \times PWM_OFS_AUTO \times (IRUN + 1)/32$ $PWM_SCALE_SUM < 255$ ヒント：代表的な範囲は 60RPM～300RPM です。	±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 V_s 電圧を変更すると、自動調整プロセスの結果は無効になります。モータ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 電圧がスケールリングされます。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。

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対応するためには、自動調整モードで動作させてください。

非自動モード（pwm_autoscale = 0）を考慮するのは、モータ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。

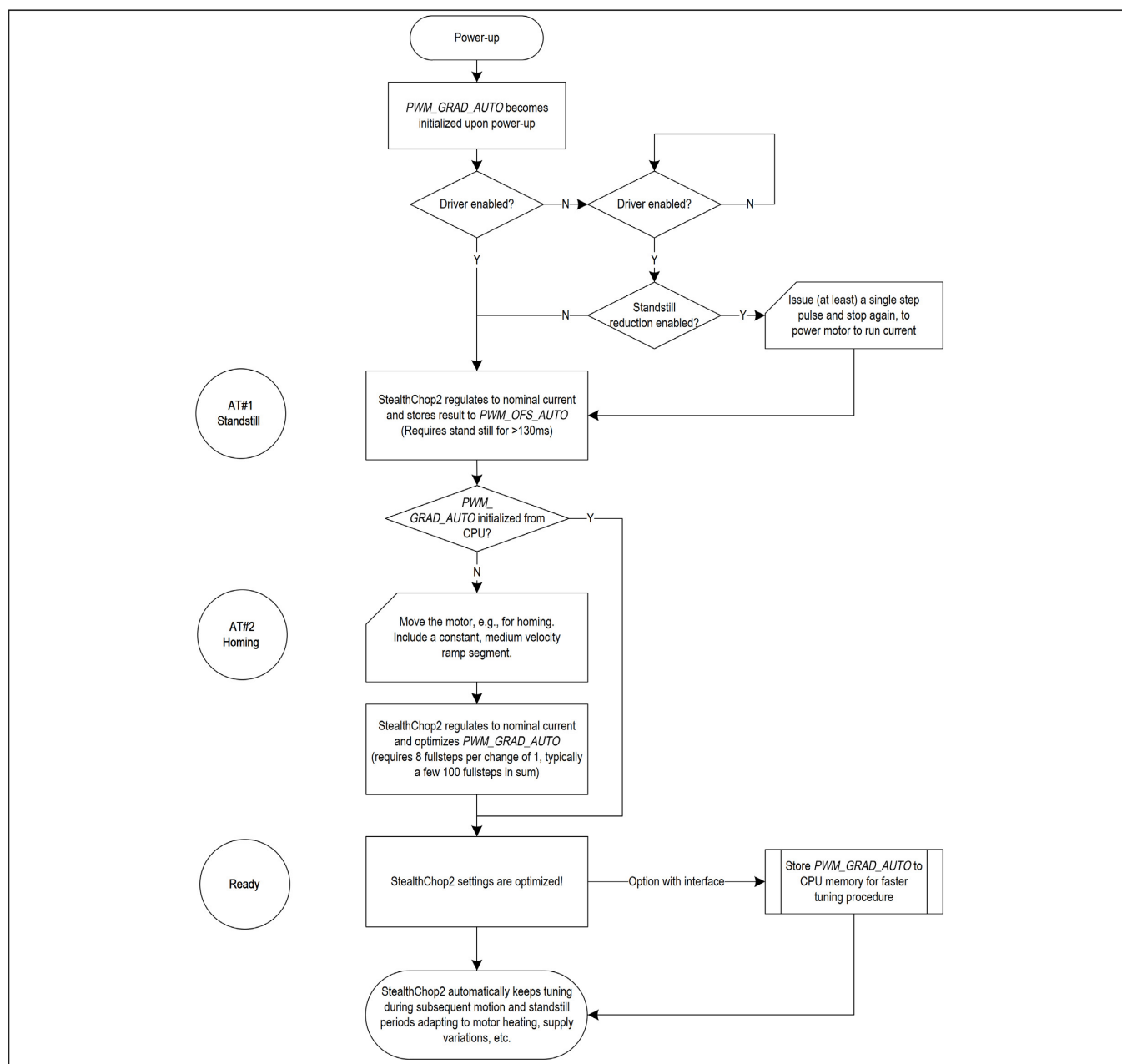


図 9. StealthChop2 の自動調整手順

表 11. StealthChop2 用の PWM 周波数の選択 (太字を推奨)

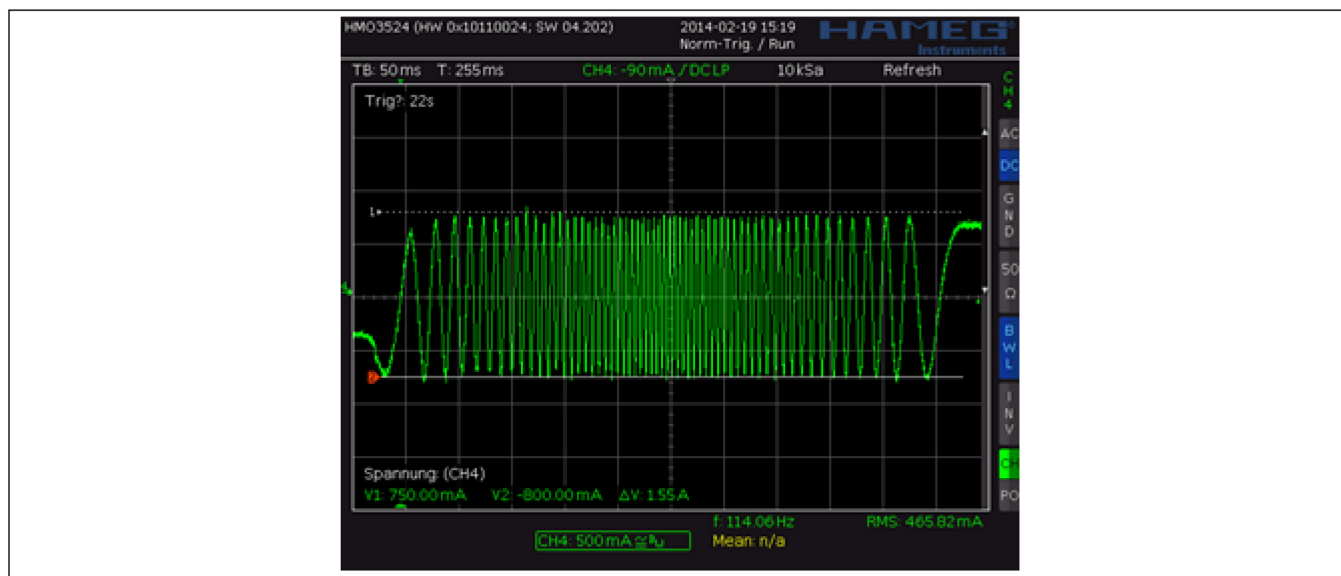
CLOCK FREQUENCY f_{CLK}	PWM_FREQ = %00 $f_{PWM} = 2/1024 f_{CLK}$	PWM_FREQ = %01 $f_{PWM} = 2/683 f_{CLK}$	PWM_FREQ = %10 $f_{PWM} = 2/512 f_{CLK}$	PWM_FREQ = %11 $f_{PWM} = 2/410 f_{CLK}$
20MHz	39.1kHz	58.1kHz	78.1kHz	97.6kHz
18MHz	35.2kHz	52.7kHz	70.3kHz	87.8kHz
16MHz	31.3kHz	46.9kHz	62.5kHz	78.0kHz

表 11. StealthChop2 用の PWM 周波数の選択（太字を推奨）（続き）

12.5MHz (internal)	24.4kHz	36.6kHz	48.8kHz	61.0kHz
10MHz	19.5kHz	29.3kHz	39.1kHz	48.8kHz
8MHz	15.6kHz	23.4kHz	31.2kHz	39.0kHz

StealthChop2 電流レギュレータ

StealthChop2 電圧 PWM モードでは、自動スケーリング機能 ($pwm_autoscale = 1$, $pwm_auto_grad = 1$) がモータ電流を目的の電流設定値に安定化します。自動スケーリングは、AT プロセスの一部として用いられ、また、その後のモータ・パラメータの変化をトラッキングするために用いられます。ドライバはチョッパのオン時間中にモータ電流を測定し、モータ電流が目標電流に一致するように、比例レギュレータを用いて PWM_SCALE_AUTO を安定化します。 PWM_REG は、このレギュレータの比例係数です。基本的に、安定した穏やかなレギュレーション特性を得るために、比例係数はできる限り小さくする必要がありますが、ドライバがモータの目標電流の変動による変化 (V_{REF} の変化など) に素早く反応できるだけの大きさを持っていないとなりません。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 10. StealthChop2 : PWM_REG の設定は良好

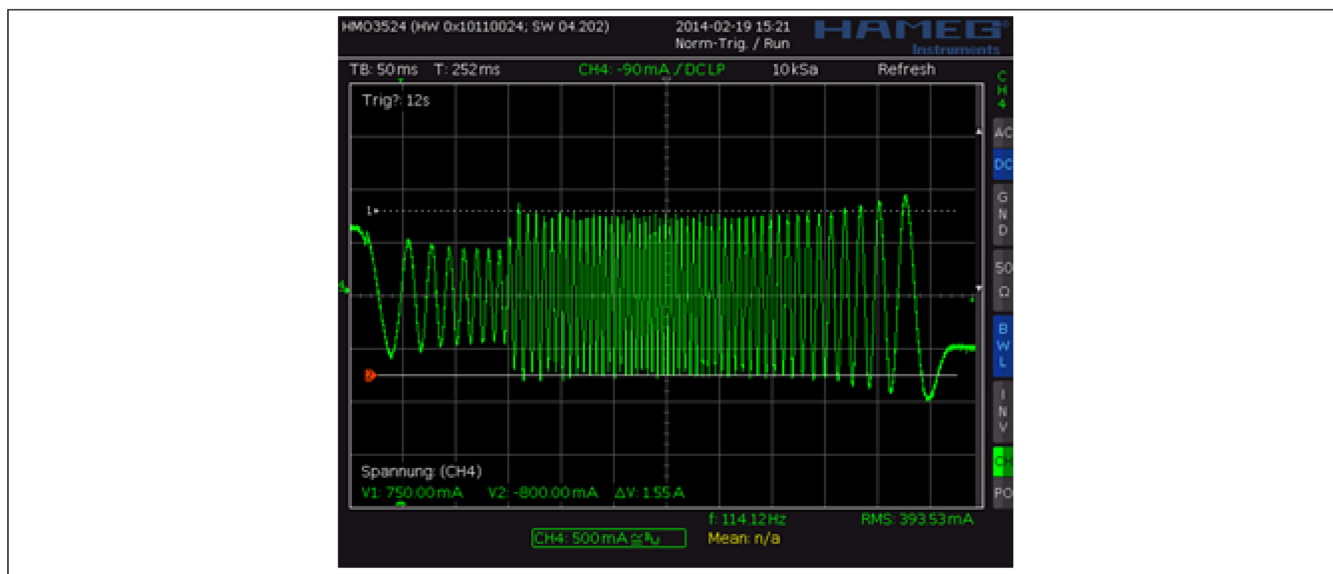


図 11. StealthChop2 : AT#2 時の PWM_REG の設定が過小

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することができます。

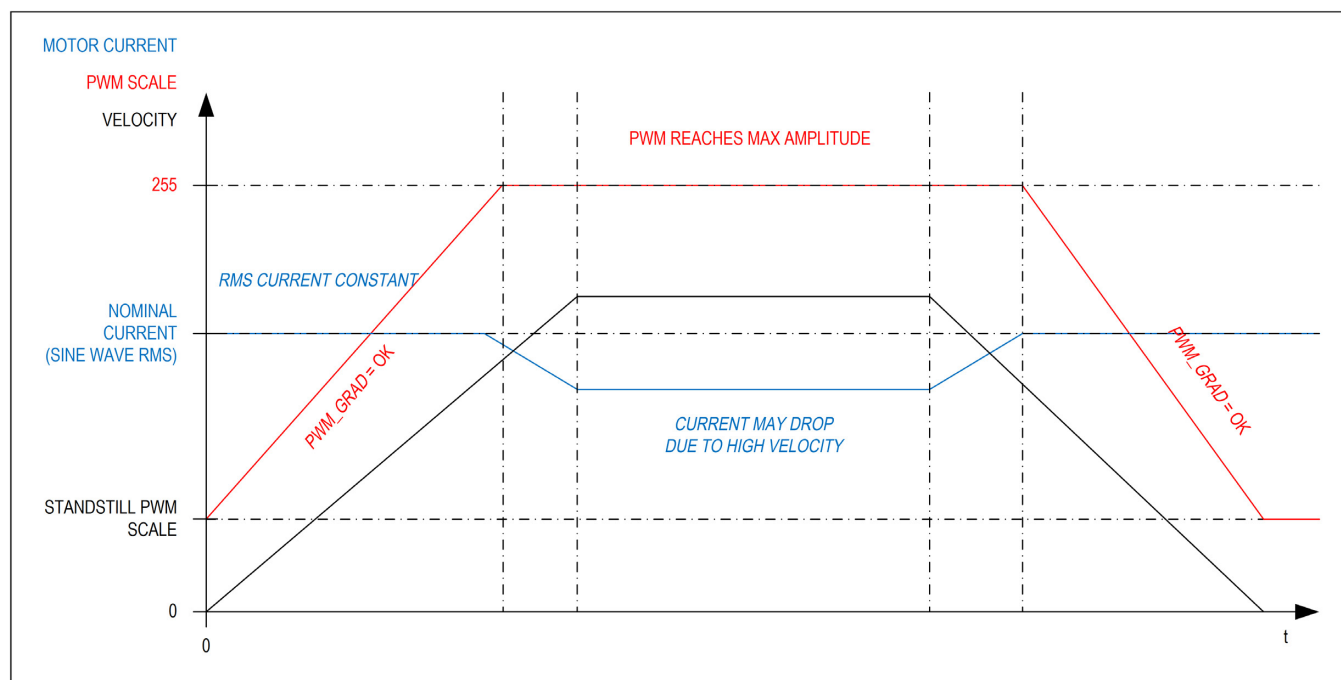


図 12. 正しく決定された PWM_GRAD(_AUTO)および PWM_OFS(_AUTO)

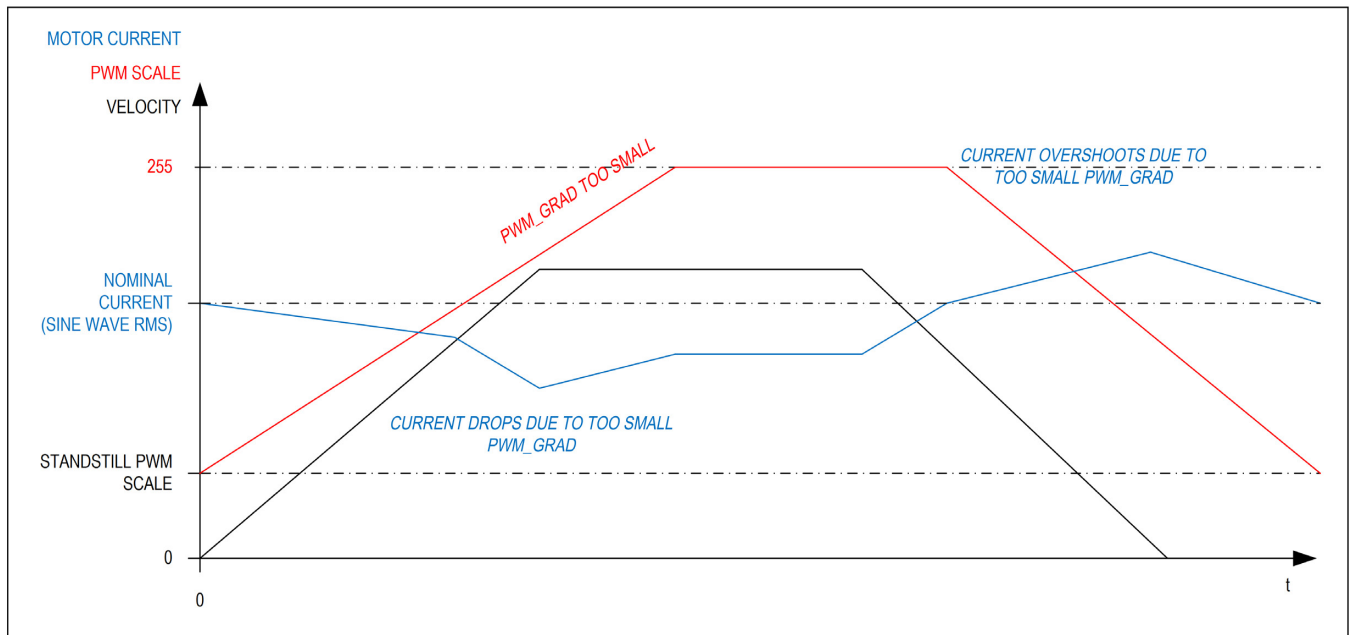


図 13. 過小な PWM_GRAD 設定の例

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・デューティサイクルは、TBL とチョッパ周波数設定値によって設定されるブランキング時間で与えられます。そのため、StealthChop2 自動スケーリング・モードでのモータ固有最小コイル電流は、電源電圧およびチョッパ周波数と共に増加します。ブランキング時間が短いほど、電流制限値は小さくなります。PWM_OFS_AUTO を正しく定めるには、AT#1 時に、稼働電流、GLOBALSCALER、IRUN がレギュレーション範囲内に十分収まっていることが重要です。電流の低下（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。

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

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

$I_{\text{LowerLimit}}$ は、最小公称 IRUN モータ電流設定値のおおよその目安として扱うことができます。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周波数です。

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

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

これは、関連する設定をすべて考慮すると、自動調整のモータ目標電流値は 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。

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

$I_{\text{LOWERLIMIT}}$: 自動調整用の $pwm_meas_sd_enable$ (レジスタ $PWM_CONF[22]$) ビットの設定に応じて下限コイル電流値が適用されます。自動調整フェーズ AT#1 におけるモータ電流は、この下限値より大きくなければなりません。 $I_{\text{LOWERLIMIT}}$ を計算するか、電流プローブを用いてそれを測定してください。 $IRUN$ および 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ルリング

速度ベースのスケールリングは、クロック・サイクル単位で測定した 1 ステップおきの時間に基づき (例えば T_{STEP} に基づき)、StealthChop2 振幅をスケールリングします。このコンセプトは、基本的に、電流測定を必要としません。レギュレーション・ループが不要であるためです。純粋な速度ベースのスケールリングは、 $pwm_autoscale$ を 0 に設定した場合のみ、プログラミングを通じて可能です。基本的な考え方は、モータ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 が求まります。 PWM_OFS の初期値は次式で計算できます。

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

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} ($1/\text{SQRT}(2) \times \text{ピーク値}$) 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。

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

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。TMC2240 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。

$$PWM_SCALE_SUM = PWM_OFS \times \left(\frac{CS_ACTUAL + 1}{32} \right) + PWM_GRAD \times \frac{256}{T_{\text{STEP}}}$$

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、実際の電流スケールリングが考慮されています。 f_{STEP} は、相当する 256 マイクロステップ分解能と等価なマイクロステップ周波数、 f_{CLK} はドライバに共有されるクロック周波数または実際の内部周波数です。

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

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

C_{BEMF} 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。

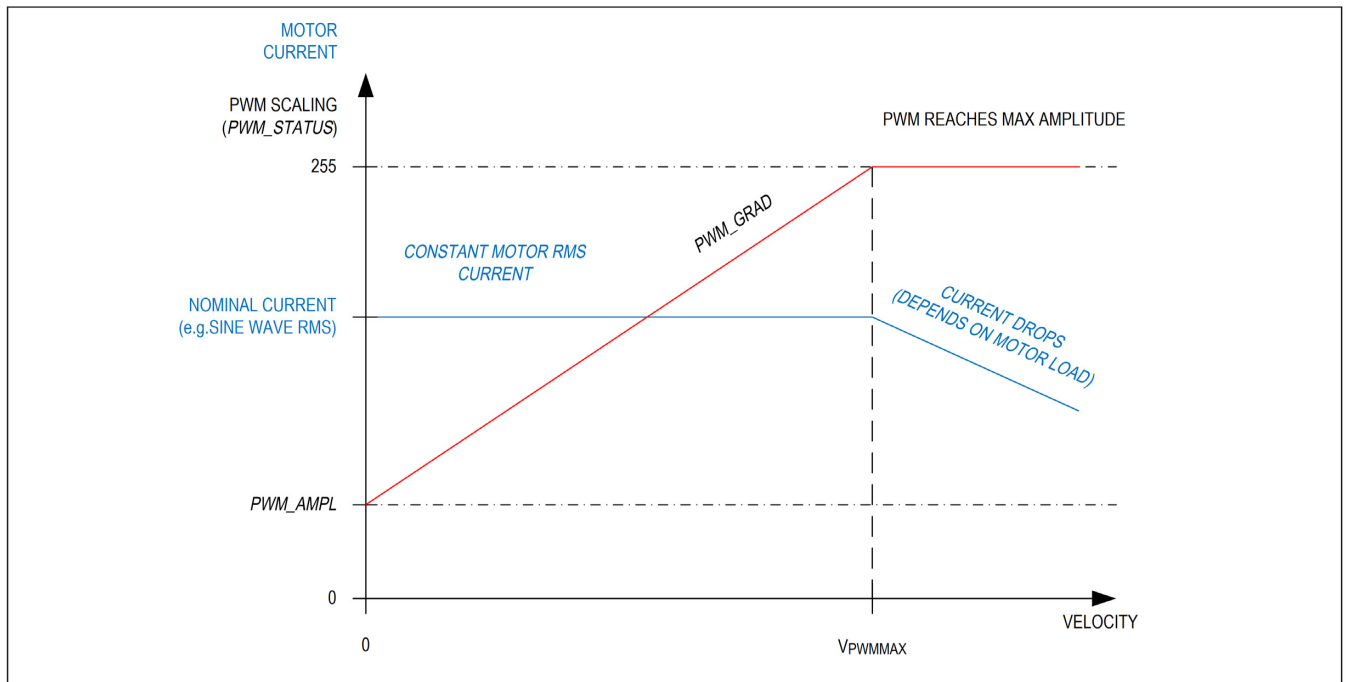


図 14. 速度ベースの PWM スケーリング (pwm_autoscale = 0)

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

モータのバック 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 = 毎秒 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。

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

$I_{COILNOM}$ 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。

$HoldingTorque$ 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を組み合わせるため、TMC2240 では、速度スレッシュホールドに基づき StealthChop2 と SpreadCycle を組み合わせることができます。この場合、StealthChop2 は低速時のみ有効です。

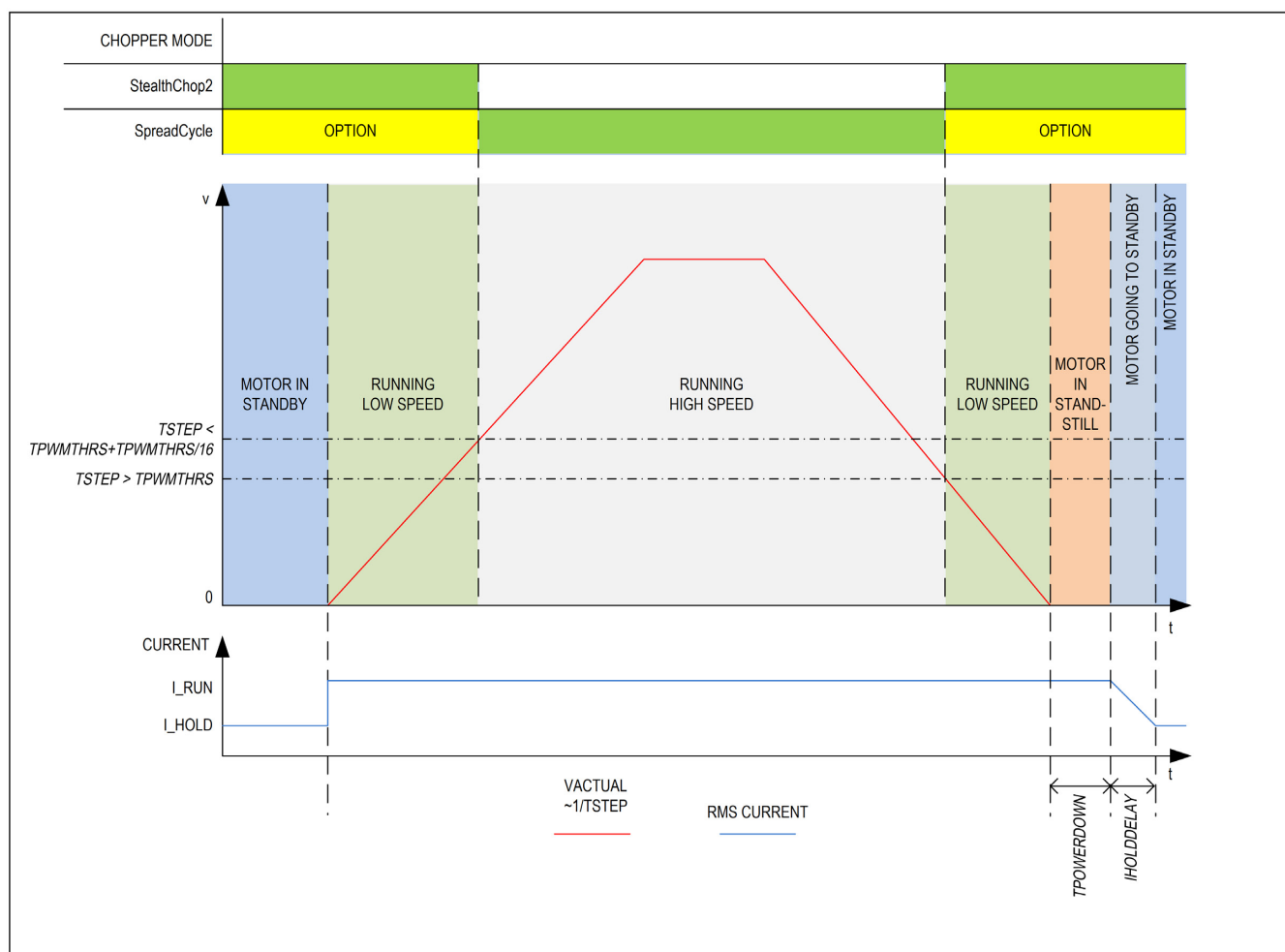


図 15. SpreadCycle への切り替えオプションのための TPWMTHRS

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）が生じる場合すらあります。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ュホー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、チョッパが別のモードに切り替わる位相の間に、速度スレッシュホールドも電源電圧も大きく変化してはなりません。変化すると、モータはステップを喪失するか、あるいは瞬時電流が過大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ったりする可能性があり、その場合は回復ができません。この状況から回復するには、エラー・フラグをクリアし、モータを速度ゼロから再始動します。

初めて **StealthChop2** に切り替える場合、**StealthChop2** が初期停止電流制御を行うことができるよう、モータを停止状態にし、その状態を 128 チョッパ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 の時間にわたり大きく増加します。低速時、バック EMF は電源電圧に比べてごくわずかにすぎないため、短絡検出がトリガされるおそれはありません。

ドライバ段を、モータをサポートする最低電流レンジ (**DRV_CONF** の **current_range**) に切り替えてください。これは過電流スレッシュホールドを 3 つのステップで自動的に適応させ、それによって突然モータがストールした場合のピーク電流を低減します。

開放負荷フラグ

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

- OLA と OLB がセットされない場合、これは、電流レギュレーションが両コイル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。

一方のコイルが開放負荷状態にある場合は、他方のコイルの電流レギュレーションが目標電流を超えて過電流検出トリップ・ポイントに達している可能性があります。これは、電流レギュレーションが測定上の制約のために一定の状況になっていて、目標電流が高い方のコイルの電流だけがレギュレーションされていることによります。重要なアプリケーションの場合は、まず **SpreadCycle** の開放負荷をチェックしてください。

必要に応じて、**SpreadCycle** チョッパ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持続できません。

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 **I_{HOLD}** 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 **I_{HOLD}DELAY** 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 チョップパー・モードに関連したすべてのパラメータを示します。

表 12. 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（速度スレッシュホールドに依存）。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。目標のスレッシュホー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クを制限するための制限値。電流ジャークを低下させるにはこの値を減じます。 デフォルト = 12	0 ... 15	8 ビットの振幅制限の上位 4 ビット
pwm_autoscale	電流測定を用いる自動電流スケーリングをイネーブル。オフの場合、フォワード制御速度ベースのモードを使用します。 デフォルト = 1	0	フォワード制御モード
		1	電流レギュレータを使用する自動スケーリング
pwm_autograd	PWM_GRAD_AUTO の自動調整をイネーブル デフォルト = 1	0	ディスエーブル。代わりにレジスタの PWM_GRAD を使用
		1	イネーブル
PWM_FREQ	PWM 周波数の選択。最小設定にすると良い結果が得られます。チョップパー出力ごとに測定される周波数は、実効的なチョップパー周波数 f_{PWM} の半分です。 デフォルト = 0	0	$f_{\text{PWM}} = 2/1024 f_{\text{CLK}}$
		1	$f_{\text{PWM}} = 2/683 f_{\text{CLK}}$
		2	$f_{\text{PWM}} = 2/512 f_{\text{CLK}}$
		3	$f_{\text{PWM}} = 2/410 f_{\text{CLK}}$

表 12. StealthChop2 を制御するパラメータ（続き）

PWM_REG	ユーザ定義による PWM 振幅レギュレーション・ループの P 係数。pwm_autoscale = 1 の場合に、値が高いほど適応速度も高くなります。 デフォルト = 4	1 ... 15	フルステップあたりの PWM_SCALE_AUTO レギュレータに対する結果は 0.5~7.5 ステップになります。
PWM_OFS	ユーザ定義の PWM 振幅（オフセット）。PWM_OFFS_AUTO の自動調整に対し、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	通常動作
		1	フリーホイーリング
		2	コイルが LS ドライバを介して短絡
		3	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	ドライバ・オフ
		1 ... 15	ドライバ・イネーブル
TBL	コンパレータのブランキング時間。標準的なアプリケーションでは 1 または 2 の設定を選択します。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、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	16 t _{CLK}
		1	24 t _{CLK}
		2	36 t _{CLK}
		3	54 t _{CLK}

SpreadCycle と一般的なチョッパ

StealthChop2 は電圧モード PWM 制御のチョッパであるのに対し、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を用いて制御されます。チョッパは互いに独立に動作します。次の図に、様々なチョッパ・フェーズを示します。

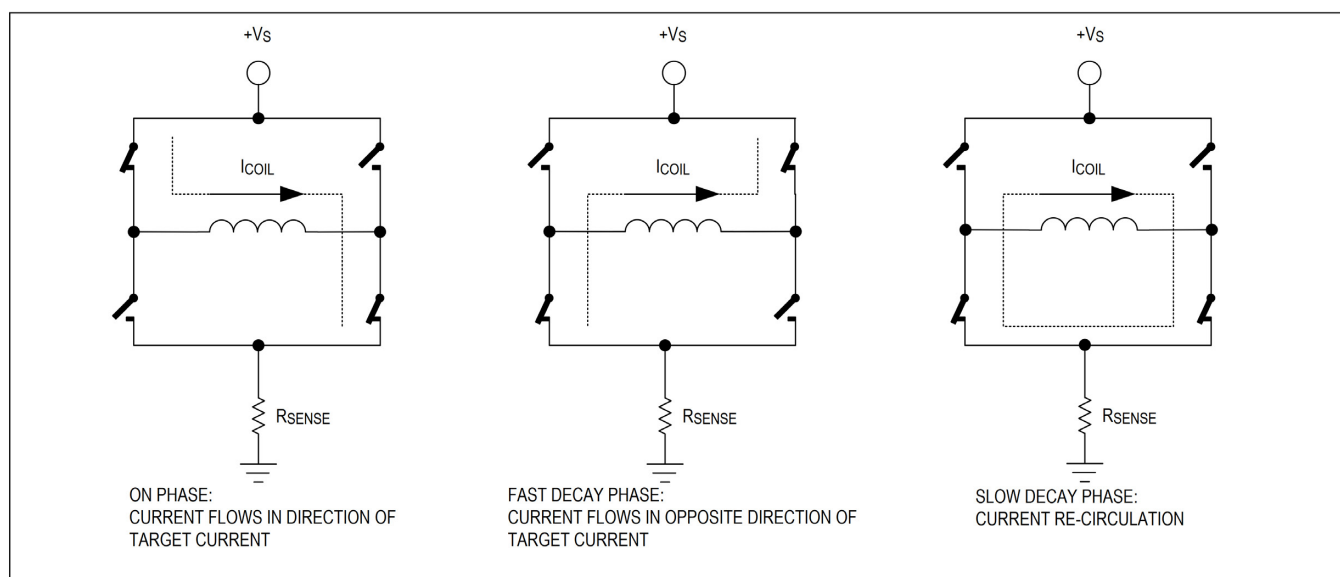


図 16. 代表的なチョッパ減衰フェーズ

電流はオン・フェーズと高速減衰フェーズのみを用いて安定化できますが、低速減衰フェーズを挿入することは、モータの電氣的な損失や電流リップルを低減するうえで重要です。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。ブランキングとは、これらのスパイクをブロックするためにコンパレータへの入力がマスクされている時間のことです。

使用できるサイクルごとのチョッパ・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・アルゴリズムと、定評のある定オフ時間チョッパ・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。

チョッパ周波数は、チョッパ・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 13. SpreadCycle および一般的な定オフ時間チョッパを制御するパラメータ

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、最大チョッパ周波数も制限されます。 StealthChop2 での動作の場合、このパラメータは使いませんが、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。 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	0	チョッパ・オフ
		1...15	オフ時間の設定値 $N_{CLK} = 24 + 32 \times TOFF$ （1 に設定すると 24 クロックの最小ブランキング時間で動作します）

表 13. SpreadCycle および一般的な定オフ時間チョッパを制御するパラメータ（続き）

TBL	コンパレータのブランキング時間を選択します。この時間は、スイッチング・イベントとリングングの時間を余裕を持ってカバーする必要があります。ほとんどのアプリケーションでは、1 または 2 の設定で十分です。フィルタ・ネットワークを用いるなど、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。	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：内部クロックまたは 13MHz 以下の外部クロックを適用する場合に使用できます。
		2	36 t _{CLK}
		3	54 t _{CLK}
chm	チョッパ・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

SpreadCycle（特許取得済み）チョッパ・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・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・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あたり 2 つのブランキング時間によって、チョッパ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・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：

- 目標チョッパ周波数：25kHz
 - $t_{OFF} = 1 / 25\text{kHz} \times 50 / 100 \times 1 / 2 = 10\mu\text{s}$
 - 前提条件：2 つの低速減衰サイクルが全チョッパ・サイクル時間の 50%を使用
- TOFF* の設定に関しては、次のようになります： $TOFF = (t_{OFF} \times f_{CLK} - 12) / 32$
- クロックが 12MHz の場合、*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は、この電流リップルが、モータの抵抗性損失を原因とする電流リップルより大きいことが必要です。これによって、チョッパは、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、チョッパ周波数の減少ももたらします。そのため、ヒステリシスの設定値を高くすると、チョッパ周波数は低下します。モータのインダクタンスにより、チョッパがモータ電流の変化に追従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、ブランキング時間より長くなくてはなりません。これは、ブランキング時には電流コンパレータがディスエーブルされるためです。

最適な設定値を得るための最も簡単な方法は、低ヒステリシス設定（例えば、 $HSTRT=0$ 、 $HEND=0$ ）から始め、モータ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 フルスステップ～400 フルスステップ）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、コイル抵抗も低いからです。そのため、ヒステリシスのデフォルト値として低い値から中程度の値を選択すると（例えば、実効ヒステリシス=4）、通常、ほとんどのアプリケーションに適します。この設定は、モータ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キング時間よりわずかに長くなると、設定は最適なものとなります。これ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・デクリメンタ（HDEC）は、ヒステリシス値を段階的に 16 システム・クロックごとにデクリメントすることで、両方の設定の間の補間を行います。ヒステリシスは、各チョッパー・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、周波数が低すぎるようになっても、大振幅かつ低電源電圧の状態で安定化します。これにより、周波数が可聴範囲に達するのが防止されます。

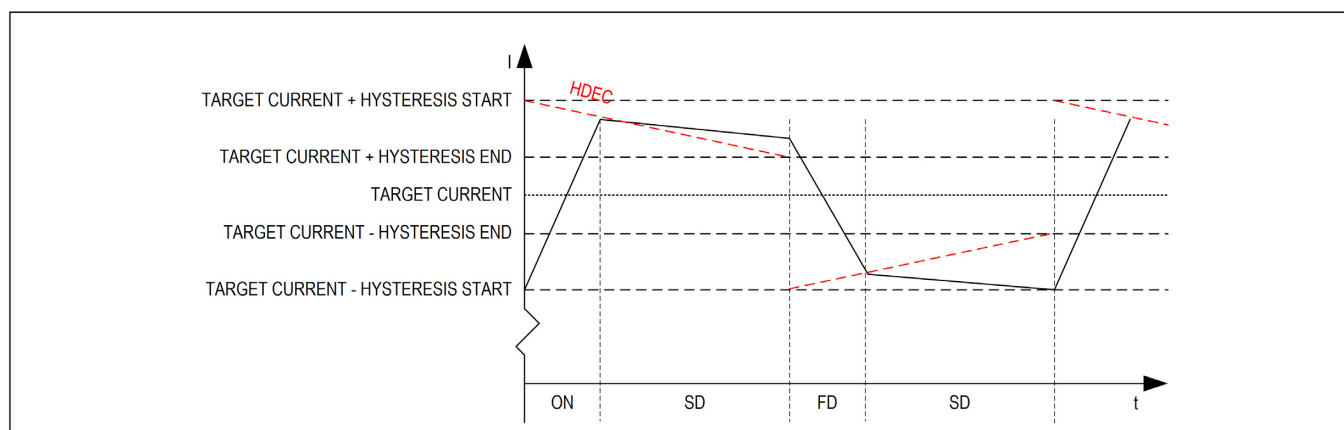


図 17.1 チョッパー・サイクルの間のコイル電流を示す、SpreadCycle チョッパーの概略図

表 14. 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	-3...-1 : $HEND$ は負 0 : $HEND$ はゼロ
		4...15	1...12 : $HEND$ は正

HSTRT = 0 かつ HEND = 0 の場合でも、TMC2240 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

一般的な定オフ時間チョッパを、SpreadCycle の代替として用いることができます。完全に調整すれば、このチョッパでも良好な結果が得られます。外部検出抵抗を使わない RDSon 電流検出と組み合わせることで、このチョッパ・モードでは可聴のハイピッチ・チョッパ・ノイズに関する利点を得ることができます。

従来型の定オフ時間チョッパでは、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・コンパレータによって決定される一方、高速減衰時間はドライバがサイン波の立下がり勾配に追従できるだけの長さであることが必要です。しかし、これは過大なモータ電流リップルおよび消費電力の原因となるため、長すぎたはけません。これを調整するには、オシロ・スコープを用いるか、様々な速度でモータ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。

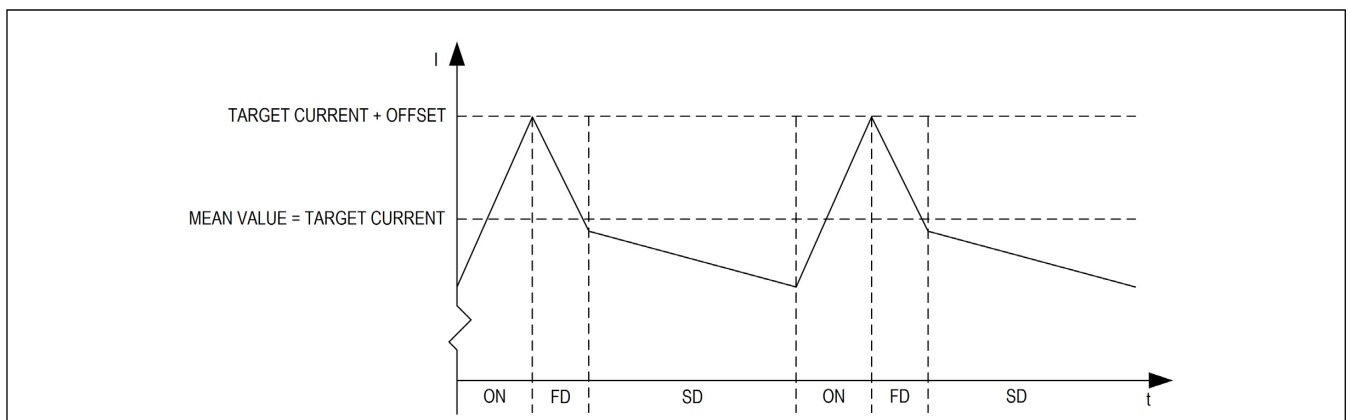


図 18. コイル電流を表すオフセットを持つ一般的な定オフ時間チョッパ

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。

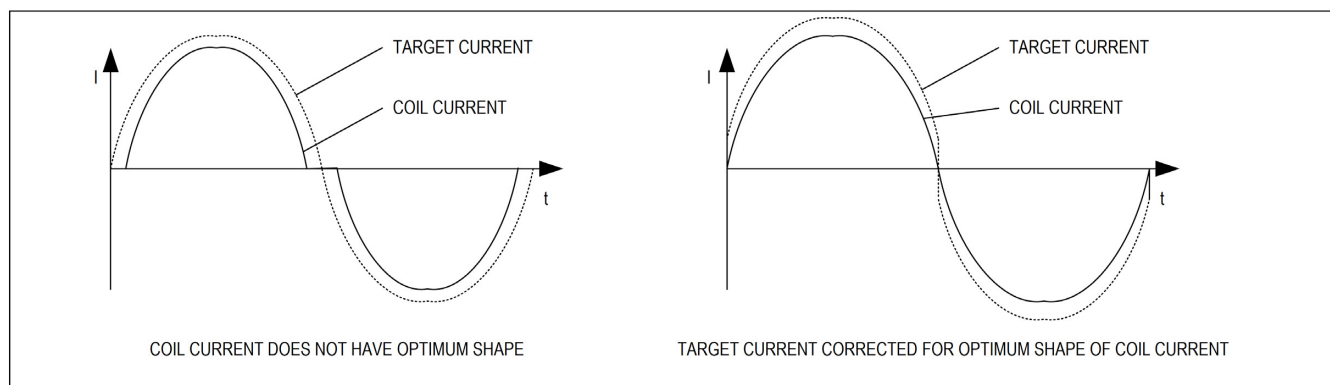


図 19. 一般的なチョッパのゼロ交差とサイン波オフセットを使用した補正

表 15. 定オフ時間チョッパ・モードを制御するパラメータ

パラメータ	説明	設定	コメント
TFD (fd3 and HSTRT)	高速減衰時間設定。CHM=1 の場合、これらのビットが各チョッパ・サイクルの高速減衰の部分を制御します。	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。	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。	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

TMC2240 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

TMC2240 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 16. 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。内蔵のサイン波テーブルから取得したコイル電流値をスケーリングします。高精度のモータ動作のためには、電流スケーリング・ファクタを 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

表 16. 帰還抵抗の推奨値

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 IFS 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に、3 種類のフルスケール電流レンジが設定可能です。

これは、できる限り最高の電流制御分解能を活用するために必要です。

そのため、IREF 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 IFS 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 IFS を、IREF ピンに接続された RREF 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 KIFS 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 RREF の範囲は 12k Ω ~60k Ω です。

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

表 17. IFS フルスケール・ピーク・レンジの設定値 (RREF = 12k Ω の場合の例)

REGISTER CONFIG DRV_CONF bits 1..0	KIFS (A $\times$ k Ω)	MAX. FS SETTING (PEAK)	TYPICAL RDS(ON) (HS + LS)	NOTES
11	36	3A	0.23 Ω	Optimized efficiency and extended operating range up to 3A _{FS} .
10	36	3A	0.23 Ω	Optimized efficiency and extended operating range up to 3A _{FS} .
01	24	2A	0.27 Ω	Reduced operating range up to 2A _{FS} . When high accuracy at lower current is required.

表 17. I_{FS} フルスケール・ピーク・レンジの設定値 (R_{REF} = 12kΩ の場合の例) (続き)

00 (default)	11.75	1A	0.40Ω	Reduced operating range up to 1A _{FS} . 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 18. 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
16	1,59	1,59	1,06	0,52
18	1,41	1,41	0,94	0,46
22	1,16	1,16	0,77	0,38
24	1,06	1,06	0,71	0,35
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
48	0,53	0,53	0,35	0,17
56	0,45	0,45	0,30	0,15

速度ベースのモード制御

TMC2240 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・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。

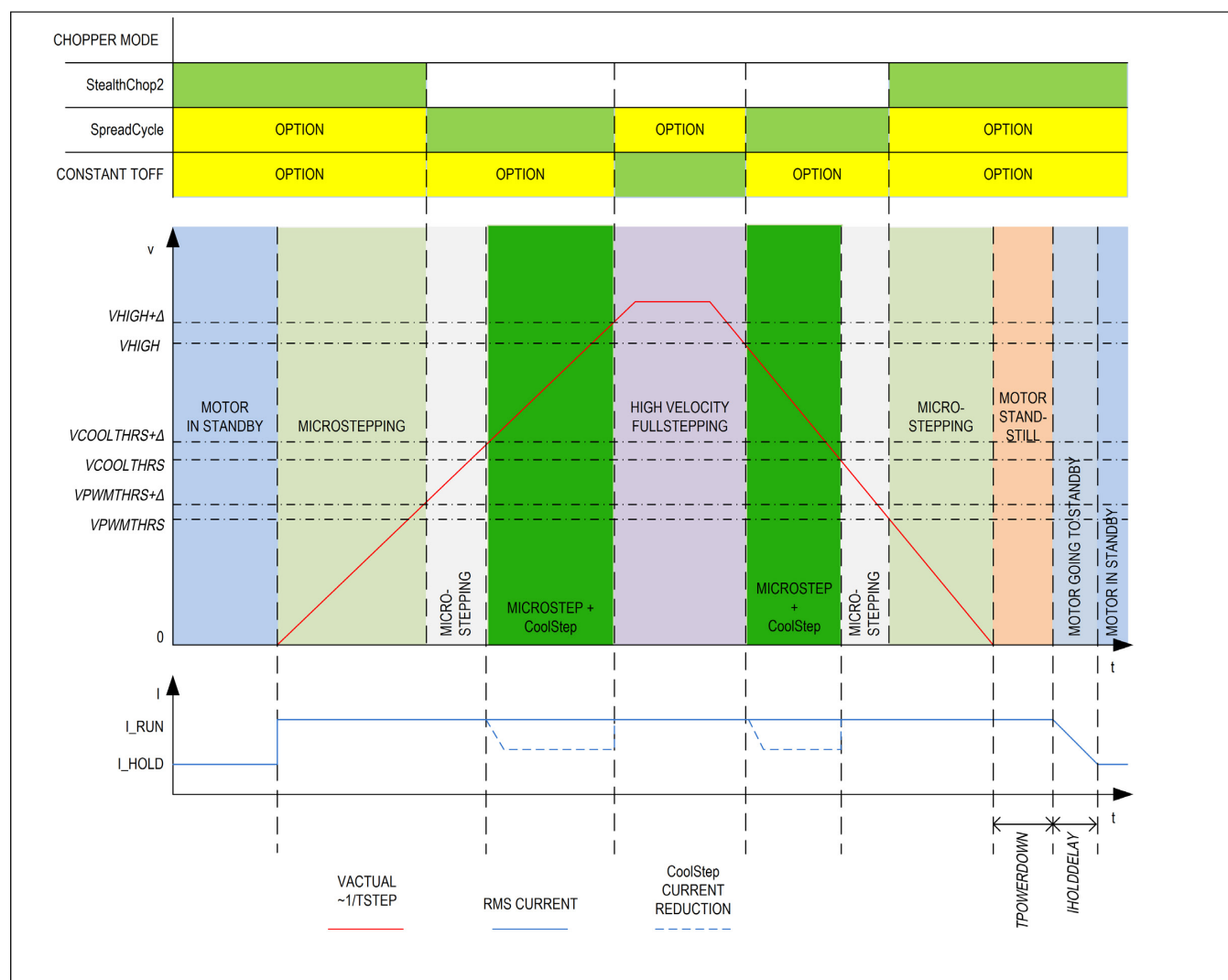


図 20. 速度依存モードの選択

図に、使用可能なすべてのスレッシュホールドと必要な順序を示します。VPWMTHRS、VHIGH、VCOOLTHRS は、TPWMTHRS、THIGH、TCOOLTHRS の設定によって決まります。速度は、各 2 ステップ・パルス間の時間間隔 $TSTEP$ で記述されます。それにより、外部ステップ源を用いた場合でも速度を決定できます。 $TSTEP$ は、常に 256 マイクロステップに正規化されます。したがって、マイクロステップの分解能が変化した場合でも、スレッシュホールドをそれに合わせて変化させる必要はありません。スレッシュホールドは、マイクロステップの設定値とは無関係に、同じモータ速度を表します。 $TSTEP$ は、これらのスレッシュホールド値と比較されます。 $TSTEP$ の測定でジッタが生じた場合に、比較した結果が連続的に切り替わるのを避けるために、 $1/16TSTEP$ または $1/32TSTEP$ のヒステリシスが適用されます。スイッチング速度の上限値は、スレッシュホールドの設定値より $1/16$ または $1/32$ だけ高くなります。モータ電流は、停止フラグ *stst* に応じて、動作レベルとホールド・レベルにプログラムできます。

自動速度スレッシュホールドを用いるとアプリケーションを様々な速度レンジに調整できます。CoolStep などの機能は、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 19. 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
<i>stst</i>	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 2 ²⁰ クロックです。	0/1	ステータス・ビット、読み出し専用
<i>TPOWER DOWN</i>	これは、モータの停止 (<i>stst</i>) 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	0...255	2 ¹⁸ t _{CLK} の倍数で表される時間
<i>TSTEP</i>	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/t _{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。	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
<i>TPWMTHRS</i>	<i>TSTEP</i> ≥ <i>TPWMTHRS</i> • StealthChop2 PWM モードが有効化されます (設定されている場合)	0...1048575	StealthChop2 の動作上限速度スレッシュホールドを制御するための設定値
<i>TCOOLTHRS</i>	<i>TCOOLTHRS</i> ≥ <i>TSTEP</i> ≥ <i>THIGH</i> : • CoolStep がイネーブルされます (設定されている場合) • StealthChop2 電圧 PWM モードは無効化されます <i>TCOOLTHRS</i> ≥ <i>TSTEP</i> • ストール出力信号がイネーブルされます (設定されている場合)	0...1048575	CoolStep と StallGuard2 の動作下限速度スレッシュホールドを制御するための設定値
<i>THIGH</i>	<i>TSTEP</i> ≤ <i>THIGH</i> : • CoolStep は無効化されます (モータは通常電流スケールで動作) • StealthChop2 電圧 PWM モードは無効化されます • <i>vhighchm</i> が設定されていると、チョップパーは TFD = 0 (低速減衰のみの定オフ時間) の場合に <i>chm</i> = 1 に切り替わります。 • チョップパー同期はオフに切り替わります (SYNC = 0) • <i>vhighfs</i> 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、フルステップ・ストール検出に切り替わります。	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ュホールドを制御するための設定値
<i>small_hysteresis</i>	<i>TSTEP</i> (下限速度スレッシュホールド) と (<i>TSTEP</i> × 15/16) - 1 または (<i>TSTEP</i> × 31/32) - 1 (上限速度スレッシュホールド) に基づくステップ周波数比較のヒステリシス	0	ヒステリシスは 1/16
		1	ヒステリシスは 1/32
<i>vhighfs</i>	このビットは、 <i>VHIGH</i> 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。	0	フルステップへの切替えなし
		1	高速時はフルステップ
<i>vhighchm</i>	このビットは、 <i>VHIGH</i> を超えた場合に <i>chm</i> = 1 および <i>fd</i> = 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。 <i>vhighfs</i> = 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、チョップパー周波数が 2 倍になるのを防止するため、 <i>TOFF</i> 設定は高速動作時に自動的に 2 倍になります。	0	チョップパー・モードの変化なし
		1	高速時は一般的な定 <i>Toff</i> チョップパー
<i>en_pwm_mode</i>	StealthChop2 電圧 PWM イネーブル・フラグ (速度スレッシュホールドに依存)。停止時にのみオフ状態からオン状態に切り替わります。	0	StealthChop2 不使用
		1	VPWMTHRS 未満では StealthChop2

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、TMC2240 には 2 つの基本チョップパー・モードに対応する、2 種類の StallGuard センサーレス負荷検出機構があります。StallGuard2 は SpreadCycle 動作時に機能し、StallGuard4 は StealthChop2 動作に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 のスレッシュホールド (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。

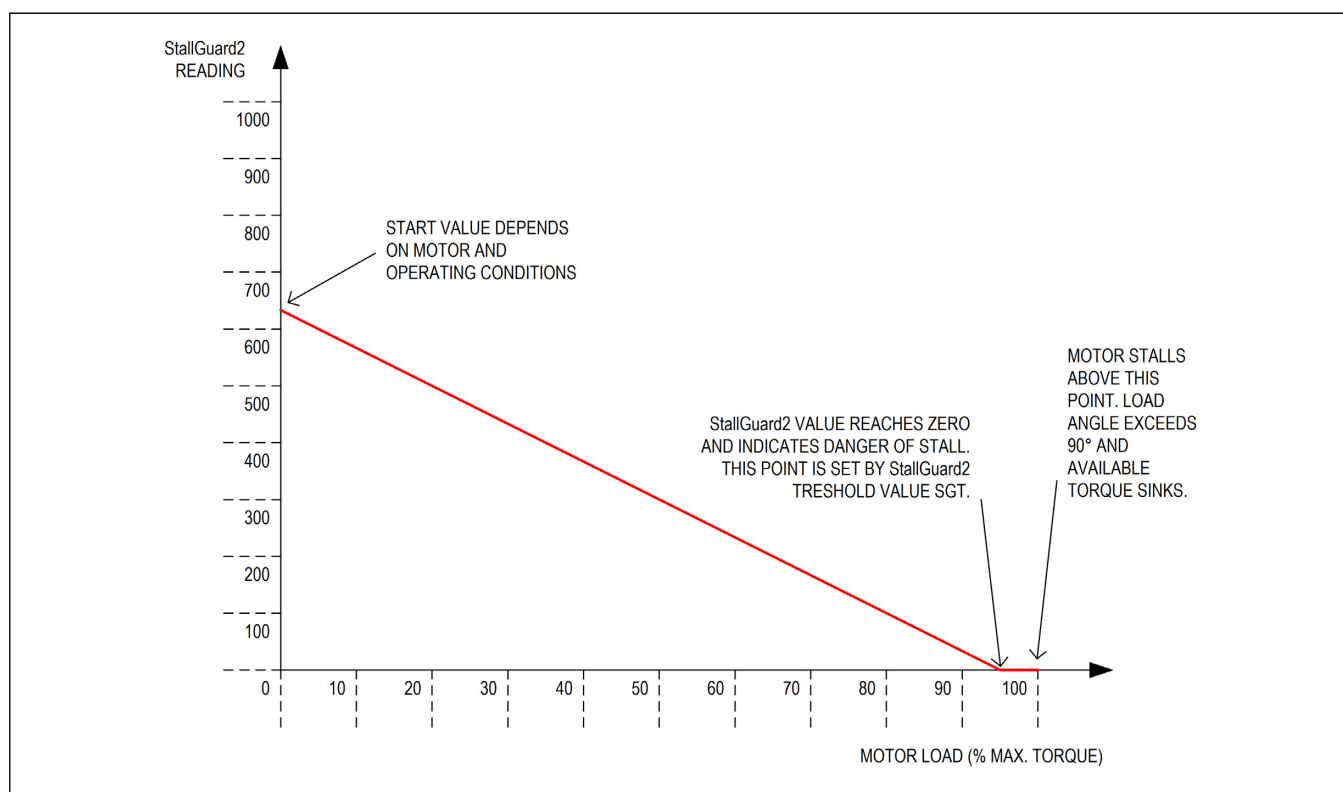


図 21. StallGuard2 の動作原理

表 20. StallGuard2 関連のパラメータ

パラメータ	説明	設定	コメント
SGT	この符号付きの値は、ストール検出用の StallGuard2 スレッシュホールド・レベルを制御し、読出し用に最適な測定範囲を設定します。値を小さくすると感度が高くなります。ゼロはほとんどのモータ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氣的周期（4 フルステップ）あたり 1 測定に減少します。	0	標準モード
		1	フィルタリング・モード
ステータス・ワード	説明	範囲	コメント
SG_RESULT	これは StallGuard2 の結果です。読出し値が大きいと機械的負荷は小さいことを示します。読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 の更新レートおよびフィルタ

StallGuard2 の測定値 *SG_RESULT* は、モータのフルステップごとに更新されます。ストールでは必ず 4 つのフルステップが失われるので、この方法によりストールを十分確実に検出できます。実際のアプリケーション、特に、CoolStep を用いる場合は、より高精度の測定を行うことの方がフルステップごとに更新を行うより重要となる場合があります。機械的な負荷は決してステップごとに瞬時に変化することはないためです。こうしたアプリケーションでは、*sfilt* ビットによって、4 回の負荷測定にわた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 相の磁石のミスアライメントによるモータ構造のばらつきを補償します。負荷の増加に対して短時間で対応する必要がある場合や、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・スレッシュホールドを決定する必要があります。そのため、モータをストールさせずに駆動できる最大負荷を決定する必要があります。同時に、この負荷での *SG_RESULT* の値 (0~100 の範囲のいずれかの値) をモニタします。ストール・スレッシュホールドは、パラメータの浮遊を許容できるよう、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 の読出し値が得られるように SGT 値を設定している場合は、ストールは自動的に検出されてモータ停止を指示します。ステップ・ロスの原因となるステップの瞬間には、最小の読出し値が示されます。ステップ・ロスの後、モータは振動し、*SG_RESULT* の読出し値は増加します。

StallGuard2 を使用したホーミング

線形駆動のホーミングでは、モータをハード・ストップの方向に移動させることが必要です。StallGuard2 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 をセットアップしたらハード・ストップ方向への動作を開始し、*diag0_stall* または *diag1_stall* のどちらかを使って外部コントローラへストール状態を示すために、これらの診断出力を設定します。ストールが検出された場合はコントローラがモータを停止します。ストップ条件は、DRV_STATUS のフラグ STALLGUARD によっても示されます。

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づくにつれ、値はモータ固有の低い値まで減少します。これは、ローターのコイルと磁石の磁界が 90° の負荷角になることに対応します。また、これは、モータにとって最もエネルギー効率の高い動作点でもあります。

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

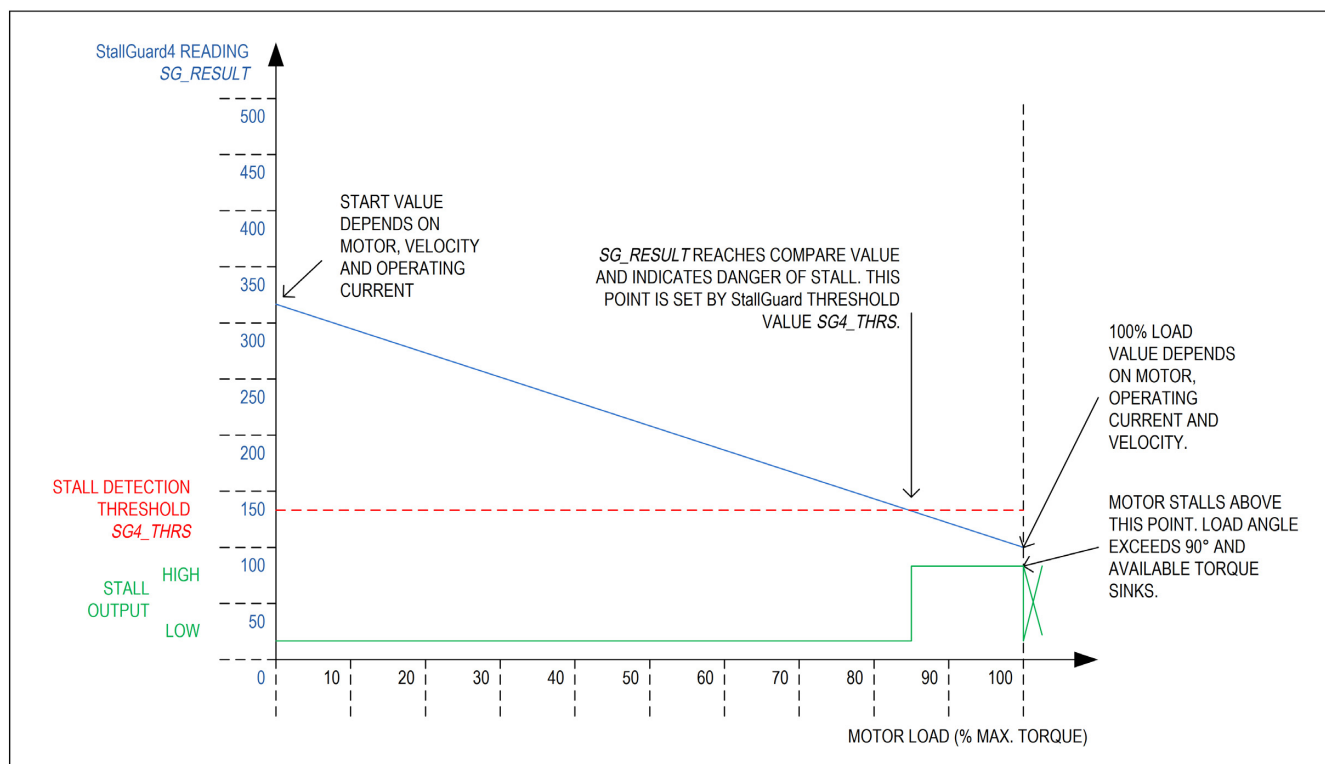


図 22. StallGuard4 の動作モード

表 21. 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 スレッシュホールドを制御します。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
ステータス・ワード	説明	範囲	コメント
SG4_RESULT	これは StallGuard4 の結果です。読出し値が大きいと機械的負荷は小さいことを示します。読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、実際のチョップパルス・モードのようなイネーブル条件や VCOOLTHRS のような速度スレッシュホールドとは独立に生成されます。結果は SG4_IND_x の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

表 21. StallGuard4 関連のパラメータ（続き）

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
sg4_filt_en	0：フィルタを使用する動作、SG4_RESULT フルスステップごとに更新 1：フィルタリング動作、SG4_IND_0...3 を使用可能、SG4_RESULT は最後の 4 つの SG4_IND_x 測定値の平均を示します	0 1	0：フィルタ・オフ 1：フィルタリング動作、SG4_IND の値を使用可能
sg_angle_offset	このフラグは、StealthChop2 の位相ラグを決定し SpreadCycle の電圧制御動作から電流制御動作への切り替わり時の位相不連続を補償するために SG4_RESULT を用いることで、StealthChop2 と SpreadCycle の間の最適化された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 のスレッシュホールド SG4_THRS を特定のモータ・タイプおよび動作条件に合わせて調整するための最も簡単な方法は、実際のアプリケーションでインタラクティブに調整を行うことです。

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。

1. モータをアプリケーションの通常の動作速度で動作させ、SG4_RESULT をモニタします。
2. モータに印加する機械的負荷を徐々に増加します。モータがストールする前の SG4_RESULT の最低値をチェックします。この値を SG4_THRS の開始値として用います（この値の半分を適用）。
3. 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。
4. 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 ≥ 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単一パルスに反応することが必要です。StallGuard が良好な結果をもたらすことのできる下限速度スレッシュホー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ので、この方法によりストールを十分確実に検出できます。

StallGuard4 には測定用に次の 2 つのオプションがあります。

1. sg4_filt_en = 0：単一測定、各フルステップ後に更新、1 フルス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。
2. sg4_filt_en = 1：このモードでは、次に示す 4 つの個別の信号が生成されます。コサイン波（コイル A）の立下がり 0 遷移時の SG4_IND_0、コサイン波の立上がり 0 遷移時の SG4_IND_1、サイン波（コイル B）の立下がり 0 遷移時の SG4_IND_2、サイン波の立上がり 0 遷移時の SG4_IND_3 です。SG4_RESULT の実際の値は、4 つのすべての測定値の平均値で、フルステップごとに更新されます。そのため、各フルステップが全体の結果に及ぼす影響は 25% だけです。このモードは、柔らかい障害物がある場合の検出、あるいは、不明確なモータでの CoolStep の使用に最適です。フィルタリング・モードでは、負荷の突然の増加（ハード・モータ障害）に対する感度は減少します。

モータ・ストールの検出

モータのストールを確実に検出するには、特定の *SG4_THRS* 設定値と特定のモータ速度または速度範囲を用いてストール・スレッシュホールドを決定する必要があります。更に、モータ電流の設定値には一定の影響力があり、一度最適値を決定したら変更はすべきではありません。そのため、所定のアプリケーションに対し、モータをストールさせずに駆動できる最大負荷を決定する必要があります。同時に、この負荷での *SG4_RESULT* をモニタします。ストール・スレッシュホールドは、パラメータの浮遊を許容できるよう、動作制限範囲に十分収まる値とします。より精緻に行われた評価も、固定スレッシュホールドと比較する場合より、*SG4_RESULT* の変化の影響を受ける可能性があります。そのため、絶対値に影響するいくつかの特定の効果が除外されます。

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を行い、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 22. CoolStep の重要パラメータ

パラメータ	説明	範囲	コメント
SEMIN	下限スレッシュホールドを設定する 4 ビットの符号なし整数。 <i>SG_RESULT</i> がこのスレッシュホールドを下回る（負荷が大きいかを示す）場合、CoolStep は両方のコイルへの電流を増加します。4 ビットの <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	スレッシュホールドは <i>SEMIN</i> × 32
SEMAX	上限スレッシュホールドを制御する 4 ビットの符号なし整数。このスレッシュホールド以上の値の <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。上限スレッシュホールドは、(<i>SEMIN</i> + <i>SEMAX</i> + 1) × 32 です。	0...15	スレッシュホールドは(<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。

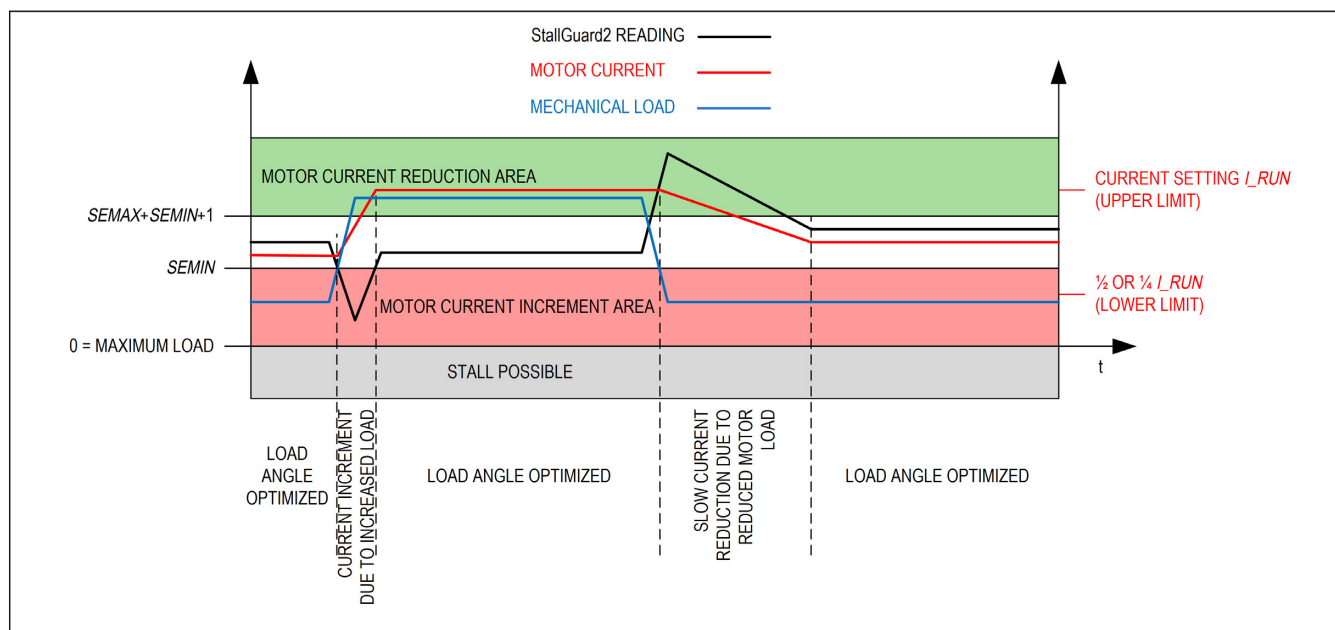


図 23. モータ電流を負荷に適合させる CoolStep

表 23. 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ュホールド未満の新しい 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、上限スレッシュホールドを上回る StallGuard2 / StallGuard4 読み出し値の数を設定します。	0...3	デクリメントあたりの StallGuard2 測定値の数 : 32、8、2、1
SEIMIN	IRUN 電流設定値をスケリングして、CoolStep 動作の下限モータ電流制限を設定します。 StealthChop2 を使用する場合、特に 25% まで低減することが必要な場合は、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。	0	0 : IRUN の 1/2 (StealthChop と使用するとき は IRUN ≥ 16 とする)
		1	1 : IRUN の 1/4 (StealthChop と使用するとき は IRUN ≥ 28 とする)
TCOOLTHRS	CoolStep に切り替えるための下限速度スレッシュホールド。この速度を下回ると CoolStep はディスエーブルされます。StallGuard2 が安定な結果を与える速度範囲の下限に合わせてください。 ヒント : VCOOLTHRS を VMAX と同じ値に設定することで、加速フェーズおよび減速フェーズ時に CoolStep をディスエーブルするよう修正できます。	1...2 ²⁰ - 1	スレッシュホールド値を TSTEP と比較することで、下限 CoolStep 速度を指定します。
THIGH	CoolStep 用の上限速度スレッシュホールド値。この速度を超えると CoolStep はディスエーブルされます。StallGuard2 / 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 SEIMIN で指定される IRUN の一部です。	0...31	1/32、2/32、... 32/32

CoolStep の調整

CoolStep を SpreadCycle と併用して調整する前に、まず StallGuard2 のスレッシュホールド・レベル *SGT* を調整します。これは、負荷測定値 *SG_RESULT* の範囲に影響します。CoolStep では *SG_RESULT* を使用して、モータを+90°の最適負荷角度付近で動作させます。CoolStep は、StealthChop2 と共に、*SG4_RESULT* を使用します。このモードでは、レベル調整は *SEMIN* を介して行われます。

電流のインクリメント速度は *SEUP* で指定され、電流のデクリメント速度は *SEDN* で指定されます。これらは、異なる応答を必要とする可能性のある異なるイベントでトリガされるため、個別に調整できます。これらのパラメータのエンコーディングでは、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。下限スレッシュホールドに達することは、高速応答を必要とする可能性のある、より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、上限スレッシュホー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。

注意：

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

StallGuard2 を CoolStep と組み合わせる方が望ましい場合は、必要に応じ CoolStep の下限閾値 *SEMIN* を増やしてみてください。

応答時間

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

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ュホー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、時間が短いので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、ドライバには下限速度スレッシュホー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。このスレッシュホールド未満では、*IRUN* または *IHOLD* を介した通常の電流設定が有効です。上限スレッシュホールドは *VHIGH* の設定により与えられます。速度制限値 *VHIGH* および *VCOOLTHRS* は、*THIGH* および *TCOOLTHRS* の設定値で決まります。

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

診断出力

外部モーション・コントローラによる動作では、多くの場合、特定のステッパ・モータ・ドライバ状態に対して迅速に反応することが求められます。したがって、DIAG 出力は、STEP/DIR インターフェイスを補完する様々なリアルタイム情報のセットを提供します。このセットは様々な設定が可能です。

DIAG0 と DIAG1 に出力される情報はどちらも選択できる他、出力のタイプ（ロー・アクティブ・オープン・ドレイン - デフォルト設定、またはハイ・アクティブ・プッシュプル）も選択できます。ドライバのリセットを決定するために、リセット状態時の DIAG0 は、ローにプルダウンすることによって常にパワーオン・リセット状態を示します。使用できる信号と制御ビットを下の図に示します。

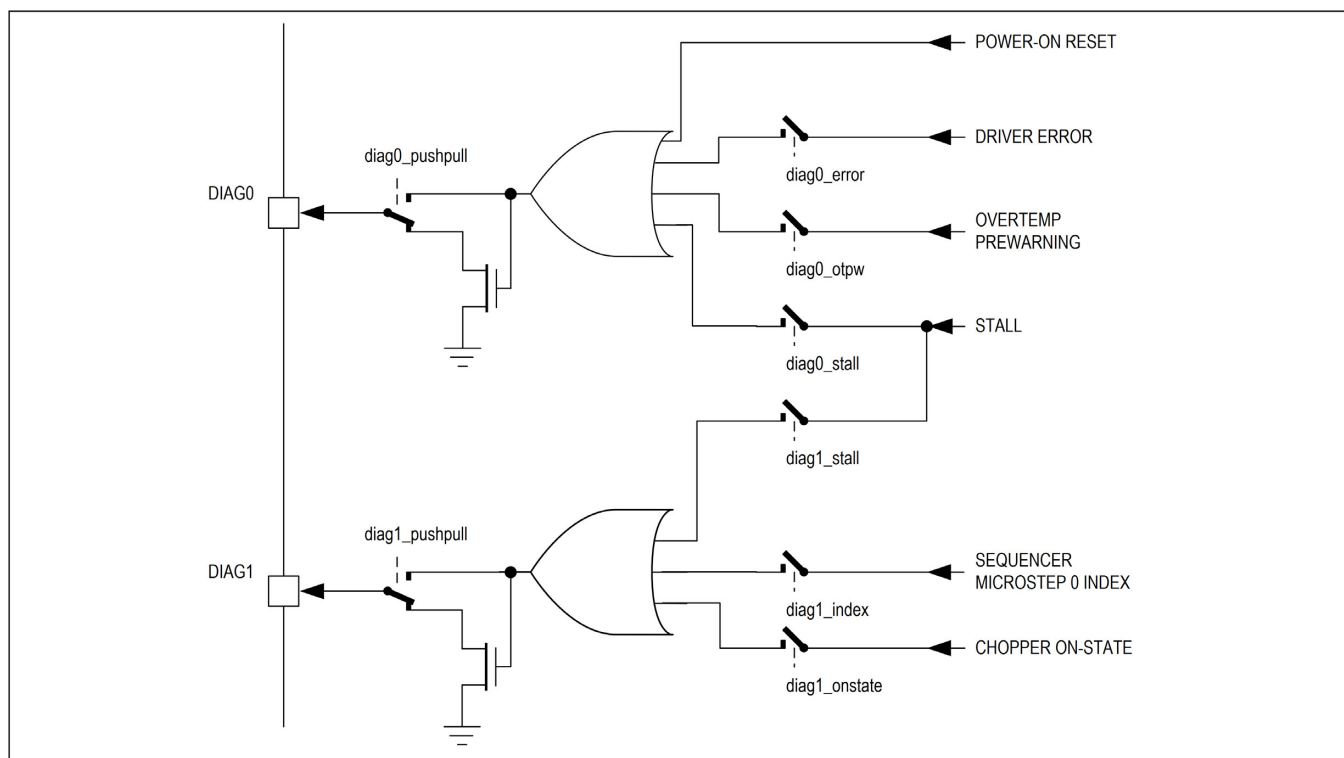


図 24. DIAG0 と DIAG1 の出力オプション

ストール出力信号を使用すると、停止スイッチのような外部モーション・コントローラによって StallGuard を扱うことができます。

チョッパ・モードに応じ、StallGuard の SG_RESULT 値がゼロになるか $SG4_RESULT$ が $SG4_THRS$ 未満に低下し、同時に速度条件が満たされると ($TSTEP \leq TCOOLTHRS$)、常にこの信号がアクティブになります。

チョッパのオン状態は、デューティ・サイクルを決定するために SpreadCycle または固定オフ時間で動作しているときに、両方のコイル・チョッパが（交互に）オン状態となることを示します。

INDEX 出力は、マイクロステップ・カウンタのゼロ位置を知らせて、アプリケーションが特定の電流パターンを基準に駆動できるようにします。インデックス・パルスの幅はマイクロステップの幅に対応しています。256 未満のマイクロステップで補間を行わずに駆動するときは、インデックス時間が 2 クロック・サイクルに低下します。インデックス出力は、コイル B マイクロステップ波の正のゼロ遷移を知らせます。

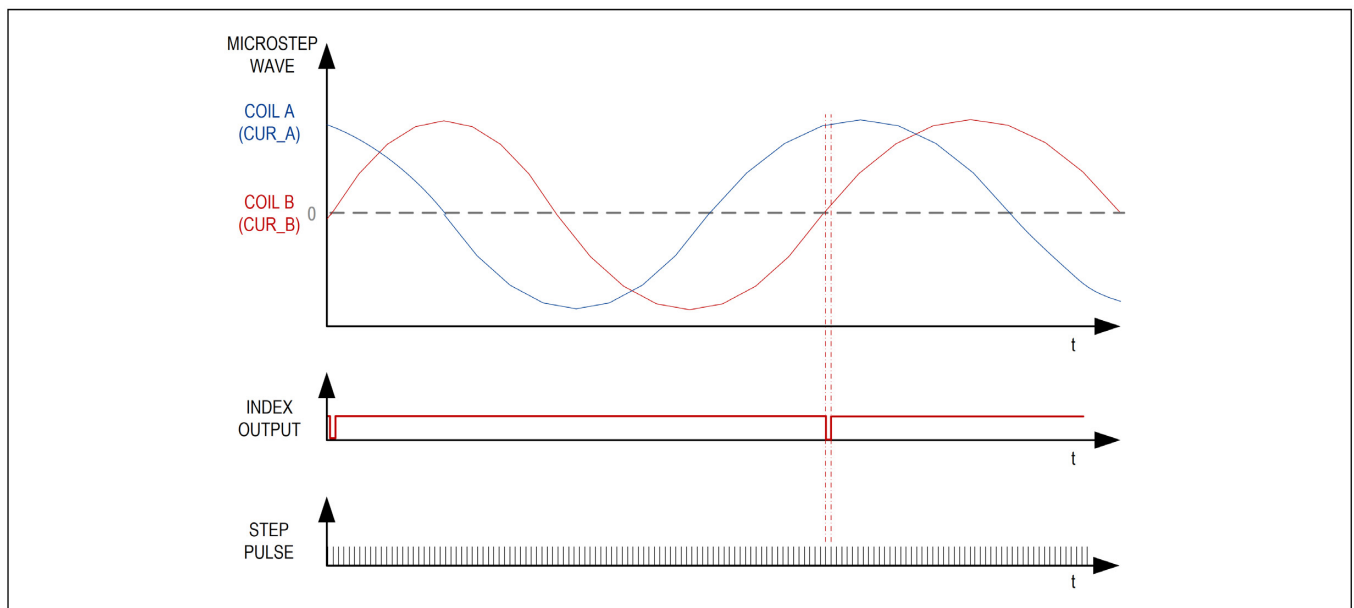


図 25. コイル B マイクロステップ波の正のゼロ遷移におけるインデックス信号（オープン・ドレイン設定時）

サイン波ルックアップ・テーブル

TMC2240 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트리ごとに 1 ビットを使用して、昇順に保存されています。そのため、波形の 1/4 を保存するのに必要なのは、わずか 256 ビット (*ofs00*~*ofs255*) のみです。これらのビットが 8 個の 32 ビット・レジスタに割り当てられます。各 *ofs* ビットは、テーブルで 1 ステップ進行する際に、勾配 *Wx* を追加するか *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スペースのチョッパーがオフセットを追加できるヘッドルームを残しておくために、変更の結果得られる波形の最大振幅が -248~248 の範囲となるよう調整する必要があります。

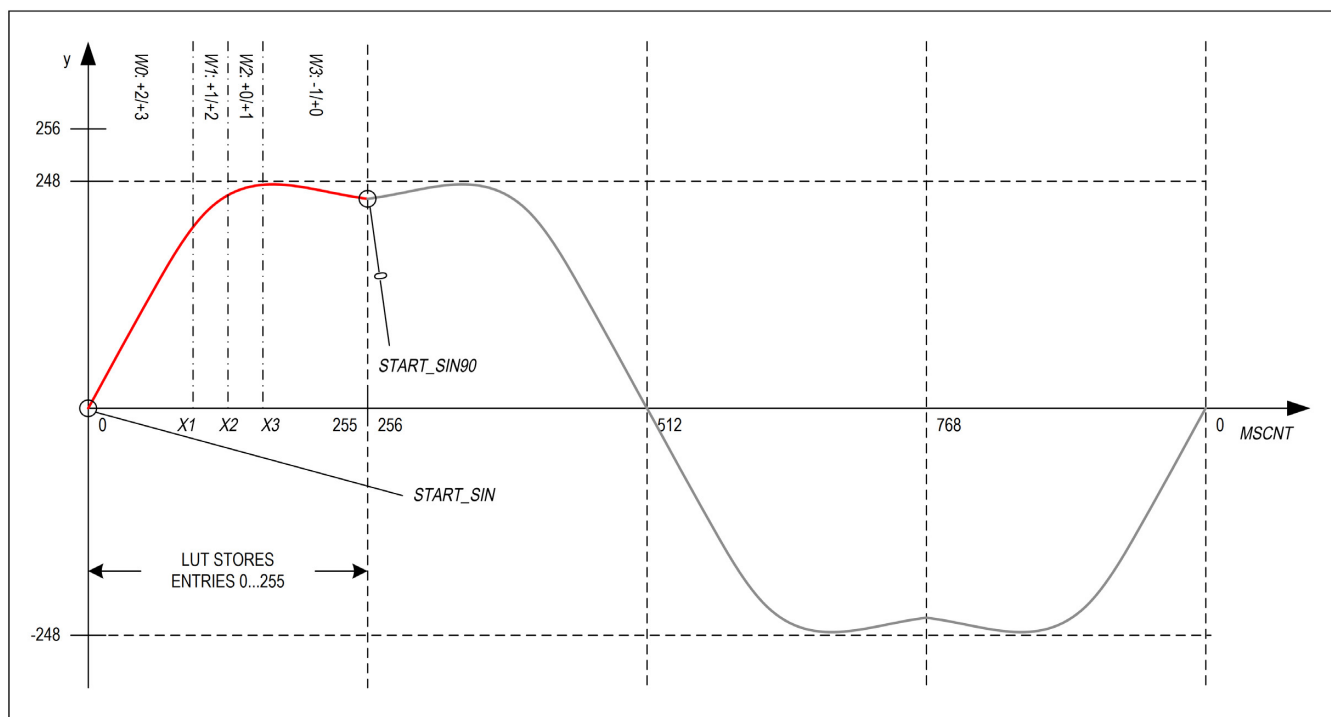


図 26. LUT プログラムの例

マイクロステップ・シーケンサはテーブル内を進行する際に、モータ・コイルの実際の電流値をマイクロステップごとに計算し、それらをレジスタ *CUR_A* および *CUR_B* に保存します。ただし、インクリメンタル・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。

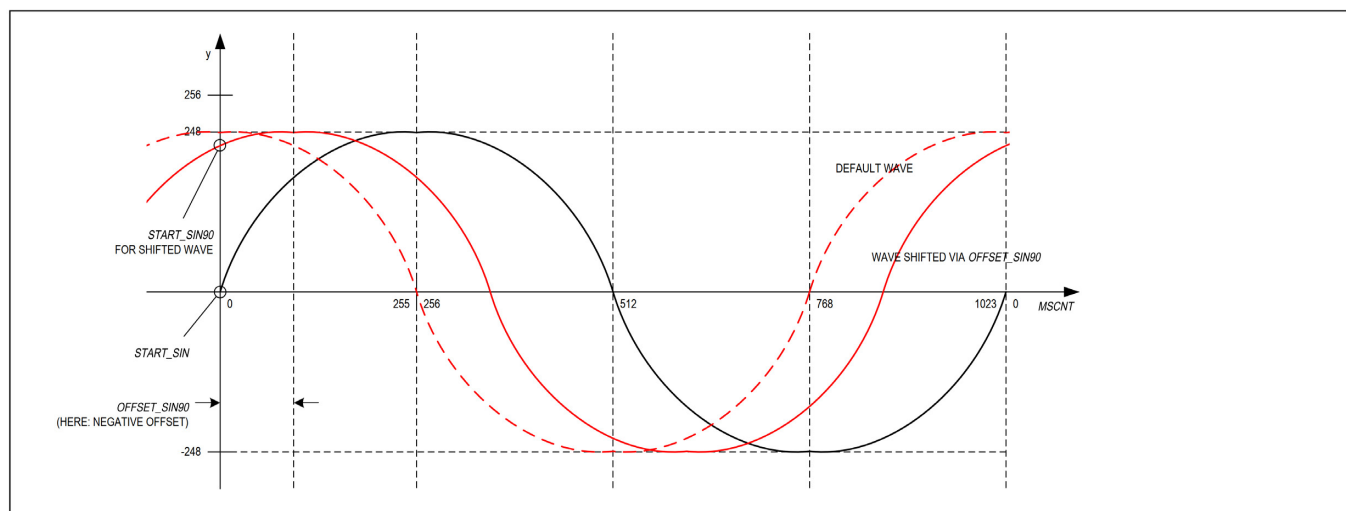


図 27. OFFSET_SIN90 を使用したコサイン波のシフト

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] = %1010101010101010101010101010100 = 0xAAAAB554`

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

`MSLUT[2] = %0010010001001001001001001001001 = 0x24492929`

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

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

`MSLUT[5] = %1011010110111011011101110111101 = 0xB5BB777D`

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

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

`MSLUTSEL = 0xFFFF8056;`

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

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

`MSLUTSTART = 0x00F70000;`

`START_SIN_0 = 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

TMC2240 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セ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・カウンタを示します。連続的な回転のため、この値は増加／減少し、常に変化します。
- オプション 2: 割込み出力がアクティブであるかどうか確認し、アクティブな割込み出力に続くフラグのみ読み出します。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。

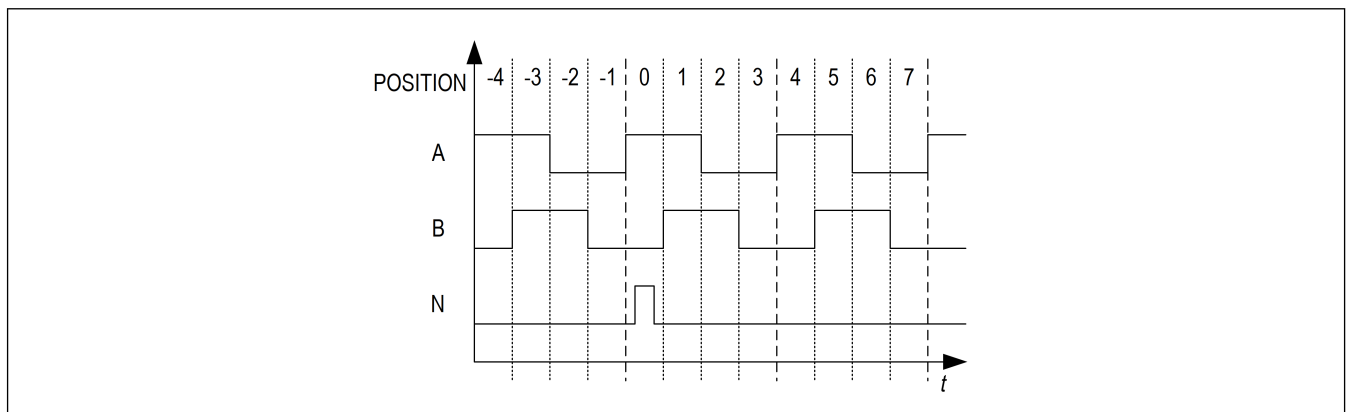


図 28. インクリメンタル・エンコーダの ABN 信号の概要

エンコーダ・カウンタ *X_ENC*

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

レジスタ *ENC_STATUS*

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

エンコーダ定数 *ENC_CONST*

エンコーダ定数（またはエンコーダ・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、 $(2^{16} - (\text{FACTOR} + 1)) \times (2^{16} - \text{FRACTION})$ 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 桁）
- 10 進数モードのエンコーダ・ファクタ -25.6： $(2^{16} - (25 + 1)) \times (10000 - 6000) = (2^{16} - 26) \times (4000) = 0xFFE6.0x0FA0$
- 負のエンコーダ定数は、次式を用いて計算できます： $(2^{16} - (\text{FACTOR} + 1)) \times (10000 - \text{DECIMALS})$

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 × USC/エンコーダ分解能

表 24. 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。

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

リセット、ディスエーブル/停止、パワー・ダウン

緊急停止

ドライバは、負のアクティブ・イネーブル・ピン DRV_ENN を備えており、すべてのパワー MOSFET を確実にスイッチ・オフします。これにより、モータ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・ダウン状態になります。

外部リセットおよびスリープ・モード

リセットおよびスリープ・モードは SLEEPN ピンで制御されます。

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

30μs 未満の非常に短いパルスはフィルタ除去され、動作には影響しません。

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

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

パワーアップ後、あるいはスリープ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 V_{CC_IO}（これは高電圧ピンです）に接続します。

ポジション・ロスのないステッパ・モータの再起動

セルフロックング駆動では、ポジション・ロスなしでモータを完全にスイッチ・オフできます。ロックングは機械的摩擦とステッパ・モータのコギング・トルクによって発生することがあります。ほとんどのステッパ・モータには公称トルクの数パーセント程度のコギング・トルクがあり、これはモータを特定のポジションにロックするために利用することもできます。その構造上、ほとんどのモータはフルステップ・ポジションでロックされます。フルステップ・ポジションの特性は、同じ絶対電流のときに両方のコイルによって得られるポジションによって決まります。n 回のマイクロステッピングの場合は n ステップごとにフルステップ位置に達します。ドライバのパワーアップ後に n/2 ステップの実行が完了すると、最初のフルステップ・ポジションに達します。フルステップ・ポジションに達したときの内部マイクロステップ・カウンタの値は、128、384、640、または 896 です。

外部的な力が加わってモータを特定の方向に動かさない限り、ロータのポジションと電気的ポジションの違いが±2 フルステップ以下であれば、パワーアップ後のモータは同じステップになります。オフセットは最大でも 1 フルステップとするのが最も安全です。

ドライバをパワーアップすると、すべてのレジスタがゼロにリセットされます。これは内部ポジション・カウンタにも影響します。したがって、ポジション・カウンタはパワーアップ後にゼロから再開されます。イネーブル・ピンを「1」に固定した状態でモータ電流が流れると、モータはこの（ハーフステップ）ポジションになります。これにより、モータ・ポジションの結果の記録については複数のオプションを選択できます。

表 25. ポジション回復の方法

イネーブル・ピン DRV_ENN	パワー・ダウン前の動作	パワーアップ時の動作
固定（GND）	最初のパワーアップ後のステップ数をカウントすることによってモータ・ポジションを記録します。パワー・ダウン前に、「4 × マイクロステップ分解能」で割り切れるポジションへ移動します。これらのポジションにおける MSCNT は 0 です。この位置を保存します。	MSCNT は自動的にクリアされて 0 になります。 必要に応じてモータの動作を開始します。
CPU により制御	MSCNT を読み出して（モータの絶対ポジションと共に）保存します。	モータ・ドライバをイネーブルする前に、MSCNT を保存値に戻すためのステップ数を適用します。 ステップ・パルス数 = ポジション・モジュール（4 × マイクロステップ分解能） 例：32 マイクロステップ設定では、各ステップ・パルスが MSCNT を 256/32 = 8 ずつインクリメントします。 ポジション・モジュール 128 を計算して必要なステップ数を求めます。 DIR = 0 で 10 ステップを加えると、MSCNT の値が 80 に増加します。

保護およびドライバ診断

TMC2240 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・コイルの接続が切断されているかどうかをテストできます。詳細については [DRV_STATUS](#) レジスタの表を参照してください。

ステータス・フラグの他、TMC2240 では、チップ温度の測定および読出しや、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、TMC2240 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 スレッシュホールドは、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ュホールド値については[電氣的特性](#)の表を参照してください。

フルスケール・レンジは、*DRV_CONF* レジスタの *CURRENT_RANGE* パラメータで選択できます。

デグリッチ時間（ブランキング時間）より長い間にわたって出力電流が OCP スレッシュホールドを上回る場合、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*、*s2vsa*、*s2vsb*）がセットされブリッジが連続的にディスエーブルされます。

デバイスは依然として活性で、設定やステータスの読出しが可能です。

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

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

過熱保護およびサーマル・シャットダウン

TMC2240 は過熱保護機能を内蔵しています。

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

更に、TMC2240 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、TMC2240 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

TMC2240 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。

$$ADC_TEMP = 7.7 \times TEMP + 2038$$

$$TEMP[^\circ C] = \frac{ADC_TEMP - 2038}{7.7}$$

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

ステップング・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、TMC2240 では、過電圧の検出および保護のための機構を備えています。

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

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

電源電圧は内蔵 ADC で常にモニタされています。

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

電源電圧の実際の ADC 値は、レジスタ `ADC_VSUPPLY_AIN` のパラメータ `ADC_VSUPPLY` で読み出せます。

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

$$V_S = \text{ADC_VSUPPLY} \times 9.732\text{mV}$$

標準的なアプリケーションの場合、モータから電源へ還流される最大電流は、1つのコイルの RMS コイル電流未満です。

適切な抵抗値は次式で求めることができます。

$$R_{\text{Dump}} = \frac{U_{\text{supply}}}{I_{\text{Coil}}}$$

U_{supply} は公称ドライバ電源電圧 V_S 、 I_{Coil} はモータの公称コイル電流です。

MOSFET が最終的な電流をスイッチングできるようにしてください。

OV 出力ピンは、過電圧モニタの実際の状態を示します。

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

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

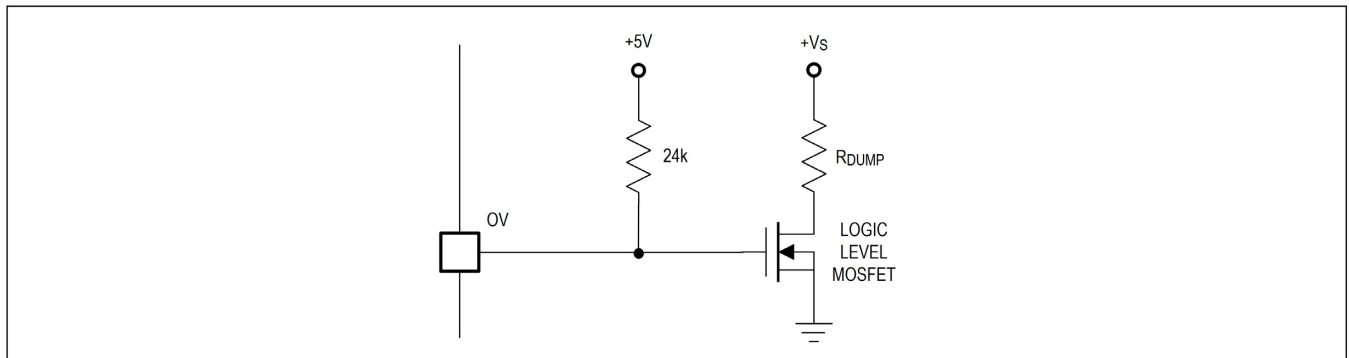


図 29. ブレーキ・チョッパー回路の例

短絡保護（GND への短絡および VS への短絡）

TMC2240 の電力段は、ハイサイド MOSFET を流れる電流を追加測定することで、回路の短絡状態に対し保護されます。大半の短絡状態は、システムのグラウンドに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ュホー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ュホールド

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。TMC2240 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、SpreadCyle 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

TMC2240 は、 V_M 、 V_{CC_IO} 、チャージ・ポンプに対する UVLO 保護機能を備えています。

V_M の UVLO 条件は 4.15V（最大値）未満でトリガされます。

V_{CC_IO} 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_M 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。UVLO 状態は、ピン設定に応じて *DIAG0* ピンでも示されます。

V_{CC_IO} の UVLO 時には、IC との通信は不可能です。*DIAG0* ピンはアクティブ・ロー（オープン・ドレイン）になります。

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。

TMC2240 のモータ位相出力ピンは、正電圧電源 (V_S ピン) に 1μF 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

TMC2240 は、外部アナログ入力 AIN を備えており、内部 ADC で連続的にサンプリングされます。

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

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

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

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

クロック発振器とクロック入力

内部クロックの使用

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

内部クロックは、12.5MHz (代表値) の周波数で動作しています。

外部クロックの使用

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

クロック信号の必要な最小および最大のデューティ・サイクルは、[電氣的特性](#)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、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 の読出しビット `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。

クイック設定ガイド

このガイドは、最初のレジスタ設定を決定してドライバ調整のための測定と決定をできるだけ少なくすることのできる、実用的なツールを提供することを目的としています。ここではすべての先進的機能とオプションをカバーしているわけではなく、モータをスムーズに動作させるための基本的な機能セットに重点を置いています。モータが動作するようになった後は、追加的な機能とその他の機能を決定し、より詳細な検討を試みることができます。モータ・コイルへの電流プローブ使用は、最良の設定を見つけるための良い助けとなります。

電流の設定

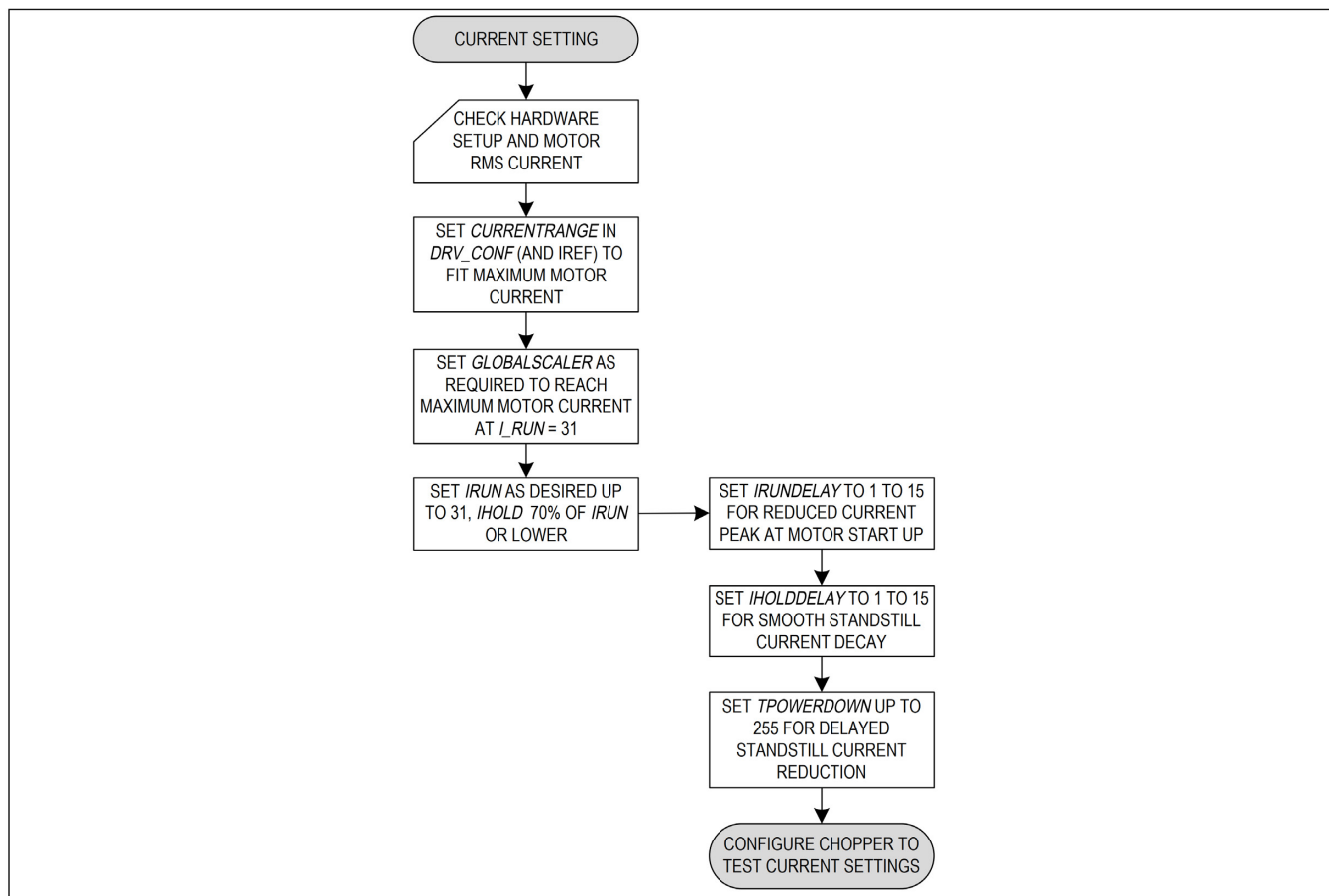


図 30. 電流設定のためのクイック設定ガイド

StealthChop2 の設定

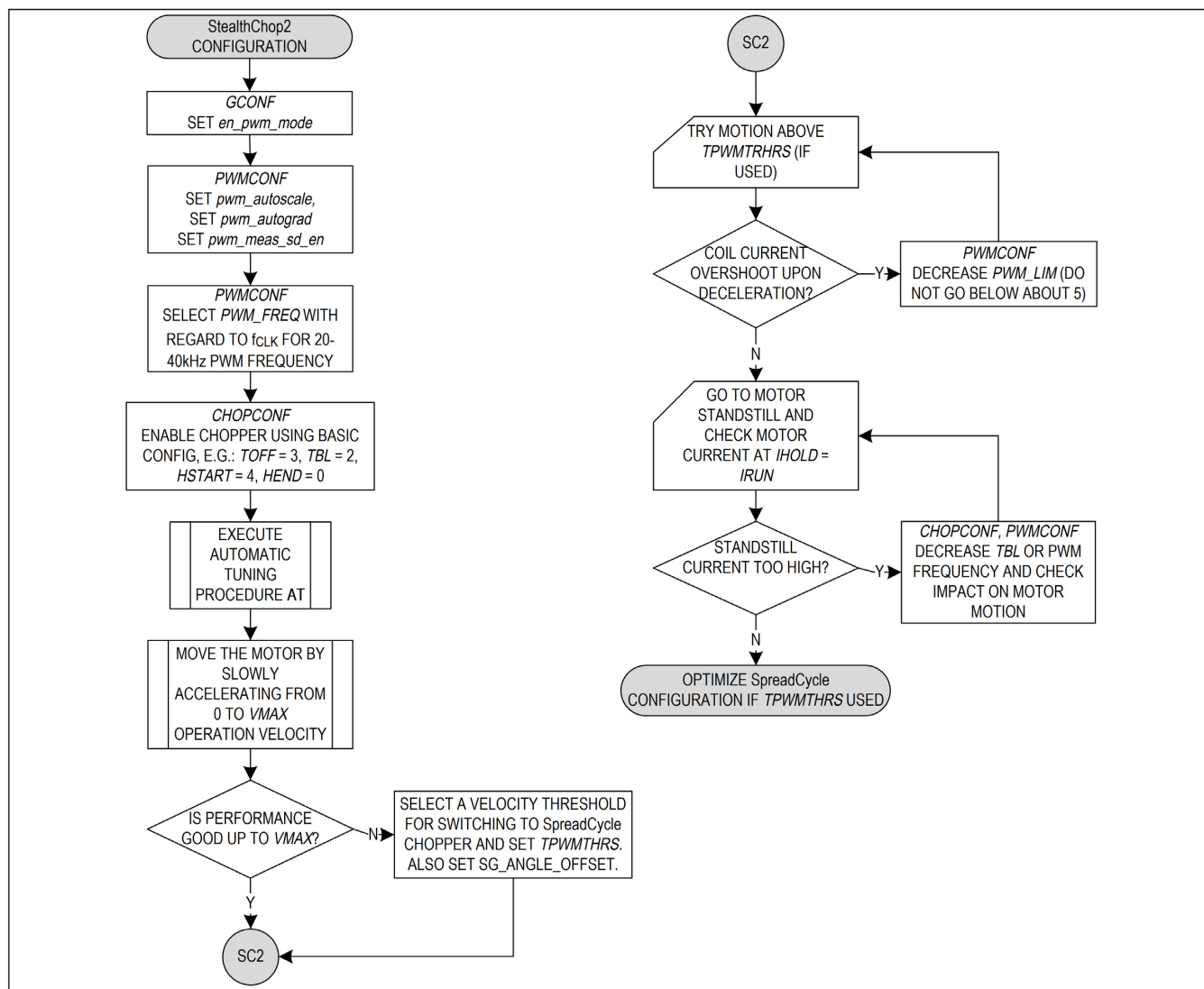


図 31. StealthChop2 のクイック設定ガイド

SpreadCycle の設定

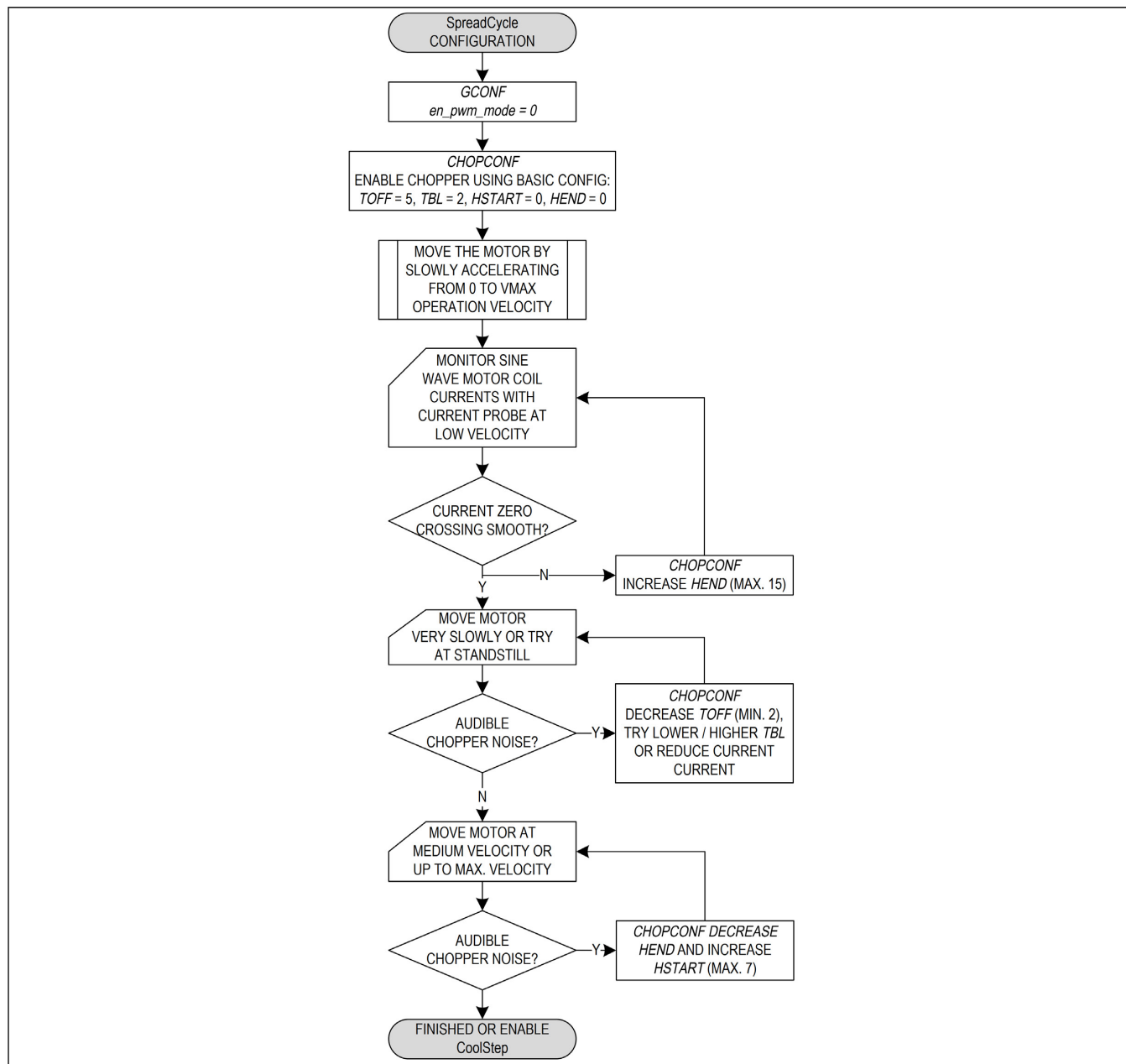


図 32. SpreadCycle のクイック設定ガイド

StealthChop2 との組み合わせで CoolStep をイネーブル

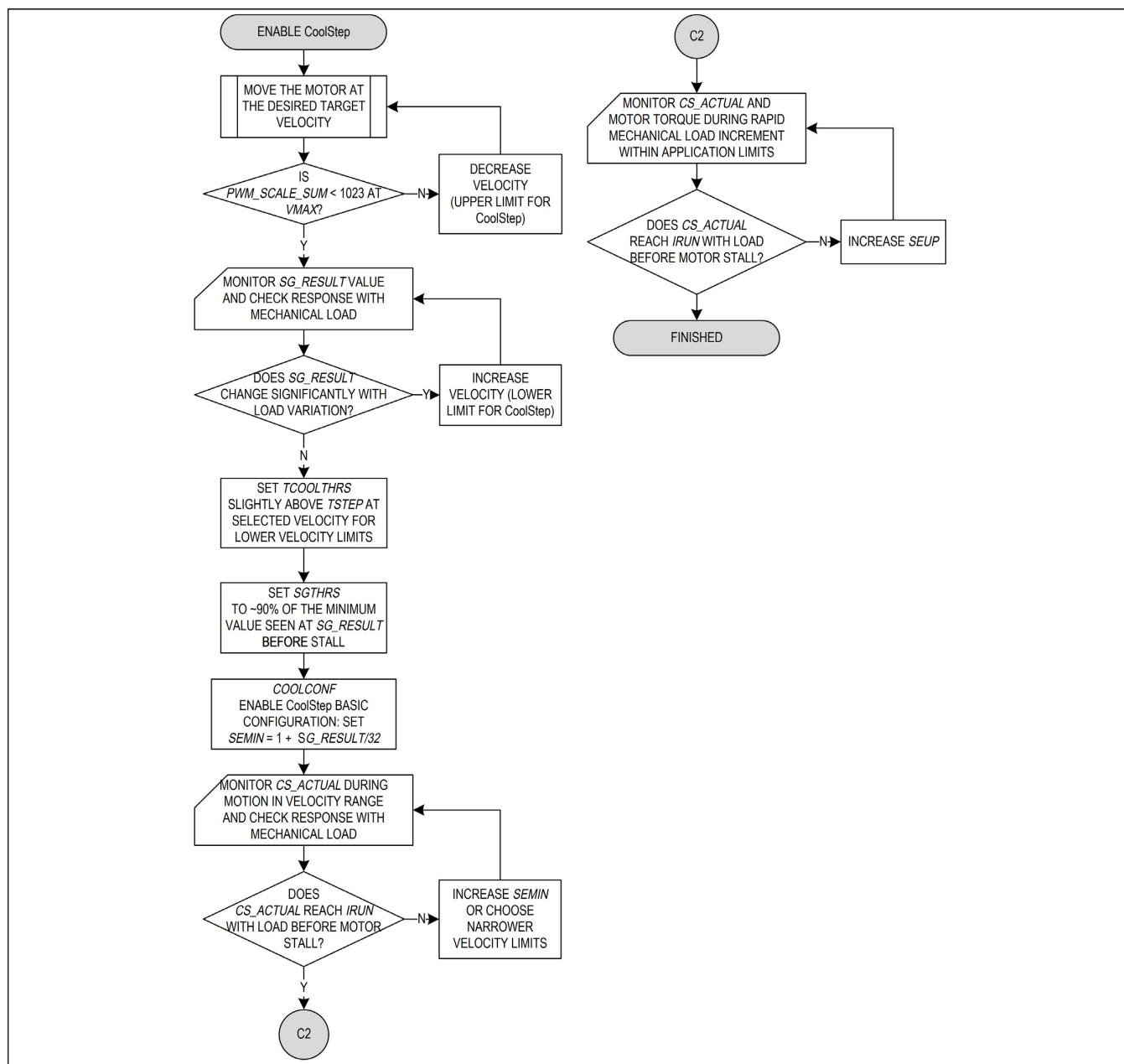


図 33. StealthChop2 との組み合わせによる CoolStep のクイック設定ガイド

SpreadCycle との組み合わせで CoolStep をイネーブ

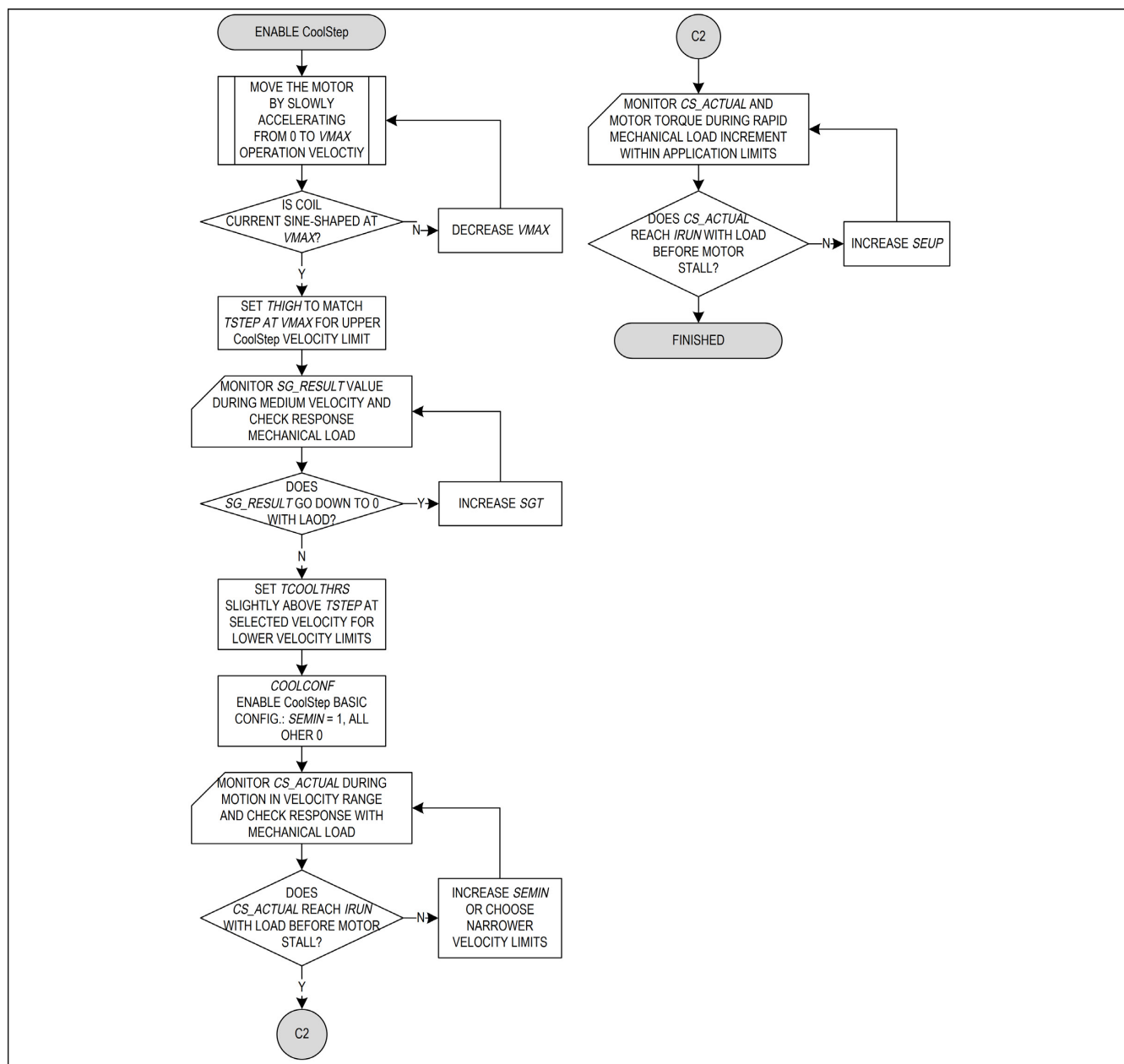


図 34. SpreadCycle との組み合わせによる CoolStep のクイック設定ガイド

全般的なレジスタ・マッピングおよびレジスタ情報

このセクションでは、レジスタ・マップに関する全般的な情報を示します。

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

- すべてのレジスタは、特に指定のない限り、パワー・アップ時に 0 にリセットされます。
- 書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 • グローバル設定 • グローバル・ステータス・フラグ • インターフェイス設定 • I/O 信号設定
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・グループは以下を目的とするレジスタで構成されます • ドライバ電流制御 • CoolStep 動作スレッシュホールドの設定 • 様々なチョッパー・モード用スレッシュホールドの設定
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・グループは、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・グループは以下を目的とするレジスタで構成されます • 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 • チョッパーおよびドライバの設定 • CoolStep および StallGuard の設定 • StallGuard の値およびドライバ・エラー・フラグの読出し

レジスタ・マップ

TMC2240

ADDRESS	NAME	MSB							LSB
General Configuration Registers									
0x00	GCONF[31:24]	–	–	–	–	–	–	–	–
	GCONF[23:16]	–	–	–	–	–	–	–	direct_m ode
	GCONF[15:8]	stop_ena ble	small_hy steresis	diag1_pu shpull	diag0_pu shpull	–	diag1_on state	diag1_in dex	diag1_st all
	GCONF[7:0]	diag0_st all	diag0_ot pw	diag0_er ror	shaft	multistep _filt	en_pwm _mode	fast_stan dstill	–
0x01	GSTAT[31:24]	–	–	–	–	–	–	–	–
	GSTAT[23:16]	–	–	–	–	–	–	–	–
	GSTAT[15:8]	–	–	–	–	–	–	–	–
	GSTAT[7:0]	–	–	–	vm_uvlo	register_ reset	uv_cp	drv_err	reset
0x02	IFCNT[31:24]	–	–	–	–	–	–	–	–
	IFCNT[23:16]	–	–	–	–	–	–	–	–
	IFCNT[15:8]	–	–	–	–	–	–	–	–
	IFCNT[7:0]	IFCNT[7:0]							
0x03	NODECONF[31:24]	–	–	–	–	–	–	–	–
	NODECONF[23:16]	–	–	–	–	–	–	–	–
	NODECONF[15:8]	–	–	–	–	SENDDelay[3:0]			
	NODECONF[7:0]	NODEADDR[7:0]							
0x04	IOIN[31:24]	VERSION[7:0]							
	IOIN[23:16]	–	–	–	–	–	SILICON_RV[2:0]		
	IOIN[15:8]	ADC_ER R	EXT_CL K	EXT_RE S_DET	OUTPUT	COMP_ B1_B2	COMP_ A1_A2	COMP_ B	COMP_ A
	IOIN[7:0]	reserved	UART_E N	ENCN	DRV_EN N	ENCA	ENCB	DIR	STEP
0x0A	DRV_CONF[31:24]	–	–	–	–	–	–	–	–
	DRV_CONF[23:16]	–	–	–	–	–	–	–	–
	DRV_CONF[15:8]	–	–	–	–	–	–	–	–
	DRV_CONF[7:0]	–	–	SLOPE_CONTROL[1:0]		–	–	CURRENT_RANGE[1:0]	
0x0B	GLOBAL SCALER[31:24]	–	–	–	–	–	–	–	–
	GLOBAL SCALER[23:16]	–	–	–	–	–	–	–	–
	GLOBAL SCALER[15:8]	–	–	–	–	–	–	–	–
	GLOBAL SCALER[7:0]	GLOBALSCALER[7:0]							
Velocity Dependent Configuration Registers									
0x10	IHOLD_IRUN[31:24]	–	–	–	–	IRUNDELAY[3:0]			
	IHOLD_IRUN[23:16]	–	–	–	–	IHOLDDELAY[3:0]			

ADDRESS	NAME	MSB							LSB
	IHOLD_IRUN[15:8]	–	–	–	IRUN[4:0]				
	IHOLD_IRUN[7:0]	–	–	–	IHOLD[4:0]				
0x11	TPOWERDOWN[31:24]	–	–	–	–	–	–	–	–
	TPOWERDOWN[23:16]	–	–	–	–	–	–	–	–
	TPOWERDOWN[15:8]	–	–	–	–	–	–	–	–
	TPOWERDOWN[7:0]	TPOWERDOWN[7:0]							
0x12	TSTEP[31:24]	–	–	–	–	–	–	–	–
	TSTEP[23:16]	–	–	–	–	TSTEP[19:16]			
	TSTEP[15:8]	TSTEP[15:8]							
	TSTEP[7:0]	TSTEP[7:0]							
0x13	TPWMTHRS[31:24]	–	–	–	–	–	–	–	–
	TPWMTHRS[23:16]	–	–	–	–	TPWMTHRS[19:16]			
	TPWMTHRS[15:8]	TPWMTHRS[15:8]							
	TPWMTHRS[7:0]	TPWMTHRS[7:0]							
0x14	TCOOLTHRS[31:24]	–	–	–	–	–	–	–	–
	TCOOLTHRS[23:16]	–	–	–	–	TCOOLTHRS[19:16]			
	TCOOLTHRS[15:8]	TCOOLTHRS[15:8]							
	TCOOLTHRS[7:0]	TCOOLTHRS[7:0]							
0x15	THIGH[31:24]	–	–	–	–	–	–	–	–
	THIGH[23:16]	–	–	–	–	THIGH[19:16]			
	THIGH[15:8]	THIGH[15:8]							
	THIGH[7:0]	THIGH[7:0]							
Direct Mode Register									
0x2D	DIRECT_MODE[31:24]	–	–	–	–	–	–	–	DIRECT_COIL_B[8]
	DIRECT_MODE[23:16]	DIRECT_COIL_B[7:0]							
	DIRECT_MODE[15:8]	–	–	–	–	–	–	–	DIRECT_COIL_A[8]
	DIRECT_MODE[7:0]	DIRECT_COIL_A[7:0]							
Encoder Registers									
0x38	ENCMODE[31:24]	–	–	–	–	–	–	–	–
	ENCMODE[23:16]	–	–	–	–	–	–	–	–
	ENCMODE[15:8]	–	–	–	–	–	enc_sel_decimal	–	clr_enc_x
	ENCMODE[7:0]	pos_neg_edge[1:0]		clr_once	clr_cont	ignore_A_B	pol_N	pol_B	pol_A
0x39	X_ENC[31:24]	X_ENC[31:24]							
	X_ENC[23:16]	X_ENC[23:16]							
	X_ENC[15:8]	X_ENC[15:8]							
	X_ENC[7:0]	X_ENC[7:0]							
0x3A	ENC_CONST[31:24]	ENC_CONST[31:24]							
	ENC_CONST[23:16]	ENC_CONST[23:16]							

ADDRESS	NAME	MSB							LSB
	ENC_CONST[15:8]	ENC_CONST[15:8]							
	ENC_CONST[7:0]	ENC_CONST[7:0]							
0x3B	ENC_STATUS[31:24]	–	–	–	–	–	–	–	–
	ENC_STATUS[23:16]	–	–	–	–	–	–	–	–
	ENC_STATUS[15:8]	–	–	–	–	–	–	–	–
	ENC_STATUS[7:0]	–	–	–	–	–	–	–	n_event
0x3C	ENC_LATCH[31:24]	ENC_LATCH[31:24]							
	ENC_LATCH[23:16]	ENC_LATCH[23:16]							
	ENC_LATCH[15:8]	ENC_LATCH[15:8]							
	ENC_LATCH[7:0]	ENC_LATCH[7:0]							
ADC Registers									
0x50	ADC_VSUPPLY_AIN[31:24]	–	–	–	ADC_AIN[12:8]				
	ADC_VSUPPLY_AIN[23:16]	ADC_AIN[7:0]							
	ADC_VSUPPLY_AIN[15:8]	–	–	–	ADC_VSUPPLY[12:8]				
	ADC_VSUPPLY_AIN[7:0]	ADC_VSUPPLY[7:0]							
0x51	ADC_TEMP[31:24]	–	–	–	RESERVED[12:8]				
	ADC_TEMP[23:16]	RESERVED[7:0]							
	ADC_TEMP[15:8]	–	–	–	ADC_TEMP[12:8]				
	ADC_TEMP[7:0]	ADC_TEMP[7:0]							
0x52	OTW_OV_VTH[31:24]	–	–	–	OVERTEMPPREWARNING_VTH[12:8]				
	OTW_OV_VTH[23:16]	OVERTEMPPREWARNING_VTH[7:0]							
	OTW_OV_VTH[15:8]	–	–	–	OVERVOLTAGE_VTH[12:8]				
	OTW_OV_VTH[7:0]	OVERVOLTAGE_VTH[7:0]							
Motor Driver Registers									
0x60	MSLUT_0[31:24]	MSLUT_0[31:24]							
	MSLUT_0[23:16]	MSLUT_0[23:16]							
	MSLUT_0[15:8]	MSLUT_0[15:8]							
	MSLUT_0[7:0]	MSLUT_0[7:0]							
0x61	MSLUT_1[31:24]	MSLUT_1[31:24]							
	MSLUT_1[23:16]	MSLUT_1[23:16]							
	MSLUT_1[15:8]	MSLUT_1[15:8]							
	MSLUT_1[7:0]	MSLUT_1[7:0]							
0x62	MSLUT_2[31:24]	MSLUT_2[31:24]							
	MSLUT_2[23:16]	MSLUT_2[23:16]							
	MSLUT_2[15:8]	MSLUT_2[15:8]							
	MSLUT_2[7:0]	MSLUT_2[7:0]							
0x63	MSLUT_3[31:24]	MSLUT_3[31:24]							
	MSLUT_3[23:16]	MSLUT_3[23:16]							
	MSLUT_3[15:8]	MSLUT_3[15:8]							
	MSLUT_3[7:0]	MSLUT_3[7:0]							

ADDRESS	NAME	MSB							LSB
0x64	MSLUT_4[31:24]	MSLUT_4[31:24]							
	MSLUT_4[23:16]	MSLUT_4[23:16]							
	MSLUT_4[15:8]	MSLUT_4[15:8]							
	MSLUT_4[7:0]	MSLUT_4[7:0]							
0x65	MSLUT_5[31:24]	MSLUT_5[31:24]							
	MSLUT_5[23:16]	MSLUT_5[23:16]							
	MSLUT_5[15:8]	MSLUT_5[15:8]							
	MSLUT_5[7:0]	MSLUT_5[7:0]							
0x66	MSLUT_6[31:24]	MSLUT_6[31:24]							
	MSLUT_6[23:16]	MSLUT_6[23:16]							
	MSLUT_6[15:8]	MSLUT_6[15:8]							
	MSLUT_6[7:0]	MSLUT_6[7:0]							
0x67	MSLUT_7[31:24]	MSLUT_7[31:24]							
	MSLUT_7[23:16]	MSLUT_7[23:16]							
	MSLUT_7[15:8]	MSLUT_7[15:8]							
	MSLUT_7[7:0]	MSLUT_7[7:0]							
0x68	MSLUTSEL[31:24]	X3[7:0]							
	MSLUTSEL[23:16]	X2[7:0]							
	MSLUTSEL[15:8]	X1[7:0]							
	MSLUTSEL[7:0]	W3[1:0]	W2[1:0]		W1[1:0]		W0[1:0]		
0x69	MSLUTSTART[31:24]	OFFSET_SIN90[7:0]							
	MSLUTSTART[23:16]	START_SIN90[7:0]							
	MSLUTSTART[15:8]	–	–	–	–	–	–	–	–
	MSLUTSTART[7:0]	START_SIN[7:0]							
0x6A	MSCNT[31:24]	–	–	–	–	–	–	–	–
	MSCNT[23:16]	–	–	–	–	–	–	–	–
	MSCNT[15:8]	–	–	–	–	–	–	MSCNT[9:8]	
	MSCNT[7:0]	MSCNT[7:0]							
0x6B	MSCURACT[31:24]	–	–	–	–	–	–	–	CUR_A[8]
	MSCURACT[23:16]	CUR_A[7:0]							
	MSCURACT[15:8]	–	–	–	–	–	–	–	CUR_B[8]
	MSCURACT[7:0]	CUR_B[7:0]							
0x6C	CHOPCONF[31:24]	diss2vs	diss2g	dedge	intpol	MRES[3:0]			
	CHOPCONF[23:16]	TPFD[3:0]				vhighchm	vhighfs	–	TBL[1]
	CHOPCONF[15:8]	TBL[0]	chm	–	disfdcc	fd3	HEND_OFFSET[3:1]		
	CHOPCONF[7:0]	HEND_OFFSET[0]	HSTRT_TFD210[2:0]			TOFF[3:0]			
0x6D	COOLCONF[31:24]	–	–	–	–	–	–	–	sflt
	COOLCONF[23:16]	–	sgt[6:0]						
	COOLCONF[15:8]	seimin	sedn[1:0]		–	semax[3:0]			

ADDRESS	NAME	MSB							LSB
	COOLCONF[7:0]	–	seup[1:0]		–	semin[3:0]			
0x6F	DRV_STATUS[31:24]	stst	olb	ola	s2gb	s2ga	otpw	ot	stallguard
	DRV_STATUS[23:16]	–	–	–	CS_ACTUAL[4:0]				
	DRV_STATUS[15:8]	fsactive	stealth	s2vsb	s2vsa	–	–	SG_RESULT[9:8]	
	DRV_STATUS[7:0]	SG_RESULT[7:0]							
0x70	PWMCONF[31:24]	PWM_LIM[3:0]				PWM_REG[3:0]			
	PWMCONF[23:16]	pwm_dis_reg_stst	pwm_meas_sd_enable	FREEWHEEL[1:0]		pwm_autograd	pwm_autoscale	PWM_FREQ[1:0]	
	PWMCONF[15:8]	PWM_GRAD[7:0]							
	PWMCONF[7:0]	PWM_OFS[7:0]							
0x71	PWM_SCALE[31:24]	–	–	–	–	–	–	–	PWM_SCALE_AUTO[8]
	PWM_SCALE[23:16]	PWM_SCALE_AUTO[7:0]							
	PWM_SCALE[15:8]	–	–	–	–	–	–	PWM_SCALE_SUM[9:8]	
	PWM_SCALE[7:0]	PWM_SCALE_SUM[7:0]							
0x72	PWM_AUTO[31:24]	–	–	–	–	–	–	–	–
	PWM_AUTO[23:16]	PWM_GRAD_AUTO[7:0]							
	PWM_AUTO[15:8]	–	–	–	–	–	–	–	–
	PWM_AUTO[7:0]	PWM_OFS_AUTO[7:0]							
0x74	SG4_THRS[31:24]	–	–	–	–	–	–	–	–
	SG4_THRS[23:16]	–	–	–	–	–	–	–	–
	SG4_THRS[15:8]	–	–	–	–	–	–	sg_angle_offset	sg4_filt_en
	SG4_THRS[7:0]	SG4_THRS[7:0]							
0x75	SG4_RESULT[31:24]	–	–	–	–	–	–	–	–
	SG4_RESULT[23:16]	–	–	–	–	–	–	–	–
	SG4_RESULT[15:8]	–	–	–	–	–	–	SG4_RESULT[9:8]	
	SG4_RESULT[7:0]	SG4_RESULT[7:0]							
0x76	SG4_IND[31:24]	SG4_IND_3[7:0]							
	SG4_IND[23:16]	SG4_IND_2[7:0]							
	SG4_IND[15:8]	SG4_IND_1[7:0]							
	SG4_IND[7:0]	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
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	direct_mode
Reset	–	–	–	–	–	–	–	0x0
Access Type	–	–	–	–	–	–	–	Write, Read
BIT	15	14	13	12	11	10	9	8
Field	stop_enable	small_hysteresis	diag1_pushpull	diag0_pushpull	–	diag1_onstate	diag1_index	diag1_stall
Reset	0x0	0x0	0x0	0x0	–	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	–	Write, Read	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	diag0_stall	diag0_otpw	diag0_error	shaft	multistep_filt	en_pwm_mode	fast_standstill	–
Reset	0x0	0x0	0x0	0x0	0x1	0x0	0x0	–
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	–

ビット・フィールド	ビット	説明	デコード
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 <i>DIRECT_MODE</i> (0x2D) specifies signed coil A current (bits 8..0) and coil B current (bits 24..16). In this mode, the current is scaled by <i>IHOLD</i> 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_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_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)

ビット・フィールド	ビット	説明	デコード
diag1_onstate	10	DIAG1の出力設定。	0x0: Disable DIAG1 active on chopper on. 0x1: <i>diag1_onstate</i> Enable DIAG1 active when chopper is on (for the coil which is in the second half of the fullstep)
diag1_index	9	DIAG1の出力設定。	0x0: Disable DIAG1 active on index position. 0x1: <i>diag1_index</i> Enable DIAG1 active on index position (microstep look up table position 0)
diag1_stall	8	DIAG1の出力設定。	0x0: <i>diag1_stall</i> Motor stall not indicated at DIAG1 0x1: <i>diag1_stall</i> TCOOLTHRS before using this feature)
diag0_stall	7	DIAG0の出力設定。	0x0: <i>diag0_stall</i> Motor stall not indicated at DIAG0 0x1: <i>diag0_stall</i> Enable DIAG0 active on motor stall (set TCOOLTHRS before using this feature)
diag0_otpw	6	DIAG0の出力設定。	0x0: Disable DIAG0 active on driver over temperature prewarning 0x1: Enable DIAG0 active on driver over temperature prewarning (<i>otpw</i>)
diag0_error	5	DIAG0の出力設定。 DIAG0は常にリセット・ステータスを示します。つまり、リセット状態ではアクティブ・ローです。	0x0: Disable DIAG0 active on driver errors. 0x1: Enable DIAG0 active on driver errors: Over temperature (<i>ot</i>), short to GND (<i>s2g</i>), undervoltage chargepump (<i>uv_cp</i>)
shaft	4	モータの方向／方向符号の変更	0x0: Default motor direction 0x1: Inverse motor direction
multistep_filt	3	StealthChop2のステップ入力フィルタリングをイネーブル。	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.
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
Field	—	—	—	—	—	—	—	—
Reset	—	—	—	—	—	—	—	—
Access Type	—	—	—	—	—	—	—	—

BIT	23	22	21	20	19	18	17	16
Field	—	—	—	—	—	—	—	—
Reset	—	—	—	—	—	—	—	—
Access Type	—	—	—	—	—	—	—	—
BIT	15	14	13	12	11	10	9	8
Field	—	—	—	—	—	—	—	—
Reset	—	—	—	—	—	—	—	—
Access Type	—	—	—	—	—	—	—	—
BIT	7	6	5	4	3	2	1	0
Field	—	—	—	vm_uvlo	register_reset	uv_cp	drv_err	reset
Reset	—	—	—	0x1	0x1	0x1	0x0	0x1
Access Type	—	—	—	Write 1 to Clear, Read	Write 1 to Clear, Read	Write 1 to Clear, Read	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
vm_uvlo	4	1: 最後のリセット後にV _S 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.
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込みアクセスが正常に行われるたびにインクリメントします。これを読み出すことでシリアル伝送にデータ喪失がないことを確認できます。読出しアクセスでは内容は変化しません。SPI 動作ではディスエーブルされます。カウンタは 255 の後は 0 に戻ります。

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

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

ビット・フィールド	ビット	説明
IFCNT	7:0	インターフェイス伝送カウンタ。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。読出しアクセスでは内容は変化しません。 SPI動作ではディスエーブルされます。カウンタは255の後は0に戻ります。

NODECONF (0x3)

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

ビット・フィールド	ビット	説明	デコード
SENDDelay	11:8	SWUARTのノード設定	0x0: 8 bit times (not allowed with multiple nodes) 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	NODEADDR : 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、CSNによる定義に従って1ずつ7までインクリメントします。 CSN、SCK、SDI 000 : +0 001 : +1 010 : +2 011 : +3 100 : +4 101 : +5 110 : +6 111 : +7 範囲 : 0~254 (254より先へはインクリメントしません)	

IOIN (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
Field	VERSION[7:0]							
Reset								
Access Type	Read Only							
BIT	23	22	21	20	19	18	17	16
Field	—	—	—	—	—	SILICON_RV[2:0]		
Reset	—	—	—	—	—	0x0		
Access Type	—	—	—	—	—	Read Only		
BIT	15	14	13	12	11	10	9	8
Field	ADC_ERR	EXT_CLK	EXT_RES_DET	OUTPUT	COMP_B1_B2	COMP_A1_A2	COMP_B	COMP_A
Reset	0x0	0x0	0x0	0x1	0x0	0x0	0x0	0x0
Access Type	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
Field	reserved	UART_EN	ENCN	DRV_ENN	ENCA	ENCB	DIR	STEP
Reset		0x0		0x0	0x0	0x0	0x0	0x0
Access Type	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・コンパレータB、ICテスト用)
COMP_A	8	COMP_A (チョッパ・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
DIR	1	
STEP	0	

DRV_CONF (0xA)

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

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

ビット・フィールド	ビット	説明	デコード
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
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	7	6	5	4	3	2	1	0
Field	GLOBALSCALER[7:0]							
Reset	0x0							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
GLOBALSCALER	7:0	モータ電流のグローバル・スケーリング。ドライブを特定のモータ・タイプに合わせるために、この値が電流スケーリングに乘じられます。この値を選択してから、他の設定値を調整してください。この値はチョッパのヒステリシスにも影響するためです。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。 0 : フルスケール（または256を書込み） 1~31 : 動作には許容されません 32~255 : 最大電流の32/256~255/256 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。

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				

ビット・フィールド	ビット	説明
IRUNDELAY	27:24	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。 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
IHOLDDELAY	19:16	停止が検出され（stst = 1）TPOWERDOWNの時間が経過すると直ちにモーション後のモータ・パワー・ダウンを行うための、クロック・サイクル数を制御します。滑らかな遷移にすることで、パワー・ダウン時のモータ・ジャークを回避できます。 0 : 即時パワー・ダウン 1~15 : 2¹⁸クロックの倍数で表した電流低減ステップごとの遅延
IRUN	12:8	モータ動作電流（0 = 1/32から31 = 32/32まで） ヒント：最良のマイクロステップ性能を実現するには、通常時のIRUNが16~31となるように検出抵抗を選択します。

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

ビット・フィールド	ビット	説明
TPOWERDOWN	7:0	TPOWERDOWNは、モータの停止 (stst) 後モータ電流がパワー・ダウンするまでの遅延時間を設定します。時間範囲は、約0~4秒です。 注意 : StealthChop2のPWM_OFFS_AUTOを自動調整するには、少なくとも2に設定することが必要です。 リセット後のデフォルト = 10 $0 \dots ((2^8) - 1) \times 2^{18} t_{CLK}$

TSTEP (0x12)

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

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

ビット・フィールド	ビット	説明
TSTEP	19:0	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、(2²⁰)-1です。 TSTEPに関連するスレッシュホールドはすべて、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。 (T_{xxx} × 15/16) – 1、または (T_{xxx} × 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。 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。

TPWMTHRS (0x13)

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

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

ビット・フィールド	ビット	説明
TPWMTHRS	19:0	これは、StealthChop2の電圧PWMモードでの上限速度です。 $TSTEP \geq TPWMTHRS$ StealthChop2 PWMモードが有効化されます（設定されている場合）

TCOOLTHRS (0x14)

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

ビット・フィールド	ビット	説明
TCOOLTHRS	19:0	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ュホルド速度です。（符号なし） 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がディセーブルされます。この速度を超えるとストール出力信号がイネーブルされ、速度がこの閾値未満になると再度ディセーブルされます。 $TCOOLTHRS \geq TSTEP \geq THIGH$: - CoolStepがイネーブルされます（設定されている場合） $TCOOLTHRS \geq TSTEP$ - 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）

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	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) ストール検出機能は、THIGHスレッシュホールドを通過する場合は必ず2〜3電氣的周期分スイッチ・オフされ、スイッチング・モードの影響を補償します。 $TSTEP \leq THIGH$: - CoolStepは無効化されます (モータは通常電流スケールで動作) - StealthChop2電圧PWMモードは無効化されます <i>vhighchm</i>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 <i>chm</i> = 1に切り替わります。 <i>vhighfs</i>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、フルステップ・モードのストール検出に切り替わります。

DIRECT_MODE (0x2D)

BIT	31	30	29	28	27	26	25	24
Field	—	—	—	—	—	—	—	DIRECT_C OIL_B[8]
Reset	—	—	—	—	—	—	—	
Access Type	—	—	—	—	—	—	—	Write, Read

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

ビット・フィールド	ビット	説明
DIRECT_COIL_B	24:16	GCONFでダイレクト・モードが選択されているとき : 符号付きのコイルB電流
DIRECT_COIL_A	8:0	GCONFでダイレクト・モードが選択されているとき : 符号付きのコイルA電流

ENCMODE (0x38)

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

ビット・フィールド	ビット	説明	デコード
enc_sel_decimal	10	エンコーダ・プリスケアラのモード選択	0x0: Encoder prescaler divisor binary mode: Counts <i>ENC_CONST</i> (fractional part) /65536 0x1: Encoder prescaler divisor decimal mode: Counts in <i>ENC_CONST</i> (fractional part) /10000
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
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
Field	ENC_CONST[15:8]							
Reset	0x10000							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	ENC_CONST[7:0]							
Reset	0x10000							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
ENC_CONST	31:0	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 X_ENCで加算される量は以下のとおりです $\pm \text{ENC_CONST} / (2^{16} \times X_ENC)$（2進数） または $\pm \text{ENC_CONST} / (10^4 \times X_ENC)$（10進数） ENC_MODEのenc_sel_decimal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。 2進数： $\pm [\mu\text{steps}/2^{16}]$ $\pm(0 \dots 32767.999847)$ decimal: $\pm(0.0 \dots 32767.9999)$ リセット後のデフォルト= 1.0 (=65536)

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

ビット・フィールド	ビット	説明	デコード
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
Field	ENC_LATCH[31:24]							
Reset	0x0							
Access Type	Read Only							
BIT	23	22	21	20	19	18	17	16
Field	ENC_LATCH[23:16]							
Reset	0x0							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	ENC_LATCH[15:8]							
Reset	0x0							
Access Type	Read Only							

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

ビット・フィールド	ビット	説明
ENC_LATCH	31:0	Nイベント時にラッチされたエンコーダ位置X_ENC

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							

ビット・フィールド	ビット	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	VS電圧の実際の値（ローパス・フィルタを使用）。 更新レート：2048クロックごと $V_S = ADC_VSUPPLY * 9.732mV$

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

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	過熱警告スレッシュホールド・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)
OVERVOLTAGE_VTH	12:0	出力OV用の過電圧スレッシュホールド。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							

ビット・フィールド	ビット	説明
MSLUT_0	31:0	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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
Field	MSLUT_2[23:16]							
Reset	0x24492929							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_2[15:8]							
Reset	0x24492929							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_2[7:0]							
Reset	0x24492929							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
MSLUT_2	31:0	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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
Field	MSLUT_3[31:24]							
Reset	0x10104222							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_3[23:16]							
Reset	0x10104222							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_3[15:8]							
Reset	0x10104222							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_3[7:0]							
Reset	0x10104222							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
MSLUT_3	31:0	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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
Field	MSLUT_4[31:24]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_4[23:16]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_4[15:8]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_4[7:0]							
Reset	0xFBFFFFFF							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
MSLUT_4	31:0	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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

ビット・フィールド	ビット	説明
MSLUT_5	31:0	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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
Field	MSLUT_6[31:24]							
Reset	0x49295556							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	MSLUT_6[23:16]							
Reset	0x49295556							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_6[15:8]							
Reset	0x49295556							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_6[7:0]							
Reset	0x49295556							
Access Type	Write, Read							

ビット・フィールド	ビット	説明
MSLUT_6	31:0	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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	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。 0: W= %00: -1 %01: +0 %10: +1 %11: +2 1: W= %00: +0 %01: +1 %10: +2 %11: +3 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。 ofs31, ofs30, ..., ofs01, ofs00 ... ofs255, ofs254, ..., ofs225, ofs224 リセット後のデフォルト = サイン波テーブル

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	LUTのセグメント1の開始 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Wxを用いて、最大4つのセグメントに分割できます。セグメントの境界は、X1、X2、X3で選択できます。 セグメント0は0～X1-1の範囲です。 セグメント1はX1～X2-1の範囲です。 セグメント2はX2～X3-1の範囲です。 セグメント3はX3～255の範囲です。 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。 0<X1<X2<X3
X2	23:16	LUTのセグメント1の開始 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Wxを用いて、最大4つのセグメントに分割できます。セグメントの境界は、X1、X2、X3で選択できます。 セグメント0は0～X1-1の範囲です。 セグメント1はX1～X2-1の範囲です。 セグメント2はX2～X3-1の範囲です。 セグメント3はX3～255の範囲です。 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。 0<X1<X2<X3
X1	15:8	LUTのセグメント1の開始 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Wxを用いて、最大4つのセグメントに分割できます。セグメントの境界は、X1、X2、X3で選択できます。 セグメント0は0～X1-1の範囲です。 セグメント1はX1～X2-1の範囲です。 セグメント2はX2～X3-1の範囲です。 セグメント3はX3～255の範囲です。 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。 0<X1<X2<X3
W3	7:6	ofs(X3)～ofs255でのLUT幅の選択 幅制御ビットのコーディングW0～W3： %00： MSLUTの項目が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
W2	5:4	ofs(X2)～ofs(X3-1)でのLUT幅の選択 幅制御ビットのコーディングW0～W3： %00： MSLUTの項目が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

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~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 幅制御ビットのコーディングW0~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
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	MSCNT[9:8]	
Reset	–	–	–	–	–	–	0x0	
Access Type	–	–	–	–	–	–	Read Only	
BIT	7	6	5	4	3	2	1	0
Field	MSCNT[7:0]							
Reset	0x0							
Access Type	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_Bのマイクロステップ・テーブルに示します。CUR_AI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
Field	–	–	–	–	–	–	–	CUR_A[8]
Reset	–	–	–	–	–	–	–	0xF7
Access Type	–	–	–	–	–	–	–	Read Only
BIT	23	22	21	20	19	18	17	16
Field	CUR_A[7:0]							
Reset	0xF7							
Access Type	Read Only							
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	–	CUR_B[8]
Reset	–	–	–	–	–	–	–	0x0
Access Type	–	–	–	–	–	–	–	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（電流によるスケールリングなし）。
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（電流によるスケールリングなし）。

CHOPCONF (0x6C)

BIT	31	30	29	28	27	26	25	24
Field	diss2vs	diss2g	dedge	intpol	MRES[3:0]			
Reset	0x0	0x0	0x0	0x1	0x0			
Access Type	Write, Read	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	電源短絡保護をディスエーブル	0x0: Short to VS protection is on 0x1: Short to VS protection is disabled
diss2g	30	GND短絡保護をディスエーブル	0x0: Short to GND protection is on 0x1: Short to GND protection is disabled
dedge	29	ダブル・エッジ・ステップ・パルスをイネーブル	0x0: disabled 0x1: Enable step impulse at each step edge to reduce step frequency requirement.
intpol	28	256マイクロステップへの補間	0x0: no interpolation 0x1: The actual microstep resolution (MRES) becomes extrapolated to 256 microsteps for smoothest motor operation

ビット・フィールド	ビット	説明	デコード
MRES	27:24	マイクロステップの分解能の選択 この分解能は、1/4サイン波あたりのマイクロステップ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 = 2^{MRES} [microsteps]	0x0: Native 256 microstep setting. 0x1: 128 0x2: 64 0x3: 32 0x4: 16 0x5: 8 0x6: 4 0x7: 2 0x8: FULLSTEP 0x9: unused
TPFD	23:20	受動高速減衰時間 TPFDを用いることで、モータの中間領域での共振の緩和が可能です。 受動高速減衰時間の設定により、ブリッジの極性変更後に挿入される高速減衰フェーズの時間が制御されます。 $N_{\text{CLK}} = 128 \times \text{TPFD}$ %0000 : 無効 %0001 ... %1111: 1 ... 15	
vhighchm	19	高速チョッパー・モード このビットは、VHIGHを超えた場合にchm = 1およびfd = 0への切り替わりをイネーブルします。したがって、より高い速度が達成できます。vhighfs = 1と組み合わせることができます。このビットがセットされた場合、チョッパー周波数が2倍になるのを防止するため、TOFF設定は高速動作時に自動的に2倍になります。	
vhighfs	18	高速フルステップ選択 このビットは、VHIGHを超えた場合にフルステップへの切り替わりをイネーブルします。切り替わりは45°の位置でのみ生じます。フルステップの目標電流は、45°位置でのマイクロステップ・テーブルからの電流値を用います。	
TBL	16:15	TBLブランキング時間設定。 コンパレータのブランキング時間をクロック・サイクル数単位で設定します。 ヒント：ほとんどのアプリケーションには24クロックまたは36クロックを推奨します。 TBL = 0x0の場合の制限：8MHz以下の外部クロック発振器を組み合わせる場合のみ使用できます TBL = 0x1の場合の制限：内部クロックまたは13MHz以下の外部クロックを適用する場合に使用できます。	0x0: 16 clocks 0x1: 24 clocks 0x2: 36 clocks 0x3: 54 clocks
chm	14	チョッパー・モードの選択。 これが有効になるのは、en_pwm_modeが0に設定されている場合か $\text{TSTEP} < \text{TPWMTHRS}$ 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

ビット・フィールド	ビット	説明	デコード
fd3	11	TFD[3] <i>chm</i> = 1の場合 : 高速減衰時間設定 <i>TFD</i> の MSB	
HEND_OFFSET	10:7	<i>chm</i> = 0の場合 : <i>HEND</i> = 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・チョッパ用 に用いられるヒステリシス値です。 <i>chm</i> = 1の場合 : <i>OFFSET</i> = 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。	
HSTRT_TFD210	6:4	<i>chm</i> = 0の場合 : <i>HEND</i> に加算されるヒステ リシス開始値 <i>HSTRT</i> %000~%111 : ヒステリシスの低い値 <i>HEND</i> 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) <i>注意</i> : 実効 $HEND + HSTRT \leq 16$ 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 <i>chm</i> = 1の場合 : 高速減衰時間設定値 <i>TFD</i> [2..0] 高速減衰時間設定値 (MSB : <i>fd3</i>) : %0000~%1111 : $N_{CLK} = 32 \times TFD$ の高速減衰時間設定値 <i>TFD</i> (%0000 : 低速減衰のみ)	
TOFF	3:0	オフ時間 <i>TOFF</i> 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 $N_{CLK} = 24 + 32 \times TOFF$ %0000 : ドライバ・ディスエーブル、全ブリ ッジ・オフ %0001 : 1 – <i>TBL</i> ≥ 2の場合にのみ使用 %0010~%1111 : 2~15	

COOLCONF (0x6D)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	sfilt
Reset	–	–	–	–	–	–	–	0x0
Access Type	–	–	–	–	–	–	–	Write, Read

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

ビット・フィールド	ビット	説明	デコード
sfilt	24	StallGuard2フィルタをイネーブ。ル。	0x0: Standard mode, high time resolution for StallGuard 0x1: Filtered mode, StallGuard signal updated for each four fullsteps only to compensate for motor pole tolerances
sgt	22:16	StallGuard2のスレッシュホルド値 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>IRUN</i>) (when used with StealthChop requires <i>IRUN</i> ≥ 16) 0x1: 1/4 of current setting (<i>IRUN</i>) (when used with StealthChop requires <i>IRUN</i> ≥ 28)
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($SEMIN + SEMAX + 1$) × 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および電流イネーブルのための最小StallGuard2値 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	

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	s2vsa	–	–	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	0x0: normal operation 0x1: Open load detected on phase B. <i>Hint:</i> 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	0x0: normal operation 0x1: Open load detected on phase A. <i>Hint:</i> 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.
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 (<i>TOFF</i> = 0) or by the ENN input.
s2ga	27	位相Aのグラウ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 (<i>TOFF</i> = 0) or by the 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 operatoin) or fullstep stall (in fullstep 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、 <i>COOLCONF</i> 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

ビット・フィールド	ビット	説明	デコード
stealth	14	StealthChop2インジケータ	0x0: StealthChop2 not active 0x1: Driver operates in StealthChop2 mode
s2vsb	13	位相Bの電源への短絡インジケータ	0x0: no error 0x1: Short to supply detected on phase B. The driver becomes disabled. The flags stay active, until the driver is disabled by software (<i>TOFF</i> = 0) or by the ENN input.
s2vsa	12	位相Aの電源への短絡インジケータ	0x0: no error 0x1: Short to supply detected on phase A. The driver becomes disabled. The flags stay active, until the driver is disabled by software (<i>TOFF</i> = 0) or by the ENN input.
SG_RESULT	9:0	StallGuard2の結果またはStallGuard4の結果（実際のチョッパ・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 機械的負荷測定： StallGuard2/4の結果は、機械的なモータ負荷を測定する手段を与えます。値が大きいと機械的負荷が小さいことを意味します。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。 StallGuard2は、マイクロステップ動作で最もよく機能します。 SpreadCycleモード時の温度測定： 停止時には、StallGuard2の結果は取得できません。代わりに、SG_RESULTは、モータ・コイルAのチョッパ・オン時間を示します。モータを定められたマイクロステップ位置に特定の電流設定値で動作させると、チョッパ・オン時間を読み出すことでモータ温度を大まかに推定できます。モータ温度が上昇するにつれ、コイル抵抗が増し、チョッパ・オン時間も増加します。StallGuard4の詳細については、SG4_RESULTを参照してください。	

PWMCONF (0x70)

BIT	31	30	29	28	27	26	25	24
Field	PWM_LIM[3:0]				PWM_REG[3:0]			
Reset	0xC				0x4			
Access Type	Write, Read				Write, Read			

BIT	23	22	21	20	19	18	17	16
Field	pwm_dis_reg_stst	pwm_meas_sd_enable	FREEWHEEL[1:0]		pwm_autograd	pwm_autoscale	PWM_FREQ[1:0]	
Reset	0x0	0x0	0x0		0x1	0x1	0x0	
Access Type	Write, Read	Write, Read	Write, Read		Write, Read	Write, Read	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	PWM_GRAD[7:0]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	PWM_OFS[7:0]							
Reset	0x1D							
Access Type	Write, Read							

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

ビット・フィールド	ビット	説明	デコード
FREEWHEEL	21:20	様々な停止モードを可能にします 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。 %00 : 通常動作 %01 : フリーホイーリング %10 : LSドライバを用いたコイル短絡 %11 : HSドライバを用いたコイル短絡	
pwm_autograd	19	PWM自動勾配調整	0x0: Fixed value for PWM_GRAD ($PWM_GRAD_AUTO = PWM_GRAD$) 0x1: Automatic tuning (only with $pwm_autoscale = 1$) (Reset default) PWM_GRAD_AUTO is initialized with PWM_GRAD while $pwm_autograd=0$ and becomes optimized automatically during motion. <u>Preconditions</u> PWM_OFS_AUTO has been automatically initialized. This requires standstill at $IRUN$ for >130ms in order to a) detect standstill b) wait > 128 chopper cycles at $IRUN$ and c) regulate PWM_OFS_AUTO so that $-1 < PWM_SCALE_AUTO < 1$ - Motor running and $1.5 \times PWM_OFS_AUTO \times (IRUN+1)/32 < PWM_SCALE_SUM < 4 \times PWM_OFS_AUTO \times (IRUN+1)/32$ and $PWM_SCALE_SUM < 255$. <u>Time required for tuning PWM_GRAD_AUTO</u> About 8 fullsteps per change of +/-1. Also enables use of reduced chopper frequency for tuning PWM_OFS_AUTO.
pwm_autoscale	18	PWM自動振幅スケーリング	0x0: User defined feed forward PWM amplitude. The current settings $IRUN$ and I_{HOLD} have no influence! The resulting PWM amplitude (limited to 0...255) is: $PWM_OFS * ((CS_ACTUAL+1) / 32) + PWM_GRAD * 256 / TSTEP$ 0x1: Enable automatic current control (Reset default)
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: $f_{PWM} = 2/410 f_{CLK}$	

ビット・フィールド	ビット	説明	デコード
PWM_GRAD	15:8	PWM振幅の速度依存勾配： $PWM_GRAD \times 256 / TSTEP$ 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PWM_OFSに加算されます。 自動スケーリングの初期値としてPWM_GRADを用いると、自動調整プロセスを加速できます。これを行うには、$pwm_autoscale = 0$とし、PWM_GRADを定められたアプリケーション固有の値に設定します。その後のみ、$pwm_autoscale = 1$に設定します。終了後はStealthChop2をイネーブルします。 ヒント： 初期調整の後、必要な初期値はPWM_GRAD_AUTOで読み出せます。	
PWM_OFS	7:0	停止時のモータのフル電流（CS_ACTUAL = 31）に関連したユーザ定義によるPWM振幅オフセット（0~255）。 （リセット後のデフォルト = 30） 自動スケーリングの初期値としてPWM_OFSを用いると、自動調整プロセスを加速できます。これを行うには、$pwm_autoscale = 0$とし、PWM_OFSを定められたアプリケーション固有の値に設定します。その後のみ、$pwm_autoscale = 1$に設定します。終了後はStealthChop2をイネーブルします。 PWM_OFS = 0とすると、モータ電流をモータ固有の下限測定スレッシュホー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なくなります。 PWM_OFS > 0とすると、PWMデューティ・サイクルを自動的にスケール・ダウンできます。この場合、下限レギュレーション・スレッシュホールド未満にすることも可能です。それにより、実際の（ホールド）電流スケール（IHOLD_IRUNレジスタ）に基づいて低（停止）電流設定値にすることができます。	

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

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

PWM_AUTO (0x72)

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

BIT	31	30	29	28	27	26	25	24
Field	—	—	—	—	—	—	—	—
Reset	—	—	—	—	—	—	—	—
Access Type	—	—	—	—	—	—	—	—
BIT	23	22	21	20	19	18	17	16
Field	PWM_GRAD_AUTO[7:0]							
Reset	0x0							
Access Type	Read Only							

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

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

SG4_THRS (0x74)

BIT	31	30	29	28	27	26	25	24
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	sg_angle_of fset	sg4_filt_en
Reset	–	–	–	–	–	–	0x1	0x0
Access Type	–	–	–	–	–	–	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	SG4_THRS[7:0]							
Reset	0x0							
Access Type	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

ビット・フィールド	ビット	説明
SG4_THRS	7:0	ストールの検出スレッシュホールド。StallGuard4値SG4_RESULTがこのスレッシュホールドと比較されます。 ストール信号は $SG4_RESULT \leq SG4_THRS$ で出力されます。 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
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	23	22	21	20	19	18	17	16
Field	–	–	–	–	–	–	–	–
Reset	–	–	–	–	–	–	–	–
Access Type	–	–	–	–	–	–	–	–
BIT	15	14	13	12	11	10	9	8
Field	–	–	–	–	–	–	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	StallGuard4専用のStallGuard結果。 SG4_RESULTは、TCOOLTHRSおよびSG4THRSとは無関係に、フルステップごとに更新されます。値が高いと、モータ負荷が低く、トルク・ヘッドルームが大きいことを意味します。 StealthChop2モードのみを対象としたものです。ビット9およびビット0は常に0を示します。10ビットにスケールアップされているのはStallGuard2との互換性を確保するためです。

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	SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定3の値を示します。
SG4_IND_2	23:16	SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定2の値を示します。
SG4_IND_1	15:8	SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定1の値を示します。
SG4_IND_0	7:0	SG4測定の値を示します SG4_filt_en = 1の場合 : フィルタ入力として用いられるSG4測定0の値を示します。

標準アプリケーション回路

標準的なアプリケーション回路

標準的なアプリケーション回路では、最低限のセットの追加コンポーネントを使用します。電源をフィルタにより除去するには、低 ESR の電解コンデンサを使用します。コンデンサは、チョッパ動作によって生じる電流リップルを処理する必要があります。最適な性能を得るために、 V_S に最小 100 μ F のコンデンサを用いることを推奨します。電源コンデンサの電流リップルは、電源の内部抵抗とケーブル長にも依存します。 V_{CC_IO} は、低ドロップアウト 3.3V 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、べた膜の共通グラウンド・プレーンを用います。 V_{DD1V8} フィルタリング・コンデンサは、 V_{DD1V8} ピンに直接接続します。

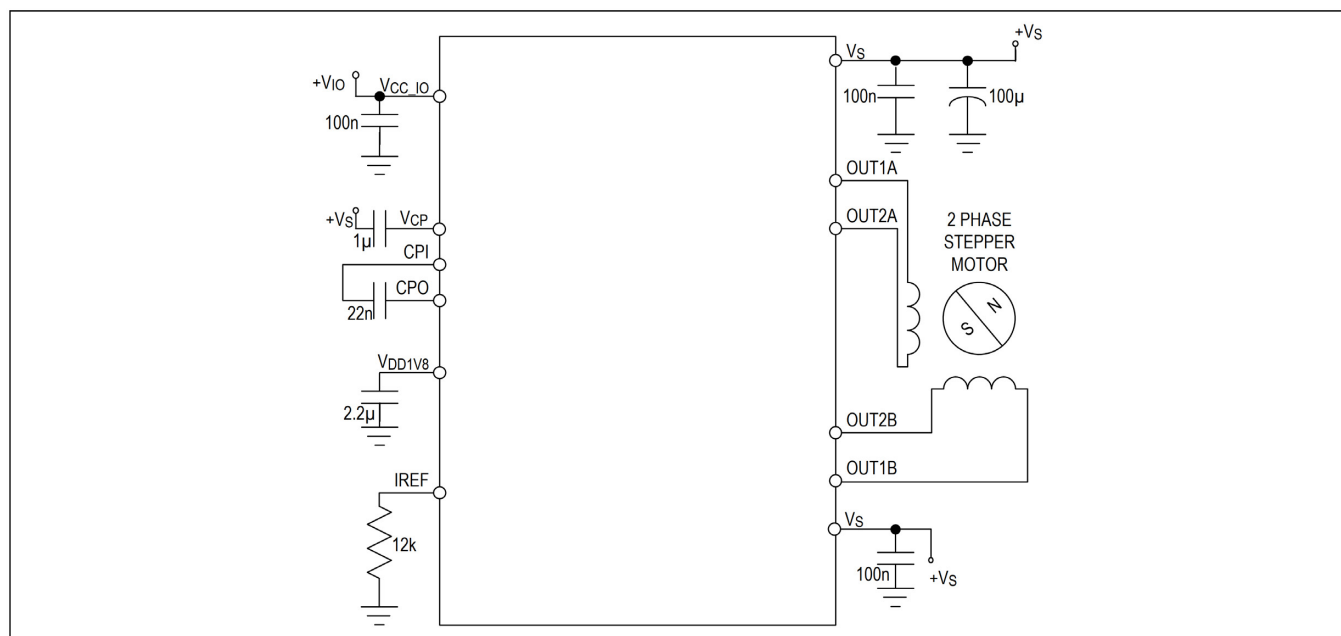


図 35. 標準的なアプリケーション回路

大モータ電流

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

おおよその目安として、PCB 設計の熱的性質は、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・チップは ESD 回路を内蔵していますが、動作時に発生する ESD イベントは、エネルギーによっては、リセットの原因となったり、モータ・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 グラウンドに取り付けたり、導電性プラスチック部品を用いたりするなど、取り付け部品にはすべて導電体で構成されたものを用いることで、ESD イベントを防止するのがベスト・プラクティスです。更に、ドライバは、ESD イベントまたは、モータ・コネクタ端子に高電圧大電流が供給される原因にもなるモータの活線挿抜に対して、ある程度までは保護可能です。

標準アプリケーション回路（続き）

簡単な方法は、ドライバ出力にコンデンサを用いて ESD イベントによる dV/dt を低減することです。コンデンサの値が大きいほど、ESD 抑制に関する利点は大きくなりますが、各チョッパ・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。

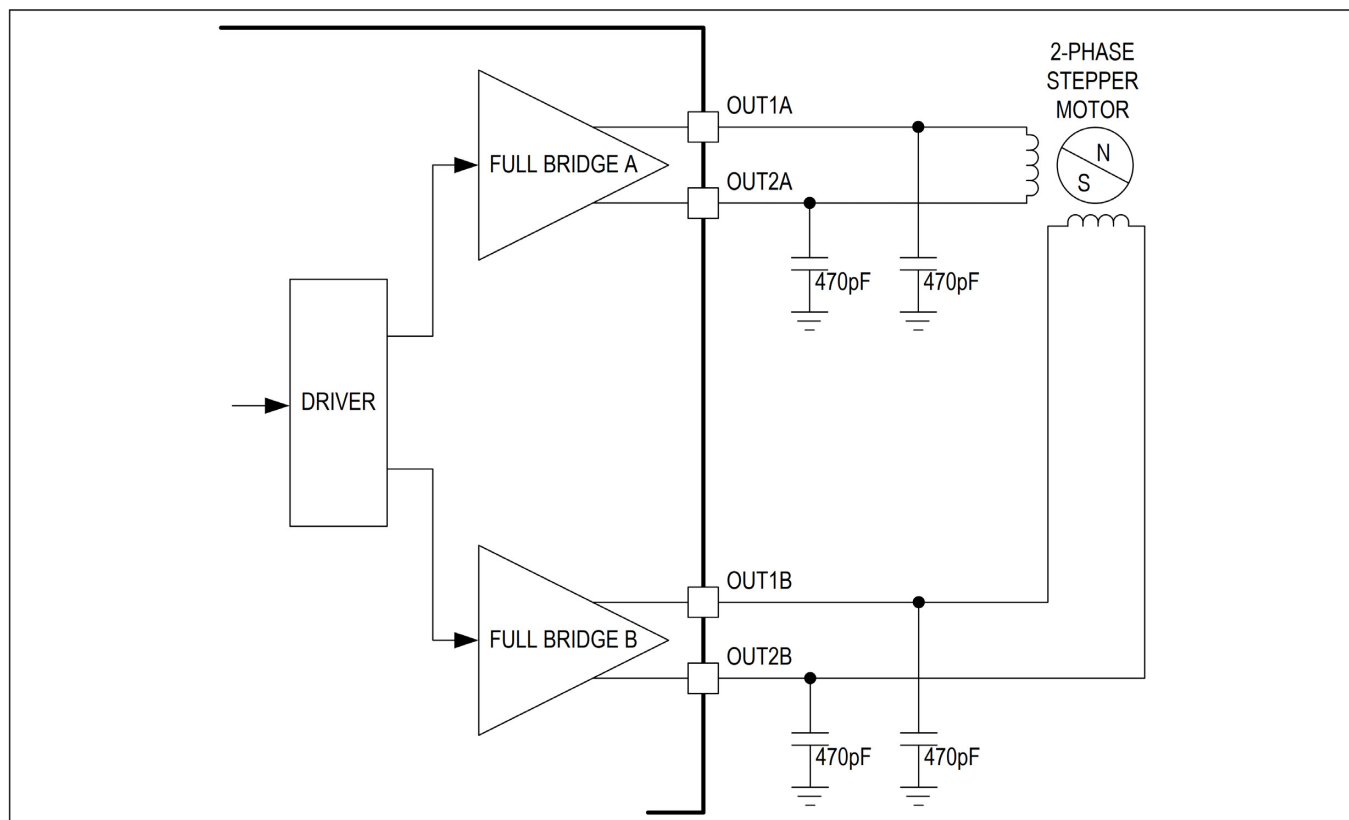


図 36. 簡単な ESD 対策

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・コネクタとデカップリングすることです。コイル端子間のバリスタ V1 および V2 は、活線接続に起因するコイルの過電圧をなくします。オプションで、バリスタ (V1A、V1B、V2A、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）

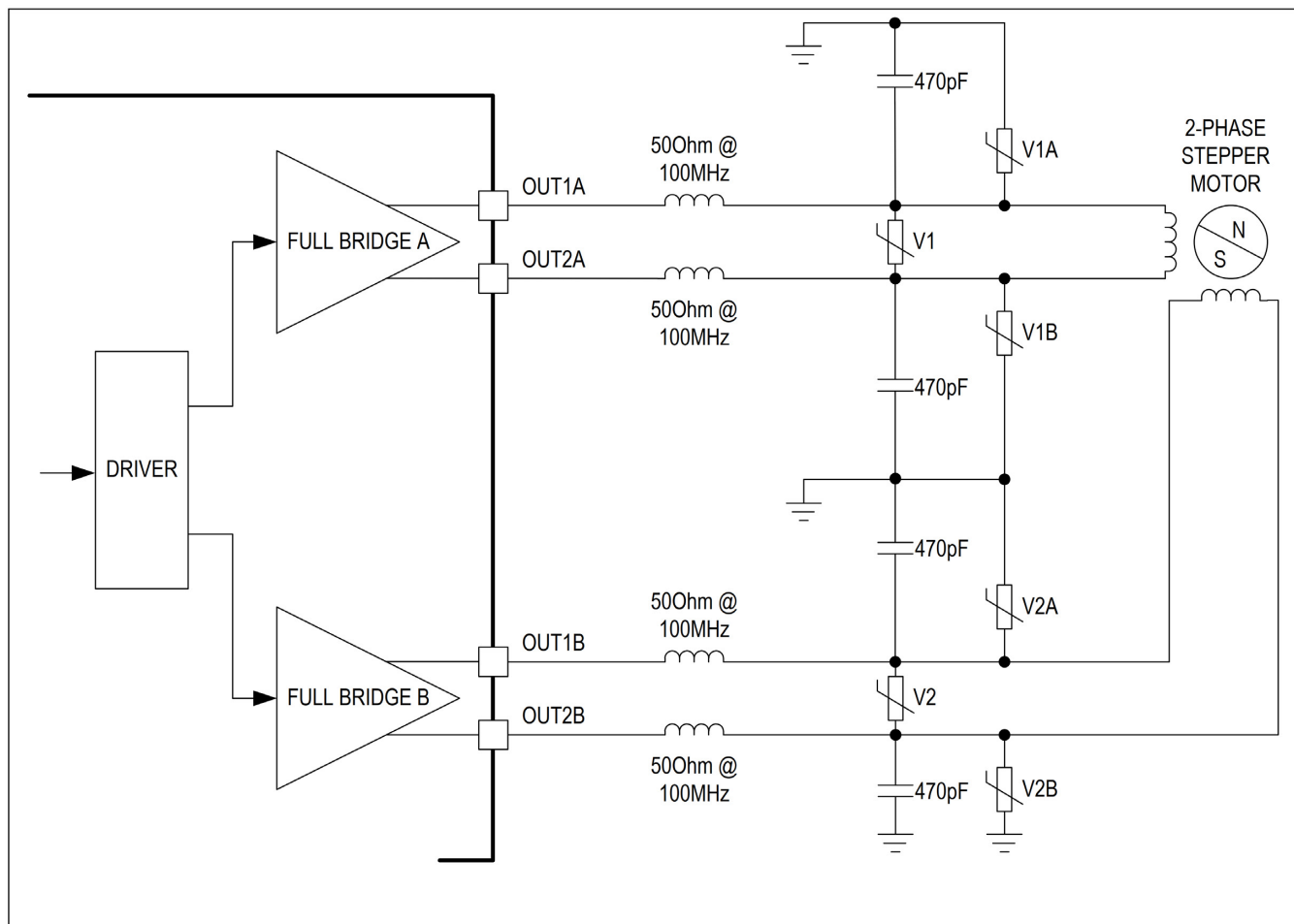


図 37. モータ出力保護の強化

型番

PART NUMBER	TEMPERATURE RANGE	PIN-PACKAGE
TMC2240ATJ+	-40°C to +125°C	32 TQFN - 5mm x 5mm
TMC2240ATJ+T	-40°C to +125°C	32 TQFN - 5mm x 5mm
TMC2240AUU+	-40°C to +125°C	38 TSSOP-EP 4.4mm x 9.7mm
TMC2240AUU+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 = テープ & リール。

改訂履歴

版数	改訂日	説明	改訂ページ
0	7/22	市場投入のためのリリース	–
1	2/23	電气的特性、端子説明、レジスタ・マップ、型番を変更。 図 2、図 9、図 12、図 22 を更新。 表 11～表 16、および表 18～表 22 を更新。 StealthChop2、フルスケール電流レンジの設定、StallGuard4 の負荷測定、CoolStep の負荷適応型電流スケールリング、過電圧保護と OV ピンのセクションを更新、標準アプリケーション回路を変更。 外部アナログ入力 AIN のモニタリング、クイック設定ガイドのセクションを新規追加。	9-13, 15-17, 75-123, 126 19, 32, 34, 53 32, 40, 42, 44, 46, 48, 50, 51, 53, 55 30-41, 47-48, 52-57, 66, 124 67-68, 68-73
2	11/23	開放負荷フラグのセクションを更新 – 説明を追加、表 18 と表 23 を更新、CoolStep の調整のセクションに注を追加、過電圧保護と OV ピンのセクションを更新、図 33 を更新、COOLCONF レジスタを更新	39, 48, 56, 57, 66, 72, 113