提供评估套件

TMC5240

36V 3Amax 智能集成步进驱动+控制器

基本描述

TMC5240是一款智能高性能步进电机控制器和驱动芯片，具有串行通信接口(SPI, UART)和广泛的诊断能力. 结合了一个灵活的, 优化的抑制抖动斜坡发生器 用于自动定位，采用行业最先进的步进电机驱动技术，该驱动基于 256 微步细分内置分度器和完全集成的 36V、3.0AMAX H桥以及非耗散集成电流传感器 (ICS).

TRINAMIC 精密的stealthChop2TM 斩波器确保绝对无噪音运行，同时具有最高效率和最佳电机转矩.

高集成度、高能效和小外形尺寸可实现小型化和可扩展系统，从而实现经济高效的解决方案。 完整的解决方案可将开发工作量降至最低，同时提供一流的性能。

H 桥 FET 具有非常低的阻抗，从而带来高驱动效率和最少的热量产生。典型的总 RDson（高侧 + 低侧）为 0.23Ω。

每个 H 桥的最大输出电流为 IMAX=5.0AMAX，受过流保护 (OCP) 限制。

假设 4 层 PCB，室温下每个 H 桥的最大 RMS 电流为 IRMS=2.1ARMS。

由于该电流受热因素的限制，实际最大 RMS 电流将取决于应用的热特性（PCB 接地层、散热器、通风等）。

每个 H 桥的最大满量程电流为 IFS= 3.0A，可以通过连接到 IREF 的外部电阻器进行设置. 此电流定义为嵌入式电流驱动调节电路的最大电流设置。

与基于外部检测电阻器的主流应用相比，非耗散 ICS 消除了笨厚的外部功率电阻器，从而显着节省空间和功耗。

TMC5240 具有丰富的诊断和保护功能，例如短路保护/OCP、热关断、欠压锁定。

在热关断和欠压锁定事件期间，驱动器被禁用。

此外，TMC5240 还提供测量驱动器温度、监测电机温度以及测量一个外部模拟输入的功能。TMC5240采用小型TQFN32 5x5mm封装和经过散热优化的TSSOP38 9.7x4.4

应用

- 纺织机、缝纫机、针织机
- 实验室和工厂自动化
- 3D 打印机、ID 打印机/卡片打印机
- 移液处理，医疗设备
- 办公自动化和纸张处理
- POS、赌博机、按摩椅、
- ATM、点钞机、票据验证器、自动取款机
- CCTV, 安防
- 泵和阀门控制
- 定日镜和天线定位

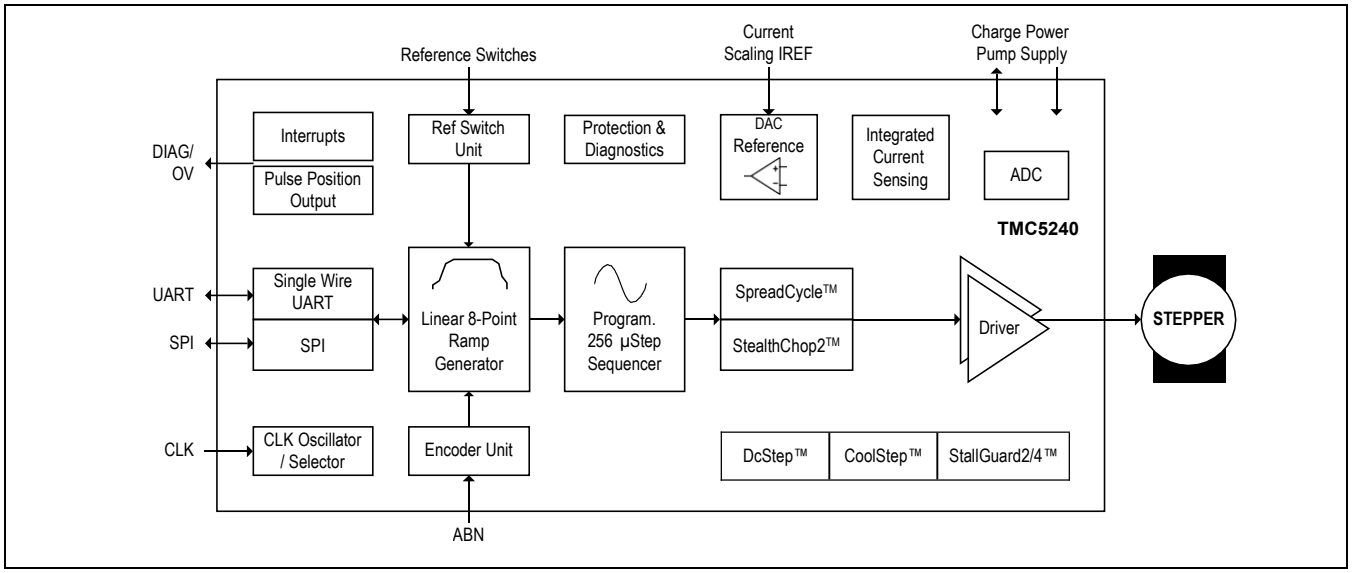
优点和特点

- 电压范围 4.5… 36V DC
- 低 Rdson (HS+LS)值：典型值为 230 mΩ (TA=25C)
- 每个 H 桥的额定电流（25C 时的典型值）：
  - IMAX=5.0A (bridge peak current)
  - IRMS=2.1ARMS (3A sine wave peak)
- 完全集成的无损电流检测 (ICS)
- EightPoint™ 运动轨迹曲线，减少定位过程中的抖动
- SPI & 单线 UART
- 编码器接口和 2x 参考开关输入
- 最高分辨率 256 微步/整步
- 灵活的波形表 & 相序以匹配电机
- StealthChop2™ 静音电机操作
- SpreadCycle™ 高动态电机控制斩波器
- StealthChop 和 SpreadCycle 的无抖动切换
- DcStep™速度自适应负载控制
- StallGuard2™ & StallGuard4™无传感器电机负载检测
- CoolStep™ 电流控制可节省高达 75% 的能源
- 被动制动和自由旋转模式
- 电机相和芯片温度测量
- 通用模拟量输入
- 全面保护和诊断
- 过压保护输出
- 紧凑的 5x5 QFN32 或 9.7x4.4 TSSOP38封装

PRELIMINARY CONFIDENTIAL

**本中文版数据手册是英文版手册的译文，敬请谅解翻译中可能存在语言组织或翻译错误，如需确认任何语言的准确性，请参考最新英文版书册确认！

简化框图



PRELIMINARY CONFIDENTIAL

目 录

PRELIMINARY CONFIDENTIAL

目 录

PRELIMINARY CONFIDENTIAL

PRELIMINARY CONFIDENTIAL

目 录

PRELIMINARY CONFIDENTIAL

LIST OF FIGURES

PRELIMINARY CONFIDENTIAL

LIST OF TABLES

PRELIMINARY CONFIDENTIAL

绝对最大额定值

VS to GND.....V to 41V
VDD to GND.....V to min(2.2,VS+0.3)V
AGND to GND..... V to +0.3V
OUT1A, OUT2A, OUT1B, OUT2B.....V to VS+0.3V
VCP to GND.....VS-0.3V to min(44, VS+6)V
CPO to GND.....VS-0.3V to min(44, VS+6)V
CPI to GND..... V to min(41, VS+0.3)V
SLEEPN to GND.....V to VS+0.3V

IREF, AIN to GND..... V to min(2.2,VDD+0.3)V
VCC_IO to GND..... V to 5.5V
Logic input/output voltage to GND..... V to VCC_IO+0.3V
OV to GND.....V to 6V
工作温度范围.....-40°C to 125°C
结温..... +160°C
储存温度范围.....-65°C to +150°C
焊接温度（回流）.....+260°C

注意，超出上述绝对最大额定值可能会导致器件永久性 损坏。这只是额定最值，并不能以这些条件或者在任何其 他超出本技术规范操作章节中所示规格的条件下，推断器 件能否正常工作。长期在绝对最大额定值条件下工作会影 响器件的可靠性。

封装信息

TQFN32 5x5

包装代码	T3255+5C
Outline Number	21-0140
Land Pattern Number	90-0013
热阻，单层板：	
PN结到环境的温度(θ_{JA})	47°C/W
PN结至器件外壳的温度 (θ_{JC})	1.7°C/W
热阻，四层板：	
PN结到环境的温度 (θ_{JA})	29°C/W
PN结至器件外壳的温度 (θ_{JC})	1.7°C/W

TSSOP38 9.7x4.4 EP

包装代码	U38E+3C
Outline Number	21-0714
Land Pattern Number	90-0435
热阻，四层板：	
PN结到环境的温度 (θ_{JA})	25°C/W
PN结至器件外壳的温度 (θ_{JC})	1°C/W

For the latest package outline information and land patterns (footprints), go to *www.maximintegrated.com/packages*. Note that a "+", "#", or "-" in the package code indicates RoHS status only. Package drawings may show a different suffix character, but the drawing pertains to the package regardless of RoHS status.

Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to *www.maximintegrated.com/thermal-tutorial*.

电气特性

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ , 典型值假设 TA=25C and VS=24V, Limits are 100% tested at T_A = +25°C. 工作温度范围和相关电源电压范围的限制由设计和特性保证。
标有"GBD"的规格由设计保证，未经生产测试.)

参数	符号	条件适用	最小值	典型值	最大值	单位
供电电源						
供电电压范围	VS		4.5		36	V
睡眠模式电流消耗	IVS	V(SLEEPN)=0		4	18	uA

PRELIMINARY CONFIDENTIAL

电气特性（待续）

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ , 典型值假设 TA=25C and VS=24V, Limits are 100% tested at T_A = +25°C. L工作温度范围和相关电源电压范围的限制由设计和特性保证。标有“GBD”的规格由设计保证，未经生产测试。）

参数	符号	条件适用	最小值	典型值	最大值	单位
静态电流消耗	IVS	V(SLEEPN)=1, V(DRV_ENN)=1		2.3	4	mA
1.8V稳压器输出电压	VVDD	VS=4.5V		1.8		V
VDD 电流限制	$IV18_{LIM}$		20			mA
电荷泵电压	VCP			VS+2.7		V
逻辑 I/O 电源电压范围	VCC_IO		2.2		5.5	V
睡眠模式电流消耗	IVCC	V(SLEEPN)=0		5	10	uA
静态电流消耗	IVCC	V(SLEEPN)=1		35	60	uA
逻辑电平输入-输出						
输入电压电平 - 高	VIH		0.7 x VCC_IO			V
输入电压电平 - 低	VIL				0.3 x VCC_IO	V
输入滞后	VHYS			0.15 x VCC_IO		V
上拉/下拉电阻	R_{PD}	to GND or to VCC_IO	60	100	140	KΩ
输入漏电流	In_{Leak}	无上拉/下拉电阻的输入	-1		1	uA
输出逻辑低电压	VOL	ILOAD=5mA			0.4	V
推挽输出逻辑-高电压	VOH	ILOAD=5mA	VCC_IO - 400mV			
开漏输出逻辑高泄漏电流	IOH	V(PIN)=5.5V	-1		1	uA
休眠电压电平高	VIH_{SLEEPN}		0.9			V
休眠电压电平低	VIL_{SLEEPN}				0.6	V
SLEEPN 下拉输入电阻	RPD_{SLEEPN}		0.8	1.5		MΩ
输出参数						
输出导通电阻低压侧	RonLS	Full scale bits = 10		0.11	0.2	Ω
		Full scale bits = 01		0.15	0.28	
输出导通电阻低测	RonLS	Full scale bits = 00		0.28	0.54	Ω
输出导通电阻高端	RonHS			0.12	0.22	Ω
输出漏电流	ILEAK		-5		5	μA

电气特性 (待续)

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ , 典型值假设 TA=25C and VS=24V, Limits are 100% tested at T_A = +25°C. 工作温度范围和相关电源电压范围的限制由设计和特性保证。标有“GBD”的规格由设计保证，未经生产测试。）

参数	符号	条件适用	最小值	典型值	最大值	单位
输出转换速率	SR	Slew-rate bits = 00		100		V/us
		Slew-rate bits = 01		200		
		Slew-rate bits = 10		400		
		Slew-rate bits = 11		800		
Over Current Protection Threshold过流保护阈值	OCP	Full scale bits = 10	5.0			A
		Full scale bits = 01	3.33			
		Full scale bits = 00	1.67			
过流保护 Blanking Time	TOCP		0.9	1.5	2.3	µs
VS 上的 UVLO 阈值	UVLO	VS下降	3.75	3.9	4.05	V
VS 迟滞的 UVLO 阈值	UVLOHYS			0.12		V
VCC_IO 上的 UVLO 阈值	UVLO	VCC_IO falling	0.9	1.5	1.95	
VCC_IO 欠压锁定滞后	UVLOVCCH			100		mV
热保护阈值温度	TSD			165		°C
热保护温度滞后				20		°C

电流调节

IREF 引脚电阻范围	RREF		12		60	kΩ
IREF 输出电压	VREF		0.882	0.9	0.918	V
满量程电流常数	KIFS	IFS = 1A		11.75		*A* * kΩ
满量程电流常数	KIFS	IFS = 2A		24		*A* * kΩ
满量程电流常数	KIFS	IFS = 3A		36		*A* * kΩ
电流调节精度	DITRIP1	ITRIG 从 7% 到 100% FS，RREF=12kΩ	-5		5	%

计时功能

休眠时间	tSLEEP	SLEEPN=0到 OUT_ tristate			50	us
睡眠唤醒时间	TWAKE	SLEEPN=1 to normal operation			2.5	ms
开启时间	TEN	从 DRV_ENN 引脚下降沿到驱动器打开的时间			1.5	us
关闭时间	TEN	从 DRV_ENN 引脚上升沿到驱动器关闭的时间			6	us

保护电路

PRELIMINARY CONFIDENTIAL

电气特性（待续）

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ , 典型值假设 TA=25C and VS=24V, Limits are 100% tested at T_A = +25°C. 工作温度范围和相关电源电压范围的限制由设计和特性保证。标有“GBD”的规格由设计保证，未经生产测试。）

参数	符号	条件适用	最小值	典型值	最大值	单位
CLOCK时钟						
内部时钟频率	fCLKOSC		11.9	12.5	13.2	MHz
外部时钟频率	fCLK		12	16	20	MHz
外部时钟占空比	tCLKL		40		60	%
外部时钟检测周期			4		8	
内部 fCLKOSC 周期中的外部时钟超时检测			12		16	
外部时钟检测低频阈值	fCLKLO		4			MHz
SPI时序						
SCK 在 CSN 更改之前或之后有效	tCC		T_{SCLK}			ns
CSN High Time	tCSH		$4*T_{CLK}$			ns
SCK Low Time	t_{CL}		20			ns
SCK High Time	tCH		20			ns
SCK频率	f_{SCK}				10	MHz
SDI setup time before SCK rising edge	tDU		10			ns
SCK 上升沿后的 SDI 保持时间	tDH		10			ns
SCK下降沿后数据输出有效时间	tDO	VCCIO = 3.3V		27	40	ns
SDI、SCK 和 CSN 滤波器延迟时间	t_{FILT}	上升沿和下降沿		10		ns
编码器时序						
编码器计数频率	fCNT			<2/3 f_{CLK}	f_{CLK}	
A/B/N 输入低电平时间	tABNL		$3t_{CLK}$ + 20			ns
A/B/N 输入高电平时间	tABNH		$3t_{CLK}$ + 20			ns
A/B/N 尖峰滤波时间	tFILTABN	上升沿和下降沿		$3t_{CLK}$		
ADC / 模拟输入 / 温度						
ADC 分辨率				13		Bit
模拟输入电压范围	VAIN		0		1.25	V
模拟输入漏电流	AINleak		-1		1	uA

PRELIMINARY CONFIDENTIAL

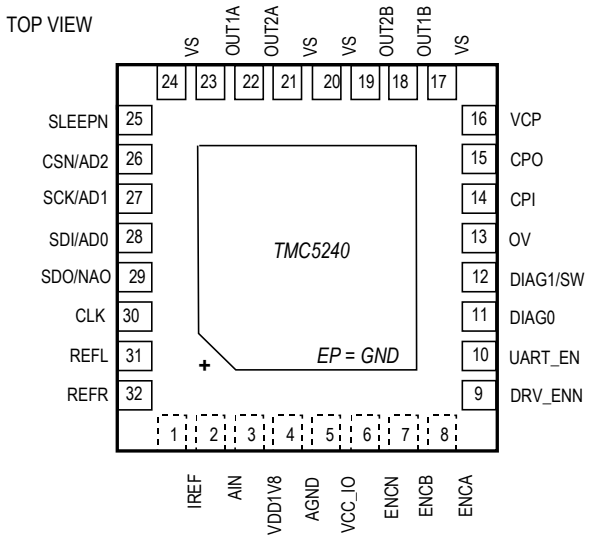
电气特性（待续）

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ , 典型值假设 TA=25C and VS=24V, Limits are 100% tested at T_A = +25°C. 工作温度范围和相关电源电压范围的限制由设计和特性保证。标有“GBD”的规格由设计保证，未经生产测试。）

参数	符号	条件适用	最小值	典型值	最大值	单位
模拟输入频率	fAIN				70	kHz
驱动器温度精度	T_{driver}			+/-10		°C
电源电压测量精度			-5		5	%
ADC 采样率	$f_{sample, ADC}$			f_{CLK} / 2048		

引脚定义

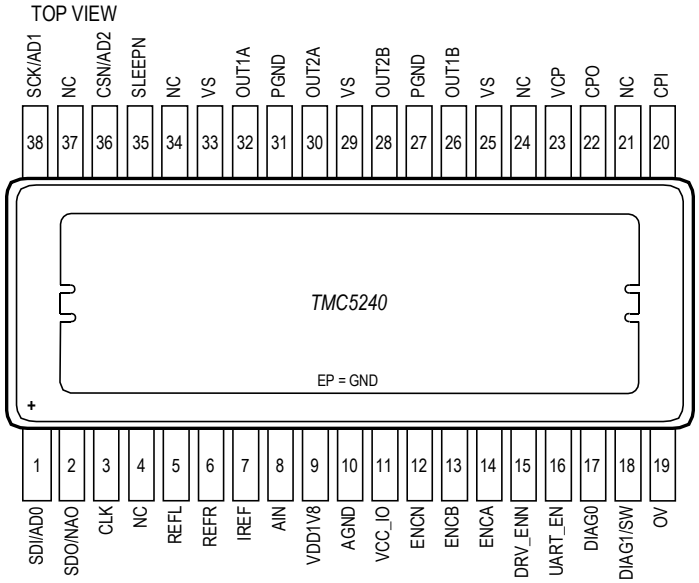
TMC5240 TQFN 引脚定义



TQFN
5mm x 5mm

PRELIMINARY CONFIDENTIAL

TMC5240 TSSOP 引脚定义



TSSOP38
4.4x9.7mm

PRELIMINARY CONFIDENTIAL

引脚说明

引脚		名称	功能	REF SUPPLY	TYPE
TQFN32	TSSOP38				
4	10	AGND	模拟地。 连接至接地层。		地层
17, 20, 21, 24	25, 29, 33	VS	电机电源电压. 放置滤波电容与贴近芯片引脚处并与地层实现最短路径		电源正
3	9	VDD1V8	内部 1.8V 稳压器的输出。将 2.2µF 或更大的陶瓷电容连接到靠近引脚的 AGND 以获得最佳性能。		Supply
16	23	VCP	电荷泵电压。 使用 1.0µF 电容连接到 VS。 将电容器的正端连接到靠近 VS 引脚的位置，以避免电感峰值。		输出
15	22	CPO	电荷泵电容输出。		输出
14	20	CPI	电荷泵电容输入。 使用 22nF 50V 电容器连接到 CPO。		输出
30	3	CLK	时钟输入。 使用短线将内部时钟连接到 GND 或提供外部时钟。内部时钟故障切换电路可防止外部时钟信号丢失。	VCC_IO	数字输入
31	5	REFL	用于内部斜坡发生器的左限位输入	VCC_IO	数字输入
32	6	REFR	用于内部斜坡发生器的右限位输入	VCC_IO	数字输入
26	36	CSN/AD2	SPI 片选输入（负有效） (UART_EN=0) 或 UART 模式下的地址输入 2 (+4) (UART_EN = 1)	VCC_IO	DI(pd)
27	38	SCK/AD1	SPI 串行时钟输入 (UART_EN=0) 或 UART 模式 (UART_EN=1) 下的地址输入 1 (+2)	VCC_IO	DI(pd)
28	1	SDI/AD0	SPI 数据输入 (UART_EN=0) 或地址输入 0 (+1) 用于单线接口 (UART_EN=1)。	VCC_IO	DI(pd)
29	2	SDO/NAO	SPI 数据输出（三态）（UART_EN=0）或单线接口（UART_EN=1）的下一个地址输出（NAO）。	VCC_IO	DI(pd)
1	7	IREF	用于电流缩放的模拟参考电流。 提供外部电阻到 GND。	VCC_IO	模拟输入
10	16	UART_EN	接口选择引脚。 当拉低时，SPI 接口被启用。 当拉高时，UART接口被启用。 集成下拉电阻。	VCC_IO	DI (pd)
7	13	ENCB	编码器 B 通道输入	VCC_IO	DI
8	14	ENCA	编码器 A 通道输入	VCC_IO	DI
6	12	ENCN	编码器 N通道输入	VCC_IO	DI
9	15	DRV_ENN	使能输入。 当该引脚被驱动为高电平时，功率级将关闭（所有电机输出被切断）。	VCC_IO	DI (pu)

PRELIMINARY CONFIDENTIAL

引脚说明（续）

PIN		名称	功能	参考供电	TYPE
TQFN32	TSSOP38				
11	17	DIAG0	诊断输出 DIAG0. 中断输出或来自内部运动控制器的 STEP 输出，用于外部驱动器. 在开漏模式下使用外部上拉电阻。 在系统复位状态下，此引脚被主动拉低，以向外部控制器指示复位条件。	VCC_IO	DO
12	18	DIAG1/SW	诊断输出 DIAG1。 位置比较或来自内部运动控制器的 DIR 输出，用于外部驱动器。 在开漏模式下使用外部上拉电阻。 UART 模式下的单线 I/O。	VCC_IO	DIO
25	25	SLEEPN	低电平有效掉电输入/复位输入. 提供连续低电平使设备进入睡眠模式。 SLEEPN 有一个内部上拉电阻。 如果不使用连接到 VS 或 VCC_IO（这是一个高压引脚）。 一旦 IC 从睡眠模式/复位中恢复，它必须重新配置才能再次使用。在睡眠模式下不存储寄存器内容。 在重新配置 IC 时，建议仍然使用 DRV_ENN 禁用功率桥驱动器。 所有 TMC5240 上拉/下拉输入都切换到电平保持模式，以避免 VCC_IO 处的电流消耗。 不要在高电机速度下使用！	VS	AI (pd)
19	28	OUT2B	电机线圈 B 输出 2	VS	A
18	26	OUT1B	电机线圈 B 输出 1	VS	A
22	30	OUT2A	电机线圈 A输出 2	VS	A
23	32	OUT1A	电机线圈 A输出 1	VS	A
EP	EP	GND	裸露的芯片焊盘. 将裸露芯片焊盘连接到 GND 平面。 提供尽可能多的过孔以将热量传递到 GND 平面。 用作功率级和内部电路的接地引脚。		GND
—	4, 21, 24, 34, 37	NC	没有内部连接。 保持此引脚开路或将其连接到 GND 以改善冷却效果。		NC

PRELIMINARY CONFIDENTIAL

引脚说明（续）

PIN		名称	功能	参考供电	TYPE
TQFN32	TSSOP38				
13	19	OV	具有可编程阈值电压的过压指示器输出（漏极开路）。连接带有负载电阻的外部 MOSFET 以限制电源电压。需要外部上拉电阻。由 ADC 更新 f_{CLK} / 2048	VCC_IO	DO (OD)
2	8	AIN	通用模拟输入通过测量 fCLK/2048 内部 12 位 ADC 输入范围 0 至 1.25V 可通过 SPI/UART 获取值。	VCC_IO	AI
5	11	VCC_IO	由外部电源提供的数字 IO 电源电压，用于定义电路 IO 电平。 需要在输出引脚上进行正确的电压电平设置。	VCC_IO	AI

PRELIMINARY CONFIDENTIAL

功能图

TMC5240

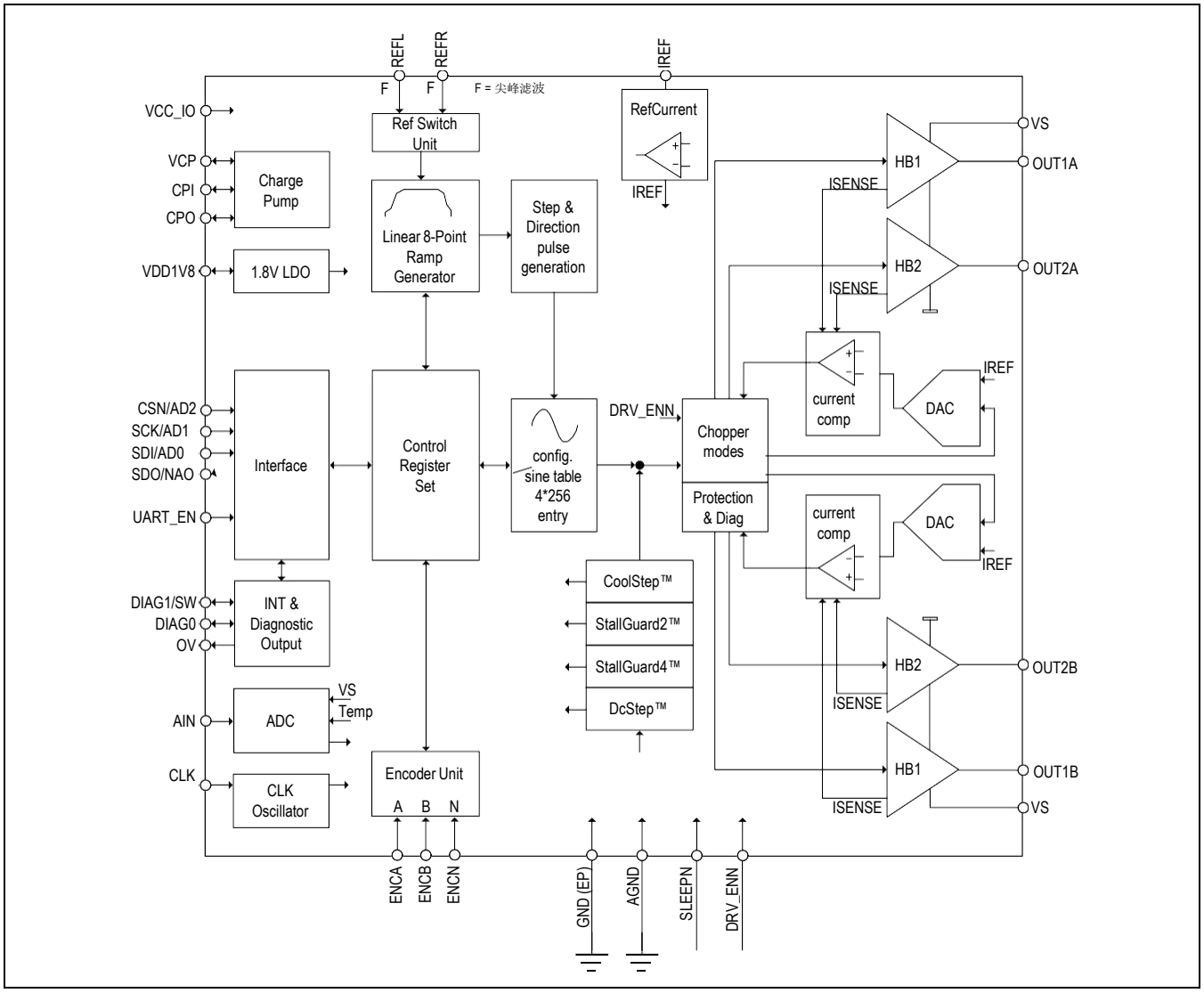


Figure 1. TMC5240 block diagram

PRELIMINARY CONFIDENTIAL

详细说明

操作原理

全功能运动驱动控制器

TMC5240 运动控制和驱动芯片是 CPU 和步进电机之间接口的智能功率元件。所有步进电机逻辑都完全包含在 TMC5240 中。无需软件即可控制电机-只需提供目标位置即可。 TMC5240 通过驱动器和控制器的片上系统集成提供了许多独特的增强功能。TMC5240 的八点斜坡发生器使用 StealthChop、DcStep、CoolStep 和 StallGuard 自动优化每个电机运动。

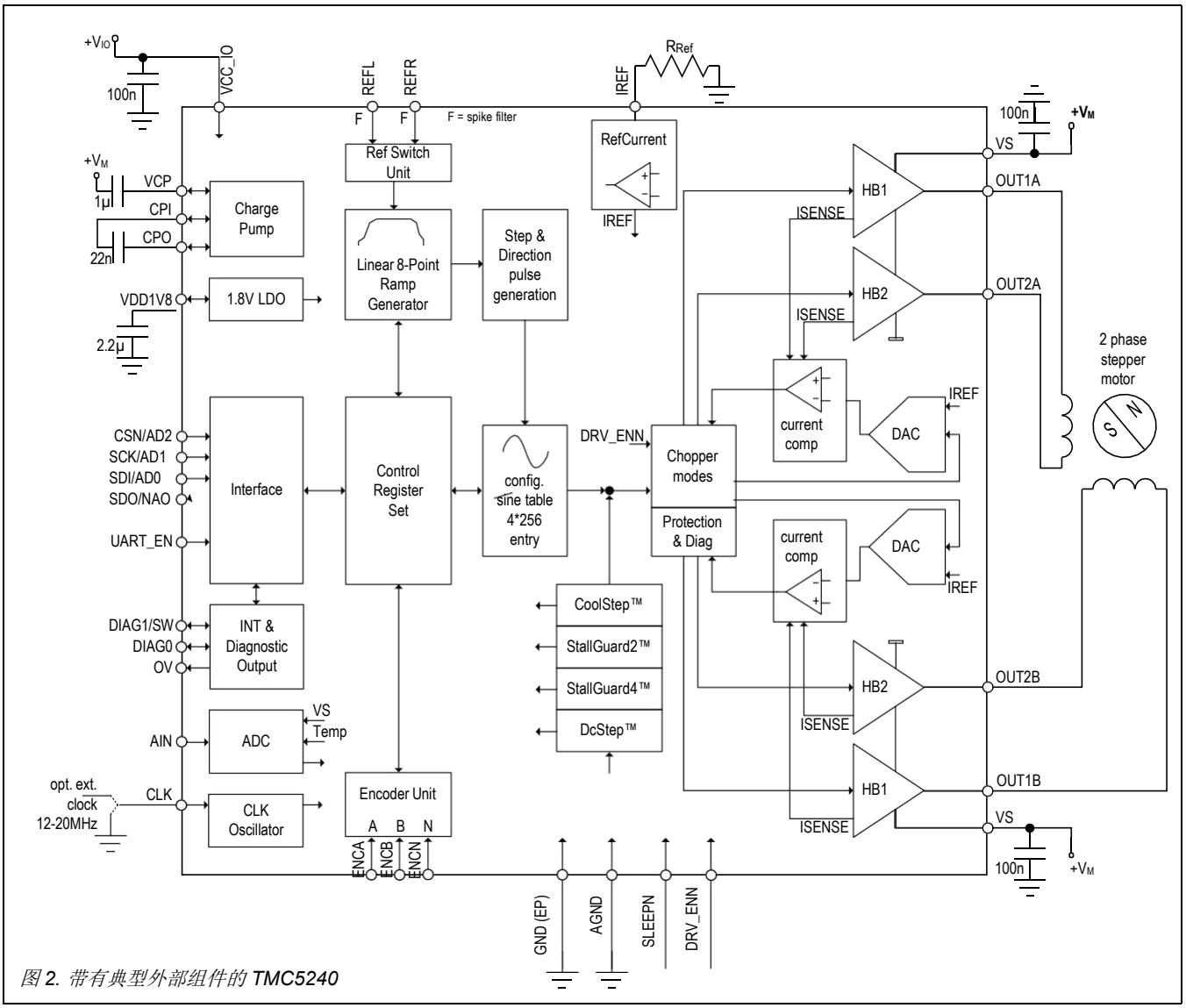


图 2. 带有典型外部组件的 TMC5240

PRELIMINARY CONFIDENTIAL

主要概念

TMC5240 实现了 ADI-Trinamic 产品独有的高级功能。这些特性有助于在许多步进电机应用中实现更高的精度、更高的能源效率、更高的可靠性、更平滑的运动和发热量更低的操作。

StealthChop2™	无噪音 静音的高精度斩波算法 以及电机在静止时候没有噪音. 允许比 StealthChop™ 有更快的电机加速和减速 并将 StealthChop 扩展到低静止电机电流。
SpreadCycle™	高精度逐周期电流控制，可实现最高动态运动。
StallGuard2™	无传感器失速检测和机械负载测量。
StallGuard4™	无传感器回零保护终端开关并在电机过载时发出警告。
CoolStep™	使用 StallGuard 测量来调整电机电流，以实现电机和驱动器的最佳效率和最低发热量。
DcStep™	自适应于负载的速度控制。 电机以尽可能快的速度移动，确保不失步。

除了这些性能增强之外，ADI-Trinamic 电机驱动器提供保护措施来检测和防止短路输出、输出开路、过热和欠压情况，以提高安全性和设备故障恢复。

控制接口

TMC5240 既支持 SPI 接口，也支持基于 UART 的单线接口和 CRC 校验。 实际接口组合的选择是通过 UART_EN 引脚完成的，根据所需的接口选择，该引脚可以硬接线到 GND 或 VCC_IO。

SPI 接口是一个与总线时钟同步的位串行接口。对于从总线主机发送到总线从机的每一位，另一位同时从从机发送到主机。SPI 主机和 TMC5240 从机之间的通信始终包括发送一个 40 位命令字和接收一个 40 位状态字。单线接口允许双向单线通讯。 它可以由任何标准UART驱动。 无需配置波特率。

集成运动控制器

集成的 32 位运动控制器自动驱动电机到目标位置，或加速到目标速度。所有运动参数都可以动态更改， 运动控制器立即执行新的参数。最小的配置数据集由加速度和减速度值以及最大运动速度组成.支持启动和停止速度以及由速度阈值选择的第二和第三加速和减速设置。 这些设置允许运动曲线适应电机扭矩曲线，以及接近 S-ramp 性能减少抖动。集成运动控制器支持对机械参考开关和无传感器失速检测 StallGuard2 和 StallGuard4 的即时反应。

优势：

- 灵活的可以编程斜坡控制器
- 有效利用电机转矩的加速和减速允许更高的机器生产效率
- 用于减少加加速度的伪 S 斜坡
- 对停止和失速情况立即做出反应

自动停止电源关闭

自动电流降低功能可显着降低应用功耗和冷却要求。 将运行电流减少到一半可将静态功耗降低到大约 25%。 静止电流，延迟时间和衰减参数可以通过串行控制接口配置。

提供可自由空转和被动电机制动作为静止的选项。被动制动将电机静止功耗降至零，同时仍提供有效的阻尼和制动！

PRELIMINARY CONFIDENTIAL

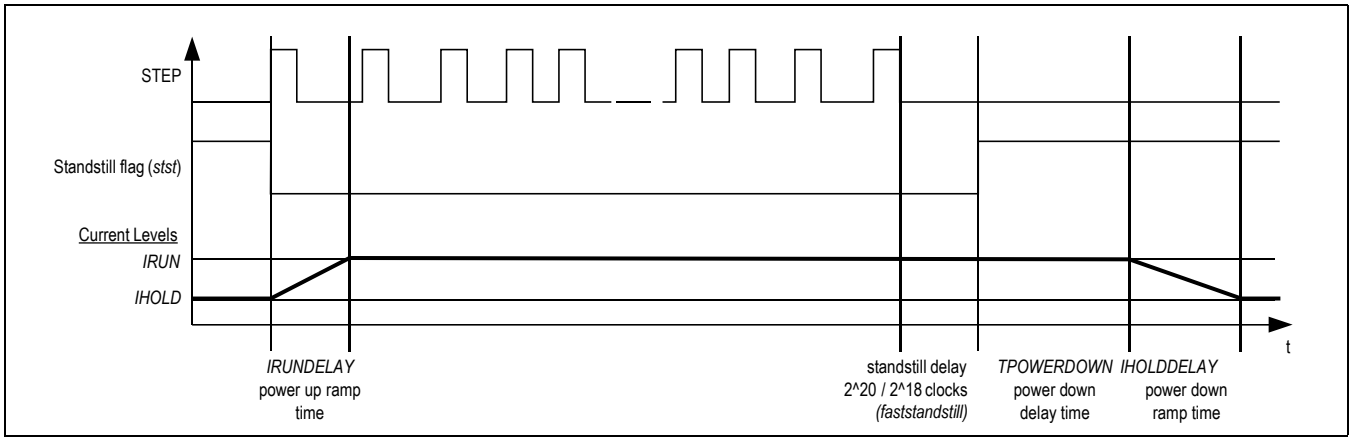


图 3. 静止和加速时的自动电机电流控制

StealthChop2 和 SpreadCycle 驱动器

StealthChop是基于电压的斩压原理。它特别保证了电机在静止和慢速运动中绝对安静，除了滚珠轴承产生的噪音之外。与其他电压模式斩波器不同，StealthChop2 不需要任何配置。它会在开机后的第一个动作中自动学习最佳设置，并在后续动作中进一步优化设置。设备开机回零的过程足以用于自动整定。 可选地，初始整定参数可以加载到寄存器组中。StealthChop2 通过立即对电机速度的变化做出快速反应，实现电机高动态控制。

对于最高速度的应用，SpreadCycle 是 StealthChop2外的另一个选项。 StealthChop2 和 SpreadCycle 可以在组合配置同时使用，以获得两全其美的效果：StealthChop2 用于无噪音静止、静音和平稳的中低速性能，SpreadCycle 以更高的速度实现高动态，在低振动下达到最高峰值速度。

SpreadCycle 是一种先进的逐周期斩波模式。它在广泛的速度和负载范围内提供平稳的操作和良好的共振阻尼。SpreadCycle 斩波器方案自动集成和调整快速衰减周期，以保证平滑的过零性能。

优势：

- 使用低成本电机显着改善微步性能
- 电机运行平稳安静
- 绝对安静的无噪音电机待机状态
- 减少机械共振提高扭矩输出

StallGuard – 机械负载检测

StallGuard2 可准确测量电机负载。它可用于失速检测以及在低于使电机失速的负载下的其他用途，例如 CoolStep 电流自适应负载动态调节。这提供了有关驱动器的更多信息，可实现无传感器回原点和机械诊断等功能。StallGuard2 与 SpreadCycle 斩波器相结合，而 StallGuard4 使用不同的原理与 StealthChop 结合。

CoolStep – 电流自适应负载的控制

CoolStep 以最佳电流驱动电机。它使用 StallGuard2 或 StallGuard4 负载测量信息将电机电流调整到实际负载情况下所需的最小电流值。这可以节省能源并保持电机不发热。由于 CoolStep采用自适应负载的动态电流调节控制，与常规需要预留 50% 扭矩储备的恒流相比，提高了电机效率。

优势：

- 最高能效，耗电量降低75%
- 电机产生的热量更少

PRELIMINARY CONFIDENTIAL

- 提高机械精度
- 采取少的冷却措施或不用冷却措施
- 提高可靠性
- 可以使用更小的电机，需要更少的扭矩储备
- 由于更少的能量激励电机共振，电机噪音更低

DcStep – 速度自适应负载的控制

DcStep 允许电机在其负载极限和速度极限附近运行，而不会产生丢步。如果电机上的机械负载增加到会导致失步的负载， 电机会自动降低速度，以便它仍然可以驱动负载。使用此功能，电机将永远不会失步。 除了在较低速度下增加扭矩外，动态惯性将使电机能够通过减速克服机械过载。DcStep 直接与斜坡发生器集成，即使由于机械负载增加而需要降低电机速度，也能到达目标位置。 DcStep 可以覆盖高达 10 倍或更多的动态范围，而不会出现任何失步。 通过优化高负载情况下的运动速度，此功能进一步提高了整体系统效率。

优 势：

- 电机在过载情况下不失步
- 应用程序尽可能快地工作
- 尽可能高的加速度
- 发挥限速时最高能效
- 使用整步驱动的发挥最大可能电机扭矩
- 使用较便宜的电机即可发挥最大效率

编码器接口

TMC5240 为外部增量编码器提供编码器接口。编码器可用于运动控制器的归位（也可用于参考开关）以及在编码器位置和斜坡发生器位置之间进行即时一致性确认。可编程预分频器允许编码器分辨率与电机分辨率相适应。 提供了一个 32 位编码器计数器。

SPI 接口

SPI 数据报结构

TMC5240 使用 40 位 SPI™（串行外设接口，SPI 是摩托罗拉的商标）数据报与微控制器通信。配备硬件 SPI 的微控制器通常能够使用 8 位的整数倍进行通信。设备的 CSN 线必须在数据报传输的整个持续时间内保持有效（保持低电平）。

发送到设备的每个数据报都由一个地址字节和四个数据字节组成。 这允许与寄存器组进行直接的 32 位数据字通信。

每个寄存器都通过 32 个数据位访问，即使它使用的数据位少于 32 个。

为简单起见，每个寄存器都由一个字节地址指定：

- 对于读访问，地址字节的最高有效位为 0。
- 对于写访问，地址字节的最高有效位为 1。

所有寄存器都是可读的，其中大部分是读写，一些只读，一些写 1 来清除（例如 GSTAT 寄存器）。

表 1. SPI 数据报结构

MSB (先发)	40 bit	bit LSB (最后发送)
写：8位地址 读取：8 位 SPI 状态	39 ... 0 读/写 32 位数据	
39 ... 32	31 ... 0	

表 1. SPI 数据报结构（续）

写给 RW + 7 位地址 读自 8位SPI状态		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

写入/读取的选择 (WRITE_notREAD)

读写选择由地址字节的 MSB（SPI 数据报的第 39 位）控制。 该位为 0 表示读访问，1 表示写访问。所以，名为 W 的位是一个 WRITE_notREAD 控制位。高电平有效写位是地址字节的 MSB。因此，必须将 0x80 添加到地址才能进行写访问。SPI 接口始终将数据传送回主机，与 W 位无关。 传回的数据是从与前一个数据报一起传输的地址读取的数据，如果先前的访问是读取访问的话。如果先前的访问是写访问，则回读的数据反映先前接收的写数据。因此，读访问和写访问之间的区别在于，读访问不将数据传输到寻址寄存器，而是仅传输地址，并且它的 32 个数据位是虚拟的，并且，接下来的读或写访问将数据返回 从前一个读取周期中传输的地址读取。

读访问请求数据报使用虚拟写数据。 读取数据通过随后的读取或写入访问传输回主机。 因此，可以以流水线方式读取多个寄存器。

每当从 TMC5240 读取或写入数据时，返回的 MSB 都包含 SPI 状态。 SPI_STATUS 是八个选定状态位的数量。

例子：

对于地址为 0x21 的寄存器 (XACTUAL) 的读访问，在读取访问之前，地址字节必须设置为 0x21。对于寄存器 (VACTUAL) 的写访问，地址字节必须设置为 0x80 + 0x22= 0xA2。对于读访问，数据位可能具有任何值 (-)。 因此，可以将它们设置为 0。

表 2. SPI 读/写示例流程

Action	Data sent to	Data received from
read *XACTUAL*	0x2100000000	0xSS & unused data*
read *XACTUAL*	0x2100000000	0xSS & XACTUAL
write *VMAX*:= 0x00ABCDEF	0xA700ABCDEF	0xSS & XACTUAL
write *VMAX*:= 0x00123456	0xA700123456	0xSS00ABCDEF

*)SS：是状态位 SPI_STATUS 的占位符

每个数据报回读传输的 SPI 状态位

新的状态信息在每次访问结束时被锁存，并可用于下一次 SPI 传输。

T表 3. SPI_STATUS – 每次 SPI 访问传输的状态标志位 39 至 32

Bit	Name	Comment
7	*status_stop_r*	*RAMP_STAT*[1] – 1: 信号停止右开关状态（仅限运动控制器）
6	*status_stop_l*	*RAMP_STAT*[0] – 1: 信号停止左开关状态（仅限运动控制器）
5	*position_reached*	*RAMP_STAT*[9] – 1: 到达目标位置信号（仅限运动控制器）
4	*velocity_reached*	*RAMP_STAT*[8] – 1: 达到目标速度信号（仅限运动控制器）
3	*standstill*	*DRV_STATUS*[31] – 1: 电机静止信号
2	*sg2*	*DRV_STATUS*[24] – 1: 信号StallGuard标志激活
1	*driver_error*	*GSTAT*[1] – 1: 驱动程序 报错1 （通过读取 GSTAT 清除）

PRELIMINARY CONFIDENTIAL

表 3. SPI_STATUS – 每次 SPI 访问传输的状态标志位 39 到 32（续）

0	*reset_flag*	*GSTAT*[0] – 1: 复位触发信号（通过读取 GSTAT 清除）

数据校准

所有数据都右对齐。 一些寄存器表示无符号（正）值，一些将整数值（有符号）表示为二进制补码，单个位或位组分别表示为单个位作为整数组。

SPI 信号

TMC5240上的SPI总线有四个信号:

- SCK – 总线时钟输入
- SDI – 串行数据输入
- SDO – 串行数据输出
- CSN – 芯选输入(低有效)

通过芯选输入 CSN 上的低电平使从机进行 SPI 通讯。 位传输与总线时钟 SCK 同步，从机在 SCK 的上升沿锁存来自 SDI 的数据，并在下降沿之后将数据驱动到 SDO。 首先发送最高有效位。 与 TMC5240 的总线事务至少需要 40 个 SCK 时钟周期。

如果驱动的时钟超过 40 个，则移入 SDI 的额外位在 40 个时钟延迟后通过内部移位寄存器在 SDO 上移出。这可用于以菊花链方式连接多个芯片。

在整个总线传输期间，CSN 必须为低电平。当 CSN 变为高电平时，内部移位寄存器的内容被锁存到内部控制寄存器中，并被识别为从主设备到从设备的命令。如果发送超过 40 位，则只有在 CSN 上升沿之前接收到的最后 40 位被识别为命令。

SPI 时序

The SPI 最大频率在10 MHz. SCK 独立于系统的时钟频率，而唯一取决于时钟频率的参数是最小 CSN 高电平时间。 All SPI inputs are internally filtered to avoid triggering on pulses shorter than 10ns. The figure shows the timing parameters of an SPI bus transaction. Timing values are given in the EC table.

The SPI interfaces uses SPI MODE 3.

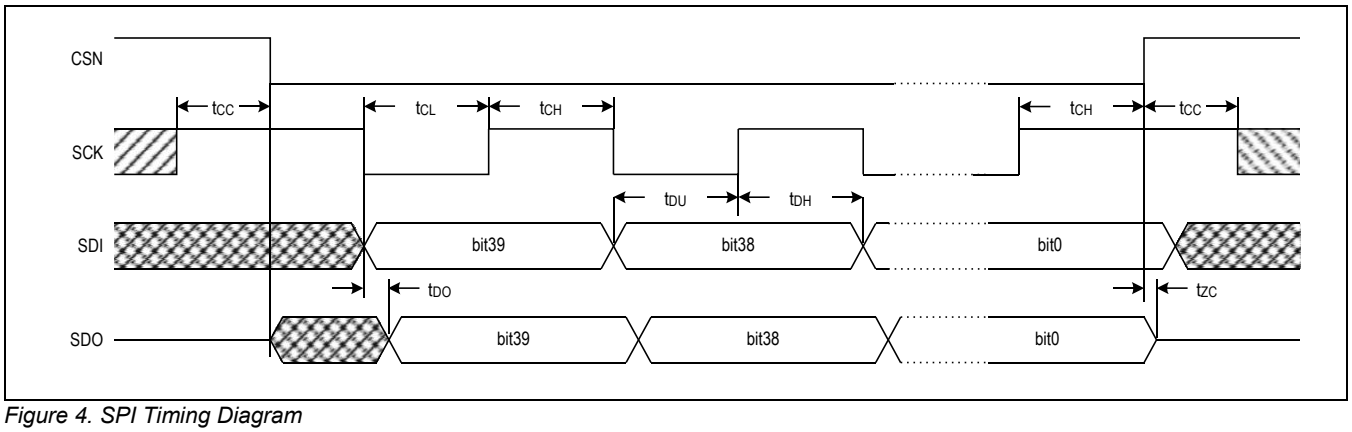


Figure 4. SPI Timing Diagram

UART Single Wire Interface

The UART single wire interface allows control of the TMC5240 with any microcontroller UART. It shares transmit and receive line like an RS485 based interface. Data transmission is secured using a cyclic redundancy check, so that

PRELIMINARY CONFIDENTIAL

increased interface distances (e.g. over cables between two PCBs) can be bridged without danger of wrong or missed commands even in the event of electro-magnetic disturbance. The automatic baud rate detection makes this interface easy to use.

Datagram Structure

Write Access

Table 4. UART Write access datagram structure

each byte is LSB…MSB, highest byte transmitted first																				
0 … 63																				
sync + reserved								**8 bit slave 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)				*SLAVEADDR*			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

A sync nibble precedes each transmission to and from the TMC5240 and is embedded into the first transmitted byte, followed by an addressing byte. Each transmission allows a synchronization of the internal baud rate divider to the master clock. The actual baud rate is adapted and variations of the internal clock frequency are compensated. Thus, the baud rate can be freely chosen within the valid range. Each transmitted byte starts with a start bit (logic 0, low level on DIAG1/SW) and ends with a stop bit (logic 1, high level on DIAG1/SW). The bit time is calculated by measuring the time from the beginning of start bit (1 to 0 transition) to the end of the sync frame (1 to 0 transition from bit 2 to bit 3). All data is transmitted byte wise. The 32 bit data words are transmitted with the highest byte first.

A minimum baud rate of 9000 baud is permissible, assuming 20 MHz clock (worst case for low baud rate). Maximum baud rate is $f_{CLK}/16$ due to the required stability of the baud clock.

The initial slave address *SLAVEADDR* is selected by CSN_AD2, SCK_AD1, SDI_AD0 in the range 0 to 7.

The slave address is determined by the sum of the register *SLAVEADDR* and the pin selection given above. This means, that a high level on SDI (with CSN low and SCK low) increments the *SLAVEADDR* setting by one.

Bit 7 of the register address identifies a Read (0) or a Write (1) access. Example: Address 0x10 is changed to 0x90 for a write access.

The communication becomes reset if a pause time of longer than 63 bit times between the start bits of two successive bytes occurs. This timing is based on the last correctly received datagram. In this case, the transmission needs to be restarted after a failure recovery time of minimum 12 bit times of bus idle time. This scheme allows the master to reset communication in case of transmission errors. Any pulse on an idle data line below 16 clock cycles will be treated as a glitch and leads to a timeout of 12 bit times, for which the data line must be idle. Other errors like wrong CRC are also treated the same way. This allows a safe re-synchronization of the transmission after any error conditions. Remark, that due to this mechanism an abrupt reduction of the baud rate to less than 15 percent of the previous value is not possible.

Each accepted write datagram becomes acknowledged by the receiver by incrementing an internal cyclic datagram counter (8 bit). Reading out the datagram counter allows the master to check the success of an initialization sequence or single write accesses. Read accesses do not modify the counter.

Read Access

Table 5. UART Read access request datagram structure

each byte is LSB…MSB, highest byte transmitted first																				
sync + reserved								**8 bit slave 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)				*SLAVEADDR*			register address			0			CRC		
0	1	2	3	4	5	6	7	8	…	15	16	…	23				24	…	31

PRELIMINARY CONFIDENTIAL

The read access request datagram structure is identical to the write access datagram structure, but uses a lower number of user bits. Its function is the addressing of the slave and the transmission of the desired register address for the read access. The TMC5240 responds with the same baud rate as the master uses for the read request.

In order to ensure a clean bus transition from the master to the slave, the TMC5240 does not immediately send the reply to a read access, but it uses a programmable delay time after which the first reply byte becomes sent following a read request. This delay time can be set in multiples of eight bit times using *SENDDELAY* time setting (default=8 bit times) according to the needs of the master. In a multi-slave system, set *SENDDELAY* to min. 2 for all slaves. Otherwise a non-addressed slave might detect a transmission error upon read access to a different slave.

Table 6. UART Read access reply datagram structure

each byte is LSB…MSB, highest byte transmitted first																				
0..... 63																				
sync + reserved								**8 bit slave 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

The read response is sent to the master using address code %1111. The transmitter becomes switched inactive four bit times after the last bit is sent.

Address %11111111 is reserved for read accesses going to the master. A slave cannot use this address.

CRC Calculation

An 8 bit CRC polynomial is used for checking both read and write access. It allows detection of up to eight single bit errors. The CRC8-ATM polynomial with an initial value of zero is applied LSB to MSB, including the sync- and addressing byte. The sync nibble is assumed to always be correct. The TMC5240 responds only to correctly transmitted datagrams containing its own slave address. It increases its datagram counter for each correctly received write access datagram.

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

Serial calculation example

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

C-Code Example for CRC calculation

```
void swuart_calcCRC(UCHAR* datagram, UCHAR datagramLength)
{
  int i,j;
  UCHAR* crc = datagram + (datagramLength-1); // CRC located in last byte of message
  UCHAR currentByte;

  *crc = 0;

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

PRELIMINARY CONFIDENTIAL

```
      }
      currentByte = currentByte >> 1;
    } // for CRC bit
  } // for message byte
}
```

UART Signals

The UART interface on the TMC5240 comprises five signals. In UART mode the slave checks the single wire pin DIAG1/SW for correctly received datagrams with its own address continuously. The pin is switched as input during this time. It adapts to the baud rate based on the sync nibble, as described before. In case of a read access, it switches on its output driver on DIAG1/SW and sends its response using the same baud rate.

Table 7. TMC5240 UART Interface Signals

DIAG1/SW	Data input and output
CSN/AD2	Bit 2 of initial UART address increment (+4)
SCK/AD1	Bit 1 of initial UART address increment (+2)
SDI/AD0	Bit 0 of initial UART address increment (+1), tie to NAO of previous IC in chain
SDO/NAO	Next address output (NAO) pin for chained sequential addressing scheme (reset default= high)

Addressing Multiple Slaves

If only one or up to eight TMC5240 are addressed by a master using a single UART bus interface, a simple hardware address selection can be used. The individual UART addresses are set by connecting the UART address pins (SDI, SCk, CSN) to VCC_IO and GND.

If more than eight slaves need to be connected to the same UART bus a different approach must be used. This approach can address up to 255 devices by using the output NAO (SDO) as a selection pin for the bit 0 address pin of the next device. Proceed as follows:

- Tie all address pins as well as SDI/AD0 of your first TMC5240 to GND.
- Connect SDO/NAO output of the first TMC5240 to the next drivers address[0] pin (SDI/AD0). Connect further drivers in the same fashion.
- Now, the first driver responds to address 0. Following drivers are set to address 1.
- Program the first driver to its dedicated slave address. Note: once a driver is initialized with its slave address, its SDO/NAO output, which is tied to the next drivers address[0] pin (SDI/AD0) has to be programmed to logic 0 in order to differentiate the next driver from all following devices.
- Now, the second driver is accessible and can get its slave address. Further units can be programmed to their slave addresses sequentially.

PRELIMINARY CONFIDENTIAL

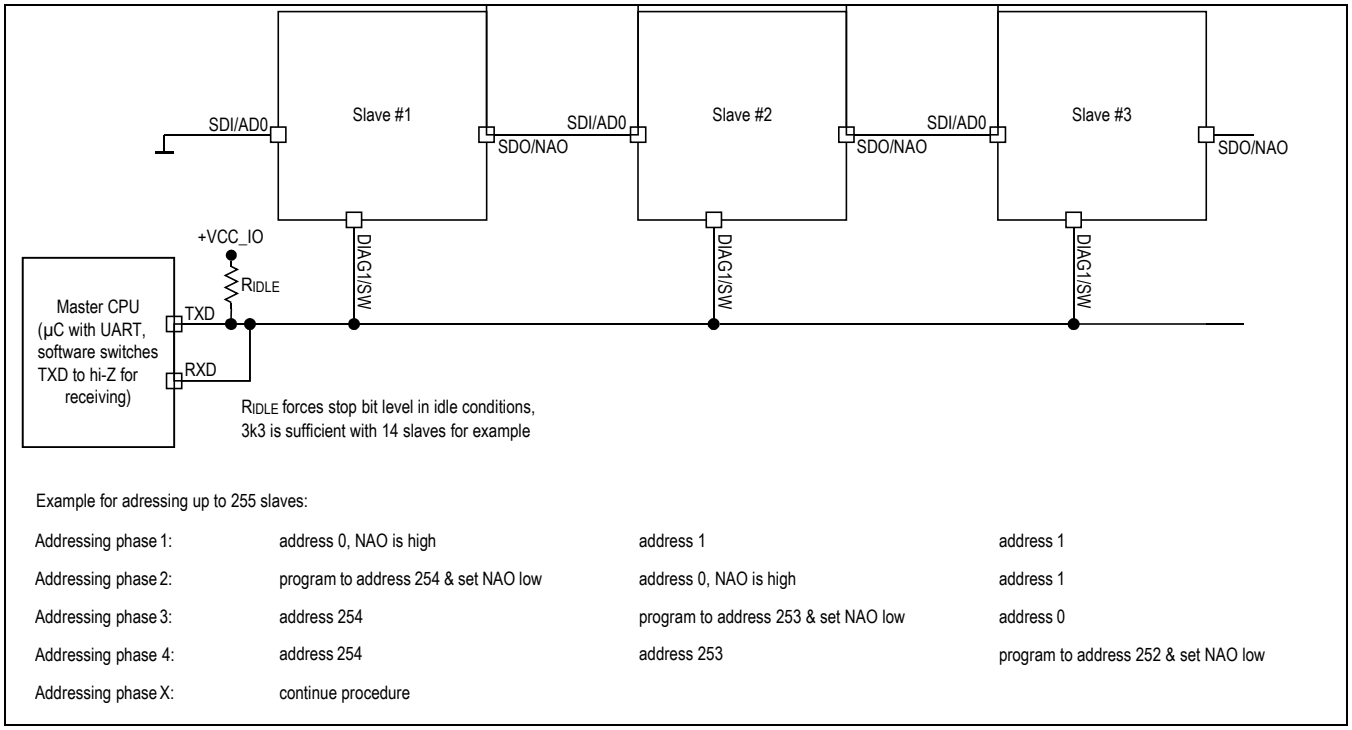


Figure 5. UART daisy chaining example

StealthChop2

StealthChop2 是一种非常安静的步进电机运行模式. 它是基于电压模式PWM. 在静止和低速的情况下，电机没有任何噪音。因此，StealthChop2 驱动的步进电机应用非常适合室内或家庭使用。电机在低速运行时完全没有振动。使用 StealthChop，电机电流通过使用电压模式 PWM 将特定有效电压驱动到线圈中来施加。借助增强的 StealthChop2，驱动程序自动适应应用程序以获得最佳性能. 无需更多配置。可选配置允许在特殊情况下调整设置，或为自动适应算法设置初始值。对于宽范围的速度控制，SpreadCycle 应与 StealthChop2 结合使用。

PRELIMINARY CONFIDENTIAL

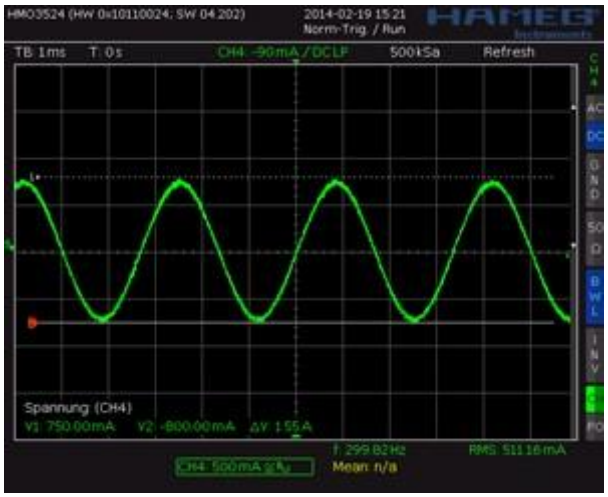


图 6. 采用 StealthChop 的电机线圈正弦波电流（用电流探头测量）

自动调整

StealthChop2 集成了一个自动调整程序 (AT)，它可以自动调整最重要的运行参数以适应电机。通过这种方式，StealthChop2 允许高电机动态，并支持将电机供应电流至极低数值。 为了获得最佳结果，只需做两个步骤：启动时电机处于静止状态，但此时的静止电流设置为和运行电流相等 (AT#1) . 以中等速度移动电机，例如 作为回零过程的一部分 (AT#2). 下图中的流程图显示了调谐过程。

表 8. StealthChop 自动调整 AT#1 和 AT#2 的限制和要求

自动调谐时机和条件			
步骤	**参数**	**状况**	**所需时间**
AT#1	*PWM_OFS_AUTO*	• 电机处于静止状态，设置实际电流刻度 (CS) 与运行电流 (IRUN) 相同。 • 如果启用了待机降流，可短暂控制电机运行以实现到达运行电流，或将 IHOLD 设置为 IRUN。 • 引脚 VS 处于供电状态。	≤ 2^20+2*2^18 t_{CLK}, ≤ 130ms (使用内部时钟)
AT#2	*PWM_GRAD_AUTO*	• 以一定速度移动电机，产生大量反电动势并且可以达到全运行电流.条件: • 1.5 * *PWM_OFS_AUTO* * (*IRUN*+1) / 32 < *PWM_SCALE_SUM* < 4 * *PWM_OFS_AUTO* * (*IRUN*+1) / 32 • *PWM_SCALE_SUM* < 255. *提示：典型范围是 60-300 RPM。*	+/-1 的变化需要 8 个全步 对于常规电机 PWM_GRAD_AUTO 最佳值在 50 或以下, 速度从默认值 0 开始时最多需要 400 个完整整步。

提示:

建议通过评估板确定自动调整的最佳条件。

使用 PWM_GRAD 和 PWM_OFS 的应用特定参数在固件中进行初始化以提供初始调整参数。

在 AT#2 调谐的恒速阶段监控 PWM_SCALE_AUTO 出现下降接近零，这表明调整成功。

请注意:

PRELIMINARY CONFIDENTIAL

在没有经过适当调整的情况下在 StealthChop 中运行可能会导致减速斜坡期间的电机电流很高，尤其是在低电阻电机和快减速设置的情况下。 按照自动调谐过程和使用评估板确认最佳优化参数。建议使用初始值来设置PWM_OFS和PWM_GRAD，这是根据电机类型确定的。

修改 GLOBALSCALER 或 VS 电压会使自动调谐过程的结果失效。 在下一个 AT#1 阶段之前，电机电流调节无法补偿显着变化。只要在以后的操作中满足 AT#1 和 AT#2 的条件，自动调谐就会适应变化的条件。

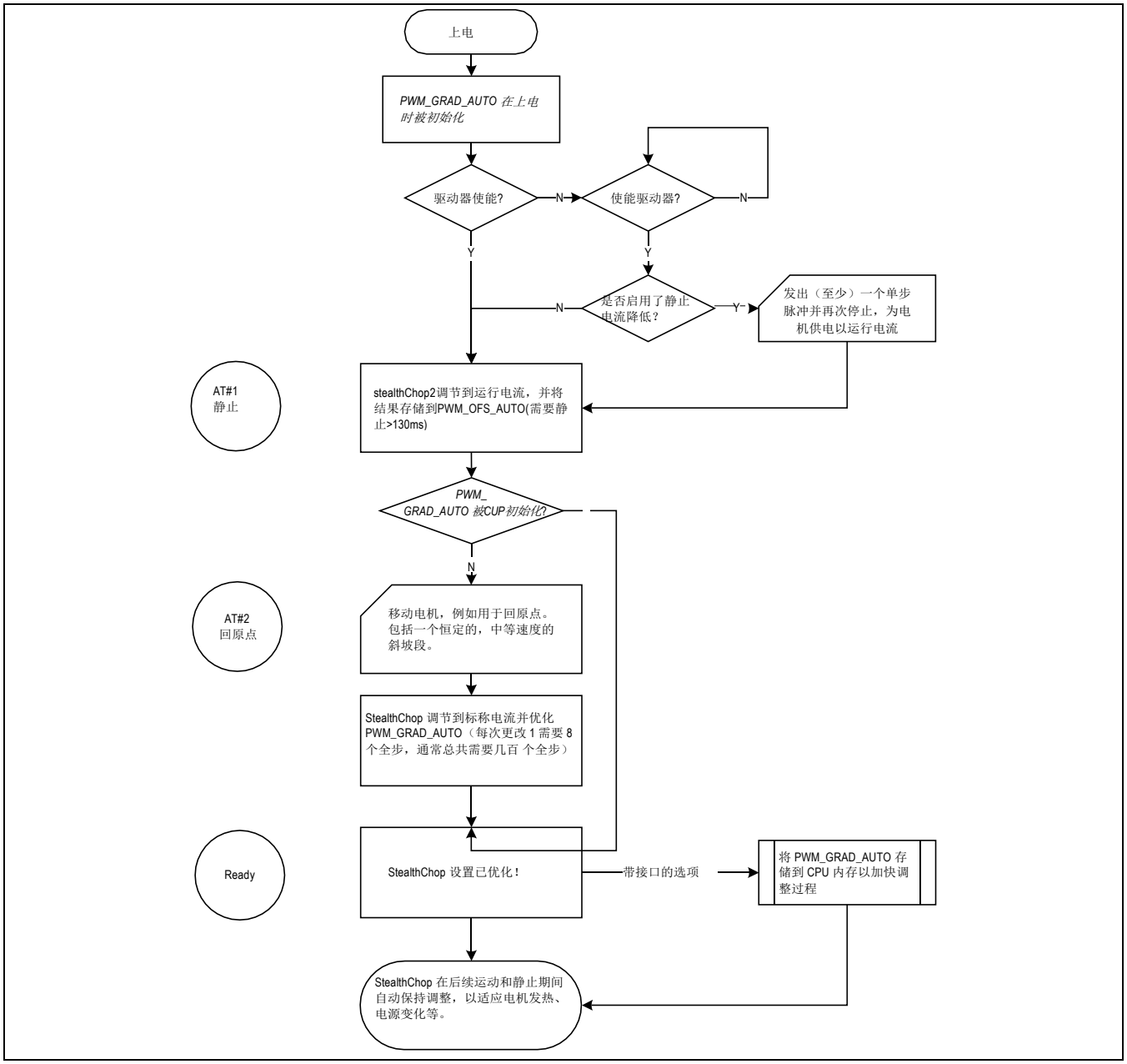


图 7. StealthChop2 自动调谐过程

PRELIMINARY CONFIDENTIAL

StealthChop 选项

为了将电机电流匹配到某个水平，有效 PWM 电压会根据实际电机速度进行缩放。 有几个额外的因素会影响以目标电流驱动电机所需的电压水平：电机电阻，它的 back EMF (即与其速度成正比) 以及供电电压的实际水平. 提供两种 PWM 调节模式：使用电流反馈的自动调谐模式 (AT) (pwm_autoscale= 1, pwm_autograd = 1) 和前馈速度控制模式（pwm_autoscale = 0）。前馈速度控制模式不会对电源电压的变化或电机失速等事件做出反应，但它提供了非常稳定的电流幅值。它不使用也不需要任何电流测量手段。 当电机类型和电源电压已知时，这种方式是完美的。我们推荐自动调谐模式，除非电流调节在给定的工作条件下不令人满意。

建议使用具体应用的初始调谐参数，配合电机类型和电源电压。此外，运行在自动调谐模式，以响应参数的变化，如电机发热或电源电压的变化。

非自动模式 (pwm_autoscale=0) 应仅在已知的电机和操作条件下考虑。在这种情况下，需要通过通讯接口仔细编程。操作参数 PWM_GRAD 和 PWM_OFS 最初可以在自动调谐模式下确定初始值。

为了使分频器适应时钟源的频率，可以分四段选择 StealthChop PWM 频率。20-50kHz 范围内的设置适用于大多数应用。它平衡了低电流纹波和良好的高速性能与动态功耗。

表 9. StealthChop 的 PWM 频率选择（粗体 = 推荐）

时钟频率 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 $f_{PWM}=2/410\ f_{CLK}$
18 MHz	**35.2 kHz**	52.7 kHz	70.3 kHz	87.8 kHz
16 MHz	**31.3 kHz**	**46.9 kHz**	62.5 kHz	78.0 kHz
12.5 MHz (内部时钟)	**24.4 kHz**	**36.6 kHz**	**48.8 kHz**	61.0 kHz
10 MHz	19.5 kHz	**29.3 kHz**	**39.1 kHz**	**48.8 kHz**
8 MHz	15.6 kHz	**23.4 kHz**	**31.2 kHz**	**39.0 kHz**

StealthChop 电流调节器

在 StealthChop 电压 PWM 模式下，自动缩放功能 (pwm_autoscale = 1, pwm_auto_grad = 1) 将电机电流调节到所需的电流设置。自动缩放被用作自动调整过程 (AT) 的一部分，并用于后续跟踪电机参数的变化。驱动器在斩波器准时测量电机电流，并使用比例调节器来调节 PWM_SCALE_AUTO，以使电机电流与目标电流匹配，PWM_REG 是该调节器的比例系数。基本上，比例系数应该尽可能小，以获得稳定和温和的调节行为，但它必须足够大，以使驱动器能够快速响应由电机目标电流变化引起的变化（例如 VREF 的变化）。在初始调谐步骤 AT#2 中，PWM_REG 还补偿电机速度的变化。 因此，在AT#2期间的高加速将需要更高的PWM_REG设置。通过仔细选择回零速度和加速度，调节梯度的最小设置通常就足够了 (PWM_REG=1)。PWM_REG设置应该优化为所需的最快加速和减速坡道(比较后面的2个图)。

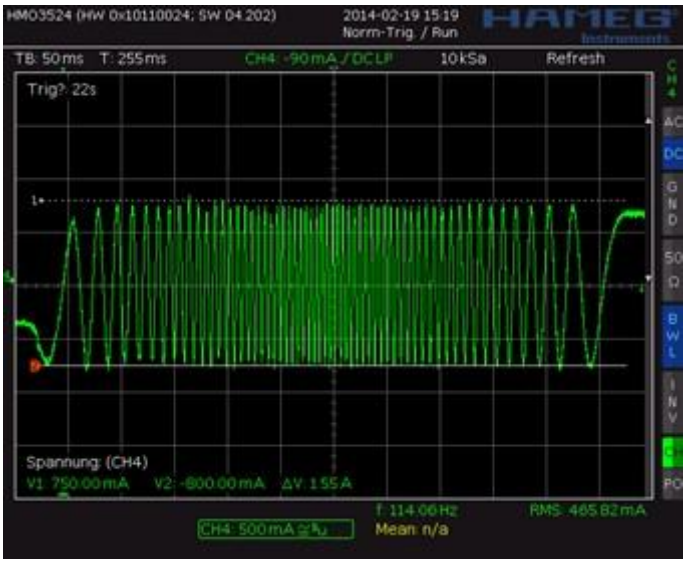


图 8. StealthChop2：PWM_REG 的良好设置

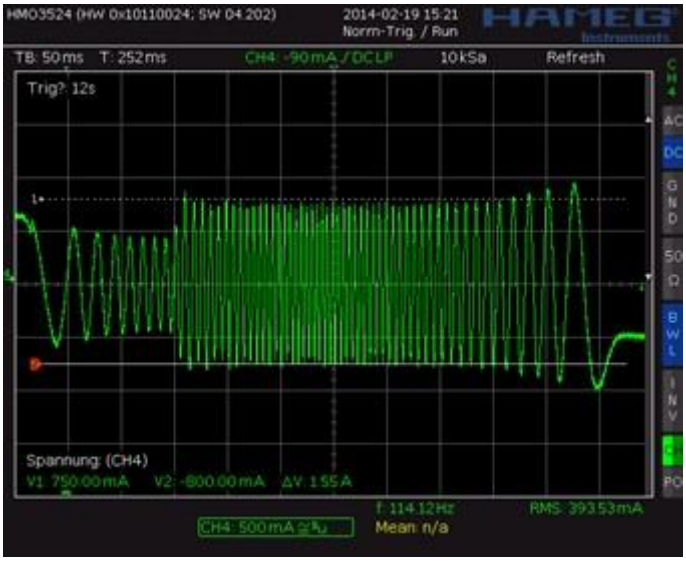


图 9. StealthChop2：在 AT#2 期间 PWM_REG 的设置太小

如下图所示，在加速阶段监测电机电流时，可以检查 AT#2 阶段 PWM_REG 设置的质量和完成的自动调谐程序（或 PWM_OFS 和 PWM_GRAD 的非自动设置）。

PRELIMINARY CONFIDENTIAL

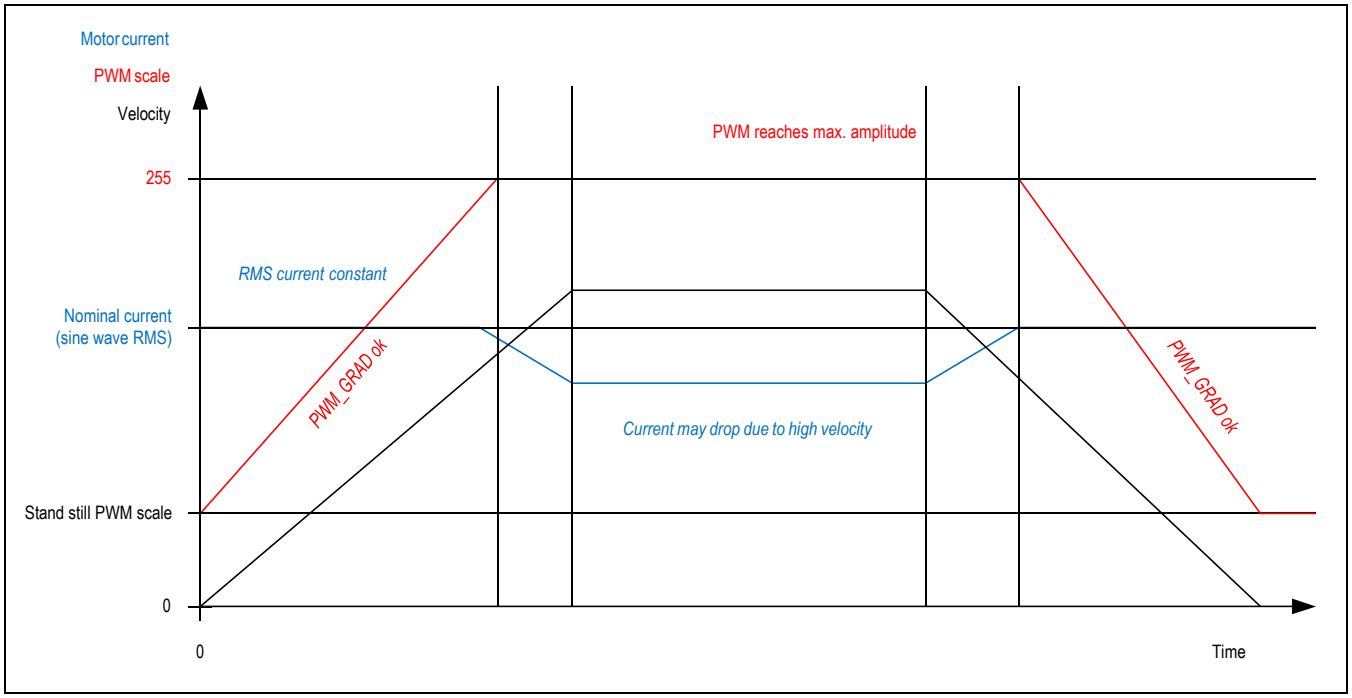


图 10. 成功确定 PWM_GRAD(_AUTO) 和 PWM_OFS(_AUTO)

电流下限

根据 pwm_meas_sd_enable 的设置，StealthChop 电流调节器原理为电机电流调节施加了下限。 由于仅在斩波期间（Chopper on）测量线圈电流（pwm_meas_sd_enable=0），允许线圈电流调节的最小斩波占空比由 TBL 设置的空白时间和斩波频率设置给出。 因此，在 StealthChop 自动缩放模式下，电机特定的最小线圈电流会随着电源电压和斩波频率而上升。较低的消隐时间(blanking time)允许较低的电流限值.正确确定PWM_OFS_AUTO很重要,AT#1 中的运行电流、GLOBALSCALER 和 IRUN 都在调节范围内。较低的电流（例如，用于静止待机）是基于 PWM_OFS_AUTO 和 PWM_GRAD_AUTO 自动实现的电流缩放，分别基于 PWM_OFS 和 PWM_GRAD的具有非自动电流缩放。 自由旋转选项允许电机电流为零。

StealthChop2 自动调谐的电机线圈电流下限 (pwm_meas_sd_enable = 0)：

$$I_{LowerLimit} = t_{BLANK} * f_{PWM} * \frac{V_M}{R_{COIL}}$$

V_M 为电机电源电压，R_{COIL} 为电机线圈电阻。

$I_{Lower\ Limit}$ 可以作为最小标称的 *IRUN* 电机电流设置. 如果电流下限不足以达到所要求的设定值，驱动程序将仅在步骤 AT#1 中以较低的斩波频率重试。

f_{PWM} 为斩波频率，由设置PWM_FREQ确定。在AT#1中，驱动器尝试一个更低的(大约一半的频率)，以防它无法到达电流。频率将在静止状态下保持活动状态，而电流标度 CS=IRUN。 使用静止电流降低，这是一个短暂的时刻。

例子: 电机线圈电阻5Ω，供电电压24V。TBL=%01, PWM_FREQ=%00, tBLANK为24个时钟周期，fPWM为2/(1024个时钟周期):

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

PRELIMINARY CONFIDENTIAL

这意味着，考虑到所有相关设置，自动调谐的电机目标电流必须为 225mA 或更大。这个较低的电流限制也适用于通过 GLOBALSCALER修改电机电流。

注意:

对于自动调谐，会有低的线圈电流限制。

IRUN ≥ 8: IRUN 的电流设置低于 8 不适用于自动调谐。

$I_{LOWER\ LIMIT}$: 根据 pwm_meas_sd_enable 位（在寄存器 PWM_CONF[22] 中）的设置，用于自动调谐，会有最低线圈电流限制。自动调谐阶段 AT#1 中的电机电流必须超过最低电流限制。计算 $I_{LOWER\ LIMIT}$ 或使用电流探头测量它。在成功的自动调整后，可以通过修改IRUN和IHOLD，将电机的运行电流或保持电流设置为低于下电流限制。 电流下限也限制了驱动器对 GLOBALSCALER 变化的响应能力。

电流下限也限制了驱动器对 GLOBALSCALER 变化的响应能力。

要克服电流下限，设置 pwm_meas_sd_enable=1。 这将允许 IC 在慢衰减阶段额外测量线圈电流。

基于速度的标定

基于速度的缩放根据每两个微步之间的时间缩放 StealthChop 幅度，即基于 TSTEP，以时钟周期测量。这个概念基本上不需要电流测量，因为不需要调节回路。只有当设置pwm_autoscale = 0时，才可以通过编程实现纯粹的基于速度的标定。基本思想是对驱动目标电流进入电机所需的电压进行线性近似。步进电机具有一定的线圈电阻，因此需要一定的电压幅度来产生基于基本公式 I=U/R 的目标电流。R 是线圈电阻，U 是按 PWM 值缩放的电源电压，电流 I 结果值。 可以计算 PWM_OFS 的初始值：

$$PWM\_OFS = \frac{374 * R_{COIL} * I_{COIL}}{V_M}$$

VM 为电机电源电压，ICOIL 为目标 RMS 电流

考虑到 8 位分辨率和 248 正弦波峰值，实际 PWM 幅度的有效 PWM 电压 UPWM (1/SQRT(2) x 峰值) 结果显示为 PWM_SCALE：

$$UPWM = V_M * \frac{PWM\_SCALE}{256} * \frac{248}{256} * \frac{1}{\sqrt{2}} = V_M * \frac{PWM\_SCALE}{374}$$

随着电机速度的提高，电机会产生越来越大的反电动势电压。反电动势电压与电机速度成正比。它降低了在线圈电阻处有效的 PWM 电压，因此电流减小。TMC5240 提供第二个速度相关因子 (PWM_GRAD) 来对此进行补偿。在这种模式下，总体有效PWM幅值(PWM_SCALE_SUM)根据微步频率自动计算如下:

$$PWM\_SCALE\_SUM = PWM\_OFS + PWM\_GRAD * 256 * \frac{f_{STEP}}{f_{CLK}}$$

fSTEP是相当于256微步分辨率的微步频率，fCLK是提供给驱动器的时钟频率或实际内部频率。

作为第一个近似值，反电动势从电源电压中减去，因此有效电流幅度减小。这样，可以计算出 PWM_GRAD 设置的第一个近似值：

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

C_{BEMF} 是电机的反电动势常数，单位为伏特/弧度/秒.

MSPR 是与 1/256 微步分辨率相关的每转微步数, 例如 对于 1.8° 电机，51200 = 256µsteps 乘以 200 fullsteps。

PRELIMINARY CONFIDENTIAL

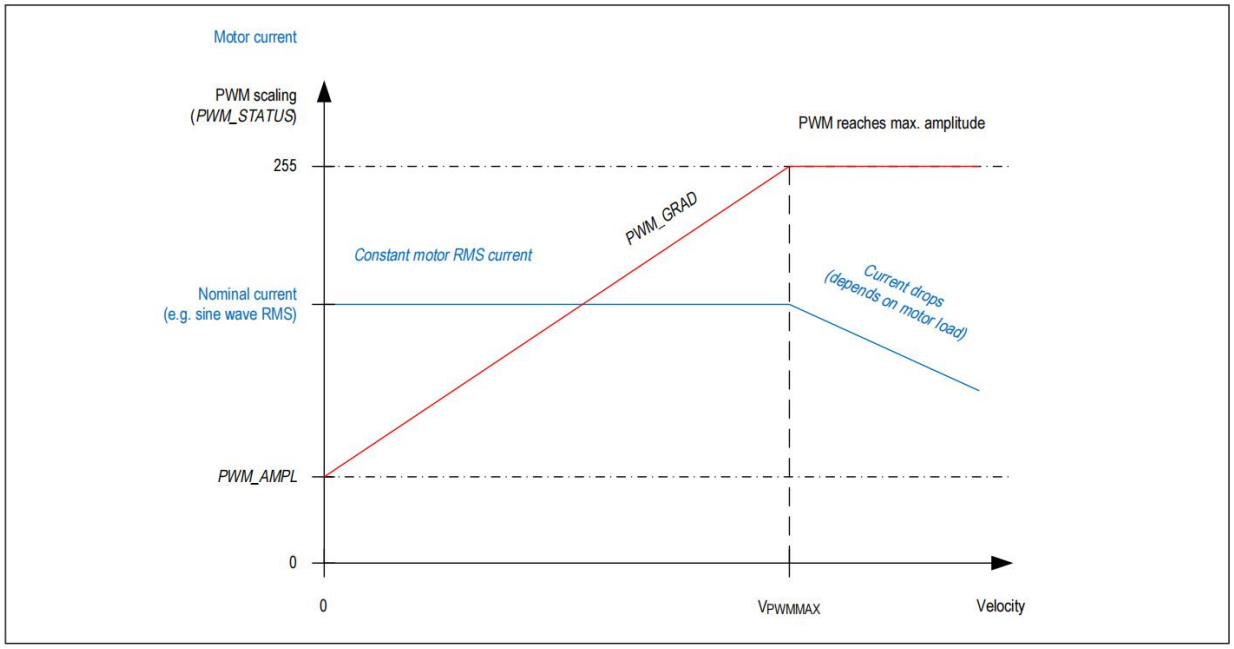


图 11. 基于速度的 PWM 标定 (pwm_autoscale=0)

PWM_OFS 和 PWM_GRAD 的值可以通过使用示波器上的电流探头跟踪电机电流来轻松优化。或者，自动调谐会确定这些值，并且可以从 PWM_OFS_AUTO 和 PWM_GRAD_AUTO 中读取它们。

了解电机的反电动势常数：反电动势常数是电机以一定速度转动时产生的电压。通常电机数据表不会指定这个值，因为它可以从电机扭矩和线圈额定电流中得到。在 SI 单位内，反电动势常数 C_{BEMF} 的数值与转矩常数的数值相同。 例如，扭矩常数为 1 Nm/A 的电机将具有 1V/rad/s 的C_{BEMF}。以 1 rps（1 rps = 每秒 1 转 = 6.28 rad/s）转动这样的电机会产生 6.28V 的反电动势电压. 因此，反电动势常数可以计算为：

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

$I_{COILNOM}$ 是指定保持转矩下电机的额定相电流

HoldingTorque 是电机特定的保持转矩，即两个线圈在 $I_{COILNOM}$ 处达到的转矩。 扭矩单位为 [Nm]，其中 1Nm = 100Ncm = 1000mNm。

电压作为每个线圈的 RMS 电压是有效的，因此在这个公式中标称电流乘以 2，因为标称电流假定一个完整的步进位置，两个线圈在运行。

结合 StealthChop 和 SpreadCycle

对于需要高速运动的应用，SpreadCycle 可能会在较高的速度范围内带来更稳定的运行。为了将无噪声运行与最高动态性能相结合，TMC5240 允许根据速度阈值组合 StealthChop 和 SpreadCycle。有了这个，StealthChop 仅在低速时有效。

PRELIMINARY CONFIDENTIAL

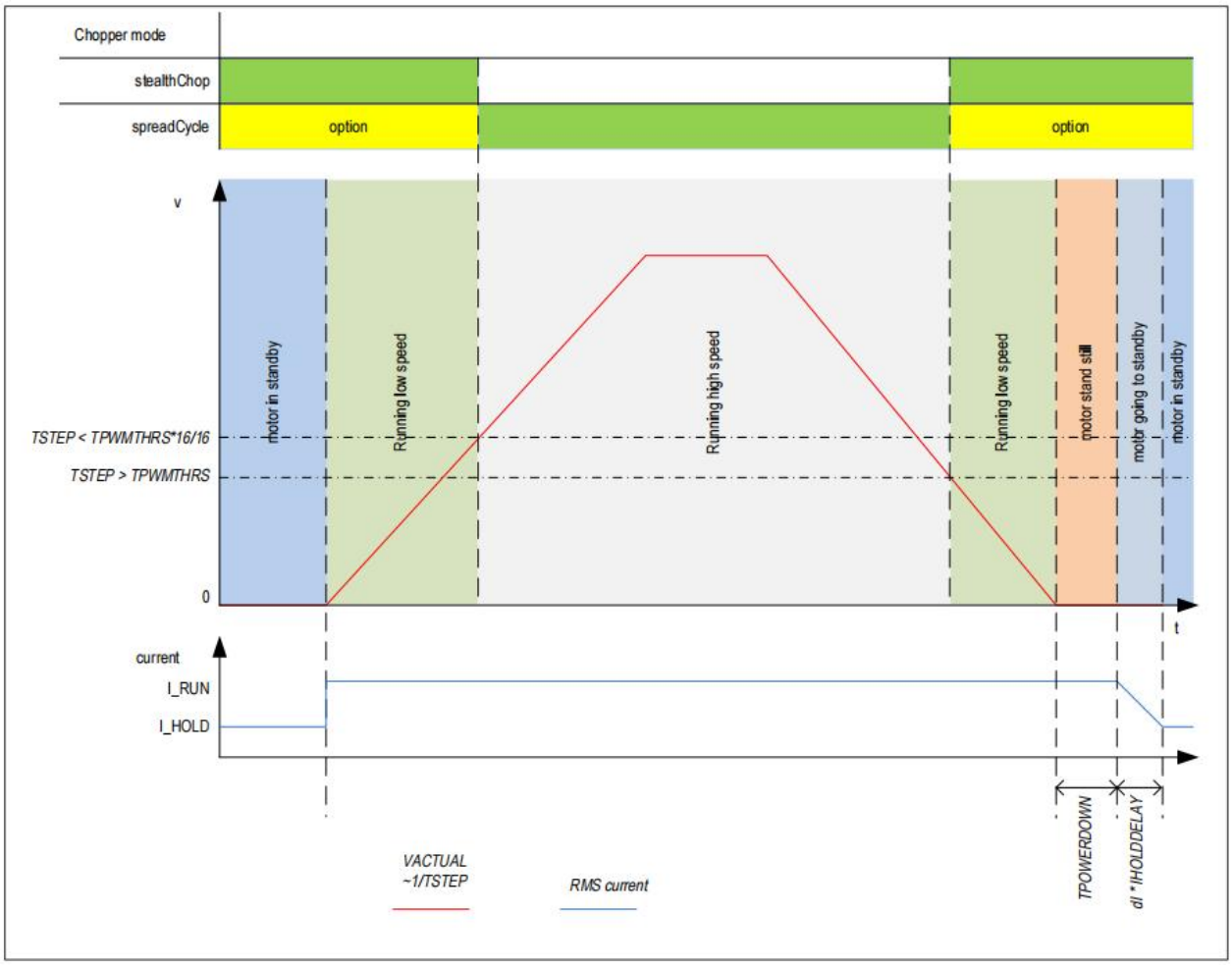


图 12. TPWMTHRS 用于可换到 SpreadCycle

首先，对两种斩波原理分别进行参数化和优化。在下一步中，必须确定转换速度。例如，StealthChop 操作用于精确的低速定位，而 SpreadCycle 将用于高动态运动。TPWMTHRS 决定切换速度。在到达所需切换速度时读出 TSTEP，并将改结果值设置到 TPWMTHRS。 使用较低的切换速度来避免切换点出现抖动。

无抖动地切换至SpreadCycle:

当在高速度切换时会出现抖动，因为电机的反电动势（随速度上升）会导致电机电压和电机电流之间产生高达 90° 的相位移。因此，当在电压 PWM 和电流 PWM 模式之间在高速度切换时，这种抖动的强度会增大。高速跳动甚至可能产生暂时的过流情况（取决于电机线圈电阻）。在低速（例如 1 到几 10 RPM）下，大多数电机可以完全忽略它。因此，在 SpreadCycle 和 StealthChop 之间切换驱动程序时，请考虑抖动。通过 TPWMTHRS 来控制的自动切换时，驱动器可以通过使用 StallGuard4 来确定相移来自动消除抖动。它将对 SpreadCycle 施加相同的相移，直到速度回落到阈值以下。设置标志 SG4_THRS.sg_angle_offset 以启用此功能。

PRELIMINARY CONFIDENTIAL

如果您只想使用 StealthChop，请将 TPWMTHRS 设置为零。

首次使用自动电流调节启用 StealthChop 模式时，电机必须处于静止状态才能进行适当的电流调节。当驱动器以较高速度切换到 StealthChop 时，StealthChop 逻辑会存储最后的电流调节设置，直到电机再次返回到较低速度。这样，当返回到较低速度时，调节具有一个已知的起点，此时 StealthChop 重新启用。因此，在斩波器切换到不同模式的阶段期间，速度阈值和电源电压都不会发生明显变化，因为否则电机可能会失步或瞬时电流可能过高或过低。

电机失速或电机速度的突然变化可能导致驱动器检测到短路或进入自动电流调节状态，无法恢复。 清除错误标志并从零速重新启动电机以从这种情况中恢复。

在第一次开启 StealthChop 时从静止状态启动电机，并使其停止至少 128 个斩波周期，以允许 StealthChop 进行初始静止电流控制。

StealthChop 中的标志

由于 StealthChop 使用电压模式驱动，基于电流测量的状态标志响应较慢，驱动器分别对反电动势的突然变化做出延迟反应，例如电机停转。

在 StealthChop 运行期间电机失速或运动突然停止可能导致过流情况。根据之前的电机速度，以及电机的线圈电阻，它会显着增加电机电流，持续数 10 毫秒。在低速下，反电动势只是电源电压的一小部分，不会触发短路检测的危险。

使用 StealthChop 时，调整低侧驱动器过流检测以安全触发电机失速。 这将避免从电源汲取高峰值电流。

Open Load Flags

在 StealthChop 模式下，状态信息与逐周期调节的 SpreadCycle 模式不同。标志 OLA 和 OLB 表示电流调节达到两个线圈的标称电流。

- 跳变的 OLA 或 OLB 可能是由于电机线圈的差异太大。
- 中断的电机线圈会导致线圈的持续激活的开路负载标记。
- 如果电流调节在最后几个完整步长内未能成功放大到完整目标电流（因为没有连接电机或高速超过 PWM 限制），则一个或两个标志处于活动状态.

如果需要，请使用 SpreadCycle 斩波器进行按需开路测试，因为它可以提供最安全的结果。使用 StealthChop，可以检查 PWM_SCALE_SUM 以检测正确的线圈电阻。

PWM_SCALE_SUM 提示电机状态

通过读取 PWM_SCALE_SUM，可通过自动缩放获得有关电机状态的信息。由于该参数反映了驱动目标电流进入电机所需的实际电压，它取决于几个因素：电机负载、线圈电阻、电源电压和电流设置。 因此，评估 PWM_SCALE_SUM 值可以检查电机工作点。当达到限值 (1023) 时，电流调节器则无法维持电机的全部电流，例如 由于提供的电压下降。

空转和被动制动

StealthChop 为电机停止提供了不同的选项。这些选项可以通过将静止电流 IHOLD 设置为零并使用 FREEWHEEL 设置选择所需的选项来启用。所需选项在 TPOWERDOWN 和 IHOLDDELAY 指定的时间段后启用。 一旦电机目标电流为零，电流调节就会冻结，以确保快速启动。通过空转选项，可以实现空转和被动制动。被动制动是一种有效的涡流电机制动，它消耗的能量最少，因为没有主动电流驱动到线圈中。然而，当施加连续扭矩时，被动制动将允许电机缓慢转动。

探索 StealthChop 时，在您的应用程序中操作电机。 带有机械负载的电机性能通常会更好，因为它可以防止电机因空载时可能发生的机械振荡而失速。

PRELIMINARY CONFIDENTIAL

StealthChop 相关参数

下表包含与 StealthChop 斩波器模式相关的所有参数。

表 10.
StealthChop 相关参数

Parameter/ 参数	Description/描述	设置	Comment/备注
en_spread_ cycle	StealthChop 的禁用 (寄存器GCONF).	1	不使用 StealthChop
		0	StealthChop 使能
pwm_meas_sd_enable	在慢衰减阶段控制电流测量 默认=0	0	仅在导通阶段（ON）测量电流。 适用电流下限.
		1	在慢衰减阶段额外测量的电流以克服电流下限。
pwm_dis_reg_stst	此选项可消除静止期间的任何调节噪音。 默认=0	0	电流调节始终开启.
		1	电机静止且电流减小（小于 IRUN）时禁用电流调节.
TPWMTHRS	指定 StealthChop 中操作的速度上限。当以所需的阈值速度切换时，输入 TSTEP 读数（两个微步之间的时间）。	0 … 1048575	如果 TSTEP 低于 TPWMTHRS，则 StealthChop被禁用
PWM_LIM	从 SpreadCycle 切换到 StealthChop 时限制电流急跳的限制值。减小该值以减小电流跳动。	0 … 15	8位限幅的高四位（默认=12）
pwm_ autoscale	启用使用电流测量的自动电流缩放。 如果关闭，则使用基于速度PWM标定模式。	0	基于速度PWM标定
		1	使用电流调节器自动缩放
pwm_ autograd	启用 PWM_GRAD_AUTO 的自动调整	0	禁用，使用寄存器中的 PWM_GRAD 代替
		1	使能
PWM_FREQ	PWM频率选择。使用最低设置可获得良好的效果。 在每个斩波器输出处测得的频率是有效斩波器频率 f PWM 的一半。	0	$f_{PWM}=2/1024\ f_{CLK}$
		1	$f_{PWM}=2/683\ f_{CLK}$
		2	$f_{PWM}=2/512\ f_{CLK}$
		3	$f_{PWM}=2/410\ f_{CLK}$
PWM_REG	用户定义的 PWM 幅度调节环 P 系数。 当 pwm_autoscale=1 时，较高的值会导致较高的适配速度。	1 … 15	PWM_SCALE_AUTOregulator 每整步产生 0.5 到 7.5 步
PWM_OFS	用户定义的 PWM 幅度（偏移量）用于基于速度的缩放和用于自动调整 PWM_OFFS_AUTO 的初始化值。	0 … 255	PWM_OFS=0 禁用基于电流设置的线性电流缩放
PWM_GRAD	用户定义的 PWM 幅度（梯度）用于基于速度的缩放和用于自动调整 PWM_GRAD_AUTO 的初始化值。	0 … 255	
FREEWHEEL	电机待机电流设置为零 (I_HOLD=0) 时的静止选项。仅在启用 StealthChop 时可用。 freewheeling自由旋转使电机易于移动，而两个线圈短路选项实现了被动制动。	0	正常模式
		1	Freewheeling自由旋转
		2	通过驱动低测短路线圈
		3	通过驱动高测短路线圈

PRELIMINARY CONFIDENTIAL

表 10.
与 StealthChop 相关的参数（待续）

PWM_SCALE _AUTO	回读由电流调节器确定的实际 StealthChop 电压 PWM 缩放修正。 调谐时应调节至接近 0。	-255 … 255	（只读）在 SpreadCycle 中操作时，缩放值变为冻结
PWM_GRAD _AUTO PWM_OFS _AUTO	允许监控自动调整和确定 PWM_OFS 和 PWM_GRAD 的初始值。	0 … 255	(read only)
TOFF慢衰减占用的时间	电机驱动器的一般使能，实际值不影响 StealthChop	0	驱动器去使能
		1 … 15	驱动器使能
TBL快衰减屏蔽开关事件时间	比较器的空白时间。 对于典型应用，选择 1 或 2 的设置。 对于更高的容性负载，可能需要 3 个。 较低的设置允许 StealthChop 调节到较低的线圈电流值。	0	16 t_{CLK}
		1	24 t_{CLK}
		2	36 t_{CLK}
		3	54 t_{CLK}

SpreadCycle 和经典斩波器

StealthChop 是一种电压模式 PWM 控制的斩波器，而 SpreadCycle 是一种逐周期电流斩波控制。 因此，它可以对电机速度或电机负载的变化做出极快的反应。使用斩波器控制通过两个电机线圈的电流。 斩波器彼此独立工作。 下图中显示了不同的斩波器相位。

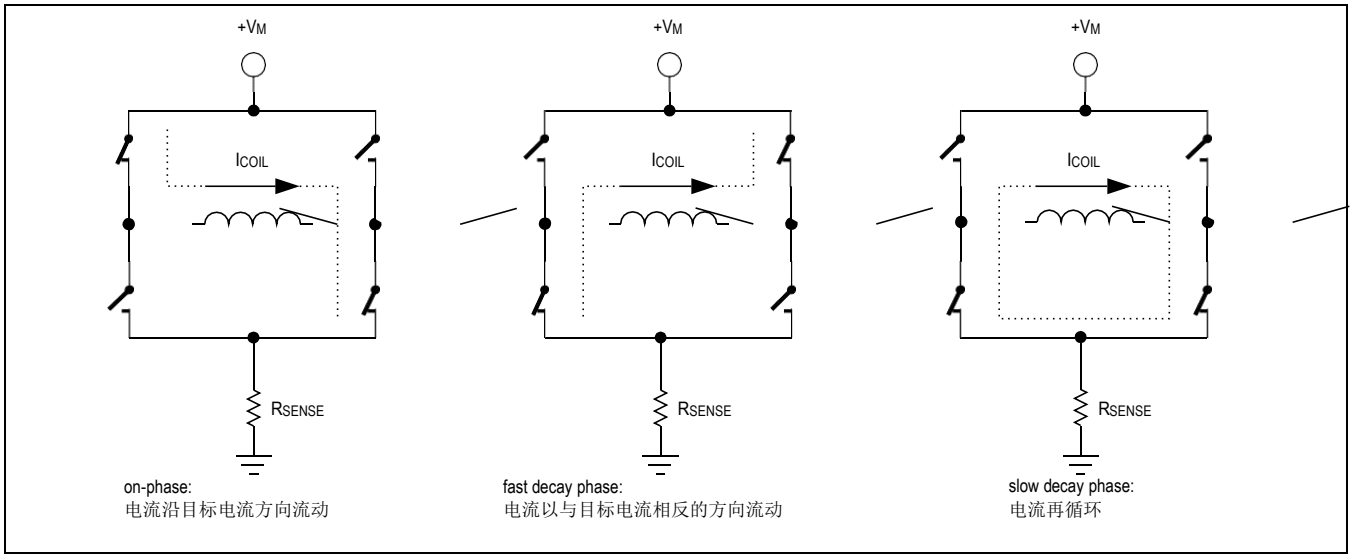


图 13. 典型斩波衰减阶段

尽管可以仅使用全开和快衰减来调节电流，但插入慢衰减对于减少电机中的电气损耗和电流纹波很重要。慢衰减阶段的持续时间在控制参数中指定，并设置斩波频率的上限。电流比较器在电流正好流过一个低边晶体管的阶段测量线圈电流，但在缓慢衰减阶段不测量。 慢衰减阶段由计时器终止。当通过线圈的电流达到目标电流时，比较器终止导通相位。快速衰减阶段可以由比较器或另一个定时器终止。

当切换线圈电流时，基于 RDSon 的电流测量会由于寄生电容的充电和放电而出现尖峰。在这段时间内，通常是一到两微秒，电流无法测量。 Blanking消隐是屏蔽比较器的输入以阻止这些尖峰的时间。

PRELIMINARY CONFIDENTIAL

有两种逐周期斩波模式可用：一种称为 SpreadCycle 的新型高性能斩波算法和一种经过验证的恒定关断时间斩波模式。恒定关闭时间模式通过三个阶段循环：开启、快速衰减和慢速衰减。 SpreadCycle 模式通过四个阶段循环：开启、慢衰减、快速衰减和第二个慢衰减。

斩波频率是斩波电机驱动器的一个重要参数。太低的频率可能会产生听得见的噪音。较高的频率会降低电机中的电流纹波，但频率过高可能会增加磁损耗。此外，驱动器中的功耗也随着频率的增加而增加，这是因为开关斜率的影响会增加，从而导致动态耗散。因此，需要找到折中方案。 大多数电机的最佳工作频率范围为 16 kHz 至 30 kHz。 斩波器频率受许多参数设置以及电机电感和电源电压的影响。

提示：在使用 SpreadCycle 时，25 kHz 到 40 kHz 范围内的斩波器频率对大多数电机来说是一个很好的结果。 较高的频率会导致增加的开关损耗。

表 11. 用于控制 SpreadCycle 和 Classic Chopper 的参数

参数	描述	设置	备注
TOFF	设置慢衰减时间 (*off time*). 该设置还限制了最大斩波频率。 使用 StealthChop 操作时，不使用此参数，但需要它来使能电机。 在仅使用 StealthChop 操作的情况下，设置任何值都可以。 将此参数设置为零会完全禁用所有功率管，并且电机可以自由转动。	0	chopper off
		1…15	off time setting N_{CLK}= 24 + 32**TOFF* (1 将在 24 个时钟的最小空白时间下工作)
TBL	快衰减屏蔽开关事件时间选择比较器空白时间Blank time。 这个时间需要安全地覆盖切换事件和振铃的持续时间。 对于大多数应用，设置为 1 或 2 即可。 对于高容性负载，例如使用滤波电路时，需要设置为 2 或 3。	0	16 t_{CLK}
		1	24 t_{CLK}
		2	36 t_{CLK}
		3	54 t_{CLK}
chm	选择斩波模式	0	SpreadCycle
		1	classic const. off time

SpreadCycle 斩波

SpreadCycle（专利）斩波器算法是一种精确且易于使用的斩波器模式，可自动确定快速衰减相位的最佳时间长度。即使在默认设置下，SpreadCycle 也能提供卓越的微步质量。 有几个参数可用于针对应用优化斩波器。

每个斩波周期由开启阶段、慢衰减阶段、快速衰减阶段和第二慢衰减阶段组成。每个斩波周期的两个慢衰减阶段和两个空白时间为斩波频率设置了上限。缓慢衰减阶段通常占静止斩波周期的 30%-70% 左右，对于低电机和驱动器功耗非常重要。

慢衰减时间 TOFF 的起始值计算示例：

- 目标斩波器频率：25kHz。
  - 假设：两个缓慢的衰减周期占整个斩波器周期时间的 50%
- t_{OFF} = 1 / 25kHz * 50 / 100 * 1 / 2 = 10 µ *s*
- 对于 TOFF 设置，这意味着: TOFF = $\left(t_{OFF} * f_{CLK} - 12\right) / 32$
- 对于 12 MHz 时钟，这会导致 TOFF=3.4，这需要设置 TOFF = 3 或 4。
- 对于 16 MHz 时钟，这会导致 TOFF=4.6，这需要设置 TOFF = 4 或 5。

提示: 最高的电机速度有时会受益于将 TOFF 设置为 1 或 2 并将 TBL 设置为 1 或 0。

迟滞启动设置迫使驱动器将最小量的电流纹波引入电机线圈。电流纹波必须高于由电机中的电阻损耗引起的电流纹波，以提供最佳微步结果。

PRELIMINARY CONFIDENTIAL

这将允许斩波器针对目标电流的上升和下降精确地调节电流。将电流纹波引入电机线圈所需的时间也会降低斩波频率。因此，较高的滞后设置将导致较低的斩波频率。电机电感限制了斩波器跟随电机电流变化的能力。此外，导通阶段和快衰减的持续时间必须长于消隐Blanking时间，因为电流比较器在消隐期间被禁用。

最容易找到最佳设置的方法是从低滞后设置（例如 HSTRT=0、HEND=0）开始并增加 HSTRT，直到电机在低速设置下平稳运行。最好在用电流探头测量电机电流时检查这一点。检查过零附近的正弦波形将在两个半波之间显示一个小凸台，以防滞后设置太小。在中等速度下（即每秒 100 到 400 个整步），过低的滞后设置会导致电机的嗡嗡声和振动增加。过高的滞后设置会导致斩波频率降低和斩波噪声增加，但不会对波形产生任何好处。

如实验所示，设置完全独立于电机，因为更高电流的电机通常也具有更低的线圈电阻。 因此，为迟滞选择低到中等的默认值（例如，有效迟滞 = 4）通常适合大多数应用。可以通过对电机进行试验来优化设置：设置太低会导致微步精度降低，而设置太高会导致斩波器噪音和电机功耗增加。当快速衰减时间比消隐时间稍长时，设置为最佳。 如果难以达到，您可以减少关闭时间设置。

迟滞原理在某些情况下会导致斩波频率变得过低, 例如，当线圈电阻与供电电压相比高时。这可以通过将滞后设置分为开始设置 (HSTRT+HEND) 和结束设置 (HEND) 来避免。自动迟滞递减器 (HDEC) 通过每 16 个系统时钟逐步递减迟滞值，在两种设置之间进行插值。在每个斩波周期开始时，迟滞以起始值和结束值之和 (HSTRT+HEND) 的值开始，并在周期期间递减，直到斩波周期结束或迟滞结束值 (HEND ） 到达。这样，如果频率变得太低，斩波器频率就会稳定在高幅度和低电源电压的情况下。 这样可以避免频率达到可听见范围。

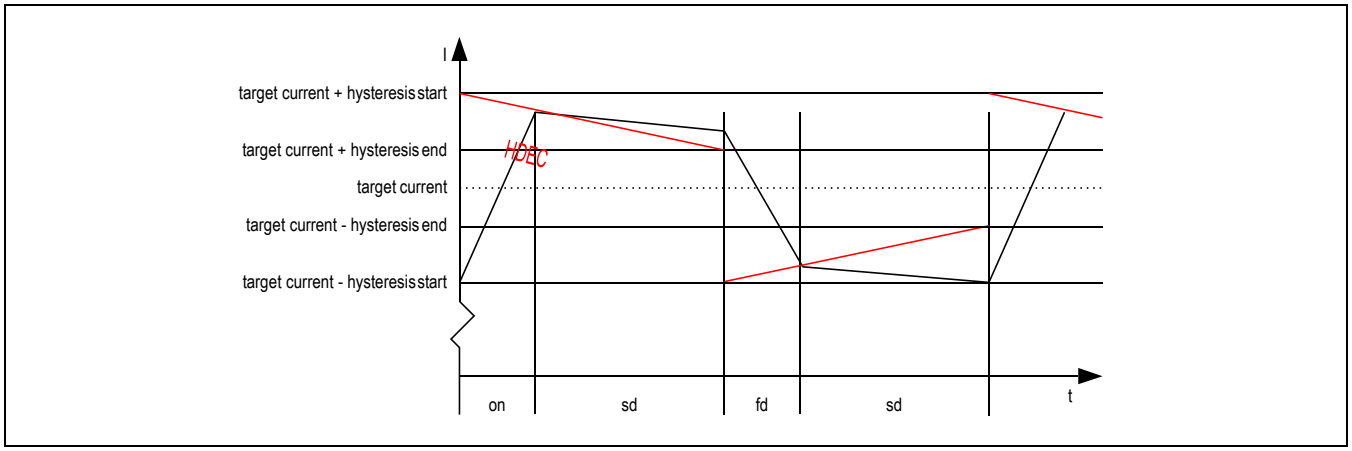


图 14. SpreadCycle 斩波器方案显示斩波器周期期间的线圈电流

表 12. SpreadCycle 模式参数

参数	描述	设置	备注
HSTRT	*迟滞启动设置。 该值是滞后结束值 HEND 的偏移量。*	0…7	*HSTRT*=1…8 该值添加 到 HEND.
HEND	*迟滞结束设置。 在递减数次后设置迟滞结束值。*HSTRT+HEND 之和必须≤16。 在最大的电流设置为30（幅度降低到240），总和不受限制。	0…2	-3…-1: 负 HEND

PRELIMINARY CONFIDENTIAL

Table 12. SpreadCycle mode parameters (continued)

		3	0: zero HEND
		4…15	1…12: 正 HEND

即使在 HSTRT=0 和 HEND=0 时，TMC5240 也会通过模拟电路设置一个最小滞后。

例子：

已选择 4 的滞后。 您可能决定不使用滞后递减。 在这种情况下设置：

HEND=6 (设置一个有效的最终值 6-3=3)

HSTRT=0 (设置最小滞后，即 1: 3+1=4)

为了利用可变滞后，我们可以将大部分值设置为 HSTRT，即 4, 剩下的 1 到迟滞结束。 生成的配置寄存器值如下：

HEND=0 (设置一个有效的最终值 -3)

HSTRT=6 (设置迟滞结束的有效起始值 +7: 7-3=4)

常规的恒定关闭时间斩波器

经典的恒定关断时间斩波器是 SpreadCycle 的替代方案。恒定关闭时间斩波器在每个开启阶段之后使用固定时间快衰减。同时导通阶段的持续时间由斩波比较器确定，快衰减时间需要足够长，以使驱动器跟随正弦波的下降斜率，但不应太长，以免导致电机电流纹波和功耗过大。这可以使用示波器或评估不同速度下的电机平滑度进行调整。 一个好的起始值是类似于慢衰减时间设置和快衰减时间设置相似。

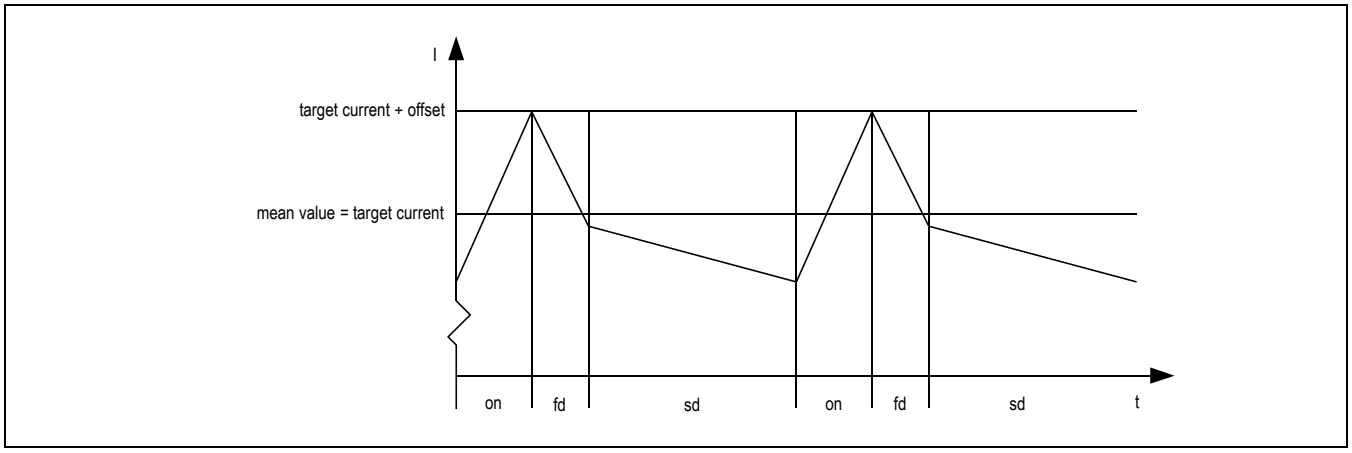


图15。经典的常量关断时间偏移显示线圈电流的关断时间斩波器

调整快速衰减时间后，应调整偏移以实现平滑的过零。这是必要的，因为快衰减阶段使电机电流的绝对值低于目标电流（参见下图）。如果零偏移过低，则电机在电流过零期间会短暂静止。 如果它设置得太高，它会产生更大的微步。 通常，需要正向偏移设置才能实现最平稳的操作。

PRELIMINARY CONFIDENTIAL

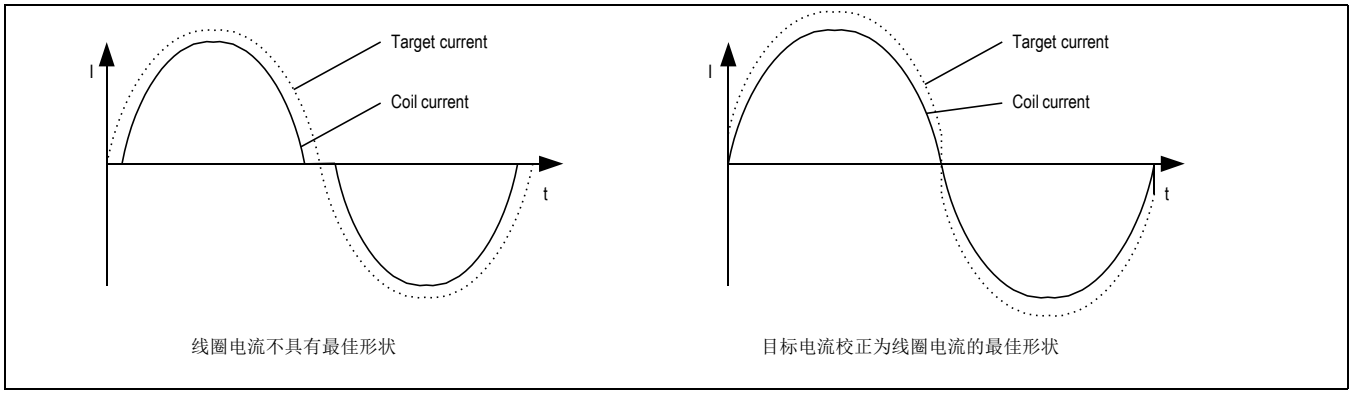


图 16. 使用经典斩波器的过零和使用正弦波偏移的校正

表 13. 控制恒定关断时间斩波器模式的参数

参数	描述	设置	备注
TFD (*fd3* & *HSTRT*)	*快速衰减时间设置。 当 CHM=1 时，这些位控制每个斩波器周期的快速衰减部分。*	0	仅慢衰减
		1…15	快衰减阶段的持续时间
OFFSET (*HEND*)	*正弦波偏移。* CHM=1 时，这些位控制正弦波偏移。 正偏移校正过零误差。	0…2	负偏移：-3…-1
		3	无偏移: 0
		4…15	正偏移：1…12
disfdcc	选择使用电流比较器来终止快速衰减周期。如果启用了电流比较器，当电流达到大于实际正值的负值时，它会终止快速衰减周期。	0	启用快衰减周期的比较器终止
		1	仅按时间结束

集成电流检测 (ICS)

集成了非耗散电流检测。此功能消除了该功能通常需要笨重的外部功率电阻器。与基于外部检测电阻的主流应用相比，ICS 显着节省了空间和功耗。

设置电机电流

表 14. 控制电机电流的参数

Parameter	Description	Setting	Comment
IRUN	电机运行时的电流刻度。 缩放取自内部正弦波表的线圈电流值。 对于高精度电机操作，使用 16 到 31 范围内的电流比例因子，因为按比例缩小电流值会通过使微步变粗来降低有效微步分辨率。 此设置还控制 CoolStep 设置的最大电流值。	0 … 31	scaling factor 1/32, 2/32, … 32/32
IHOLD	与 IRUN 相同，但用于电机在静止时候。		

PRELIMINARY CONFIDENTIAL

表 14. 控制电机电流的参数（待续）

IHOLDDELAY	允许从运行电流到保持电流的平滑电流下降。IHOLDDELAY 以 2^18 个时钟为增量控制在 TZEROWAIT 后电机电流下降的时钟周期数：: 0=瞬间下降, 1..15: 以 2^18 个时钟的倍数计算每个电流步长的电流降低延迟。 示例：当使用 IRUN=31 和 IHOLD=16 时，需要 15 个电流步骤来降低保持电流。 因此，IHOLDDELAY 设置为 4 会导致下降时间为 4*15*2^18 个时钟周期，即在 16MHz 时大约为一秒。	0	瞬间下降至 IHOLD
		1 …15	$1*2^{18}$ … $15*2^{18}$ 每个电流递减的时钟
IRUNDELAY	控制检测到启动后电机通电的时钟周期数。 允许在从保持电流 (IHOLD) 到运行电流 (IRUN) 的运动开始时平滑电流增量。 虽然快速上电对于建立完整的电机扭矩很重要，但一小段延迟时间有助于减少噪音并避免电源电流的抖动。	0	即时通电至 IRUN
		1...15	每个电流增量步长的延迟，以 IRUNDELAY 的倍数计算 * 512 个时钟

设置满量程电流范围

满量程电流通过一个外部参考电阻和 DRV_CONF 寄存器中的 2 位来选择。可配置 3 种不同的满量程电流范围，以适应不同的电机尺寸和应用。这是需要受益于最好的电流控制解决方案。

因此，在 IREF 和 GND 之间连接一个电阻，以设置满量程斩波电流 I_{FS}.

DRV_CONF 寄存器中的位 1..0 定义了驱动级的典型导通电阻，并根据外部电阻进一步控制满量程范围。

下面的等式显示了满量程电流与连接到引脚 IREF 的 RREF 分流电阻和 DRV_CONF 寄存器位设置的函数关系。

比例常数 KIFS 取决于所选的满量程设置（DRV_CONF 寄存器位 1..0），外部电阻 RREF 的范围可以在 12KΩ 和 60KΩ 之间。

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

表 15. IFS 满量程设置（RREF = 12KΩ 的示例）

寄存器配置 DRV_CONF bits 1..0	K_{IFS} (A*kΩ)	最大 FS 设置	典型 Rds ON（高测 + 低测）	Notes
11	36	3A	0.23Ω	优化的效率和扩展的工作范围高达 3A FS。
10	36	3A	0.23Ω	优化的效率和标准操作范围高达3A FS。
01	24	2A	0.27Ω	操作范围缩小至 2AFS。 当需要在较低电流下获得高精度时。
00 (default)	11.75	1A	0.40Ω	操作范围缩小至 1AFS。 当需要在较低电流下获得高精度时。

基于速度模式控制

TMC5240 允许配置不同的斩波器模式和操作模式以实现最佳的电机控制。根据电机负载，可以优化不同模式以实现最低噪声和高精度、最高动态、或最高速度时大扭矩。

PRELIMINARY CONFIDENTIAL

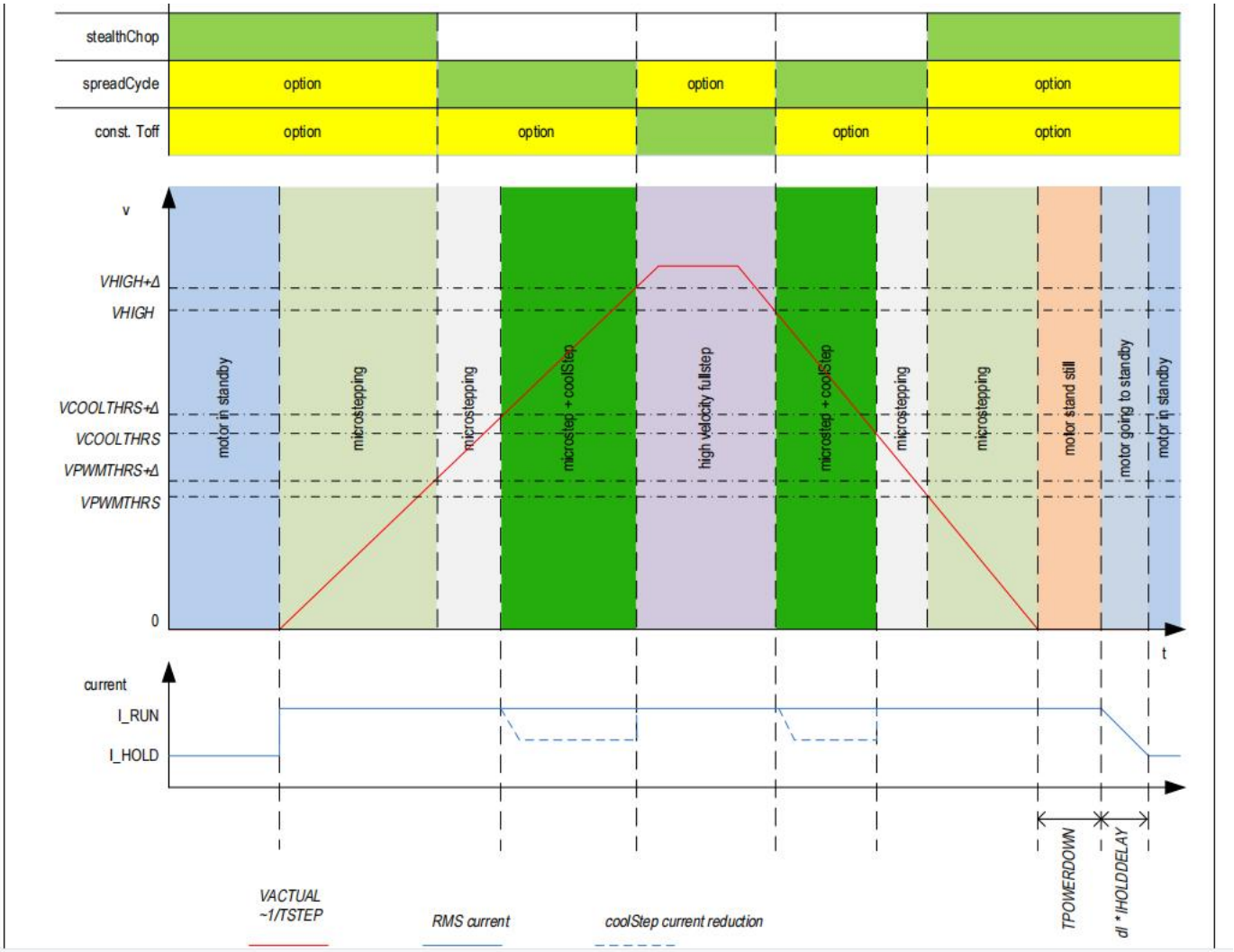


图17。速度相关模式的选择

上图显示了所有可用阈值和所需的顺序VPWMTHRS、VHIGH 和 VCOOLTHRS 由设置 TPWMTHRS、THIGH 和 TCOOLTHRS 确定。速度通过每两个步进脉冲之间的时间间隔 TSTEP 来描述。这允许在使用外部脉冲源时确定速度。*TSTEP 始终以 256 微步为标准*。这样，当微步分辨率发生变化时，阈值就不必进行调整。 阈值代表相同的电机速度，与微步设置无关。TSTEP 与这些阈值相比较。1/16 TSTEP 的滞后。 当 TSTEP 测量中出现抖动时，应用 1/32 TSTEP 以避免比较结果的连续切换。 The upper switching velocity is higher by 1/16, resp. 1/32 of the value set as threshold. StealthChop 阈值 TPWMTHRS 未显示。

它可以包含在 VPWMTHRS < VCOOLTHRS 中。根据停止标志 stst，可以将电机电流编程为运行和保持水平。

使用自动速度阈值可以调整应用程序以适应不同的速度范围。CoolStep 等功能将完全很轻易地集成到您的应用中。 这样一来，一旦参数化，它们就不需要通过软件进行任何激活或停用。

Parameter	Description	Setting	Comment
stst	显示电机在每种操作模式下静止不动。 时间是最后一步脉冲后 2^20 个时钟。	0/1	状态位，只读
TPOWERDOWN	这是电机静止后 (stst) 到电机电流下降的延迟时间. 时间范围约为 0 到 4 秒。 设置 0 为无延迟，1 为一个时钟周期延迟。 进一步的增量是 2^18 个时钟周期的离散步长。	*0…255*	时间为 2^18 的倍数 时钟
TSTEP	两个 1/256 微步之间的实际测量时间，源自，脉冲输入频率，以 1/fCLK 为单位。 溢出或静止时测量值为 (2^20)-1。	0… 1048575	状态寄存器，只读。 实际测量的脉冲时间，以 tCLK 的倍数表示
TPWMTHRS	*TSTEP* ≥ *TPWMTHRS* ● 启用 StealthChop PWM 模式（如果已配置） ● DcStep 禁用	0… 1048575	设置以控制在 StealthChop 中运行的速度上限
TCOOLTHRS	*TCOOLTHRS* ≥ *TSTEP* ≥ *THIGH*: ● CoolStep 已启用（如果已配置） ● StealthChop 电压 PWM 模式禁用 *TCOOLTHRS* ≥ *TSTEP* ● 启用失速输出信号（如果已配置）以与外部控制器一起使用	0… 1048575	设置以控制使用 CoolStep 和 StallGuard 操作的最低速度阈值
THIGH	*TSTEP* ≤ *THIGH*: ● CoolStep 被禁用（电机以正常电流缩放运行） ● StealthChop 电压 PWM 模式禁用 ● *如果vhighchm 被设置*, 斩波器切换到 chm=1 且 TFD=0（仅具有慢衰减的恒定关闭时间）。 ● chopSync2 关闭 (*SYNC*=0) ● 如果 *vhighfs 被设置*, 电机在全步模式下运行，并且失速检测切换到 DcStep 失速检测。	0… 1048575	设置以控制 CoolStep 和 StallGuard 操作的速度上限阈值以及可选的高速步进模式
small_hysteresis	分别基于 TSTEP（下限速度阈值）和 (TSTEP*15/16)-1 (TSTEP*31/32)-1（上限速度阈值）的步进频率比较滞后	0	Hysteresis is 1/16
		1	Hysteresis is 1/32
vhighfs	当超过 VHIGH，该位启用切换到整步模式.仅在 45° 位置进行切换。整步目标电流使用来自 45° 位置的微步表中的电流值。	0	没有切换到整步
		1	高速时切换至整全
vhighchm	当超过 VHIGH 时，该位使能切换到 chm=1 和 fd=0。这样，可以实现更高的速度。可以与 vhighfs=1 结合使用。如果设置，则 TOFF 设置在高速运行期间自动加倍，以避免斩波频率加倍。	0	斩波器模式不变
		1	在高速时使用常规经典常关断时间斩波
en_pwm_mode	StealthChop 电压 PWM 启用标志（取决于速度阈值）。 仅在静止时从关闭状态切换到开启状态。	0	禁用StealthChop
		1	速度在VPWMTHRS以下StealthChop开启

斜坡发生器

斜坡发生器允许基于目标位置或目标速度的运动。 它会根据加速度和速度设置自动计算运动曲线。TMC5240 集成了一种新型斜坡发生器，

PRELIMINARY CONFIDENTIAL

与经典的线性加速斜坡相比，它提供了更快的机器操作。八点斜坡发生器允许使加速度斜坡适应步进电机的转矩曲线。它为加速阶段和减速阶段分别使用三种不同的加速设置，尽可能减小抖动。

实际单位转换

TMC5240 使用其内部或外部时钟信号作为所有内部操作的时间参考。因此，所有时间、速度和加速度设置均以 f_{CLK} 为参考。为获得最佳稳定性和可重复性，建议使用外部石英晶振作为时基，或从微控制器提供时钟信号。

参数/符号 Parame r v	单位	计算/描述/评论
f_{CLK}[Hz]	[Hz]	TMC5240 的时钟频率 [Hz]
s	[s]	second
US	µstep	
FS	fullstep	
µstep velocity v[Hz]	µsteps / s	v[Hz] = v[TMC5240] * (f_{CLK}[Hz]/2 / 2^23)
µstep acceleration a[Hz/s]	µsteps / s^2	a[Hz/s] = a[TMC5240] * f_{CLK}[Hz]^2 / (512*256) / 2^24
USC microstep count	counts	microstep resolution in number of microsteps (i.e. the number of microsteps between two fullsteps – normally 256)
rotations per second v[rps]	rotations / s	v[rps] = v[µsteps/s] / USC / FSC FSC: motor fullsteps per rotation, e.g. 200
rps acceleration a[rps/s^2]	rotations / s^2	a[rps/s^2] = a[µsteps/s^2] / USC / FSC
ramp steps[µsteps] = rs	µsteps	rs = (v[TMC5240])^2 / a[TMC5240] / 2^8 microsteps during linear acceleration ramp (assuming acceleration from 0 to v)
TSTEP, Txxx_THRS	-	TSTEP = f_{CLK} / f_{STEP} The time reference for velocity thresholds is referred to the actual microstep frequency of the clock input respectively velocity v[Hz].
Ramp generator update rate	[Hz]	f_{UPDATE} = f_{CLK} / 512 VACTUAL updates with this frequency.

在极少数情况下，加速度上限可能会限制应用程序,例如，当工作与降低时钟频率或高传动装置和低负载的电机。为了尽可能增加有效加速度，可以降低细分. 设置 CHOPCONF 选项 intpol=1 和 MRES=%0001 将使相同速度设置下的电机速度加倍，因此有效加速度和减速度也加倍。 使用此设置，电机将具有相同的平滑度，但位置分辨率降低一半。

运动曲线

快速开始

如需快速入门，请参阅快速配置指南。

斜坡发生器寄存器组请参考寄存器部分。

斜坡模式

斜坡发生器提供三段加速和三段减速斜坡以及额外的可编程启动和停止速度。

注意

PRELIMINARY CONFIDENTIAL

如果不使用，启动速度可以设置为零。

如果不使用，可以将停止速度设置为较低的值（100 或降至 10）。

注意始终将 VSTOP 设置为等于或高于 VSTART。 这确保了即使是很短的运动也能在目标位置成功停止.

将TVMAX 设置为0 来禁用急速减速.

三组不同的加减速可以自由组合。 过渡速度 V1 和 V2 允许在三个加速和减速设置之间进行速度相关的切换。因为随着电机扭矩在较高速度下下降，典型的高速应用在较高速度下使用较低的加速和减速值当考虑系统中的摩擦时，很明显，系统的减速能力比加速能力快. 因此，在许多应用中可以将减速度值设置得更高。 这样，电机在时间要求严格的应用中的运行速度将最大化。

因为目标位置和斜坡参数可以在运动过程中随时更改,运动控制器将始终使用最佳（最快）方式到达目标，同时遵守用户设置的约束。这样可能会发生运动就会自动停止，过冲后再返回。这种情况由特殊标志 second_move 标记。

斜坡发生器进一步支持自动抖动抑制，通过平滑从加速阶段到减速阶段的过渡，根据机械抖动的情况，并通过强制最短持续时间 (TVMAX) 的恒定速度段从减速阶段到加速阶段.

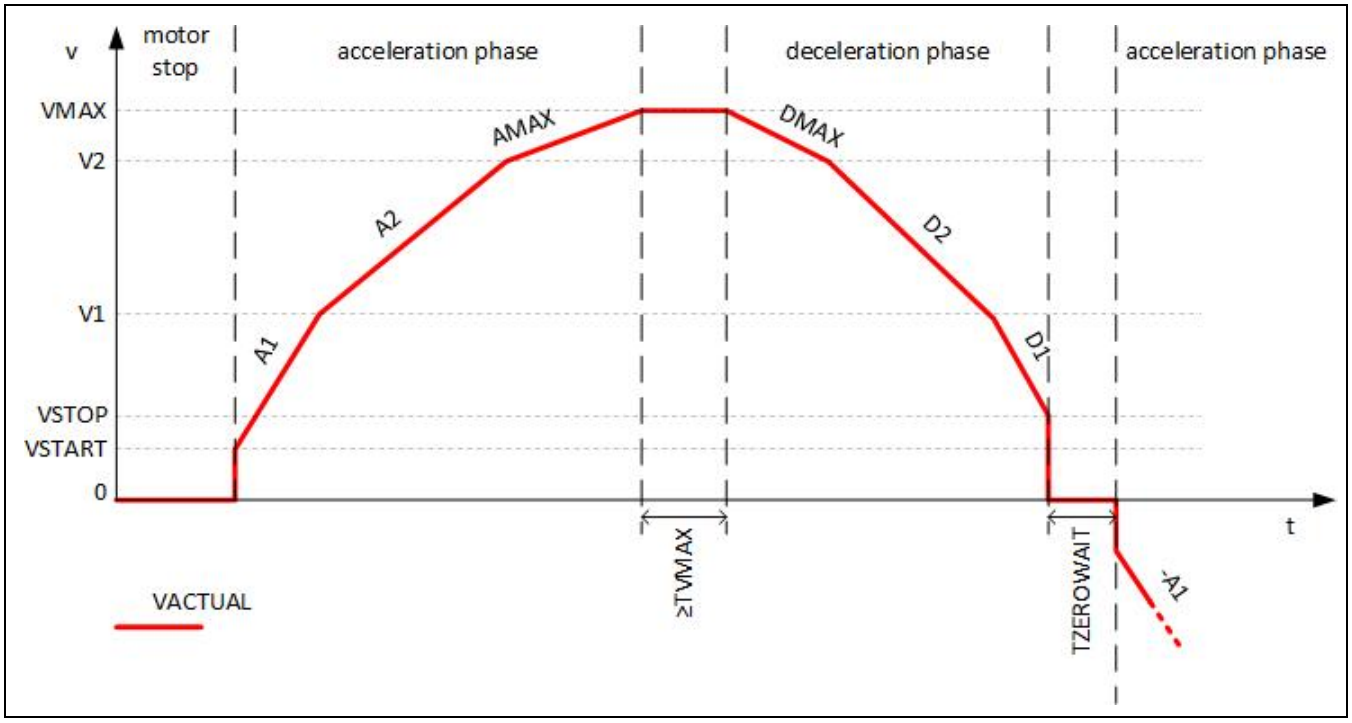


图 18. 斜坡发生器速度轨迹显示随后的负方向移动

PRELIMINARY CONFIDENTIAL

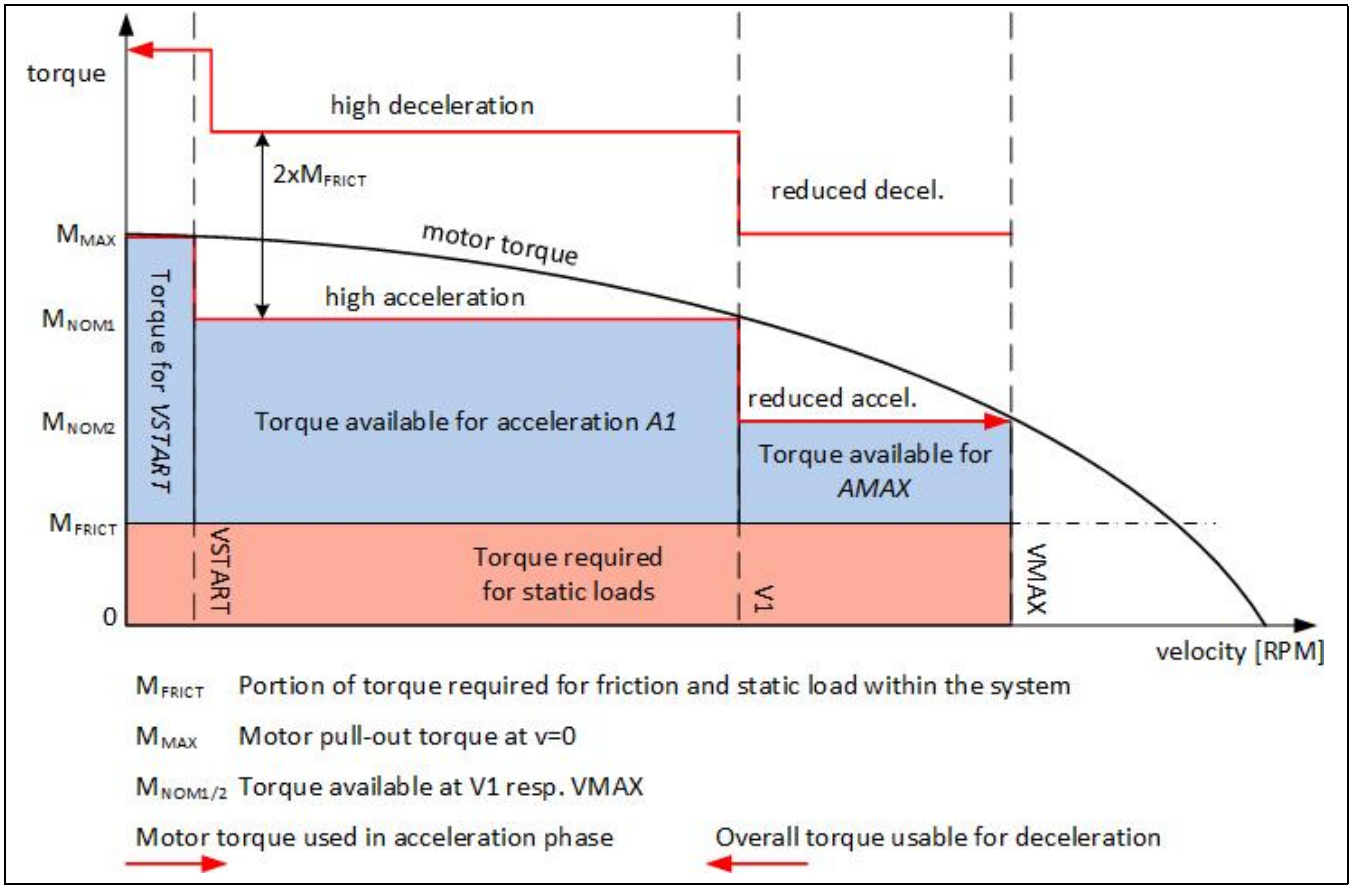


图 19. 使用斜坡发生器优化电机扭矩的图示

开始和停止速度

当使用更高水平的开始和停止速度时，很明显，随后向相反方向移动将提供与 VSTART+VSTOP 相同的加加速度，而不仅仅是 VSTART。由于电机可能无法跟随这一点，您可以通过设置 TZEROWAIT 设置后续移动的时间延迟。运动延迟时间由标志 t_zerowait_active 进行标记。一旦达到目标位置，标志 position_reached 就会激活。

八点斜坡

三个加速和减速段的集合可以以两种方式使用：通过在较低速度下使用较高加速度值来适应电机扭矩曲线，或减少从一个加速段过渡到下一个加速段时的加加速度（加速度变化）。 对于 jerk 优化坡道，通常 A1、D1、AMAX 和 DMAX 设置为低于 A2 和 D2 的值。关于 jerk 的最关键点是从加速到减速的过渡，没有恒速段，以及在目标位置动态变化的情况下从减速到加速的过渡。

为了解决这两个问题，8 点运动曲线生成器允许基于最小分段持续时间 (TVMAX) 强制执行恒速分段。如果由于位移不足而无法保持此持续时间，计算出降低的 VMAX (VMAX´) 并用于等速段. 最小 VMAX´ 与 VSTOP 相同。

以下轨迹显示了基于伪 S 形配置的不同位置位移的结果速度曲线。

PRELIMINARY CONFIDENTIAL

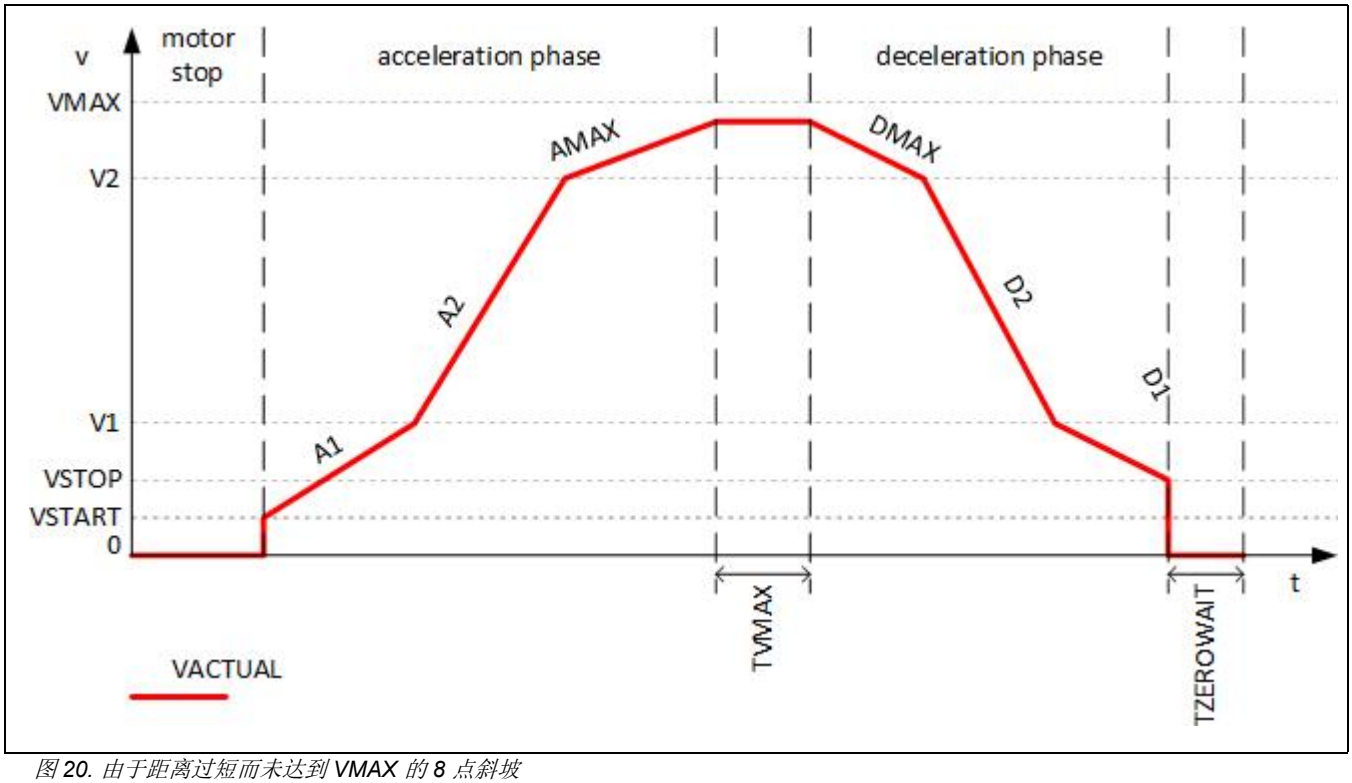


图 20. 由于距离过短而未达到 VMAX 的 8 点斜坡

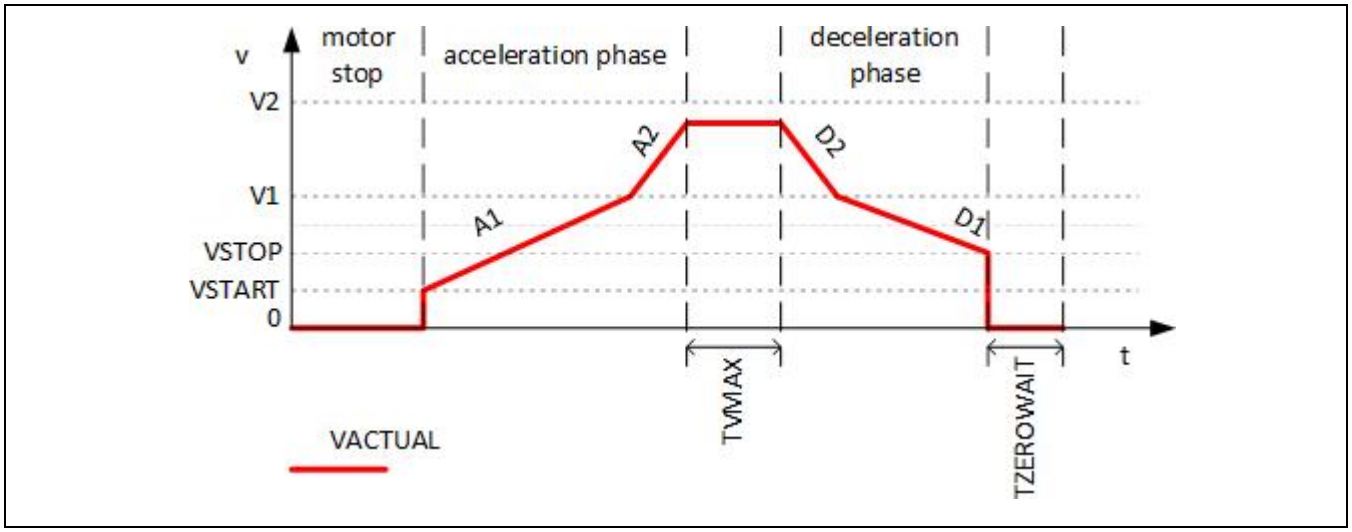


图 21. 由于距离过短低，未达到 V2 且无 AMAX 和 DMAX 阶段

PRELIMINARY CONFIDENTIAL

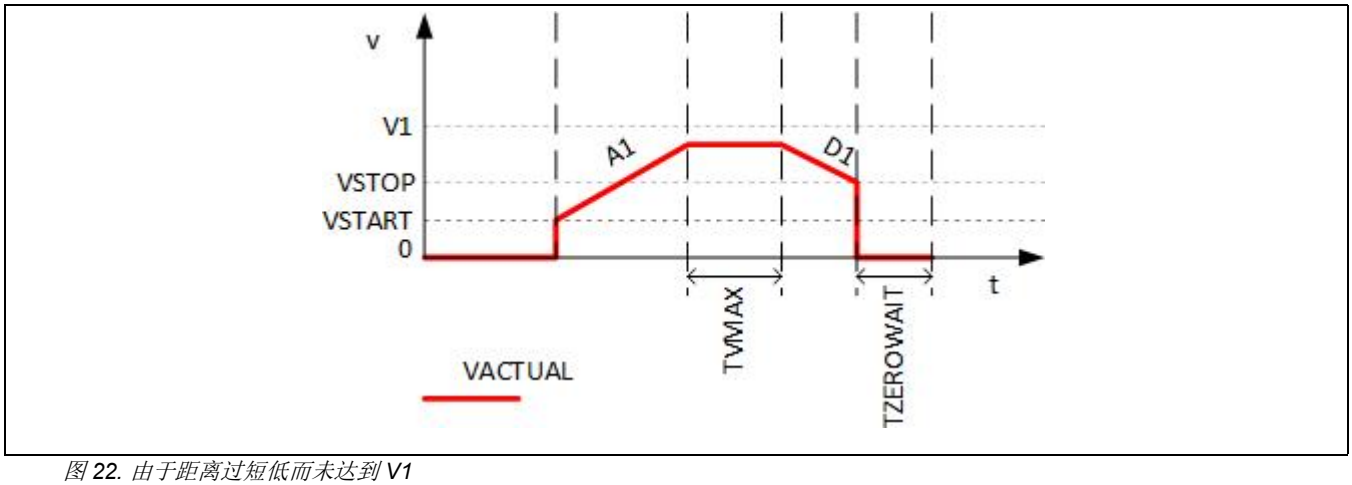


图 22. 由于距离过短低而未达到 V1

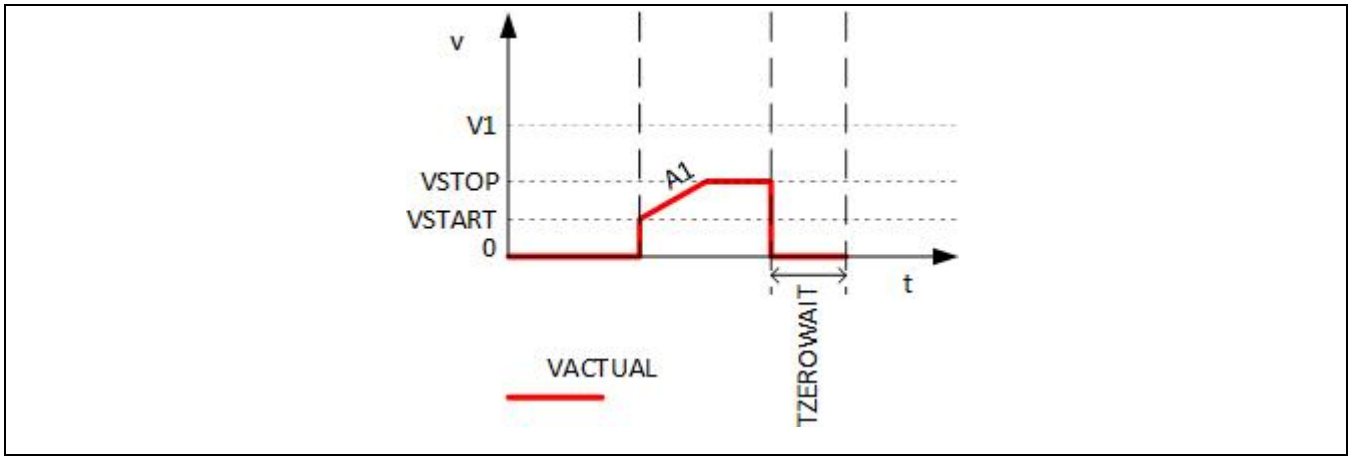


图 23. TVMAX 由于低位移而未保留

PRELIMINARY CONFIDENTIAL

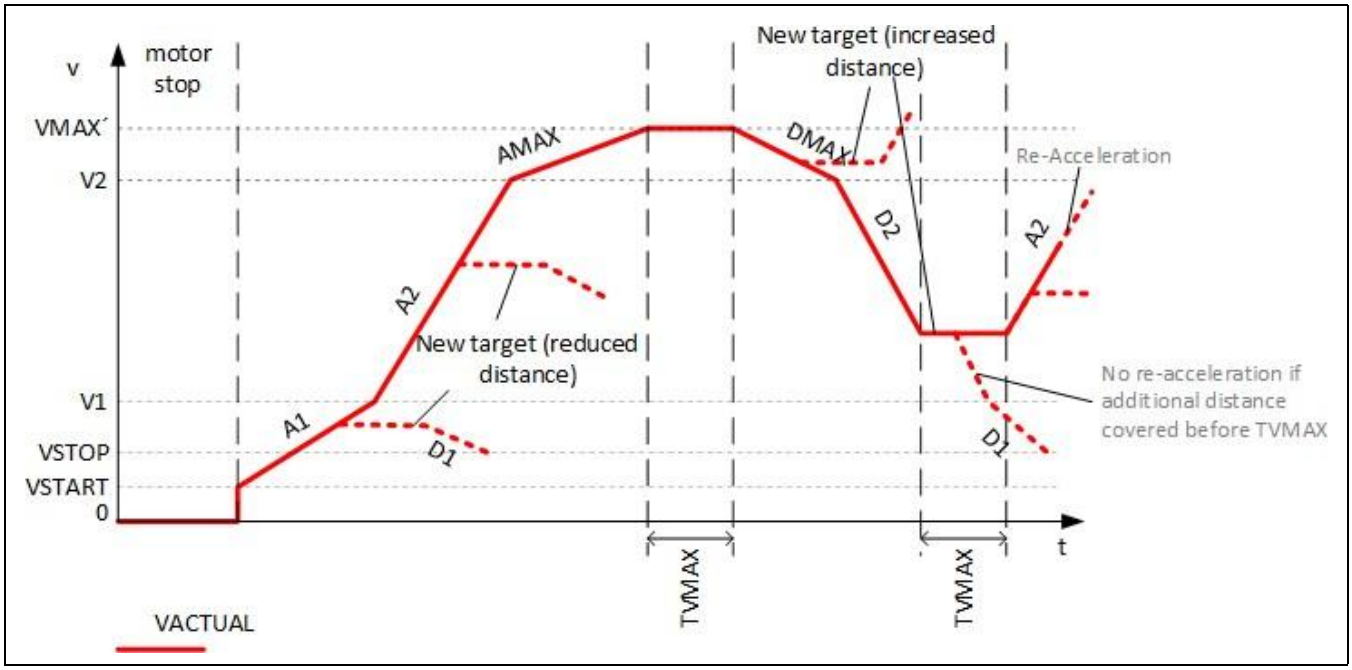


图 24. 带有动态目标位置变化的 8-Point Ramp 示例

速度模式

为方便使用，速度模式运动不使用不同的加速和减速设置。您只需为速度模式设置 VMAX 和 AMAX。在这种模式下，斜坡发生器总是使用 AMAX 加速或减速到 VMAX。

为了使电机减速停止，只需将 VMAX 设置为零即可。标志 vzero 表示电机处于停止状态。 如果velocity_reached 一直有信号，表明已经达到目标速度。

提前终止斜坡

在用户可以与系统交互的情况下，某些应用程序需要在到达目标位置之前通过降低到零速度来终止运动。

使用加速设置来终止运动的选项:

1. 切换到速度模式，将 VMAX=0 和 AMAX 设置为所需的减速度值。这将使用线性斜坡停止电机。
   在定位模式下停止，设置 VSTART=0 和 VMAX=0. *在这种情况下不使用 VSTOP*。 驱动器将使用 AMAX 和 A1（由 V1 确定）以达到零速度。
2. 对于正使用 DMAX、D1、D2 和 VSTOP 进行的停止，通过将 XACTUAL 复制到 XTARGET 来触发减速阶段。 将 TZEROWAIT 设置为足够长以允许 CPU 在此期间进行数据交互。驱动会减速并最终停下来。使用选项 a) 或 b) 在 TZEROWAIT 时间内轮询实际速度以终止运动。
3. 启动一个停止开关。这可以通过硬件输入来完成，例如 使用有线“或”连接到停止开关输入。 如果您不使用硬件输入并将 REFL 和 REFR 绑定到固定电平，请启用停止功能（stop_l_enable、stop_r_enable）并使用反相功能（pol_stop_l、pol_stop_r）来模拟开关激活停止。
4. 利用虚拟停止开关（VIRTUAL_STOP_L、VIRTUAL_STOP_R）。位置比较（X_ACTUAL VIRTUAL_STOP_L/R）将相应地触发停止。

PRELIMINARY CONFIDENTIAL

应用示例：操纵杆控制

使用运动控制器可以优化监控摄像机等应用程序：当操纵杆命令以用户定义的速度操作电机时，目标斜坡生成器确保永远不会离开有效的运动范围。

实现操纵杆控制

1. 使用位置模式来控制运动方向和设置运动极限.
2. 可以随时在VSTART 和您期待的最大速度范围之间修改 VMAX. 在 VSTART=0 时，您也可以通过设置 VMAX=0 来停止运动。运动控制器将使用由 V1 确定的 A1 和 AMAX 来适应加速和减速的速度。
3. 如果不修改加速设置，不需要重写XTARGET，修改VMAX即可。
4. DMAX、D1、D2 和 VSTOP 仅在斜坡控制器由于到达目标位置而减速时使用，或者当目标位置被修改为指向相反的方向时。

速度阈值

斜坡发生器提供与实际速度 VACTUAL 相结合的多个速度阈值。不同的范围允许将电机编程为最佳步进模式、线圈电流和加速度设置。大多数应用不需要所有阈值，但原则上可以组合所有模式。VHIGH 和 VCOOLTHRS 由寄存器 THIGH 和 TCOOLTHRS 确定，以便在使用外部脉冲源时确定速度。TSTEP 与这些阈值进行比较。滞后响应为1/16 TSTEP。1/32 TSTEP用于避免在TSTEP测量发生抖动时，比较结果的连续切换。 上限开关速度高出 1/16，

分别 设置为阈值的值的 1/32。StealthChop 阈值 TPWMTHRS 未显示。 它可以包含在 VPWMTHRS < VCOOLTHRS 中。

PRELIMINARY CONFIDENTIAL

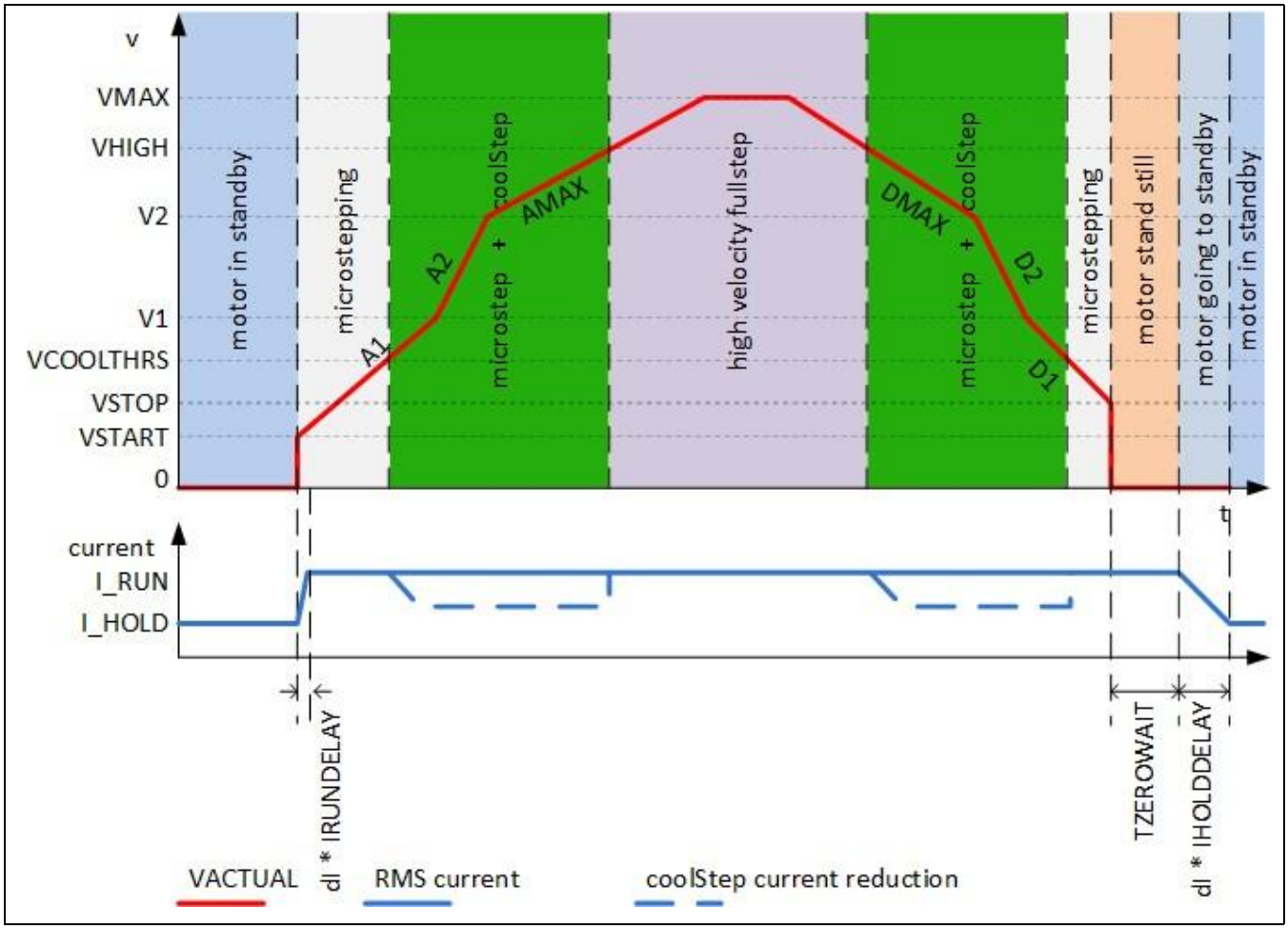


图25。斜坡发生器速度依赖电机控制

不同斩波器模式和无传感器操作特性的速度阈值与每两个微步 TSTEP 之间的时间相关联。

参考开关

在驱动器正常运行之前，必须设置一个绝对参考位置。 可以使用可以通过失速检测或参考开关检测的机械停止来找到参考位置。

在直线驱动的情况下，不得超过机械运动范围。这也可以通过启用左右参考开关的停止功能来确保异常情况。因此，斜坡发生器响应 SW_MODE 寄存器中配置的多个停止事件。有两种停止电机的方法：

- 当开关被触发时，电机可以立即停止。这在紧急情况和基于 StallGuard 的归位中很有用。
- 或者可以使用减速设置（DMAX、V2、D2、V1、D1）将电机缓慢减速到零。

提示
当一个开关事件发生时，将斜坡位置XACTUAL锁存到保持寄存器XLATCH，可以得到参考开关位置的精确判断。

PRELIMINARY CONFIDENTIAL

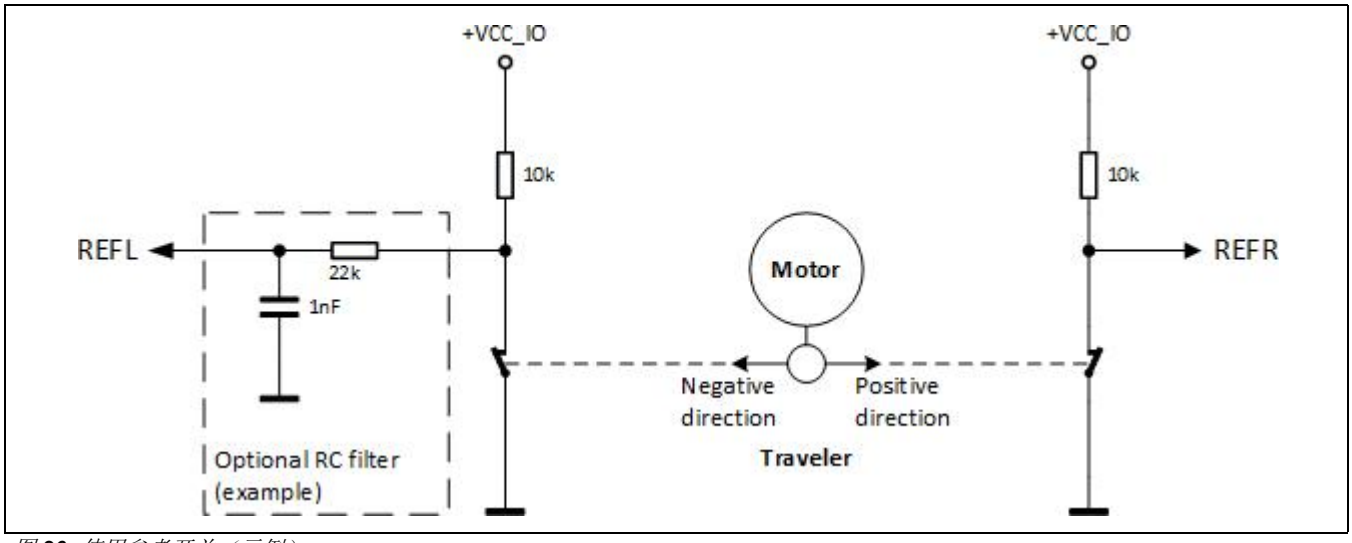


图 26. 使用参考开关（示例）

通过编程开关极性或选择上拉或下拉电阻配置，可以使用常开或常闭开关。常闭开关对于开关连接的中断是故障保护的。可以使用的开关有：

- 机械开关
- 光电传感器, 或
- 霍尔传感器.

> 请谨慎选择符合您的开关要求的参考开关电阻!
>
> 如果电缆较长，则在 TMC5240 参考输入附近可能需要额外的 RC 滤波。添加一个 RC 滤波器也将降低因接线故障而破坏逻辑电平输入的危险，但它会增加一定的延迟，这应该考虑到应用程序中。

实施回零位过程

1. 确保零位开关没有被触发，例如 通过远离零位开关.
2. 在所需的开关事件时激活位置锁定并在激活开关时激活电机（软）停止。基于 StallGuard 的归位需要使用硬停止 (en_softstop=0)。
3. 使用运动斜坡向靠近原点开关的方向运动。 (左开关对应负方向的位置，右开关对应正方向的位置). 您可以通过使用位置斜坡命令来使该运动超时。
4. 一旦到达开关位置，该位置就会被锁定，电机就会停止。 通过轮询实际速度 VACTUAL 或检查 vzero 或停止标志等到电机再次停止。
5. 将斜坡发生器切换到保持模式并计算锁定位置与实际位置之间的差异. 对于基于 StallGuard 的归位或使用硬停止时，XACTUAL 会准确地停在原点位置，因此没有差异 (0)。
6. 将计算出的差值写入实际位置寄存器。现在，回原点完成。移动到位置 0 将使电机精确地回到原点开关处。 如果 StallGuard 用于回原点，对 RAMP_STAT 的读取访问会清除 StallGuard 停止事件 event_stop_sg 并将电机从停止条件中释放出来。

使用第三个开关回原点

一些应用使用额外的原点开关，该开关独立于机械左右限位开关运行。TMC5240 的编码器功能为位置锁定提供了一个额外的来源。 它允许使用 N 通道输入通过上升沿或下降沿事件来快速获取 XACTUAL，或两者都获取。 该功能还提供中断输出。

1. 激活锁存功能（ENCMODE：设置 ignoreAB、clr_cont、neg_edge 或 pos_edge 和 latch_x_act）。

PRELIMINARY CONFIDENTIAL

然后锁存功能可以触发中断输出（当在 DIAG0 发出中断信号时，通过读取 ENC_STATUS 中的 n_event 来检查）。

2. 向 N 通道开关所在的方向移动。 如果电机在检测到原点开关之前碰到限位开关（REFL 或 REFR），反转运动方向。
3. 一旦开关被触发，读出XLATCH。它给出了开关事件的位置。
4. 检测到开关事件后，停止电机，并从实际位置中减去 XLATCH。（所需步骤的详细说明在上面的回原点过程中）

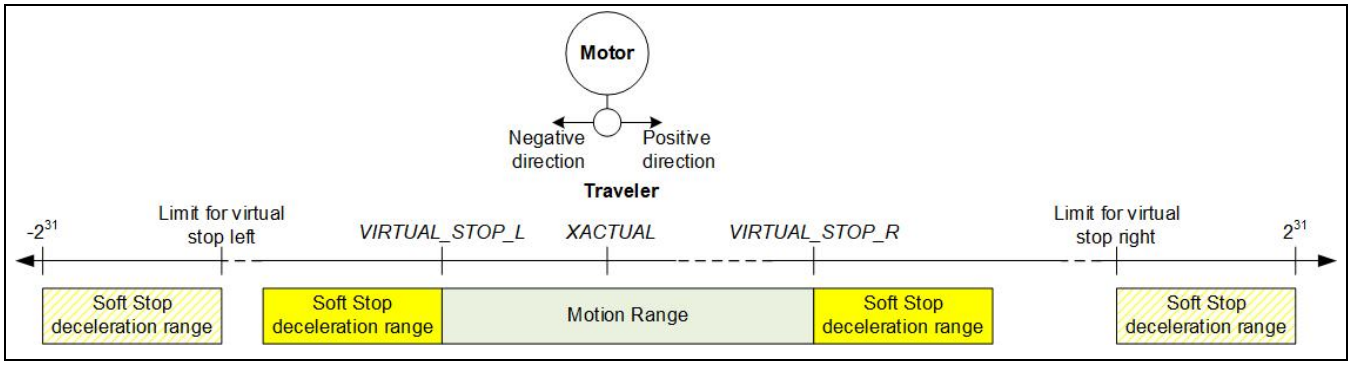
虚拟参考开关

TMC5240 支持虚拟参考开关，以支持只有 1 个或没有参考开关（StallGuard 回零）的应用，以安全地限制物理运动范围。

当实际电机位置 (XACTUAL) 在正向运动期间超过 VIRTUAL_STOP_R 或在负向运动期间低于 VIRTUAL_STOP_L 时，虚拟停止开关将激活。通过设置 en_virtual_stop_l 或 en_virtual_stop_r来启用虚拟停止开关， 每个虚拟停止开关仅阻止向相应方向的运动。

可选地，可以切换虚拟停止以监控编码器位置 (X_ENC)。 要选择，请设置 virtual_stop_enc。

在使用软减速的情况下，将虚拟停止 (VIRTUAL_STOP_R、VIRTUAL_STOP_L) 的值设置为与有符号 32 位运动范围的上溢/下溢范围有足够的距离，以允许电机减速。



控制外部支持 STEP/DIR的驱动器

TMC5240 允许使用内部斜坡发生器来控制外部 STEP/DIR 驱动器，如 TRINAMIC TMC2590、TMC262 或 TMC2240，用于强大的步进应用。 在这种配置下，内部驱动器通常不使用，但可以在外部驱动器的基础上使用，例如需要两台电机同步运行时。

通过设置 GCONF 标志 diag0_step 和 diag1_dir 来启用 DIAG0 和 DIAG1_SW 输出以用于 STEP 和 DIR 输出。 在此模式下，其他内部驱动器功能（如 DcStep 和自动电机电流控制）不可用，因为外部驱动器没有反馈到 TMC5240。

DIAG1_SW 上的低电平对应于正方向的脉冲（XACTUAL 增加），高电平对应于负方向的脉冲（XACTUAL 减少）。

可选择外部 STEP 信号的两个选项：

1. 双边沿 (*GCONF.length_step_pulse* = 0). STEP 输出的每次切换都对应一个脉冲。DIR 至少提前 1 个 CLK 周期有效。启用外部驱动程序上的 dedge 功能以匹配。
2. 单边沿 (*GCONF.length_step_pulse* > 0). STEP 输出上的每个正脉冲对应一个脉冲。脉冲长度由 CLK 周期中的 length_step_pulse 控制。DIR 信号只会在步进脉冲之间变化，并在下一个步进脉冲之前保持与 STEP 脉冲长度相等的最小设置时间。

该功能还可用于向外部逻辑提供步进同步信号，例如 触发测量。

Position Compare Functions [adapt]

[TBD]

PRELIMINARY CONFIDENTIAL

位置比较

周期性位置比较和诊断输出

StallGuard2 负载测量

为了适应不同的电机控制方案，TMC5240 提供两种 StallGuard 无传感器负载检测方案，涵盖了两种基本的斩波器模式。StallGuard2 在 SpreadCycle 操作中工作，而 StallGuard4 针对 StealthChop 操作进行了优化。

StallGuard2 可准确测量电机负载。 它可用于失速检测以及在低于使电机失速的负载下的其他用途，例如 CoolStep 负载自适应电流调节。StallGuard2 测量值在很宽的负载、速度和电流设置范围内呈线性变化。在最大电机负载下，该值变为零或接近零。这对应于线圈的磁场和转子中的磁场之间的 90° 负载角。这也是电机能效最高的运行点。

提示：为了使用 StallGuard2 和 CoolStep，应首先使用 SGT 设置调整 StallGuard2 灵敏度！

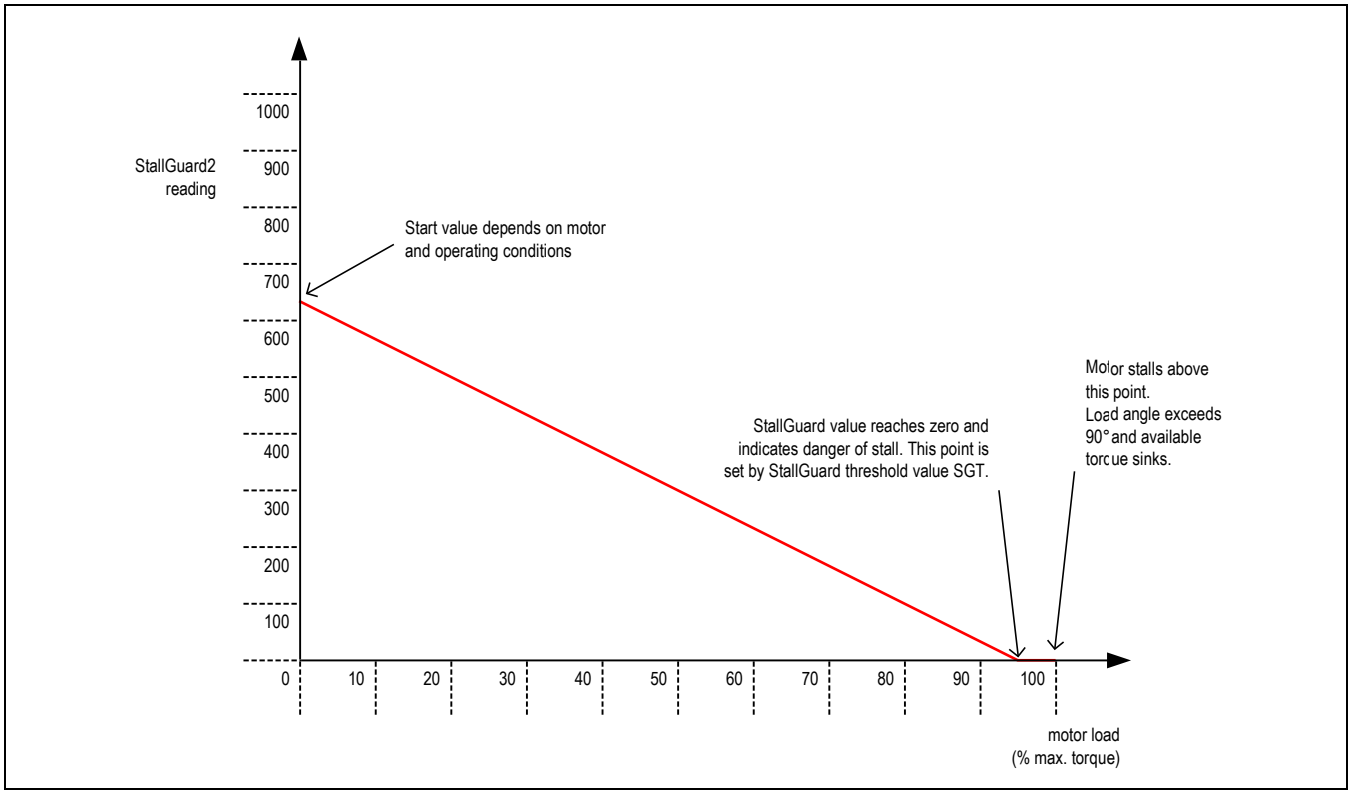


图 27 StallGuard2 功能原理

表 16. StallGuard2 相关参数

参数	描述	设置	备注
SGT	该有符号值控制失速检测的 StallGuard2 阈值水平，并设置最佳读数测量范围。 值越低，灵敏度越高。 零是适用于大多数电机的起始值。 较高的值会降低 StallGuard2 的灵敏度，并需要更大的扭矩来显示失速。	0	indifferent value
		+1… +63	灵敏度较低
		-1… -64	高灵敏度

PRELIMINARY CONFIDENTIAL

表 16. StallGuard2 相关参数（待续）

sfilt	启用 StallGuard2 过滤器以提高测量精度。如果设置，则将测量频率降低到电机的每个电气周期（4 个全步）测量一次。	0	标准模式
		1	过滤模式
状态字	**描述**	**范围**	**备注**
SG_RESULT	这是 StallGuard2 的 结果。 读数越高表明机械负载越小。 较低的读数表示较高的载荷，因此表示较高的载荷角。 调整 SGT 设置，以在电机失速前的最大负载下显示 SG_RESULT 读数大约为 0 到 100。	0…1023	0：最高负载 低值：高负载 高值：低负载

调整 StallGuard2 阈值 SGT

StallGuard2 值 SG_RESULT 受电机特定特性和应用特定负载和速度要求的影响。因此，针对特定电机类型和操作条件调整 StallGuard2 阈值 SGT 的最简单方法是在实际应用中进行交互式调整。

调整 StallGuard SGT 的初始过程

1. 以您的应用程序的正常运行速度运行电机并监控 SG_RESULT。
2. 对电机施加缓慢增加的机械负载。如果电机在 SG_RESULT 达到零之前停止，降低 *SGT*. 如果 SG_RESULT 在电机失步前达到零，则增加 SGT。良好的 SGT 起始值为零。 SGT 是有符号的，所以它可以有负值或正值。
3. 现在启用 sg_stop 并确保电机在失步前安全停止。如果电机在失速发生之前停止，则增加 SGT。通过禁用 sg_stop 或清除 RAMP_STAT 寄存器中的 event_stop_sg（写入清除）来重新启动电机。 不使用 SPI_DRIVER 复制
4. 当SG_RESULT在电机停止前不久，随着负载增加，在0 ~ 100左右时，达到最优设置，并且 SG_RESULT 在没有负载的情况下增加 100 或更多. *在大多数情况下，SGT 可以针对某个运动速度或速度范围进行调整。* 确保设置在一定范围内（例如所需速度的 80% 到 120%）以及极端电机条件（最低和最高适用温度）下可靠工作。

允许自动调节 SGT 的可选过程

SGT 设置背后的基本思想是一个因子，它补偿 StallGuard 测量的电机内部电阻损耗。在静止和非常低的速度，电阻损耗是电机能量平衡的主要因素，因为机械功率是零或接近零。通过这种方式，SGT可以在接近零速度时设置为最佳状态。该算法特别适用于在应用程序中对SGT进行调整，以获得不受环境条件、电机杂散等影响的最佳结果。

1. 以 < 10 RPM 的低速（即每秒几到几整步）和目标工作电流和电源电压运行电机。在这个速度范围内，SG_RESULT 对电机负载的依赖性不大，因为电机不会产生明显的反电动势。因此，机械负载不会对结果产生太大影响。
2. 开启 sfilt 。 现在将 SGT 从 0 开始增加到一个值，SG_RESULT 开始上升。 如果SGT较高，SG_RESULT将上升到最大值. 再次减小到最大值，其中SG_RESULT保持在0.现在，SGT值被尽可能合理地设置，当您看到SG_RESULT以更高的速度增大时，将会有有用的失速检测。

使用此设置的失速检测的上限速度由速度确定，其中电机反电动势接近电源电压，当进一步增加速度时电机电流开始下降。

*当电机失步时，SG_RESULT 变为零，并且可以通过在 SW_MODE 中启用 sg_stop 来对斜坡发生器进行编程，以在失步事件时停止电机。*设置 TCOOLTHRS 以匹配 StallGuard 提供良好结果的最低速度阈值，以便使用 sg_stop。

系统时钟频率会影响SG_RESULT。 对于高性能要求的应用，应使用外部晶体稳定时钟。电源电压也会影响 SG_RESULT，因此调制越严苛，数值越精确

PRELIMINARY CONFIDENTIAL

SG_RESULT 测量具有高分辨率，有几种方法可以提高其准确性，如下面的部分所述。

快速入门

如需快速入门，请参阅第 24 章中的快速配置指南。有关详细程序，请参阅应用笔记 AN002 - StallGuard2 和 CoolStep 的参数化

可调的速度限值 TCOOLTHRS 和 THIGH

由于前面描述的 SGT 调整而选择的 SGT 设置可用于特定的速度范围。在此范围之外，可能无法安全地检测到失速，并且 CoolStep 可能不会给出最佳结果。

在许多应用中，大部分时间都在单个操作点处或附近进行操作，单个设置就足够了。驱动程序提供了一个较低和一个较高的速度阈值来匹配这个。失速检测在确定的操作点之外禁用，例如 在将 TCOOLTHRS 设置为匹配值时，在无传感器归位过程之前的加速阶段。THIGH 可以指定一个速度上限。

在某些应用中，使用少量支持点和线性插值对 SGT 值进行与速度相关的调整会很方便。

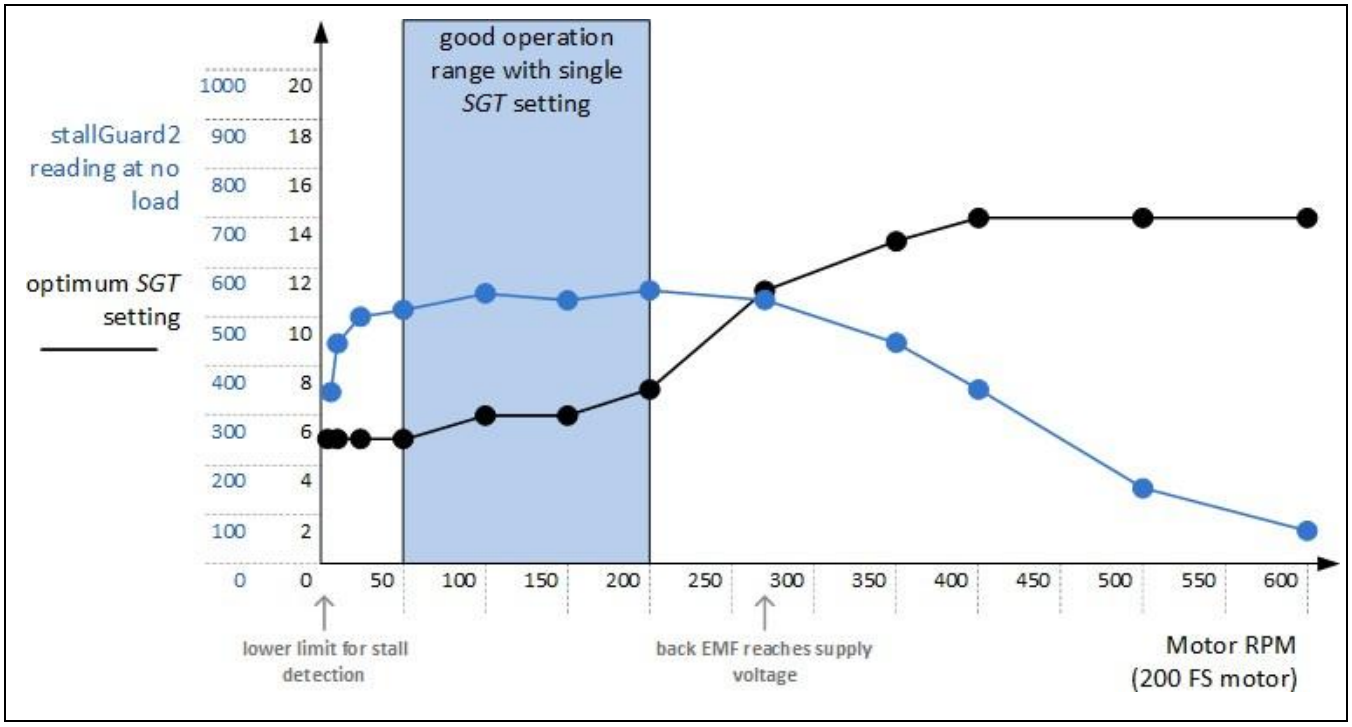


图28。例子:一个例子电机下的最佳SGT设置和StallGuard2读取

具有高转矩纹波和谐振的小型电机

具有高制动扭矩的电机显示出 StallGuard2 测量值 SG_RESULT 的变化随着电机电流的变化而增加，特别是在低电流时。对于这些电机，应检查电流相关性以获得最佳结果。

电机线圈电阻的温度相关性

在宽温度范围内工作的电机可能需要温度校正，因为电机线圈电阻会随着温度升高而增加。这可以作为 SG_RESULT 在温度升高时的线性降低进行校正，因为电机效率降低。

PRELIMINARY CONFIDENTIAL

StallGuard2 测量的准确性和可重复性

在生产环境中，可能希望在应用中为一种电机类型使用固定的 SGT 值。StallGuard2 测量中的大多数单元与单元之间的差异是由电机结构的制造公差造成的。StallGuard2 的测量误差——只要所有其他参数保持稳定——可以低至：

$$\text{StallGuard 测量误差} = +/-\max(1,|SGT|)$$

StallGuard2 更新率和过滤

StallGuard2 测量值 SG_RESULT 会随着电机的每一个完整整步而更新。 这足以安全地检测到失速，因为失速总是意味着丢失四个完整的整步。在实际应用中，尤其是在使用 CoolStep 时，更精确的测量可能比更新每个完整步更重要，因为机械负载永远不会从一个步骤到下一个步骤瞬时变化。对于这些应用程序，sfilt位启用了四个负载测量的过滤功能。当需要高精度测量时，应始终启用过滤器。它补偿了电机结构中的变化，例如由于 A 相与 B 相磁体未对准而造成的。当需要对增加的负载进行快速响应时，应禁用过滤器，并使用 StallGuard 获得无传感器回原点的最佳结果。

检测电机失速

为获得最佳失速检测，请在不使用 StallGuard 过滤 (sfilt=0) 的情况下工作。为了安全地检测电机失速，必须使用特定的 SGT 设置来确定失速阈值。因此，需要确定电机可以在不失速的情况下驱动的最大负载。同时，监控此负载下的 SG_RESULT 值，例如 0 到 100 范围内的某个值。 失速阈值应该是一个安全地在操作范围内的值，以允许参数偏离。 在SGT设置为0或接近0时的响应给出了信号质量的一些概念:在无负载和最大负载时检查SG_RESULT值。 它们应该显示至少100或几百的差异，这应该比补偿大。如果您以某种方式设置 SGT 值，即在最大电机负载时出现读数 0，则可以自动检测到失速以发出让电机停止。在步进导致失步的那一刻，最低读数将可见。失步后，电机会振动并显示更高的 SG_RESULT 读数。

使用 StallGuard 回零点

直线运动的回零需要将电机往硬停止的方向移动。由于 StallGuard 需要一定的速度才能工作（由 TCOOLTHRS 设置），因此请确保起点距离硬停止足够远，以提供加速阶段所需的距离。设置 SGT 和斜坡发生器寄存器后，启动硬停止方向的运动并激活停转功能（在 SW_MODE 中设置 sg_stop）。一旦检测到失速，斜坡发生器停止运动并将 VACTUAL 设置为零，从而停止电机。 DRV_STATUS 中的标志 StallGuard 也指示停止状态。 在设置新的运动参数以防止电机立即重新启动后，StallGuard可以被禁用，或者通过读取RAMP_STAT重新使能电机。 RAMP_STAT 中的 event_stop_sg 标志的读取和清除功能将在 TZEROWAIT 到期后重新启动电机，以防运动参数没有被修改。

StallGuard2 的操作限制

StallGuard2 无法在极高的电机速度下可靠运行：非常低的电机速度（对于许多电机来说，每秒不到一转）产生的反电动势很低，使测量不稳定，并且取决于环境条件（温度等）。上面描述的自动调谐过程将对此进行补偿。其他情况也会导致 SGT 设置过大，测量值 SG_RESULT 对电机负载的响应不佳。非常高的电机速度，无法将完整的正弦电流驱动到电机线圈中，也会导致响应不佳。这些速度通常以电机反电动势达到电源电压为特征。

StallGuard4 负载测量

StallGuard4 是为与 StealthChop 结合使用而开发的。它提供电机负载的准确测量，可用于失速检测、负载估计以及 CoolStep 负载自适应电流调节。StallGuard4 测量值在很宽的负载、速度和电流设置范围内呈线性变化，如图 14.1 所示。当接近最大电机负载时，该值下降到特定于电机的较低值。这对应于线圈的磁场与转子中的磁场之间的 90° 负载角。

PRELIMINARY CONFIDENTIAL

这也是电机能效最高的运行点。

为了使用 StallGuard4，请检查电机在边界条件下的灵敏度。

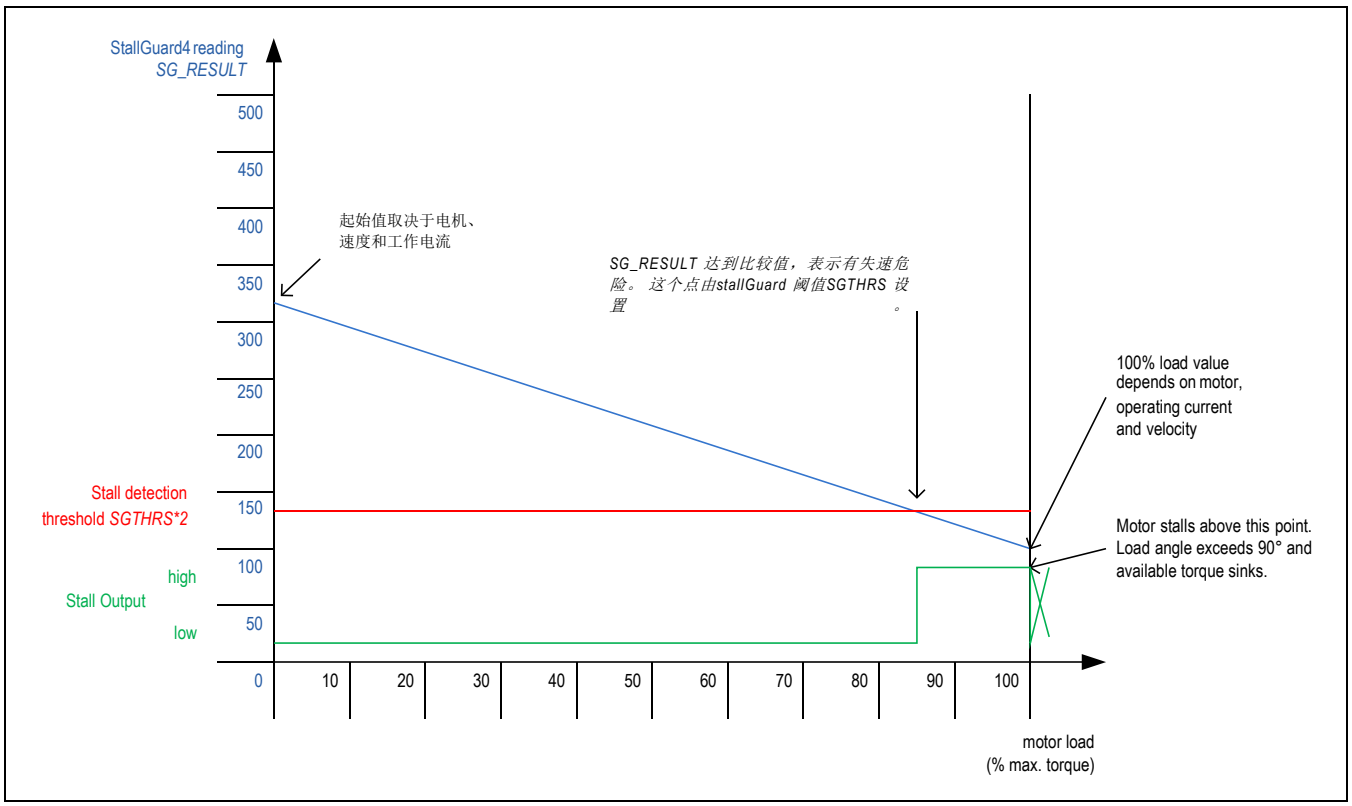


图 29. StallGuard4 操作模式

表 17. StallGuard4 相关参数

参数	描述	设置	备注
SG4THRS	该值控制失速检测的 StallGuard4 阈值水平. 它可以补偿电机的特定特性并控制灵敏度。 值越高，灵敏度越高。 较高的值使 StallGuard4 更灵敏，并且需要更少的扭矩来指示失速。	0… 255	将该值的双倍与 SG4_RESULT 进行比较。 如果 SG4_RESULT 低于此值，则电机停止被激活。
状态字	**描述**	**范围**	**备注**
SG4_RESULT	这是 StallGuard4 结果。 读数越高表明机械负载越小。 较低的读数表示较高的负载，因此表示较高的负载角。该值的生成与启用条件（如实际斩波器模式）和速度阈值（如 VCOOLTHRS）无关。结果是根据 SG4_IND_x 测量值计算得出的，添加一位以获得更高的精度和与 StallGuard2 相似的范围。	0…510	低值：最高负载 高值：低/空载
SG4_IND_3 *SG4_IND_2* *SG4_IND_1* *SG4_IND_0*	单独测量运动A相下降(SG4_IND_0) /上升(SG4_IND_1)的过渡响应。B相下降(SG4_IND_2) /上升(SG4_IND_3)相变。过滤模式下可以使用单个度量值(sg4_filt_en=1)。SG4_IND_0覆盖未过滤模式下的所有情况(sg4_filt_en=0)	0....255	低值：最高负载 高值：低/空载

PRELIMINARY CONFIDENTIAL

Table 17. StallGuard4 related parameters (continued)

sg4_filt_en	0: 未经过滤的操作，SG4_RESULT 数值每4个更整步更新一次 1: 过滤操作，SG4_IND_0...3 可用， SG4_RESULT 给出最后四个 SG4_IND_x 测量值的平均值	0 1	0: 过滤关闭 1: 过滤操作， SG4_IND 值可用
sg_angle_offset	此标志启用 StealthChop 和 SpreadCycle 之间的优化切换， 通过使用 SG4_RESULT 确定 StealthChop 中的相位滞后 并在SpreadCycle中从电压控制切换到电流控制操作时补偿相位跳跃.当切换回 StealthChop 时，相位偏移将被存储并再次减去。	0 1	0: 无角度校正 1: 优化 StealthChop 和 SpreadCycle 之间的切换

StallGuard4 vs. StallGuard2

StallGuard4 针对与 StealthChop 的操作进行了优化，其上一代 StallGuard2 与 SpreadCycle 一起使用。 功能类似：两者都提供一个负载值，低负载时的高值到高负载时的低值。 在 StallGuard2 中调整到数值为0时预示着失步，StallGuard4 使用比较值来触发失速检测，而不是通过应用偏移来移动测量结果。

调节 StallGuard4

StallGuard4 值 SG4_RESULT 受电机特定特性和负载、线圈电流和速度的特定应用需求的影响。因此，针对特定电机类型和操作条件调整 StallGuard4 阈值 SG4_THRS 的最简单方法是在实际应用中进行交互式调整。

调整 StallGuard SG4_THRS 的初始过程

1. 以适合您应用的正常运行速度运行电机并监控 此时SG4_RESULT。
2. 对电机施加缓慢增加的机械负载。 在电机失步之前检查 SG4_RESULT 的最小值。将此值用作 SG4_THRS 的起始值（使用该值的一半）。
3. 现在通过 DIAG 输出监控 StallGuard 输出信号（正确配置，同时设置 TCOOLTHRS 以匹配运行的速度下限）并在相应输出上看到脉冲时停止电机。 确保电机能安全停止， 如果电机在失速发生之前停止，则增加 SG4_THRS。
4. 当安全地检测到失速并在发生失速时在 DIAG 处产生一个脉冲时，即达到最佳设置。大多数情况下，SG4_THRS 可以针对某个运动速度或速度范围进行调整。确保设置在一定范围内（例如所需速度的 80% 到 120%）以及极端电机条件（最低和最高适用温度）下可靠工作.

当SG4_RESULT低于SG4_THRS时，StallGuard会触发DIAG输出一个脉冲。 它仅在 StealthChop 模式下启用，并且当 TCOOLTHRS ≥ TSTEP > TPWMTHRS 时启用。

如果需要，外部运动控制器应通过停止电机来响应这个脉冲.设置 TCOOLTHRS 以匹配 StallGuard 提供良好结果的较低速度阈值。

SG4_RESULT 测量具有高分辨率，有几种方法可以提高其精度，如下面的部分所述。

StallGuard4 更新速率

StallGuard4测量值SG4_RESULT随着电机的每一个完整步骤而更新。这足以安全地检测失速，因为失速总是意味着丢失四个完整的步骤。

StallGuard4提供两种测量选项:

1.) *sg4_filt_en* = 0: 单次测量，在每个整步后更新，对每个整步有效。 这种测量可以对负载变化做出最快的反应，因为 SG4_RESULT 会随着线圈电压的每次零传输而完全更新。 因此，它最适用于有硬障碍物的失速检测。

2.) *sg4_filt_en* = 1: 在这种模式下，会产生四个单独的信号: *SG4_IND_0 在余弦波（线圈 A）下降 0 过渡时*; *SG4_IND_1 在余弦波上升 0 过渡时*; *SG4_IND_2 在正弦波（线圈 B）的下降 0 过渡时*; *SG4_IND_3 在正弦波上升 0 过渡时*. SG4_RESULT 的实际值是所有四次测量的平均值，每整步更新一次。 这样，每个整步对整体结果的影响只有 25%

此模式非常适合检测软障碍物，或在不精确的电机上使用 CoolStep。在过滤模式下，对负载突然增加（电机硬阻挡）的敏感度降低。

检测电机失速

为了安全地检测电机失速，必须使用特定的 SG4_THRS 设置和特定的电机速度或速度范围来确定失速阈值. 此外，电机电流设置有一定的影响，一旦确定了最佳值，就不应修改。因此，需要确定电机可以在不失速的情况下驱动的最大负载。同时，在这个负载下监控 SG4_RESULT。失速阈值应该是在操作限制内的一个安全值，以允许参数偏离。 更精确的评估也可能对 SG4_RESULT 的变化做出反应，而不是与固定阈值进行比较。这将排除某些影响绝对值的效应。

StallGuard4 操作的限制

StallGuard4 在极高的电机速度下无法可靠运行： 非常低的电机速度（对于许多电机来说，每秒不到一转）会产生低反电动势，使测量不稳定，并且取决于环境条件（温度等）。其他情况也会导致测量值 SG4_RESULT 对电机负载的响应不佳。非常高的电机速度，其中没有将完整的正弦电流驱动到电机线圈中，也会导致响应不佳。 这些速度的典型特征是电机反电动势超过电源电压。

CoolStep 操作

CoolStep 是一种基于电机机械负载的步进电机自动智能电流优化，使其更节能。根据实际斩波器模式，CoolStep 会自动在 StealthChop 中使用 StallGuard4 负载测量结果，或在 SpreadCycle 中使用 StallGuard2。但是，必须对其中一个或另一个进行调整。 一次调整不会涵盖所有操作点。

设置 CoolStep

CoolStep 由多个参数控制，但有两个参数对于理解其工作原理至关重要：

表 18. CoolStep 关键参数

Parameter	Description	Range	Comment
SEMIN	设置下限阈值的 4 位无符号整数. 如果 SG_RESULT 低于此阈值，CoolStep 会增加两个线圈的电流。4 位 SEMIN 值缩放 32 以覆盖 10 位 SG_RESULT 值范围的下半部分。（此参数的名称来源于 smartEnergy，它是 CoolStep 的早期名称。）	0	disable CoolStep
		1…15	threshold is *SEMIN**32
SEMAX	控制上限阈值的 4 位无符号整数。如果 SG_RESULT 的值等于或高于此阈值，CoolStep 会降低两个线圈的电流。上限阈值为 (SEMIN + SEMAX + 1)*32。	0…15	threshold is (*SEMIN*+*SEMAX*+1)*32

下图显示了 CoolStep 的操作区域：

- 黑线代表 SG_RESULT 测量值。
- 蓝线表示施加到电机的机械负载。
- 红线代表进入电机线圈的电流。

当负载增加时，SG_RESULT 低于 SEMIN，CoolStep 增加电流。当负载减小时，SG_RESULT 上升到 (SEMIN + SEMAX + 1) * 32 以上，电流减小。

PRELIMINARY CONFIDENTIAL

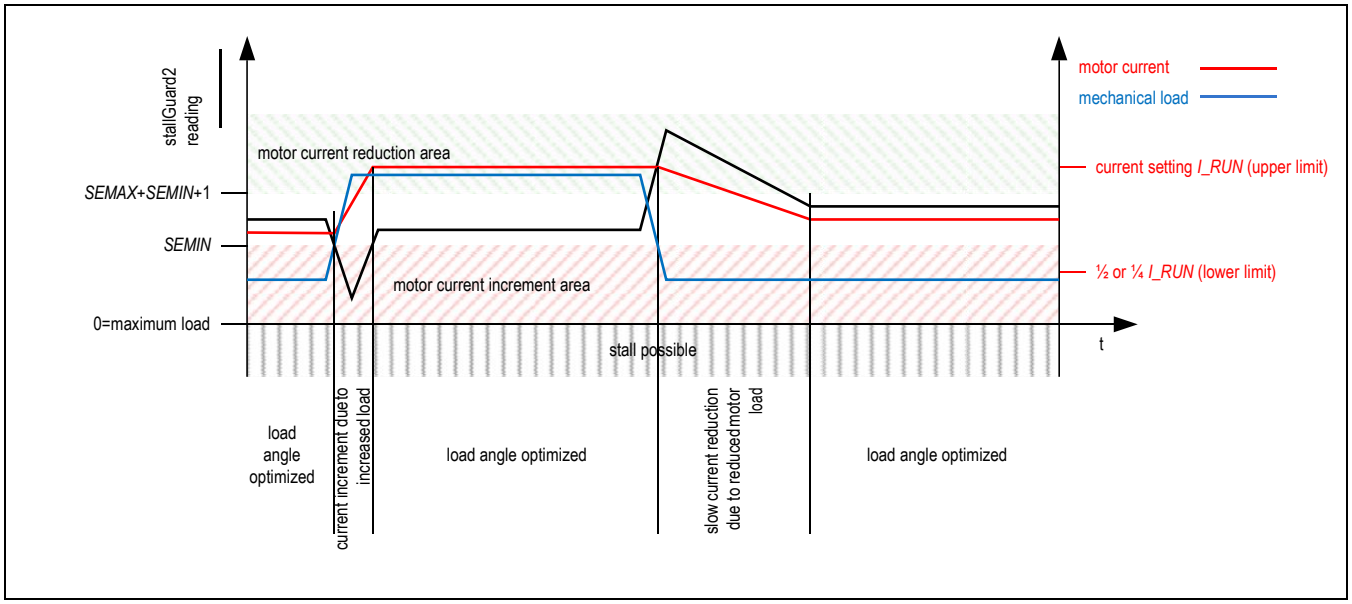


图 30. CoolStep 使电机电流适应负载

表 19. CoolStep 附加参数和状态信息

Parameter	Description	Range	Comment
SEUP	设置电流增量步长。 对于每个低于阈值下限的测量 StallGuard2 值，电流都会增加。	0…3	step width is 1, 2, 4, 8
SEDN	将 StallGuard2 读数的数量设置为电机电流每次减小所需的上限阈值。	0…3	每递减的 StallGuard2 测量次数： 32, 8, 2, 1
SEIMIN	通过缩放 IRUN 电流设置来设置 CoolStep 操作的电机电流下限。	0	0: 1/2 of IRUN
		1	1: 1/4 of IRUN
TCOOLTHRS	开启 CoolStep 功能的最低速度阈值。 低于此速度 CoolStep 将被禁用. 适应 StallGuard2 提供稳定结果的速度范围下限。 *提示:* 通过设置为与 VMAX 相同，可以调整为在加速和减速阶段禁用 CoolStep。	1… 2^20-1	通过将阈值与 TSTEP 进行比较来确定较低的 CoolStep 速度
THIGH	CoolStep 的上限速度阈值。 超过此速度 CoolStep 将被禁用。 适应 StallGuard2 提供稳定结果的速度范围。	1… 2^20-1	还控制其他功能，例如切换到整步模式。
状态字	描述	范围	**备注**
CS_ACTUAL	此状态值提供由 CoolStep 控制的实际电机电流比例。 该值上升到 IRUN 值并下降到由 SEIMIN 指定的 IRUN 部分。	0…31	1/32, 2/32, … 32/32

调整 CoolStep

在结合 SpreadCycle 调整 CoolStep 之前，首先调整 StallGuard2 阈值水平 SGT，这会影响负载测量值 SG_RESULT 的范围。 CoolStep 使用 SG_RESULT 在 +90° 的最佳负载角附近运行电机。 CoolStep 与 StealthChop 结合使用 SG4_RESULT。 在这种模式下，通过 SEMIN 进行调节。

PRELIMINARY CONFIDENTIAL

电流增量速度在 SEUP 中设定，电流减量速度在 SEDN 中设定。 它们可以单独调优，因为它们由不同的事件触发，而这些事件可能需要不同的响应。 这些参数的编码允许线圈电流增加比减少快得多，因为越过下限是一个更严重的事件，可能需要更快的响应。如果响应太慢，电机可能会失步。 相比之下，对超过上限阈值的缓慢响应并没有比错过节省电力的机会更严重的风险。

CoolStep 在电流设置参数 IRUN 和 seimin 位控制的限制之间运行。

响应时间

为了快速响应不断增加的电机负载，请使用大的电流增量步 SEUP。如果电机负载变化缓慢，可以使用较低的电流增量步长来避免电机振荡。如果启用了由 sfilt 控制的滤波器，则测量速率和调节速度会降低四倍。

建议: 最常见和最有益的用途是将CoolStep用于典型系统目标操作速度的控制，并据此设置速度阈值。由于加速和减速通常应该很快，它们将需要满电机电流，而由于它们的持续时间短，它们对整体功耗的影响很小。

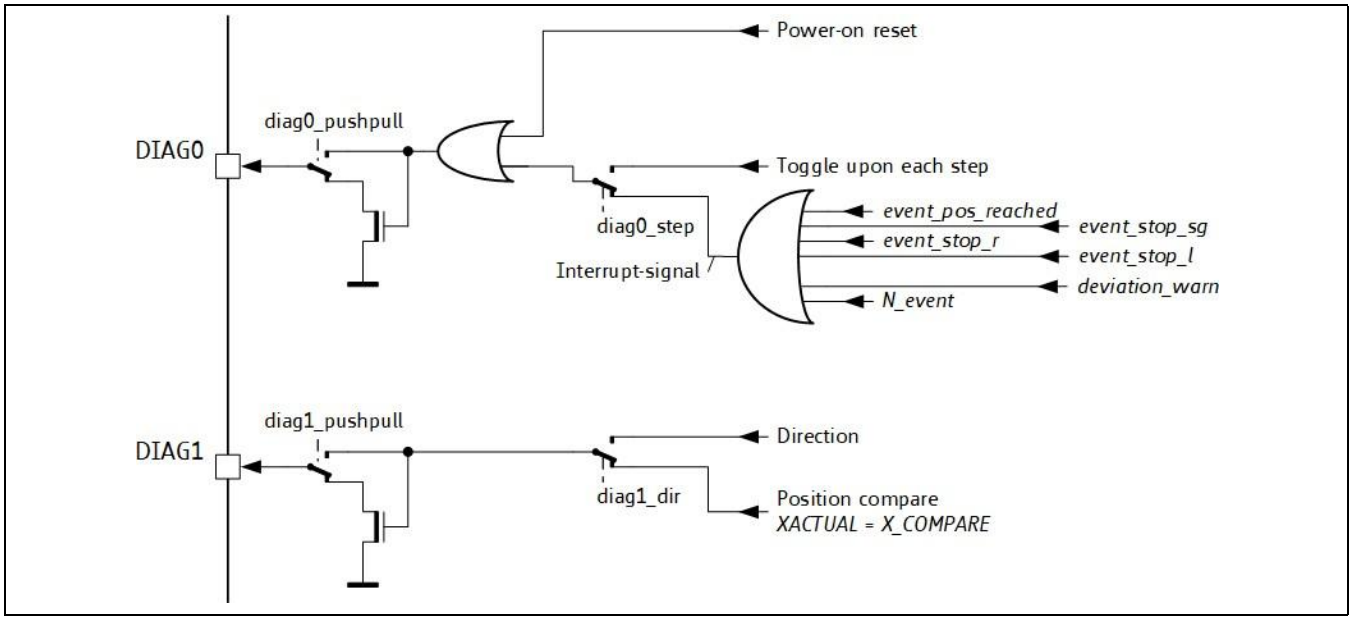
低速和待机操作

由于 CoolStep 无法在静止和非常低的转速下测量电机负载，因此斜坡发生器中提供了较低的速度阈值。 它应该被设置为一个应用程序特定的默认值。低于此阈值，通过 IRUN 或 IHOLD 进行的正常电流设置有效。VHIGH 设置提供了一个上限阈值。 作为 StallGuard2 调整过程的结果，这两个阈值都可以设置。

诊断输出

DIAG 输出提供一个位置比较信号，以允许精确触发外部逻辑, 和一个中断信号，以触发软件到运动斜坡内的某些条件. 可以选择开漏（低电平有效）输出信号（默认），或高电平有效推挽输出信号。使用开漏输出时，可以对多个驱动器输出信号进行“或”运算。需要一个 4.7kΩ 至 100kΩ 范围内的外部上拉电阻。DIAG0 在复位条件下也变为低电平。但是，在此配置中，不能通过监视 DIAG0 来确定复位条件的结束，因为 event_pos_reached 标志在复位时也变为活动状态，因此在复位条件后引脚保持为低电平。 为了安全地确定复位条件，通过 SPI 监视复位标志或读出任何寄存器以确认芯片已上电。

PRELIMINARY CONFIDENTIAL



DcStep

DcStep 是步进电机的一种自动换向模式。 只要步进电机能够应对负载，它就允许驱动器按照斜坡发生器的指令以目标速度运行。万一电机过载，它会减慢到一定速度，此时电机仍可以驱动负载。 这样，步进电机永远不会失步，并且可以尽可能快地驱动重负载。它在低速时可以提供更高的扭矩，加上飞轮质量产生的动态扭矩，可以补偿机械扭矩峰值。一旦电机完全堵转，失步标志信号会被触发。

DcStep 设置

在常规应用中，操作区域受限于最大应用速度下所需的最大扭矩。需要高达 50% 的扭矩安全余量，以补偿不可预见的负载峰值、由于共振和机械部件老化导致的扭矩损失。 DcStep 允许最大化使用电机的扭矩。使用电机和应用飞轮质量可以克服更高的短时动态负载，而不会出现电机失速的危险。使用 DcStep，标称应用负载可以扩展到更高的扭矩，仅受保持扭矩区域附近的安全裕度限制（这是电机可以提供的最高扭矩）。此外，最大应用速度可以提高到实际可达到的电机速度。

PRELIMINARY CONFIDENTIAL

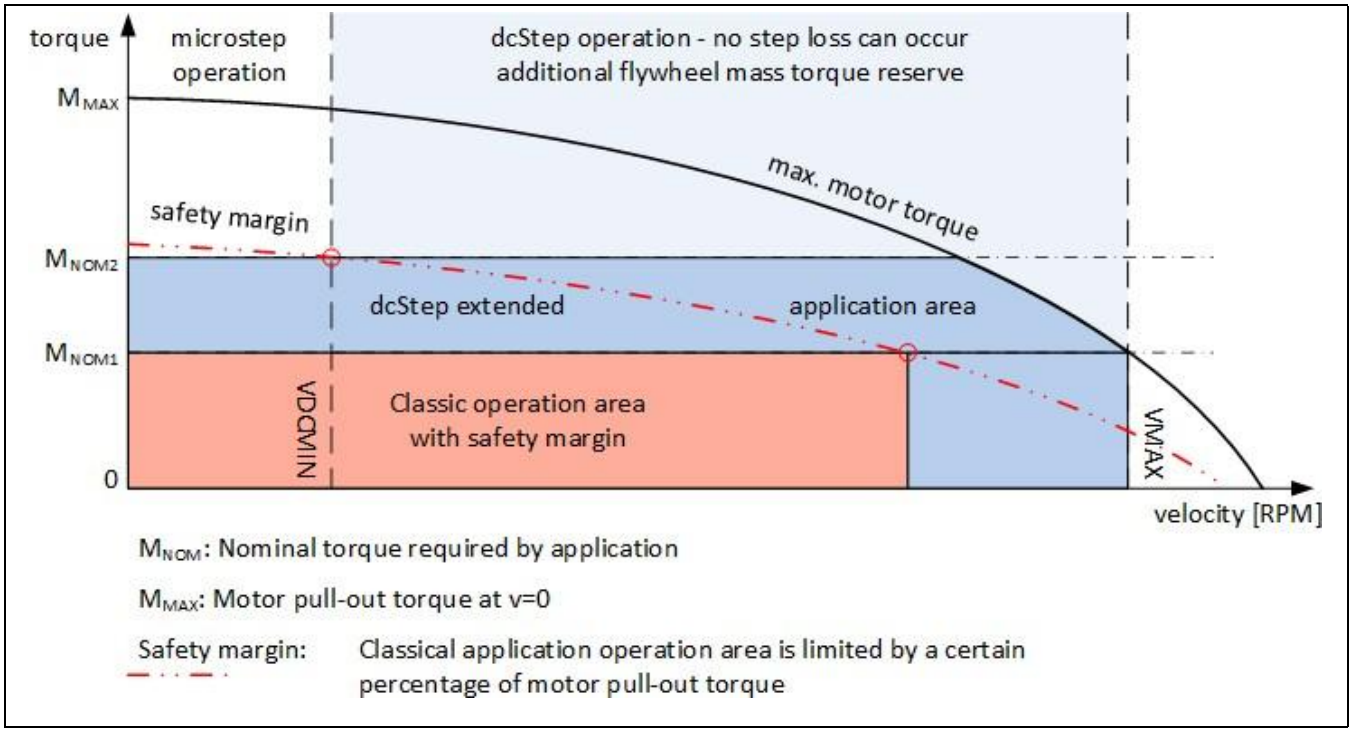


图 31. DcStep 扩展应用程序操作区域

快速入门
如需快速入门，请参阅第 24 章中的快速配置指南。
有关详细配置过程，请参见应用笔记 AN003 - DcStep

DcStep 与运动控制器的集成

DcStep 只需要几个设置. 它直接将电机运动反馈给斜坡发生器，因此它可以无缝集成到运动斜坡中，即使电机相对于目标速度变得过载。DcStep 在整步模式下以斜坡发生器目标速度 VACTUAL 运行电机，或者如果电机过载，则以降低的速度运行电机。它需要设置最小运行速度VDCMIN。 VDCMIN 应设置为 DcStep 能够可靠检测电机运行的最低运行速度。除非电机制动到低于 VDCMIN 的速度，否则电机永远不会失步。如果速度低于该值，一旦释放负载，电机将重新启动，除非启用失速检测（设置 sg_stop）。 失速检测由StallGuard2负责。

PRELIMINARY CONFIDENTIAL

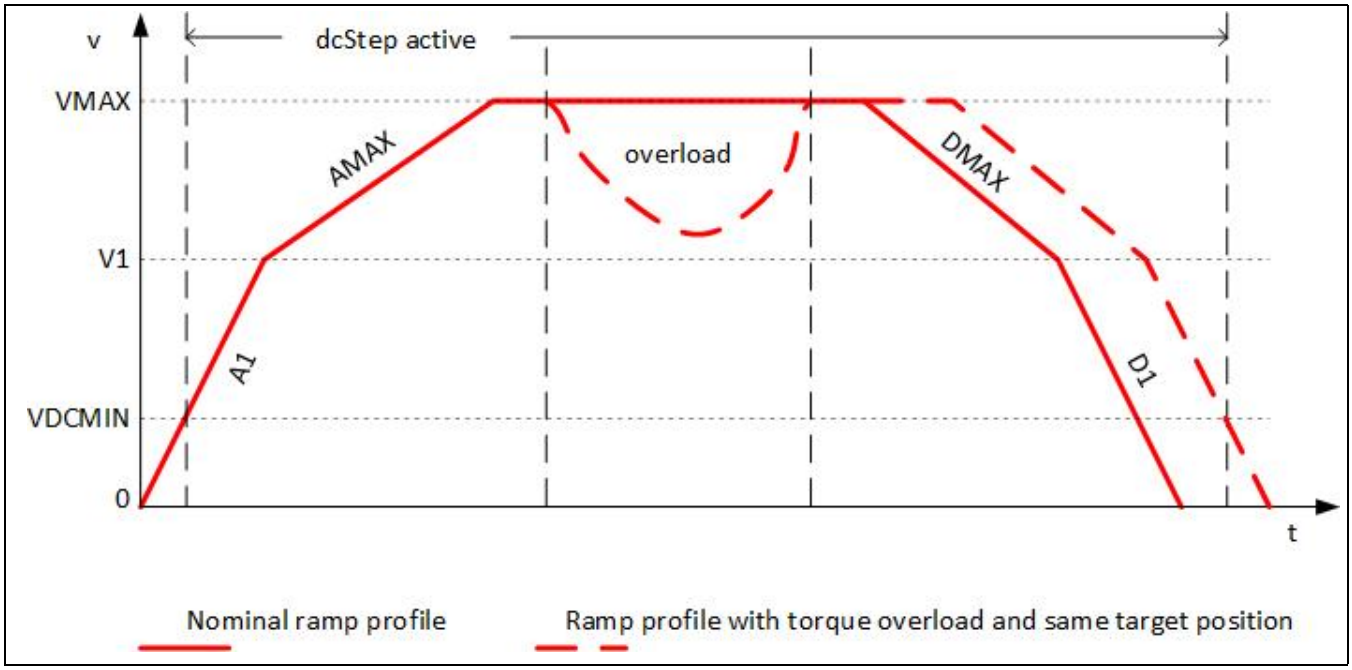


图 32. 受过载情况影响的速度曲线

注意

DcStep 要求正弦波的相位极性在 MSCNT 范围 768 到 255 内为正，在 256 到 767 内为负。 余弦极性从 0 到 511 必须为正，从 512 到 1023 必须为负。 相移 1 会干扰 DcStep 操作。 因此，建议使用默认波形。 有关使用默认表的初始化，请参阅正弦波查找表章节。

DcStep 模式下的失速检测

尽管 DcStep 能够在过载时使电机减速，但它无法避免在每种操作情况下都出现的失速。一旦电机被阻塞，或者它减速到低于电机相关的最小速度，在这种情况下，电机操作不再安全地被检测到，电机可能会停转并丢步。为了安全检测失步并避免电机重新启动，可以启用失速停止（设置标志 sg_stop）。 在这种情况下，一旦电机停止，VACTUAL 就会被设置为零。在读取 RAMP_STAT 状态标志之前，它一直保持停止状态。 标志 event_stop_sg 显示了主动停止条件。在 DcStep 操作期间还可以使用 StallGuard2 负载值。 值的范围被限制在 0 到 255，在某些情况下最多可以读出 511。为了启用 StallGuard，还需要将 TCOOLTHRS 设置为与略高于 VDCMIN 或最高 VMAX 的速度相对应。

当飞轮负载与电机轴松散耦合时，这种模式下的失速检测可能会由于共振而误触发。

参数	描述	范围	备注
vhighfs & vhighchm	需要为 DcStep 操作设置 CHOPCONF 中的这些斩波器配置标志。 一旦超过 VDCMIN，斩波器就会切换到整步。	0 / 1	设置为 1 用于 DcStep
TOFF	DcStep 通常受益于 CHOPCONF 中增加的关断时间值。 应该首选设置 >2。	2… 15	设置 8…15 与设置 8 用于 DcStep 操作没有任何区别。

PRELIMINARY CONFIDENTIAL

VDCMIN	这是使用内部斜坡发生器时 DcStep 操作的速度下限阈值。 低于此阈值，电机以正常微步模式运行。在 DcStep 操作中，电机以最小 VDCMIN 运行，即使它完全被阻塞。 与 DC_TIME 设置一起调整。 激活 StealthChop 也会禁用 DcStep。	0… 2^22	0：禁用 DcStep 设置为 DcStep 操作的速度下限.
DC_TIME	此设置控制 DcStep 负载测量的参考脉冲宽度。 它必须针对具有最大电机扭矩的稳健运行进行优化。较高的值允许较高的扭矩和较高的速度，较低的值允许运行到由 VDCMIN 设置的较低速度。 检查标称工作条件下的最佳设置，并在极端工作条件下重新检查（例如最低工作电源电压、最高电机温度和最高电源电压、最低电机温度）。	0… 1023	设置的下限为： tBLANK（由 TBL 定义），时钟周期 + n n 在 1 到 100 的范围内（对于典型的电机）
DC_SG	此设置控制 DcStep 模式下的失速检测。 提高灵敏度。 失速可以作为一个错误的条件，为电机发出一个硬停止. 启用sg_stop标志，在失速事件时停止电机。这样，一旦失速电机就会停止运行。	0… 255	设置略高于 DC_TIME / 16

在 DcStep 操作中测量实际电机速度

如果电机由于机械负载而变得低于目标速度，DcStep 能够降低电机速度。 *VACTUAL 显示斜坡发生器目标速度。*它不受 DcStep 的影响。可以根据位置计数器 XACTUAL 测量 DcStep 速度。

因此，以已知的时间差对位置计数器进行两次快速截取：

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

例子：

在 16.0 MHz 时钟频率下，0.954 秒的测量间隔将直接产生速度值，9.54 毫秒的测量间隔将产生 1/100 的实际 DcStep 速度。

为了尽可能精确地掌握时间间隔，每次从 IC 开始或结束 XACTUAL 的传输时，都对计时器进行截取。 用于 SPI 传输的 NCS 的上升沿提供了最准确的时间参考。

正弦波查询表

TMC5240 提供用于存储微步电流波的可编程查询表。默认情况下，该表预编程为正弦波，这是大多数步进电机的良好起点。将表重新编程为电机特定的波形可以显着改进微步，尤其是在使用低成本电机时。使用好处是：

微步 – 使用低成本电机得到极大改进

电机 – 运行平稳和安静

力矩 – 减少机械共振的产生，提高扭矩

低频电机噪音 - 通过调整正弦和余弦波偏移来降低由于电机的实际制造公差引起的噪音

微步表

为了最大限度地减少所需的内存和要编程的数据量，只存储了四分之一的波形. 内部微步表映射从 0° 到 90° 的微步波。它对称地延伸到 360°。当读取表10位微步计数器MSCNT地址完全扩展波表。该表以增量方式存储，每个条目使用一位。因此只需要 256 位（ofs00 到 ofs255）来存储四分之一波。 这些位映射到 8 个 32 位寄存器。当在表格中前进一微步时，每个 ofs 位控制倾斜 Wx 或 Wx+1 的递增。当 Wx 为 0 时，表中实际微步位置的 1 位表示前进到下一微步时“加一”。

PRELIMINARY CONFIDENTIAL

由于波的倾角可能高于 1，因此可以使用四分之一波内多达四个灵活的可编程段将基本倾角 Wx 编程为 -1、0、1 或 2。这样，即使是负的倾向也可以实现。 四个倾斜段由位置寄存器 X1 到 X3 控制。倾斜段 0 从微步位置 0 到 X1-1，其基倾角由 W0 控制，第 1 段从 X1 到 X2-1，其基倾角由 W1 控制，以此类推。

修改波时，必须注意确保当四分之一波扩展为全波时平滑且对称的零过渡。 波的最大摆动应调整到 -248 到 248 的范围内，以提供最佳分辨率，同时为基于滞后的斩波器留出空间以添加偏移。

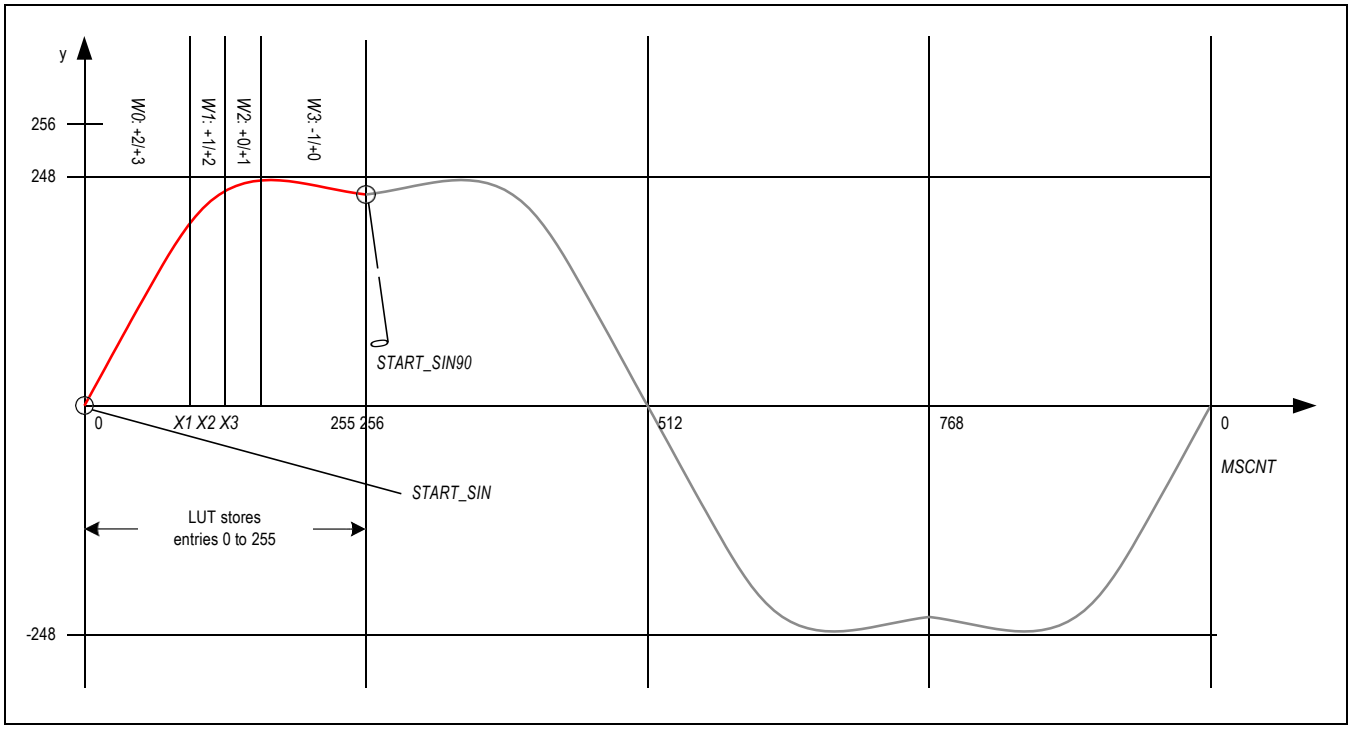


图 33. LUT 编程示例

当微步定序器在表中前进时，它会计算每个微步的电机线圈的实际电流值，并将其存储到寄存器 CUR_A 和 CUR_B。 然而，增量编码需要绝对初始化，尤其是当微步表被修改时。 因此，只要 MSCNT 通过零，就会初始化 CUR_A 和 CUR_B。

将相移与电机匹配：

两个寄存器控制表格的起始值。

- 由于零处的起始值不一定是 0（可能是 1 或 2），因此可以将其编程到起始点寄存器 START_SIN 中。
- 同理，需要将第二个电机线圈的第二个波的开始存储在 START_SIN90 中。该寄存器存储两相电机 90° 相移的结果表条目。为了适应电机容差，相移可以从 90°（256 微步）修改为 45° 到 135° 之间的任意值，方法是在 -127 到 +127 范围内添加一个微步偏移（寄存器 OFFSET_SIN90）。 电机公差将需要适度调整，最多可以调整到 10 步。 使用 StallGuard4 单个值 SG4_IND 并修整偏移量，可以找到所需的校正偏移量，直到两个线圈给出对称的结果。

PRELIMINARY CONFIDENTIAL

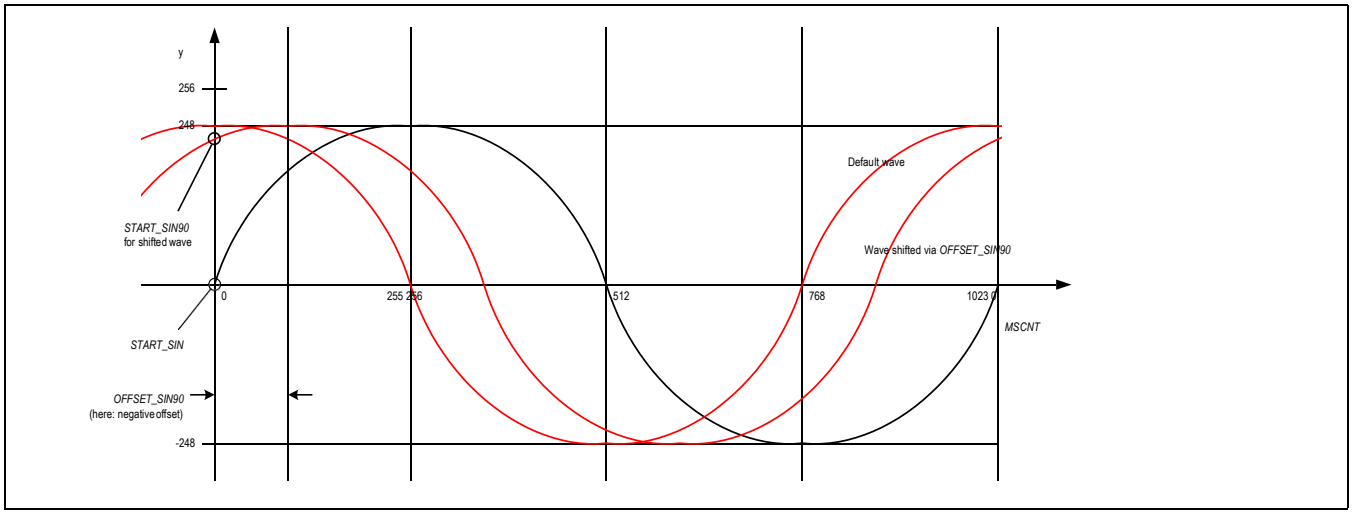


图 34. 通过 OFFSET_SIN90 移动余弦波

默认表是一个很好的设置基础。 这是一个复位默认微步表的初始化示例：：

MSLUT[0]= %10101010101010101011010101010100 =0xAAAAB554
MSLUT[1]= %01001010100101010101010010101010 = 0x4A9554AA
MSLUT[2]= %00100100010010010010100100101001 = 0x24492929
MSLUT[3]= %00010000000100000100001000100010 = 0x10104222
MSLUT[4]= %11111011111111111111111111111111 = 0xFBFFFFFF
MSLUT[5]= %10110101101110110111011101111101 = 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

为了优化电机相移，在 StealthChop 中以中等速度运行电机并设置 sg4_filt_en = 1。调整相位偏移以匹配 A 相 (SG4_IND_0+SG4_IND_1) 到 B 相 (SG4_IND_2+SG4_IND_3) 的 StallGuard 4 结果。

如果 A 相值 > B 相值，则递增 OFFSET_SIN90，否则递减。 重复直到找到最好的匹配。请务必为 START_SIN90 输入正确的值。 对于 -10 到 +9 的偏移，使用 START_SIN90=247； 高达 -17 或 +17 使用 START_SIN90=246。 START_SIN 始终为 0。

ABN 增量编码器接口

TMC5240 配备用于 ABN 编码器的增量编码器接口。 编码器通过数字增量正交信号（通常命名为 A 和 B）和索引信号（通常命名为 N 表示零，Z 表示零，或 I 表示索引）给出位置。

PRELIMINARY CONFIDENTIAL

N 信号

N 信号可用于清除位置计数器或触发捕捉. 要持续监控 N 通道并触发编码器位置清除或位置锁存，其中检测到 N 通道事件，设置标志 clr_cont。或者，可以仅对下一个编码器 N 通道事件做出反应，并在第一个 N 信号事件（标志 clr_once）之后自动禁用编码器位置的清除或锁存。这可能是需要的，因为编码器每转一圈就给出一次这个信号。

一些编码器需要通过特定的 A 和 B 极性配置来验证 N 信号。这可以通过 ENCMODE 寄存器中的 pol_A 和 pol_B 标志来控制。例如，当 pol_A 和 pol_B 都设置时，仅在 A 和 B 通道的高极性期间才接受活动 N 事件。
为了使用下一个活动的 N 事件清除编码器位置 ENC_POS，请设置 clr_enc_x = 1 和 clr_once = 1 或 clr_cont= 1。

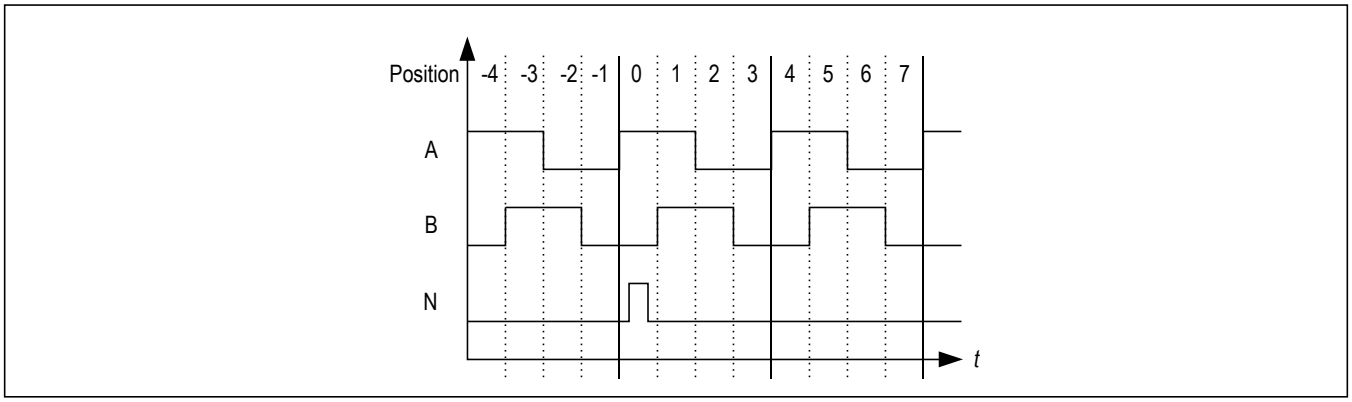


图 35. 增量编码器的 ABN 信号示意图

编码器常量ENC_CONST

在增量编码器的正交信号 AB 的每次极性改变时，编码器常数 ENC_CONST 被添加到编码器计数器或从编码器计数器中减去。编码器常数 ENC_CONST 代表一个有符号的定点数 (16.16)，以促进电机和编码器之间的通用适配。在十进制模式下，低 16 位表示 0 到 9999 之间的数字。 对于配备增量编码器的步进电机，固定数字表示可以非常适宜地进行参数化。此外，可以很容易地考虑机械传动装置。 取反 ENC_CONST 的符号允许反转计数方向以匹配电机和编码器方向。

例子:

- 编码器系数 1.0：ENC_CONST = 0x0001.0x0000 = FACTOR.FRACTION
- 编码器系数 -1.0：ENC_CONST = 0xFFFF.0x0000。 这是 0x00010000 的二进制补码。 它等于 (2^16-(FACTOR+1)).(2^16-FRACTION)
- 十进制模式编码器因子 25.6：00025.6000 = 0x0019.0x1770 = FACTOR.DECIMALS（DECIMALS=小数的前 4 位）
- 十进制模式编码器系数 -25.6：(2^16-(25+1)).(10000-6000) = (2^16 – 26).(4000) = 0xFFE6.0x0FA0。
  - 使用以下等式计算负编码器常数：(2^16-(FACTOR+1)).(10000-DECIMALS)

编码器计数器 X_ENC

编码器计数器 X_ENC 保存当前编码器位置以供读取。关于处理信号 A、B 和 N 的不同模式考虑了使用不同类型的编码器发现的低有效和高有效信号。 更多细节请参考 6.4 节中的寄存器映射。

寄存器 ENC_STATUS

寄存器 ENC_STATUS 保存有关 N 通道信号上的编码器清除事件的状态。 寄存器 ENC_LATCH 存储 N 信号事件上的实际编码器位置。

PRELIMINARY CONFIDENTIAL

检查编码器锁存事件

选项 1：检查 ENC_LATCH 是否有更改。 它从 0 开始，并在第一次开始运动后显示发生 N 事件的编码器计数。 对于连续旋转，它将显示增加/减少的值，因此总是会发生变化。

选项 2：检查中断输出是否有效，并且仅在有效中断输出之后读取标志。

请不要将 ENC_STATUS 事件标志用于主动、高频轮询，因为在并行读取事件和编码器 N 事件的情况下，该标志将同时被清除，并且会丢失。

设置编码器以匹配电机分辨率

电机参数的编码器示例设置：USC=256 µsteps, 200 fullstep motor

Factor = FSC*USC / encoder resolution

表 20. 具有 256 微步的 200 全步电机的编码器示例设置

编码器分辨率	所需的编码器系数	备注
200	256	
360	142.2222 = 9320675.5555 / 2^16 = 1422222.2222 / 10000	不可能完全匹配！
500	102.4 = 6710886.4 / 2^16 = 1024000 / 10000	与十进制设置完全匹配
1000	51.2	与十进制设置完全匹配
1024	50	
4000	12.8	与十进制设置完全匹配
4096	12.5	
16384	3.125	

例子:

编码器常数寄存器应在十进制模式下编程为 51.2。 因此，设置

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

复位、禁用/停止和掉电

紧急停止

驱动器提供了一个低电平使能引脚 DRV_ENN 以安全地关闭所有功率 MOSFET。这允许使电机自由转动。此外，无论何时需要不与软件耦合的紧急停止，它都是安全的硬件的急停功能。某些应用可能需要将驱动器置于具有主动保持电流或被动制动模式的状态。 这可以通过将引脚 ENCA 编程为步进禁用功能来实现。 设置 GCONF 标志 stop_enable 以激活此选项。每当 ENCA 被拉高并且只要它保持高电平，电机就会突然停止并进入断电状态，如通过 IHOLD、IHOLD_DELAY 和 StealthChop 停止选项配置。

外部复位和休眠模式

复位和睡眠模式由 SLEEPN 引脚控制。

SLEEPN 上持续时间 >30µs 的短脉冲会导致芯片复位（在诊断输出中也可见）。

<30µs 的极短脉冲被滤除，不会对操作产生影响。

如果 SLEEPN 保持在 GND，则 IC 进入低功耗待机状态（睡眠模式）。 所有内部电源都关闭。

在这两种情况下，复位和待机所有内部寄存器值和配置都被清除并设置为其默认值，并且功率桥关闭。

PRELIMINARY CONFIDENTIAL

在上电或离开睡眠模式和复位条件后，需要重新配置寄存器。

在重新配置 IC 寄存器时，建议仍然使用 DRV_ENN 禁用功率桥输出。请勿在电机高转速时使用，因为从电机反馈的能量可能会损坏芯片！ 如果不使用请连接到 VS 或 VCC_IO（这是一个高压引脚）。

保护和驱动诊断

TMC5240 驱动器提供一整套诊断和保护功能，例如对 GND 短路保护和欠压检测。开路负载条件的检测允许测试电机线圈连接是否被中断。 有关详细信息，请参见 DRV_STATUS 寄存器表。

除了状态标志外，TMC5240 还允许测量和读取芯片温度以及反馈电机相绕组温度。

为了提高系统可靠性和整体电路保护，TMC5240 包含一个过压比较器和一个触发输出 OV，以控制外部开关以防止电源电压过度升高。

过流保护

过流保护 (OCP) 可保护设备免受电源轨（电源电压和接地）和输出（OUT1A、OUT2A、OUT1B、OUT2B）之间的短路。

OCP 阈值取决于所选的满量程电流范围/请参阅 EC 表了解相应的阈值。

使用 DRV_CONF 寄存器中的 CURRENT_RANGE 参数选择满量程范围。

如果输出电流大于OCP阈值的时间大于去噪时间，则检测到OCP事件。

当检测到 OCP 事件时，则立即关闭 H 桥。

在设置故障标志（DRV_STATUS 寄存器中的 s2ga、s2gb、s2vsa、s2vsb）之前，短路保护会尝试 3 次，并且桥接器会持续禁用。

该设备仍处于活动状态并允许读取配置和状态。

要重新启用功率桥，必须循环DRV_ENN引脚。

另一个选项是在CHOPCONF中关闭TOFF=0的功率桥，然后重新启用TOFF>0的功率桥。

热保护和关断

TMC5240 具有内部热保护功能。

如果晶圆温度超过 165°C（典型值），则会引发故障指示和故障标志（DRV_STATUS 中的 ot），并且驱动器处于三态，直到结温降至 ca. 145°C（典型值）。之后，驱动程序被重新启用。

此外，TMC5240 支持基于 ADC 的可配置热预警级别。这可以使用参数 OVERTEMPPREWARNING_VTH 在寄存器 OTW_OV_VTH 中进行配置。ADC 感测芯片的平均温度，而驱动器级可能处于更高的温度。这只是为了指定 TMC5240 可以进入热关断状态，并且即使设置为低温，也可能不会发出预警。热量主要由电机驱动级产生，并且在电压升高时由内部电压调节器产生。 当启用对GND短路保护时，避免了驱动mosfet过热的关键情况。在许多应用中，超温预警将指示异常运行情况，可用于启动用户预警或降低电机电流等功率措施。热关断只是一种应急措施，应通过设计防止温度上升到关断水平。

温度测量

TMC5240 提供测量内部芯片温度和电机温度的功能。

这些诊断功能有助于监控芯片或 PCB 温度以及电机温度随时间的变化，以提高系统稳健性或收集额外信息以进行预测性维护.

PRELIMINARY CONFIDENTIAL

芯片温度测量

除了过热预警和过热标志外，芯片温度本身可以使用 ADC_TEMP 寄存器中的 ADC_TEMP 参数来确定。

可以使用以下公式计算以摄氏度为单位的最终温度：

ADC_TEMP = 7.7 * TEMP + 2038

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

电机温度测量

PWM_SCALE 寄存器显示 StealthChop 操作中的实际占空比。对于给定的电机电流，占空比取决于电机的相电阻。

由于相电阻与温度有关，因此 PWM_SCALE 可用于估算实际电机温度并监控电机温度随

时间的变化。这种测量最好在电机停止或缓慢移动期间进行。 通常情况下，电机温度不

会迅速变化。

过压保护和引脚OV

步进电机应用会产生明显的过电压，尤其是当电机从高速快速减速或电机停转时。

该电压由驱动器输出级反馈到电源轨。

对于典型的 NEMA17 或更大的电机，以及具有足够飞轮质量的小型电机，反馈的能量可能很大，因此电力电容器和电路消耗不足以将电源保持在其限制范围内。

为了保护驱动器和连接的电路，TMC5240 具有过压检测和保护机制。

OV 输出允许连接一个带有功率电阻（制动电阻）的 NPN 或 MOSFET，以将多余的能量转储到电阻中。

晶体管将以大约 3-4 kHz 的频率斩波（取决于时钟频率），以将电源保持在限制范围内。

电源电压由内部 ADC 持续监控。

可以使用参数 OVERVOLTAGE_VTH 在寄存器 OTW_OV_VTH 中配置给定应用的电源电压上限.

可以通过寄存器 ADC_VSUPPLY_AIN 作为参数 ADC_VSUPPLY 读取电源电压的实际 ADC 值。

OV 输出引脚显示过压监控器的实际状态。

只要 ADC_VSUPPLY 变得大于或等于 OVERVOLTAGE_VTH，OV 输出引脚就会变为三态/'Z'。

OV 输出引脚为开漏引脚。 下图显示了一个示例制动斩波器电路。

PRELIMINARY CONFIDENTIAL

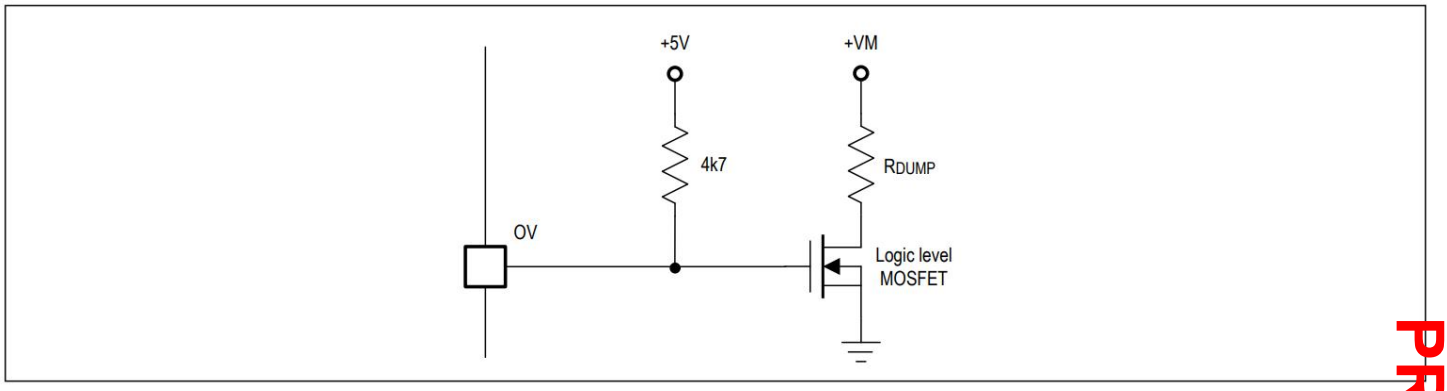


图 36. 制动斩波电路示例

对地短路保护

TMC5240 功率级通过额外测量流过高侧 MOSFET 的电流来防止发生短路情况。 这一点很重要，因为大多数短路情况是由电机电缆绝缘缺陷引起的，例如当接触到连接到系统地的导电部分时。短路检测可防止虚假触发，例如 通过 ESD 放电，在关闭电机之前重试 3 次。

一旦安全地检测到短路情况，相应的驱动桥就会关闭，并且 s2ga 或 s2gb 标志会被触发。为了重新启动电机，用户必须通过禁用和重新启用驱动器进行干预。应该注意的是，接地短路保护不能保护系统和功率级以应对所有可能的短路事件，因为短路事件是相当不确定的，并且可能涉及复杂的外部组件网络。因此，应基本避免短路。

开路诊断

电缆中断是系统故障的常见原因，例如 当连接器没有牢固地插入时。TMC5240 通过检查是否可以达到所需的电机线圈电流来检测开路负载情况。这样，欠压条件、高电机速度设置或短路和过热条件也可能导致触发负载开路标志，并通知用户电机扭矩可能会受到影响。 在电机静止时，无法测量开路负载，因为线圈最终可能具有零电流。

为了安全地检测中断的线圈连接，请在 SpreadCyle 中运行，并仅使用低速或标称电机速度操作在单个方向上以选定微步分辨率的最小四倍运动检查打开负载标志。但是，ola 和 olb 标志仅具提供信息，不会引起驱动程序的任何操作。

欠压锁定保护

TMC5240 具有针对 VM、VCC_IO 和电荷泵的欠压锁定保护 (UVLO)。

VM 上的 UVLO 条件在低于 4.15V（最大值）时触发。

VCC_IO 上的 UVLO 条件在低于 1.95V（最大值）时触发。

电荷泵上的 UVLO 条件被触发 在电荷泵出现错误条件的情况下触发，例如，由于错误的电容值。

可以从寄存器 GSTAT 中读取一个 VM UVLO 条件作为标志 vm_uvlo。 该标志是写清除标志。 必须主动设置为 1 才能清除它。

在 VCC_IO UVLO 期间，无法与 IC 进行通信。 DIAG0 引脚将低电平有效（漏极开路）。

ESD 防护

该芯片在每个引脚上都有内部 ESD 保护。

当在正电压电源（VM 引脚）上使用至少 1uF 的旁路电容器时，TMC5240 电机相位输出引脚在应用中受到高达 8KV HBM 的保护。

PRELIMINARY CONFIDENTIAL

无论如何，这对电机热插拔没有任何保护作用。

时钟振荡器与时钟输入

使用内部时钟

如果要使用内部时钟振荡器，直接将 CLK 输入引脚连接到 IC 附近的 GND。

使用外部时钟

当外部时钟可用时，建议使用 12 MHz 至 20 MHz 的频率以获得最佳性能。时钟信号的占空比并不重要，只要满足引脚的最小高电平或低电平输入时间（参见电气特性）即可。

当时钟占空比为 50% 时，最高可使用 20 MHz。

当使用高时钟频率时，请确保时钟源提供干净的 CMOS 输出逻辑电平和陡峭的斜率。

只要在 CLK 引脚上提供外部时钟，就会启用外部时钟输入。

读取寄存器 IOIN 中的位 ext_clk 可以反馈当前正在使用哪个时钟源（1 = 外部时钟）。

万一外部时钟出现故障或被关闭，内部时钟会自动无缝接管，以防止驱动器损坏。

通用寄存器映射和寄存器信息

本节提供了一些关于寄存器映射的一般信息。

- 有关所有寄存器及其内容的详细信息，请参见寄存器映射部分。
- 除非另有说明，否则所有寄存器在上电时复位为 0.
- 将 0x80 添加到地址 Addr 以进行写访问！

表 21. 寄存器映射概述（待续）

寄存器	描述
通用配置寄存器	这些寄存器包含 ● 全局配置 ● 全局状态标志 ● 接口配置 ● 和 I/O 信号配置
斜坡发生器运动控制寄存器组	该寄存器集提供如下寄存器 ● 选择斜坡模式 ● 选择速度 ● 回零位 ● 加减速 ● 目标定位 ● 参考开关和 StallGuard2 事件配置 ● 斜坡和参考开关状态
速度相关驱动器特性控制寄存器组	该寄存器集提供如下寄存器 ● 驱动电流控制 ● 设置 CoolStep 操作的阈值 ● 设置不同斩波模式的阈值 ● 设置 DcStep 操作的阈值
直接模式寄存器	该寄存器组提供用于直接线圈电流控制模式的寄存器。
编码器寄存器组	编码器寄存器集提供正确 ABN 编码器操作所需的所有寄存器。
ADC 寄存器	该寄存器组提供寄存器来控制和读取内部 ADC。

PRELIMINARY CONFIDENTIAL

表 21. 寄存器映射概述（待续）

电机驱动寄存器组	该寄存器集提供如下寄存器 ● 设置/读取微步表和计数器 ● 斩波器和驱动器配置 ● CoolStep 和 StallGuard 配置 ● DcStep配置 ● 读出 StallGuard 值和驱动器错误标志

寄存器映射

TMC5240

地址	名称	MSB							LSB
通用配置寄存器									
0x00	GCONF[23:16]				length_step_pulse[3:0]				direct_mode
	GCONF[15:8]	stop_enable	small_hysteresis	diag1_poscomp_pushpull	diag0_int_pushpull	–	–	–	diag1_noscomp_dir
	GCONF[7:0]	diag0_nint_step	–	–	shaft	–	en_pwm_mode	fast_standstill	–
0x01	GSTAT[7:0]						uv_cp	drv_err	reset
0x02	IFCNT[7:0]	IFCNT[7:0]							
0x03	SLAVECONF[15:8]					SENDDELAY[3:0]			
	SLAVECONF[7:0]	SLAVEADDR[7:0]							
0x04	IOIN[31:24]	VERSION[7:0]							
	IOIN[23:16]	–	–	–	–	–	SILICON_RV[2:0]		
	IOIN[15:8]	ADC_ERR	EXT_CLK	EXT_RES_DET	OUTPUT	COMP_B1_B2	COMP_A1_A2	COMP_B	COMP_A
	IOIN[7:0]	reserved	UART_EN	ENCN	DRV_ENN	ENCA	ENCB	REFR	REFL
0x05	X_COMPARE[31:24]	X_COMPARE[31:24]							
	X_COMPARE[23:16]	X_COMPARE[23:16]							
	X_COMPARE[15:8]	X_COMPARE[15:8]							
	X_COMPARE[7:0]	X_COMPARE[7:0]							
0x06	X_COMPARE_REPEAT[23:16]	X_COMPARE_REPEAT[23:16]							
	X_COMPARE_REPEAT[15:8]	X_COMPARE_REPEAT[15:8]							
	X_COMPARE_REPEAT[7:0]	X_COMPARE_REPEAT[7:0]							
0x0A	DRV_CONF[23:16]			–	–	–	–	–	–
	DRV_CONF[15:8]	–	–	–	–	–	–	–	–
	DRV_CONF[7:0]	–	–	SLOPE_CONTROL[1:0]		–	–	CURRENT_RANGE[1:0]	
0x0B	GLOBAL SCALER[7:0]	GLOBALSCALER[7:0]							
速度相关配置寄存器									
0x10	IHOLD_IRUN[31:24]					IRUNDELAY[3:0]			
	IHOLD_IRUN[23:16]	–	–	–	–	IHOLDDELAY[3:0]			
	IHOLD_IRUN[15:8]	–	–	–	IRUN[4:0]				
	IHOLD_IRUN[7:0]	–	–	–	IHOLD[4:0]				
0x11	TPOWERDOWN[7:0]	TPOWERDOWN[7:0]							
0x12	TSTEP[23:16]					TSTEP[19:16]			
	TSTEP[15:8]	TSTEP[15:8]							

PRELIMINARY CONFIDENTIAL

地址	名称	MSB							LSB
	TSTEP[7:0]	TSTEP[7:0]							
0x13	TPWMTHRS[23:16]					TPWMTHRS[19:16]			
	TPWMTHRS[15:8]	TPWMTHRS[15:8]							
	TPWMTHRS[7:0]	TPWMTHRS[7:0]							
0x14	TCOOLTHRS[23:16]					TCOOLTHRS[19:16]			
	TCOOLTHRS[15:8]	TCOOLTHRS[15:8]							
	TCOOLTHRS[7:0]	TCOOLTHRS[7:0]							
0x15	THIGH[23:16]					THIGH[19:16]			
	THIGH[15:8]	THIGH[15:8]							
	THIGH[7:0]	THIGH[7:0]							
斜坡发生器寄存器									
0x20	RAMPMODE[7:0]							RAMPMODE[1:0]	
0x21	XACTUAL[31:24]	XACTUAL[31:24]							
	XACTUAL[23:16]	XACTUAL[23:16]							
	XACTUAL[15:8]	XACTUAL[15:8]							
	XACTUAL[7:0]	XACTUAL[7:0]							
0x22	VACTUAL[23:16]	VACTUAL[23:16]							
	VACTUAL[15:8]	VACTUAL[15:8]							
	VACTUAL[7:0]	VACTUAL[7:0]							
0x23	VSTART[23:16]							VSTART[17:16]	
	VSTART[15:8]	VSTART[15:8]							
	VSTART[7:0]	VSTART[7:0]							
0x24	A1[15:8]	A1[15:8]							
	A1[7:0]	A1[7:0]							
0x25	V1[23:16]					V1[19:16]			
	V1[15:8]	V1[15:8]							
	V1[7:0]	V1[7:0]							
0x26	AMAX[15:8]	AMAX[15:8]							
	AMAX[7:0]	AMAX[7:0]							
0x27	VMAX[23:16]		VMAX[22:16]						
	VMAX[15:8]	VMAX[15:8]							
	VMAX[7:0]	VMAX[7:0]							
0x28	DMAX[15:8]	DMAX[15:8]							
	DMAX[7:0]	DMAX[7:0]							
0x29	TVMAX[15:8]	TVMAX[15:8]							
	TVMAX[7:0]	TVMAX[7:0]							
0x2A	D1[15:8]	D1[15:8]							
	D1[7:0]	D1[7:0]							
0x2B	VSTOP[23:16]							VSTOP[17:16]	
	VSTOP[15:8]	VSTOP[15:8]							
	VSTOP[7:0]	VSTOP[7:0]							
0x2C	TZEROWAIT[15:8]	TZEROWAIT[15:8]							

PRELIMINARY CONFIDENTIAL

地址	名称	MSB							LSB
	TZEROWAIT[7:0]	TZEROWAIT[7:0]							
0x2D	XTARGET[31:24]	XTARGET[31:24]							
	XTARGET[23:16]	XTARGET[23:16]							
	XTARGET[15:8]	XTARGET[15:8]							
	XTARGET[7:0]	XTARGET[7:0]							
0x2E	V2[23:16]					V2[19:16]			
	V2[15:8]	V2[15:8]							
	V2[7:0]	V2[7:0]							
0x2F	A2[15:8]	A2[15:8]							
	A2[7:0]	A2[7:0]							
0x30	D2[15:8]	D2[15:8]							
	D2[7:0]	D2[7:0]							
斜坡发生器驱动器功能控制寄存器									
0x33	VDCMIN[23:16]		VDCMIN[14:8]						
	VDCMIN[15:8]	VDCMIN[7:0]							
	VDCMIN[7:0]	reserved[7:0]							
0x34	SW_MODE[15:8]		virtual_stop_enc	en_virtual_stop_r	en_virtual_stop_l	en_softstop	sg_stop	en_latch_encoder	latch_r_inactive
	SW_MODE[7:0]	latch_r_active	latch_l_inactive	latch_l_active	swap_lr	pol_stop_r	pol_stop_l	stop_r_enable	stop_l_enable
0x35	RAMP_STAT[15:8]	status_virtual_stop_r	status_virtual_stop_l	status_sg	second_move	t_zerowait_active	vzero	position_reached	velocity_reached
	RAMP_STAT[7:0]	event_pos_reached	event_stop_sg	event_stop_r	event_stop_l	status_latch_r	status_latch_l	status_stop_r	status_stop_l
0x36	XLATCH[31:24]	XLATCH[31:24]							
	XLATCH[23:16]	XLATCH[23:16]							
	XLATCH[15:8]	XLATCH[15:8]							
	XLATCH[7:0]	XLATCH[7:0]							
编码器寄存器									
0x38	ENCMODE[15:8]						enc_sel_decimal	latch_x_act	clr_enc_x
	ENCMODE[7:0]	pos_neg_edge[1:0]		clr_once	clr_cont	ignore_AB	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]							
	ENC_CONST[15:8]	ENC_CONST[15:8]							
	ENC_CONST[7:0]	ENC_CONST[7:0]							
0x3B	ENC_STATUS[7:0]							deviation_warn	n_event

PRELIMINARY CONFIDENTIAL

地址	名称	MSB							LSB
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]							
0x3D	ENC_DEVIATION[23:16]							ENC_DEVIATION[19:16]	
	ENC_DEVIATION[15:8]	ENC_DEVIATION[15:8]							
	ENC_DEVIATION[7:0]	ENC_DEVIATION[7:0]							
0x3E	VIRTUAL_STOP_L[31:24]	VIRTUAL_STOP_L[31:24]							
	VIRTUAL_STOP_L[23:16]	VIRTUAL_STOP_L[23:16]							
	VIRTUAL_STOP_L[15:8]	VIRTUAL_STOP_L[15:8]							
	VIRTUAL_STOP_L[7:0]	VIRTUAL_STOP_L[7:0]							
0x3F	VIRTUAL_STOP_R[31:24]	VIRTUAL_STOP_R[31:24]							
	VIRTUAL_STOP_R[23:16]	VIRTUAL_STOP_R[23:16]							
	VIRTUAL_STOP_R[15:8]	VIRTUAL_STOP_R[15:8]							
	VIRTUAL_STOP_R[7:0]	VIRTUAL_STOP_R[7:0]							
ADC 寄存器									
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]							
电机驱动器寄存器									
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]							

PRELIMINARY CONFIDENTIAL

地址	名称	MSB							LSB
	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]							
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[15:8]							MSCNT[9:8]	
	MSCNT[7:0]	MSCNT[7:0]							
0x6B	MSCURACT[23:16]							CUR_A[1:0]	
	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]

PRELIMINARY CONFIDENTIAL

地址	名称	MSB							LSB
	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]								sfilt
	COOLCONF[23:16]	–	sgt[6:0]						
	COOLCONF[15:8]	seimin	sedn[1:0]		–	semax[3:0]			
	COOLCONF[7:0]	–	seup[1:0]		–	semin[3:0]			
0x6E	DCCTRL[23:16]	DC_SG[7:0]							
	DCCTRL[15:8]	–	–	–	–	–	–	DC_TIME[9:8]	
	DCCTRL[7:0]	DC_TIME[7:0]							
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[23:16]								PWM_SCALE_AUTO[0]
	PWM_SCALE[15:8]	–	–	–	–	–	–	PWM_SCALE_SUM[9:8]	
	PWM_SCALE[7:0]	PWM_SCALE_SUM[7:0]							
0x72	PWM_AUTO[15:8]	–	–	–	–	–	–	–	–
	PWM_AUTO[7:0]	PWM_OFS_AUTO[7:0]							
0x74	SG4_THRS[7:0]	SG4_THRS[7:0]							
0x75	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]							

Register Details

GCONF (0x0)

全局配置标志

PRELIMINARY CONFIDENTIAL

BIT位				20	19	18	17	16
领域				length_step_pulse[3:0]				direct_mode
初始值				0x0				0x0
访问类型				Write, Read				Write, Read
BIT位	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
领域	stop_enable	small_hysteresis	diag1_poscomp_pushpull	diag0_int_pushpull	–	–	–	diag1_nposcomp_dir
初始值	0x0	0x0	0x0	0x0	–	–	–	0x0
访问类型	Write, Read	Write, Read	Write, Read	Write, Read	–	–	–	Write, Read
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
领域	diag0_nint_step	–	–	shaft	–	en_pwm_mode	fast_standstill	–
初始值	0x0	–	–	0x0	–	0x0	0x0	–
访问类型	Write, Read	–	–	Write, Read	–	Write, Read	Write, Read	–

领域	BITS	DESCRIPTION	DECODE
length_step_pulse	20:17	仅 cDriver：length_step_pulse = 0：STEP 输出在每一步切换； length_step_pulse = 1...15：STEP 引脚高电平时间（以时钟周期数表示）	
direct_mode	16	通过串行接口实现直接电机相电流控制。	0x0: 正常操作 0x1: 电机线圈电流和极性通过串行接口直接编程：寄存器 XTARGET (0x2D) 指定带符号的线圈 A 电流（位 8..0）和线圈 B 电流（位 24..16）. 在此模式下，电流由 IHOLD 设置缩放。 StealthChop 基于速度的电流调节在此模式下不可用。 自动 StealthChop 电流调节仅适用于低步进电机速度。
stop_enable	15	电机硬停功能启用。	0x0: 正常操作 0x1: 紧急停止：ENCA 在连接为高电平时停止相序（相序器不执行任何步骤，电机进入静止状态）。
small_hysteresis	14		0x0: 步进频率比较的迟滞是 1/ 16 0x1:步进频率比较的滞后为 1/ 32
diag1_poscomp_pushpull	13	DIAG1 输出类型配置.	0x0: DIAG1 为集电极开路输出（低电平有效） 0x1: 启用 DIAG1 推挽输出（高电平有效）
diag0_int_pushpull	12	DIAG0 输出类型配置.	0x0: DIAG0_SW 为集电极开路输出（低电平有效） 0x1: 启用 DIAG0_SW 推挽输出（高电平有效）
diag1_nposcomp_dir	8	不使用 UART 时的 DIAG1 输出配置。	0x0: DIAG1 输出位置比较信号0x1: 启用 DIAG1 作为外部 STEP/DIR 驱动程序的 DIR 输出

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
diag0_nint_step	7	DIAG0 输出配置。	0x0: DIAG0 输出中断信号 0x1: 为外部 STEP/DIR 驱动器启用 DIAG0 作为 STEP 输出（半频、双边沿触发或 length_step_pulse 时的频率！ = 0）
shaft	4	改变电机方向/方向标志	0x0: 默认电机方向 0x1: 反向电机运动方向
en_pwm_mode	2	启用stealthChop(TM) 模式	0x0: stealthChop关闭 0x1: 启用 StealthChop 电压 PWM 模式（取决于速度阈值）。 仅在静止且 IHOLD= IRUN 电流时才可以从关闭状态切换到开启状态。
fast_standstill	1	步骤执行超时，直到检测到静止状态	0x0: Normal time: 2^20 clocks 0x1: Short time: 2^18 clocks

GSTAT (0x1)

全局状态标志

(用“1”位重写以清除相应的标志)

BIT		2	1	0
Field		uv_cp	drv_err	reset
Reset		0x0	0x0	0x1
Access Type		Write 1 to Clear, Read	Write 1 to Clear, Read	Write 1 to Clear, Read

BITFIELD	BITS	DESCRIPTION	DECODE
vm_uvlo	4	1: VM 自上次复位后发生欠压。	
register_reset	3		0x0: 正常运行 0x1: 表示寄存器映射已被重置。 所有寄存器已被清除以复位值.
uv_cp	2	电荷泵欠压状态标志	0x0: 正常运行 0x1: 指示电荷泵欠压。 驱动器在欠压期间被禁用。 该标志被锁存以获取信息。
drv_err	1	驱动错误标记	0x0: 正常运行 0x1: 表示驱动器因过热或短路检测而关闭. 阅读 DRV_STATUS 以获取详细信息. 只有当温度再次低于限制时才能清除该标志
reset	0	复位标志	0x0: 正常运行 0x1: 表示 IC 已复位.

IFCNT (0x2)

接口传输计数器。

该寄存器随着每次成功的 UART 接口写访问而递增。可以读出它以检查串行传输是否丢失数据。读取访问不会更改内容。在 SPI 操作中禁用。 计数器从 255 转回 0。

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
IFCNT	7:0	接口传输计数器。该寄存器随着每次成功的 UART 接口写访问而递增。可以读出它以检查串行传输是否丢失数据。 读取访问不会更改内容。 在 SPI 操作中禁用。 计数器从 255 转回 0。

SLAVECONF (0x3)

BIT					11	10	9	8
Field					SENDDELAY[3:0]			
Reset					0x0			
Access Type					Write, Read			
BIT	7	6	5	4	3	2	1	0
Field	SLAVEADDR[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION	DECODE
SENDDELAY	11:8	SWUART 从机配置	0x0: 8 bit times (不允许有多个从站) 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
SLAVEADDR	7:0	*SLAVEADDR*: 这八位设置UART接口的单元地址。根据 SDI、SCK、CSN 的定义，地址从 1 增加到 7。 CSN, SCK, SDI 000: +0 001: +1 010: +2 011: +3 100: +4 101: +5 110: +6 111: +7 范围：0-254（不要超过 254）	

IOIN (0x4)

读取所有可用输入引脚的状态，并以最高字节返回 IC 修订版

PRELIMINARY CONFIDENTIAL

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	REFR	REFL
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

BITFIELD	BITS	DESCRIPTION
VERSION	31:24	0x40 = 第一版 IC 相同的数字意味着完全的数字兼容性.
SILICON_RV	18:16	硅版本号
ADC_ERR	15	1: 表示 ADC 工作不正常的信号。 不要使用 ADC 功能.
EXT_CLK	14	0: 内部振荡器用于产生时钟信号 (12.5 MHz). 1: 外部振荡器用于产生时钟信号。
EXT_RES_DET	13	1: REF 和 GND 之间的外部电阻 0: 未检测到外部电阻
OUTPUT	12	通过引脚 UART_EN 使能 UART 时 SDO 引脚的输出极性。其主要目的是将 SDO 用作 NAO 下一个地址输出信号，用于多个 IC 的链式寻址。 注意：Reset Value 为 1 用作 NAO 到单线链中的下一个 IC
COMP_B1_B2	11	COMP_B1_B2 (StallGuard4 比较器 B，用于 IC 测试)
COMP_A1_A2	10	COMP_A1_A2 (StallGuard4 比较器 A，用于 IC 测试)
COMP_B	9	COMP_B (斩波比较器 B，用于 IC 测试)
COMP_A	8	COMP_A (斩波比较器A，用于 IC 测试)
reserved	7	
UART_EN	6	1 = UART 接口使能
ENCN	5	N通道状态
DRV_ENN	4	驱动器禁用/启用状态。
ENCA	3	A 通道状态
ENCB	2	B 通道状态
REFR	1	

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
REFL	0	

X_COMPARE (0x5)

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

BITFIELD	BITS	DESCRIPTION
X_COMPARE	31:0	位置比较寄存器，用于运动控制器位置选通。 X_COMPARE 是一个绝对位置。 位置脉冲可在输出 SWP_DIAG1 上获得。 *XACTUAL = X_COMPARE*: 输出信号PP（位置脉冲）变高。 如果位置不匹配，它会返回低电平状态。 如果 X_COMPARE_REPEAT >1，则 X_COMPARE 是周期性位置选通触发输出的位置参考。

X_COMPARE_REPEAT (0x6)

BIT	23	22	21	20	19	18	17	16
Field	X_COMPARE_REPEAT[23:16]							
Reset	0x0							
Access Type	Write, Read							

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
X_COMPARE_REPEAT	23:0	该寄存器以微步为单位定义相对距离（基于 MRES 配置）。 如果设置为 >1，则每次执行 X_PDISTANCE μsteps 的倍数时都会引发位置比较脉 因此，X_COMPARE 寄存器定义了 X_PDISTANCE 步长的模计算的基准位置，该位置已被设为正或负方向。

DRV_CONF (0xA)

BIT			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	

BITFIELD	BITS	DESCRIPTION	DECODE
SLOPE_CONTROL	5:4	Slope Control Setting斜坡控制设定	0x0: 100V/μs 0x1: 200V/μs 0x2: 400V/μs 0x3: 800V/μs
CURRENT_RANGE	1:0	此设置允许驱动器 RDSon 电流感应与电机电流范围匹配. 选择最低的拟合范围以获得最佳的电流精度。 该值是峰值电流设置。	0x0: 1A 0x1: 2A 0x2: 3A 0x3: 3A

PRELIMINARY CONFIDENTIAL

GLOBAL SCALER (0xB)

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

BITFIELD	BITS	DESCRIPTION
GLOBALSCALER	7:0	电机电流的全局缩放。 该值乘以电流比例，以使驱动器适应特定的电机类型。 应该在调整其他设置之前选择该值，因为它也会影响斩波器磁滞。 该值仅用于微调电机电流。 0: 满量程（或写 256）1 … 31: 不允许操作 32 … 255: 32/256 … 255/256 的最大电流。 *提示：建议值 >128 以获得最佳结果*

IHOLD_IRUN (0x10)

Test Reg

BIT					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				

BITFIELD	BITS	DESCRIPTION
IRUNDELAY	27:24	控制检测到启动后电机启动的时钟周期数。0：即时上电 1..15：以 IRUNDELAY * 512 时钟的倍数为单位的电流增量步长延迟

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
IHOLDDELAY	19:16	控制一旦检测到静止 (stst=1) 并且 TPOWERDOWN 到期后电机电流下降的时钟周期数。 平滑过渡避免了电流下降时的电机抖动。 0: 瞬间下降 1..15: 以 2^18 个时钟的倍数计算每个电流减少步骤的延迟
IRUN	12:8	电机运行电机 (0=1/32…31=32/32) *提示：使用 16 到 31 之间的设置以获得最佳微步性能。*
IHOLD	4:0	静止电流 (0=1/32…31=32/32) 与 StealthChop 模式相结合，设置 IHOLD=0 允许选择续流或线圈短路以使电机静止。

TPOWERDOWN (0x11)

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

BITFIELD	BITS	DESCRIPTION
TPOWERDOWN	7:0	*TPOWERDOWN* 设置电机静止后 (stst) 到电机电流下降的延迟时间。 时间范围约为 0 到 4 秒。 *注意：需要最小设置为 2 才能自动调整 StealthChop PWM_OFFS_AUTO。* *Reset Default = 10* *0…((2^8)-1) * 2^18 t_{CLK}*

TSTEP (0x12)

BIT					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							

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
TSTEP	19:0	两个 1/256 微步之间的实际测量时间，源自脉冲输入频率，以 1/fCLK 为单位。溢出或静止时测量值为 (2^20)-1。 所有与 TSTEP 相关的阈值都使用比较值的 1/16 的滞后来补偿时钟或步进频率中的抖动。 标志 small_hysteresis 将滞后修改为较小的值 1/32。 (Txxx*15/16)-1 或 (Txxx*31/32)-1 用作每个比较值的第二个比较值。 这意味着，较低的开关速度等于计算的设置，但较高的开关速度由滞后设置定义。 与运动控制器一起工作时，给定速度 V 的测量 TSTEP 在范围内 $(2^{24} / V) \leq TSTEP \leq 2^{24} / V - 1$. 在 DcStep 模式下，TSTEP 不会显示电机的平均速度，而是显示每个微步的速度，这可能不稳定，因此如果电机运行速度低于目标速度，则不代表电机的实际速度。

TPWMTHRS (0x13)

BIT					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							

BITFIELD	BITS	DESCRIPTION
TPWMTHRS	19:0	这是 StealthChop 电压 PWM 模式的上限速度。 ● TSTEP ≥ TPWMTHRSStealthChop PWM 模式启用（如果配置的话） ● DcStep 禁用

PRELIMINARY CONFIDENTIAL

TCOOLTHRS (0x14)

BIT					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							

BITFIELD	BITS	DESCRIPTION
TCOOLTHRS	19:0	这是开启智能能源 CoolStep 和 StallGuard 功能的下限速度。（无符号） 设置此参数可在低速时禁用 CoolStep，因为它无法可靠工作。 失速停止功能（使用内部运动控制器时通过 sg_stop 启用）和失速输出信号在超过该速度时启用。 在非 DcStep 模式下，一旦速度低于该阈值，它将再次禁用。 *TCOOLTHRS* ≥ *TSTEP* ≥ *THIGH*: ● CoolStep 已启用（如果已配置） *TCOOLTHRS* ≥ *TSTEP* ● 如果已配置的话，则启用失速时电机停止 ● 如果配置的话，失速时候输出信号 (DIAG0/1) 启用

THIGH (0x15)

BIT					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							

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
THIGH	19:0	此速度设置允许不同速度下关切换到不同的斩波器模式和整步以实现扭矩最大化。 (无符号) 只要超过 THIGH 阈值，失速检测功能就会关闭 2-3 个电气周期，以补偿切换模式的影响。 *TSTEP ≤ THIGH*: ● CoolStep 被禁用（电机以正常电流量程运行） ● StealthChop电压PWM模式被禁用 ● 如果设置了vhighchm，斩波器切换到chm=1，TFD=0（仅具有缓慢衰减的恒定关闭时间）。 ● 如果设置了 vhighfs，则电机在全步模式下运行，并且失速检测将切换到 DcStep 失速检测。

RAMPMODE (0x20)

BIT		1	0
Field		RAMPMODE[1:0]	
Reset		0x0	
Access Type		Write, Read	

BITFIELD	BITS	DESCRIPTION	DECODE
RAMPMODE	1:0	运动控制器斜坡模式	0x0: 定位模式（使用所有 A、D 和 V 参数） 0x1: 速度模式使用正 VMAX（使用 AMAX 加速） 0x2: 速度模式为负 VMAX（使用 AMAX 加速） 0x3: 保持模式（速度保持不变，除非发生停止事件）

XACTUAL (0x21)

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

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
XACTUAL	31:0	实际电机位置（有符号） *提示：这个值通常只能在驱动器回零之后修改。 在位置模式下，修改寄存器内容会启动一个动作。*

VACTUAL (0x22)

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

BITFIELD	BITS	DESCRIPTION
VACTUAL	23:0	来自斜坡发生器的实际电机速度（有符号） 该符号与运动方向相匹配。 负号意味着降低 XACTUAL 的运动。 +-(2^23)-1 [µsteps / t]

PRELIMINARY CONFIDENTIAL

VSTART (0x23)

BIT							17	16
Field							VSTART[17:16]	
Reset							0x0	
Access Type							Write, Read	
BIT	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Field	VSTART[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	VSTART[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
VSTART	17:0	电机启动速度（无符号） 通常，设置 VSTOP ≥ VSTART。 如果运动距离足以确保从 VSTART 减速到 VSTOP，则不需要这样做。 0…(2^18)-1 [µsteps / t]

A1 (0x24)

BIT	15	14	13	12	11	10	9	8
Field	A1[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	A1[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
A1	17:0	VSTART 和 V1 之间的第一次加速度（无符号） 0…(2^18)-1 [µsteps / ta²]

PRELIMINARY CONFIDENTIAL

V1 (0x25)

BIT					19	18	17	16
Field					V1[19:16]			
Reset					0x0			
Access Type					Write, Read			
BIT	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Field	V1[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	V1[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
V1	19:0	第一阶段加速 / 减速阶段阈值速度（无符号） 0：禁用 A1 和 D1 相位，仅使用 AMAX、DMAX 0…(2^20)-1 [µsteps / t]

AMAX (0x26)

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

BITFIELD	BITS	DESCRIPTION
AMAX	17:0	V1 和 VMAX 之间的第二次加速度（无符号） 这是速度模式的加速和减速值。 0…(2^18)-1[µsteps / ta²]

PRELIMINARY CONFIDENTIAL

VMAX (0x27)

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

BITFIELD	BITS	DESCRIPTION
VMAX	22:0	运动斜坡目标速度（用于定位确保 VMAX ≥ VSTART）（无符号） 这是速度模式下的目标速度。 它可以在运动过程中随时更改。 0…(2^23)-512 [µsteps / t]

DMAX (0x28)

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

BITFIELD	BITS	DESCRIPTION
DMAX	17:0	VMAX 和 V1 之间的减速（无符号） 0…(2^18)-1[µsteps / ta²]

PRELIMINARY CONFIDENTIAL

TVMAX (0x29)

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

BITFIELD	BITS	DESCRIPTION
TVMAX	15:0	以 512 个时钟为单位的等速段的最短时间。 0: 禁用等速阶段的最小持续时间设置 >0: 在从加速到减速的任何变化之间插入恒定速度的最短持续时间，反之亦然，以减少加加速度 0…(2^16)-1 * 512 t_{CLK} 注意：在位置模式和静止状态下设置 VMAX 后配置此寄存器。 在速度模式下设置 TVMAX=0 以避免在切换回斜坡模式时触发 TVMAX 延迟。

D1 (0x2A)

BIT	15	14	13	12	11	10	9	8
Field	D1[15:8]							
Reset	0xA							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	D1[7:0]							
Reset	0xA							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
D1	17:0	V1 和 VSTOP 之间的减速度（无符号） 注意：定位模式下不要设置为0，即使V1=0！ 1…(2^16)-1[µsteps / ta²] 重置默认值 = 10

PRELIMINARY CONFIDENTIAL

VSTOP (0x2B)

BIT							17	16
Field							VSTOP[17:16]	
Reset							0xA	
Access Type							Write, Read	
BIT	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Field	VSTOP[15:8]							
Reset	0xA							
Access Type	Write, Read							
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	VSTOP[7:0]							
Reset	0xA							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
VSTOP	17:0	电机停止速度（无符号） 提示：设置 VSTOP ≥ VSTART 允许短距离定位 注意：定位模式下不要设置为 0，建议至少设置为 10！ 1…(2^18)-1[μsteps / t] 重置默认值=10

TZEROWAIT (0x2C)

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

BITFIELD	BITS	DESCRIPTION
TZEROWAIT	15:0	定义减速到零速度后开始下一次运动或方向反转之前的等待时间。 时间范围约为 0 到 2 秒。 此设置可避免过度加速，例如 从 VSTOP 到 -VSTART。 0…(2^16)-1 * 512 tCLK

PRELIMINARY CONFIDENTIAL

XTARGET (0x2D)

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

BITFIELD	BITS	DESCRIPTION
XTARGET	31:0	斜坡模式的目标位置（有符号）。向该寄存器写入一个新的目标位置，以便在 RAMPMODE=0 中激活斜坡发生器位置模式。在此之前初始化所有速度、加速度和减速度参数。 *提示：该位置允许回绕，因此，XTARGET 值可以选择性地被视为一个无符号数。* *提示：最大可能的位移是 +/-((2^31)-1)。* *提示：在运动过程中增加 V1、D1 或 DMAX 时，如果需要，随后重写 XTARGET 以触发第二个加速阶段。* -2^31… +(2^31)-1

V2 (0x2E)

BIT		19	18	17	16
Field		V2[19:16]			
Reset		0x0			
Access Type		Write, Read			

PRELIMINARY CONFIDENTIAL

BIT	15	14	13	12	11	10	9	8
Field	V2[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	V2[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
V2	19:0	使用 AMAX/2 和 DMAX/2 激活加速段时与 VMAX 的速度差异。 0: 禁用 AMAX/2 和 DMAX/2 相位，仅使用 AMAX、DMAX 0…(2^20)-1 [µsteps / t]

A2 (0x2F)

BIT	15	14	13	12	11	10	9	8
Field	A2[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	A2[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
A2	17:0	V1 和 V2 之间的加速度（无符号 0…(2^18)-1 [µsteps / ta²]

D2 (0x30)

BIT	15	14	13	12	11	10	9	8
Field	D2[15:8]							
Reset	0xA							
Access Type	Write, Read							

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
D2	17:0	V2 和 V1 之间的减速度（无符号） 注意：定位模式下不要设置为0，即使V2=0！ 1…(2^18)-1[µsteps / ta²] 重置默认值 = 10

VDCMIN (0x33)

dcStep start velocity

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

BITFIELD	BITS	DESCRIPTION
VDCMIN	22:8	自动切换 DcStep 在速度 VDCMIN （无符号）以上被启用 在此模式下，实际位置由无传感器电机换向确定并反馈给 XACTUAL。 如果电机负载过重，VDCMIN 也用作最小步进速度。 激活失速停止 (sg_stop) 以检测失步。 0: 禁用，DcStep 关闭 *|VACT| ≥ VDCMIN* ≥ 256: ● 触发与超出 THIGH 设置相同的动作。. ● 开启自动 DcStep *提示：同时设置 DCCTRL 参数以操作 DcStep。 （仅位 22…8 用于值和比较）*
reserved	7:0	reads always 0

PRELIMINARY CONFIDENTIAL

SW_MODE (0x34)

Switch mode configuration

BIT		14	13	12	11	10	9	8
Field		virtual_stop_enc	en_virtual_stop_r	en_virtual_stop_l	en_softstop	sg_stop	en_latch_encoder	latch_r_inactive
Reset		0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type		Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	latch_r_active	latch_l_inactive	latch_l_active	swap_lr	pol_stop_r	pol_stop_l	stop_r_enable	stop_l_enable
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
virtual_stop_enc	14	虚拟停止源（VIRTUAL_STOP_L 和 VIRTUAL_STOP_R） 0：与斜坡发生器位置 XACTUAL 相关的虚拟停止 1：与编码器位置X_ENC相关的虚拟停止	
en_virtual_stop_r	13	1: 在有效的右虚拟停止条件下启用自动电机停止	
en_virtual_stop_l	12	1: 在有效的左侧虚拟停止条件下启用自动电机停止	
en_softstop	11	0: 硬停止 1: 软停止 软停止模式始终使用减速斜坡设置 DMAX、V1、D1、V2、D2、VSTOP 和 TZEROWAIT 来停止电机。 当速度符号与参考开关位置匹配（REFL、VIRTUAL_STOP_L 用于负速度，REFR、VIRTUAL_STOP_R 用于正速度）并且相应的开关停止功能启用时，将发生停止。 在电机释放之前，硬停止也使用 TZEROWAIT。 *注意：不要将软停止与 StallGuard2 结合使用。 对高速下的 StealthChop 操作请使用软停止。 在这种情况下，必须避免硬停止，因为它可能导致严重的过电流。*	0x0:硬停止 0x1: 软停止

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
sg_stop	10	通过 StallGuard2 启用停止（在 DcStep 模式下也可用）。 停止事件后禁止释放电机。 将 TCOOLTHRS 设置为有效的最低速度阈值。 *提示：电机启动期间不要启用Stallguard，等到电机速度超过某个值(建议比最大速度稍小一点)，此时 StallGuard2 会提供稳定的结果。 应使用 TCOOLTHRS 对该速度阈值进行编程。*	0x0: disabled 0x1: enabled
en_latch_encoder	9	1: 在发生参考开关事件时将编码器位置锁定到 ENC_LATCH。	
latch_r_inactive	8	1: 在右参考开关输入 REFR 上的非活动上升沿上激活位置锁定到 XLATCH。 活动电平由 pol_stop_r 定义。	
latch_r_active	7	1: 在右参考开关输入 REF 的有效上升沿上激活位置锁存到 LATCH。 *提示：激活latch_r_active 以通过读取 status_latch_r 来检测任何虚假停止事件。*	
latch_l_inactive	6	1: 在左侧参考开关输入 REFL 上的非活动沿上激活位置锁存到 XLATCH。 活动级别由 pol_stop_l 定义。	
latch_l_active	5	1: 在左侧参考开关输入 REFL 上的有效沿上激活位置锁存到 XLATCH。 *提示：通过读取 status_latch_l 激活 latch_l_active 以检测任何虚假停止事件.*	
swap_lr	4	1: 交换左右参考开关输入REFL和 REFR	
pol_stop_r	3	设置右参考开关输入的有效极性 0=非反相，高电平有效：REFR 上的高电平停止电机 1 = 反相，低电平有效：REFR 上的低电平停止电机	
pol_stop_l	2	设置左参考开关输入的有效极性 0=非反相，高电平有效：REFL 上的高电平停止电机 1 = 反相，低电平有效：REFL 上的低电平停止电机	
stop_r_enable	1	1：在激活右参考开关输入期间启用自动电机停止。提示：如果停止开关松开，电机将重新启动。	

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
stop_l_enable	0	1: 在活动左参考开关输入期间启用自动电机停止 提示：如果停止开关松开，电机将重新启动。	

RAMP_STAT (0x35)

斜坡状态和开关事件状态

BIT	15	14	13	12	11	10	9	8
Field	status_virtual_stop_r	status_virtual_stop_l	status_sg	second_move	t_zerowait_active	vzero	position_reached	velocity_reached
Reset	0x1	0x1	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Read Only	Read Only	Read Only	Write 1 to Clear, Read	Read Only	Read Only	Read Only	Read Only
BIT	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	event_pos_reached	event_stop_sg	event_stop_r	event_stop_l	status_latch_r	status_latch_l	status_stop_r	status_stop_l
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access Type	Write 1 to Clear, Read	Write 1 to Clear, Read	Read Only	Read Only	Write 1 to Clear, Read	Write 1 to Clear, Read	Read Only	Read Only

BITFIELD	BITS	DESCRIPTION
status_virtual_stop_r	15	虚拟右参考开关状态（1=激活）
status_virtual_stop_l	14	虚拟左参考开关状态（1=激活）
status_sg	13	1: 从 CoolStep 驱动或 DcStep 单元（如果启用）发出一个活动 StallGuard2 输入信号。 *提示：当轮询该标志时，可能会错过停顿事件——激活 sg_stop 以确保不会错过停顿事件。*
second_move	12	1：表示自动斜坡需要沿相反方向向后移动，例如 由于动态参数更改（写“1”清除）
t_zerowait_active	11	1：表示电机停止后 TZEROWAIT 激活。 在此期间，电机处于静止状态。
vzero	10	1: 表示实际速度为 0 的信号。
position_reached	9	1: 表示已到达目标位置的信号。 当 XACTUAL 和 XTARGET 匹配时，该标志被设置。
velocity_reached	8	1: 表示已达到目标速度的信号。当 VACTUAL 和 VMAX 匹配时，此标志置位。
event_pos_reached	7	1: 表示已到达目标位置的信号（position_reached 变为活动状态）。 （写入“1”以清除标志和中断条件）该位与中断输出信号进行或运算。
event_stop_sg	6	1: 发出有效的 StallGuard2 停止事件的信号。 复位寄存器将清除失速条件并且电机可以重新启动运动，除非运动控制器已经停止。（写入“1”以清除标志和中断条件）该位与中断输出信号进行或运算。

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
event_stop_r	5	1: 由于停止开关或虚拟停止而导致的主动停止右限位条件。 可以通过将 RAMP_MODE 设置为保持模式或通过命令向相反方向移动来移除停止条件和中断条件。在 soft_stop 模式下，该条件将保持有效，直到电机停止向停止开关的方向运动。 禁用停止开关或停止功能也会清除标志，但电机将继续运动。 该位与中断输出信号进行“或”运算。
event_stop_l	4	1: 由于停止开关或虚拟停止而导致的主动停止左限位条件。 可以通过将 RAMP_MODE 设置为保持模式或通过命令向相反方向移动来移除停止条件和中断条件。在 soft_stop 模式下，该条件将保持有效，直到电机停止向停止开关的方向运动。 禁用停止开关或停止功能也会清除标志，但电机将继续运动。 该位与中断输出信号进行“或”运算。
status_latch_r	3	1: Latch right ready (使用 SW_MODE 设置 latch_r_active 或latch_r_inactive 启用位置锁存) (Write ‘1’ to clear)
status_latch_l	2	1: Latch left ready (使用 SW_MODE 设置 latch_l_active 或latch_l_inactive 启用位置锁存) (Write ‘1’ to clear)
status_stop_r	1	右参考开关状态（1=激活）
status_stop_l	0	左参考开关状态（1=激活）

XLATCH (0x36)

斜坡发生器位置锁存

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

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
XLATCH	31:0	斜坡发生器锁存位置，在可编程开关事件时锁存 XACTUAL（请参阅 SW_MODE）. *提示：编码器位置可以与 XLATCH 一起锁定到 ENC_LATCH 以进行一致性检查。*

ENCMODE (0x38)

BIT						10	9	8
Field						enc_sel_decimal	latch_x_act	clr_enc_x
Reset						0x0	0x0	0x0
Access Type						Write, Read	Write, Read	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	pos_neg_edge[1:0]		clr_once	clr_cont	ignore_AB	pol_N	pol_B	pol_A
Reset	0x0			0x0	0x0	0x0	0x0	0x0
Access Type	Write, Read		Write, Read	Write, Read	Write, Read	Write, Read	Write, Read	Write, Read

BITFIELD	BITS	DESCRIPTION	DECODE
enc_sel_decimal	10	编码器预分频器模式选择	0x0: 编码器预分频器除数二进制模式：COUNTS ENC_CONST（FRACTISIAL PORT）/ 65536 0x1: 编码器预分频器除数十进制模式：CONC_CONST（FRACTIONAL PORT）/ 10000计数
latch_x_act	9	位置锁存配置	0x0: 禁用 0x1: 同时将 XACTUAL 位置与 X_ENC 一起锁定。允许在POS_EDGE和NEG_EDGE选择的N通道事件时锁存斜坡发生器位置。
clr_enc_x	8	编码器锁存器配置	0x0：发生 N 事件时，X_ENC 变为仅锁存到 ENC_LATCH 0x1：在 N 事件时锁存并额外清除编码器计数器 X_ENC
pos_neg_edge	7:6	**N通道事件灵敏度**	0x0: N 通道事件在使能 N 事件级别期间是激活的 0x1: N通道在主动进行N个事件时有效 0x2:N通道在非活动的N事件时有效 0x3: N通道在活动和非活动的N事件时有效
clr_once	5	位置锁存配置	0x0: 禁用 0x1: 在写访问后下一个事件上的锁存或锁存和清除 X_ENC

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
clr_cont	4	位置锁存配置	0x0: 禁用 0x1: 始终锁存或锁存并在N事件时清除X_ENc（每次旋转一次，建议将此设置与边缘敏感N事件组合）
ignore_AB	3	N事件配置	0x0:只有pol_N、pol_A和pol_B的极性匹配时，才会发生N事件。 0x1: N通道事件忽略A和B极性
pol_N	2	定义 N 的有效极性	0x0: low active 0x1: high active
pol_B	1	对于N个通道事件所需的B极性	0x0: neg 0x1: pos
pol_A	0	Required A polarity for an N channel event	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							

BITFIELD	BITS	DESCRIPTION
X_ENC	31:0	Actual encoder position (signed)

PRELIMINARY CONFIDENTIAL

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							

BITFIELD	BITS	DESCRIPTION
		Accumulation constant (signed)
		16 bit integer part, 16 bit fractional part
		X_ENC accumulates
		+/- *ENC_CONST* / (2^16*X_ENC) (binary)
		or
		+/-*ENC_CONST* / (10^4*X_ENC) (decimal)
ENC_CONST	31:0	*ENCMODE* bit *enc_sel_decimal* switches between decimal and binary setting. Use the sign, to match rotation direction!
		binary:
		± [µsteps/2^16]
		±(0 …
		32767.999847)
		decimal:
		±(0.0 … 32767.9999)
		reset default = 1.0 (=65536)

ENC_STATUS (0x3B)

Encoder status information

PRELIMINARY CONFIDENTIAL

BIT		1	0
Field		deviation_warn	n_event
Reset		0x0	0x0
Access Type		Write 1 to Clear, Read	Write 1 to Clear, Read

BITFIELD	BITS	DESCRIPTION	DECODE
deviation_warn	1		0x0: 没有报警 0x1: 当报警仍然存在时，不能清除偏差报警。 将 ENC_DEVIATION 设置为 0 以禁用。
n_event	0		0x0：无事件 0x1：检测到事件。要清除状态位，请在相应位置写入 1。

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							

BITFIELD	BITS	DESCRIPTION
ENC_LATCH	31:0	编码器位置 X_ENC 在 N 事件上锁存

PRELIMINARY CONFIDENTIAL

ENC_DEVIATION (0x3D)

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

BITFIELD	BITS	DESCRIPTION
ENC_DEVIATION	19:0	编码器计数器和 XACTUAL 之间的最大步数偏差用于偏差报警结果标志 ENC_STATUS.deviation_warn 0= 功能关闭。

VIRTUAL_STOP_L (0x3E)

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

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
VIRTUAL_STOP_L	31:0	基于编码器或斜坡位置的虚拟停止开关. 根据带符号的比较，提出一个止损点 *virtual_stop_enc* = 1: *X_ENC* <= *VIRTUAL_STOP_L* *virtual_stop_enc* = 0: *XACTUAL* <= *VIRTUAL_STOP_L* -2^31… +(2^31)-1

VIRTUAL_STOP_R (0x3F)

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

BITFIELD	BITS	DESCRIPTION
VIRTUAL_STOP_R	31:0	基于编码器的虚拟停止开关。 根据带符号的比较，提出一个止损点 *virtual_stop_enc* = 1: *X_ENC* >= *VIRTUAL_STOP_R* *virtual_stop_enc* = 0: *XACTUAL* >= *VIRTUAL_STOP_R* -2^31… +(2^31)-1

PRELIMINARY CONFIDENTIAL

ADC_VSUPPLY_AIN (0x50)

BIT				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							

BITFIELD	BITS	DESCRIPTION
ADC_AIN	28:16	ADC_AIN 引脚上的电压值（整数）
ADC_VSUPPLY	12:0	VS 上的实际电压值（经过低通滤波器滤波），更新率：每 2048 个时钟

ADC_TEMP (0x51)

BIT				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				

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
RESERVED	28:16	
ADC_TEMP	12:0	实际温度（低通滤波器滤波），更新率：每2048个时钟

OTW_OV_VTH (0x52)

BIT				28	27	26	25	24
Field				OVERTEMPPREWARNING_VTH[12:8]				
Reset				0d2962				
Access Type				Write, Read				
BIT	23	22	21	20	19	18	17	16
Field	OVERTEMPPREWARNING_VTH[7:0]							
Reset	0d2962							
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							

BITFIELD	BITS	DESCRIPTION
OVERTEMPPREWARNING_VTH	28:16	过热警告阈值寄存器: 0x51 >= 0x52 ADC_TEMP >= OVERTEMPPREWARNING_VTH 将触发超温预警(初始化值: 0xB92 =120°C)
OVERVOLTAGE_VTH	12:0	输出 OV 的过压阈值。 默认值：38V，36 V 等于 ADC 输入的 1.125 V

MSLUT_0 (0x60)

Microstep table entries 0…31

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION
MSLUT_0	31:0	当与相应的 MSLUTSEL W 位组合时，每个位给出条目 x 和条目 x+1 之间的差异： 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 这是第一四分之一波的差分编码。 CUR_A 和 CUR_B 的起始值存储在 START_SIN 和 START_SIN90 中的 MSCNT 位置 0。 *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

MSLUT_1 (0x61)

Microstep table entries 32…63

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_1[31:24]							
Reset	0x4A9554AA							
Access Type	Write, Read							

PRELIMINARY CONFIDENTIAL

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							

BITFIELD	BITS	DESCRIPTION
MSLUT_1	31:0	当与相应的 MSLUTSEL W 位组合时，每个位给出条目 x 和条目 x+1 之间的差异： 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 这是第一四分之一波的差分编码。 CUR_A 和 CUR_B 的起始值存储在 START_SIN 和 START_SIN90 中的 MSCNT 位置 0。 *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

MSLUT_2 (0x62)

Microstep table entries 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							

PRELIMINARY CONFIDENTIAL

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							

BITFIELD	BITS	DESCRIPTION
MSLUT_2	31:0	Each bit gives the difference between entry x and entry x+1 when combined with the corresponding *MSLUTSEL W* bits: 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 This is the differential coding for the first quarter of a wave. Start values for *CUR_A* and *CUR_B* are stored for *MSCNT* position 0 in *START_SIN* and *START_SIN90*. *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

MSLUT_3 (0x63)

Microstep table entries 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							

PRELIMINARY CONFIDENTIAL

BIT	7	6	5	4	3	2	1	0
Field	MSLUT_3[7:0]							
Reset	0x10104222							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
MSLUT_3	31:0	Each bit gives the difference between entry x and entry x+1 when combined with the corresponding *MSLUTSEL W* bits: 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 This is the differential coding for the first quarter of a wave. Start values for *CUR_A* and *CUR_B* are stored for *MSCNT* position 0 in *START_SIN* and *START_SIN90*. *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

MSLUT_4 (0x64)

Microstep table entries 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							

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
MSLUT_4	31:0	Each bit gives the difference between entry x and entry x+1 when combined with the corresponding *MSLUTSEL W* bits: 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 This is the differential coding for the first quarter of a wave. Start values for *CUR_A* and *CUR_B* are stored for *MSCNT* position 0 in *START_SIN* and *START_SIN90*. *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

MSLUT_5 (0x65)

Microstep table entries 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							

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
MSLUT_5	31:0	Each bit gives the difference between entry x and entry x+1 when combined with the corresponding *MSLUTSEL W* bits: 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 This is the differential coding for the first quarter of a wave. Start values for *CUR_A* and *CUR_B* are stored for *MSCNT* position 0 in *START_SIN* and *START_SIN90*. *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

MSLUT_6 (0x66)

Microstep table entries 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							

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
MSLUT_6	31:0	Each bit gives the difference between entry x and entry x+1 when combined with the corresponding *MSLUTSEL W* bits: 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 This is the differential coding for the first quarter of a wave. Start values for *CUR_A* and *CUR_B* are stored for *MSCNT* position 0 in *START_SIN* and *START_SIN90*. *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

MSLUT_7 (0x67)

Microstep table entries 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							

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
MSLUT_7	31:0	Each bit gives the difference between entry x and entry x+1 when combined with the corresponding *MSLUTSEL W* bits: 0: *W*= %00: -1 %01: +0 %10: +1 %11: +2 1: *W*= %00: +0 %01: +1 %10: +2 %11: +3 This is the differential coding for the first quarter of a wave. Start values for *CUR_A* and *CUR_B* are stored for *MSCNT* position 0 in *START_SIN* and *START_SIN90*. *ofs31, ofs30, …, ofs01, ofs00* … *ofs255, ofs254, …, ofs225, ofs224* *reset default= sine wave table*

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	

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
X3	31:24	LUT segment 1 start The sine wave look up table can be divided into up to four segments using an individual step width control entry *Wx*. The segment borders are selected by *X1*, *X2* and *X3*. Segment 0 goes from 0 to *X1*-1. Segment 1 goes from *X1* to *X2*-1. Segment 2 goes from *X2* to *X3*-1. Segment 3 goes from *X3* to 255. For defined response the values shall satisfy: 0<*X1*<*X2*<*X3*
X2	23:16	LUT segment 1 start The sine wave look up table can be divided into up to four segments using an individual step width control entry *Wx*. The segment borders are selected by *X1*, *X2* and *X3*. Segment 0 goes from 0 to *X1*-1. Segment 1 goes from *X1* to *X2*-1. Segment 2 goes from *X2* to *X3*-1. Segment 3 goes from *X3* to 255. For defined response the values shall satisfy: 0<*X1*<*X2*<*X3*
X1	15:8	LUT segment 1 start The sine wave look up table can be divided into up to four segments using an individual step width control entry *Wx*. The segment borders are selected by *X1*, *X2* and *X3*. Segment 0 goes from 0 to *X1*-1. Segment 1 goes from *X1* to *X2*-1. Segment 2 goes from *X2* to *X3*-1. Segment 3 goes from *X3* to 255. For defined response the values shall satisfy: 0<*X1*<*X2*<*X3*
W3	7:6	LUT width select from *ofs(X3)* to *ofs255* Width control bit coding *W0*…*W3*: %00: MSLUT entry 0, 1 select: -1, +0 %01: MSLUT entry 0, 1 select: +0, +1 %10: MSLUT entry 0, 1 select: +1, +2 %11: MSLUT entry 0, 1 select: +2, +3
W2	5:4	LUT width select from *ofs(X2)* to *ofs(X3-1)* Width control bit coding *W0*…*W3*: %00: MSLUT entry 0, 1 select: -1, +0 %01: MSLUT entry 0, 1 select: +0, +1 %10: MSLUT entry 0, 1 select: +1, +2 %11: MSLUT entry 0, 1 select: +2, +3

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
W1	3:2	LUT width select from *ofs(X1)* to *ofs(X2-1)* Width control bit coding *W0…W3*: %00: MSLUT entry 0, 1 select: -1, +0 %01: MSLUT entry 0, 1 select: +0, +1 %10: MSLUT entry 0, 1 select: +1, +2 %11: MSLUT entry 0, 1 select: +2, +3
W0	1:0	LUT width select from *ofs00* to *ofs(X1-1)* Width control bit coding *W0…W3*: %00: MSLUT entry 0, 1 select: -1, +0 %01: MSLUT entry 0, 1 select: +0, +1 %10: MSLUT entry 0, 1 select: +1, +2 %11: MSLUT entry 0, 1 select: +2, +3

MSLUTSTART (0x69)

Start values are transferred to the microstep registers *CUR_A* and *CUR_B*, whenever the reference position *MSCNT*=0 is passed.

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							

BITFIELD	BITS	DESCRIPTION
OFFSET_SIN90	31:24	Signed offset for cosine wave +-127 microsteps. Adapt *START_SIN90* to match the microstep wave table at position *MSCNT*=0
START_SIN90	23:16	*START_SIN90* gives the absolute value for cosine wave microstep table entry at *MSCNT*=0 (table position 256+*OFFSET_SIN90*).
START_SIN	7:0	*START_SIN* gives the absolute value at microstep table entry 0.

PRELIMINARY CONFIDENTIAL

MSCNT (0x6A)

BIT							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							

BITFIELD	BITS	DESCRIPTION
MSCNT	9:0	微步计数器. 表示 CUR_A 在微步表中的实际位置。 CUR_B 使用 256 的偏移量（2 相电机）。 *提示：在重新初始化 MSLUTSTART 或 MSLUT 和 MSLUTSEL 之前移动到 MSCNT 为零的位置.*

MSCURACT (0x6B)

BIT							17	16
Field							CUR_A[1:0]	
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							

BITFIELD	BITS	DESCRIPTION
CUR_A	24:16	从 MSLUT 读取的电机 A 相（余弦波）的实际微步电流（未按电流缩放）
CUR_B	8:0	从 MSLUT 读取的电机 B 相（正弦波）的实际微步电流（未按电流缩放）

PRELIMINARY CONFIDENTIAL

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		–			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			

BITFIELD	BITS	DESCRIPTION	DECODE
diss2vs	31	短路保护禁用	0x0：VS 短路保护开启 0x1：禁用 VS 短路保护
diss2g	30	对地短路保护禁用	0x0：对地短路保护开启 0x1：禁用对地短路保护
dedge	29	启用双沿阶跃脉冲	0x0：禁用 0x1：在每个阶跃边缘启用阶跃脉冲以降低阶跃频率要求。
intpol	28	插值到 256 微步	0x0：无插值 0x1：实际微步分辨率 (MRES) 内插至 256 微步，以实现最平稳的电机运行（仅适用于 STEP/DIR 运行）

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
MRES	27:24	微步分辨率选择 %0000: 原生 256 微步设置。 通常将此设置与内部运动控制器一起使用。 %0001 … %1000: 128, 64, 32, 16, 8, 4, 2, FULLSTEP 降低微步分辨率，尤其是用于 STEP/DIR 操作。 分辨率给出了每个正弦四分之一波的微步输入数。 The driver automatically uses microstep positions which result in a symmetrical wave, when choosing a lower microstep resolution. step width=2^*MRES* [microsteps]	
TPFD	23:20	被动快速衰减时间 *TPFD 可以抑制电机的中频共振。*被动快速衰减时间设置控制在电桥极性改变后插入的快速衰减阶段的持续时间 N_{CLK}= 128**TPFD* %0000: Disable %0001 … %1111: 1 … 15	
vhighchm	19	高速斩波模式 当超过 VHIGH 时，该位启用切换到 chm=1 和 fd=0。 这样，可以实现更高的速度。 可以与 vhighfs=1 结合使用. 如果设置，则 TOFF 设置在高速运行期间自动加倍，以避免斩波频率加倍。	
vhighfs	18	高速整步选择 当超过 VHIGH 时，该位启用切换到全步。 仅在 45° 位置进行切换。 全步目标电流使用来自 45° 位置的微步表中的电流值。	
TBL	16:15	*TBL /* 空白时间选择 %00 … %11: 将比较器空白时间设置为 16、24、36 或 54 个时钟 *提示：大多数应用程序建议使用 %01 或 %10* *（重置默认值：OTP %01 或 %10）*	
chm	14	斩波模式	0x0：标准模式（SpreadCycle） 0x1：具有快速衰减时间的恒定关闭时间。 当达到负标称电流时，快速衰减时间也终止。 快速衰减是按时完成的。

PRELIMINARY CONFIDENTIAL

<table>
<tr><th>BITFIELD</th><th>BITS</th><th>DESCRIPTION</th><th>DECODE</th></tr>
<tr><td>disfdcc</td><td>12</td><td>快速衰减模式

chm=1:disfdcc=1 禁用电流比较器使用以终止快速衰减周期</td><td></td></tr>
<tr><td>fd3</td><td>11</td><td>TFD[3]

与 chm=1：
MSB 快速衰减时间设置 TFD</td><td></td></tr>
<tr><td>HEND_OFFSET</td><td>10:7</td><td>with chm=0:
HEND = 迟滞低值
%0000 … %1111:
Hysteresis is -3, -2, -1, 0, 1, …, 12
(该设置的 1/512 添加到电流设置) 这是用于磁滞斩波器的磁滞值。

with chm=1:
OFFSET = sine wave offset
%0000 … %1111:
Offset is -3, -2, -1, 0, 1, …, 12
这是正弦波偏移，该值的 1/512 添加到每个正弦波条目的绝对值。</td><td></td></tr>
<tr><td>HSTRT_TFD210</td><td>6:4</td><td><table><tr><td>with chm=0

HSTRT hysteresis start value added to HEND</td><td>%000 … %111:
将 1, 2, ..., 8 添加到滞后低值 HEND
（此设置的 1/512 添加到电流设置）

注意：有效 HEND+HSTRT ≤ 16。
提示：每 16 个时钟执行一次滞后递减</td></tr><tr><td>with chm=1

TFD [2..0] fast decay time setting</td><td>快速衰减时间设置
(MSB: fd3):
%0000 … %1111:
快速衰减时间设置 TFD
NCLK= 32*TFD (%0000: 仅慢衰减)</td></tr></table></td><td></td></tr>
<tr><td>TOFF</td><td>3:0</td><td>TOFF 关闭时间和驱动使能

关闭时间设置控制慢衰减阶段的持续时间
N_{CLK}= 24 + 32*TOFF
%0000: 驱动使能禁用，所有桥都关闭
%0001: 1 – 仅用于 TBL ≥ 2
%0010 … %1111: 2 … 15</td><td></td></tr>
</table>

COOLCONF (0x6D)

BIT								
								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			

BITFIELD	BITS	DESCRIPTION	DECODE
sfilt	24	StallGuard2 和 StallGuard4 过滤器启用	0x0：标准模式，StallGuard 的高时间分辨率 0x1：滤波模式，每四个全步更新一次 StallGuard 信号，仅用于补偿电机磁极公差
sgt	22:16	StallGuard2 阈值 该有符号值控制 StallGuard2 的失速输出电平，并设置最佳读数测量范围。值越低，灵敏度越高。 零是适用于大多数电机的起始值。 -64 to +63:较高的值会使 StallGuard2 的灵敏度降低，并且需要更大的扭矩来指示失速。	
seimin	15	智能电流控制的最小电流值	0x0：电流设置的 1/2 (IRUN) 0x1：电流设置的 1/4 (IRUN)
sedn	14:13	电流下降速度 %00：每 32 个 StallGuard2 值减 1 %01: 每 8 个 StallGuard2 值减 1 %10：每 2 个 StallGuard2 值减 1 %11: 每个 StallGuard2 值减1	

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
semax	11:8	智能电流控制的 StallGuard2 迟滞值 如果 StallGuard2 结果等于或高于 (SEMIN+SEMAX+1)*32，则电机电流变小以节能。%0000 ... %1111: 0 ... 15	
seup	6:5	电流上升步幅 每个测得的 StallGuard2 值的电流增量步长 %00 … %11：1、2、4、8	
semin	3:0	智能电流控制和智能电流启用的最小 StallGuard2 值 如果 StallGuard2 结果低于 SEMIN*32，则电机电流会增加以减小电机负载角. %0000: 智能电流控制 CoolStep 关闭 %0001 … %1111: 1 … 15	

DCCTRL (0x6E)

DcStep (DC) 自动换相配置寄存器（通过引脚 DCEN 或通过 VDCMIN 启用）

提示：使用更高的微步分辨率或插值运算，DcStep 可提供更好的 StallGuard 信号。 如果激活了 vhighfs，DC_SG 在 VHIGH 以上也可用。 为获得最佳效果，还要设置 vhighchm。

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

BITFIELD	BITS	DESCRIPTION
DC_SG	23:16	在 DcStep 模式下使用 DcStep StallGuard2 进行失步检测的最大 PWM 开启时间。 (DC_SG * 16/fCLK) 设置略高于 DC_TIME/16 0=禁用
DC_TIME	9:0	PWM 换流时间上限 (DC_TIME * 1/fCLK)。设置略高于有效空白时间 TBL。

PRELIMINARY CONFIDENTIAL

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							

BITFIELD	BITS	DESCRIPTION	DECODE
stst	31	停顿指示 该标志表示电机在每种操作模式下都处于静止状态。 这发生在最后一步脉冲之后 2^20 个时钟。	
olb	30	B 相开路负载指示	0x0: 正常运行 0x1: B相检测到开路状态. *提示:* 这只是一个提供信息的标志。 驱动器对此不采取任何行动。 快速运动和静止时可能会发生错误检测。 仅在慢动作过程中检查。
ola	29	A 相开路负载指示	0x0: 正常运行 0x1: A相检测到开路状态. *提示:* 这只是一个提供信息的标志。 驱动器对此不采取任何行动。 快速运动和静止时可能会发生错误检测。 仅在慢动作过程中检查。
s2gb	28	B相对地短路指示	0x0: 正常运行 0x1: 在 B 相检测到对 GND 短路。驱动器被禁用。这些标志保持活动状态，直到驱动程序被软件 (TOFF=0) 或 ENN 输入禁用。
s2ga	27	A相对地短路指示	0x0: 正常运行 0x1: 在 A 相检测到对 GND 短路。驱动器被禁用。这些标志保持活动状态，直到驱动程序被软件 (TOFF=0) 或 ENN 输入禁用。

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
otpw	26	过热预警标志	0x0: 正常运行 0x1: 超温预警阈值已超过。 过热预警标志对两个桥都是通用的。
ot	25	过温标志	0x0: 正常运行 0x1: 已达到过温限制。 驱动器被禁用，直到 otpw 由于 IC 冷却而被清除。 过热标志对于两个电桥都是通用的。
stallguard	24	StallGuard2/StallGuard4 状态	0x0: 正常运行 0x1: StallGuard2（在 SpreadCycle 操作中）分别检测到电机失速。 通过 StallGuard4（在 StealthChop 操作中）或 DcStep 失速（在 DcStep 模式下）。
CS_ACTUAL	20:16	实际电机电流/智能能量电流 实际电流控制定标，用于监控通过寄存器 COOLCONF 设置控制的智能能源电流定标，或用于监控自动电流定标功能	
fsactive	15	激活整步模式指示	0x0: 微步模式被激活 0x1: 指示驱动程序已切换到由斩波器模式设置和速度阈值定义的整步模式
stealth	14	StealthChop 指示器	0x0: stealthChop 没有使能 0x1: 驱动器工作在 StealthChop 模式运行
s2vsb	13	B 相电源短路指示器	0x0: 无报错 0x1: 在 B 相检测到电源短路。驱动器被禁用。 这些标志保持活动状态，直到驱动程序被软件 (TOFF=0) 或 ENN 输入禁用。
s2vsa	12	A 相电源短路指示器	0x0: 无报错 0x1: 在 A相检测到电源短路。驱动器被禁用。 这些标志保持活动状态，直到驱动程序被软件 (TOFF=0) 或 ENN 输入禁用。

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
SG_RESULT	9:0	StallGuard2 结果分别为 StallGuard4 结果（取决于实际斩波器模式）和。 用于电机温度检测的SpreadCycle 静止的线圈A 的PWM 开启时间。 机械负载测量: StallGuard2/4 结果提供了一种测量机械电机负载的方法。 较高的值意味着较低的机械负载。对于 StallGuard2，值为 0 表示最高负载。 使用最佳 SGT 设置，这是电机失速的一个指标。失速检测将 SG_RESULT 与 0 进行比较以检测失速。*SG_RESULT 用作 CoolStep 操作的基础，通过将其与可编程上限和下限进行比较。*它不适用于 StealthChop 模式。 StallGuard2 与微步操作或 DcStep 配合使用效果最佳。 SpreadCycle 模式下的温度测量： 静止时，无法获得 StallGuard2 结果。*SG_RESULT 显示的是电机线圈 A 的斩波器接通时间。* 在某个电流设置下将电机移动到确定的微步位置，通过读取斩波器的接通时间来粗略估计电机温度。 随着电机升温，其线圈电阻上升，斩波器接通时间增加。有关 StallGuard4 的详细信息，请参阅 SG4_RESULT。	

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							

PRELIMINARY CONFIDENTIAL

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

BITFIELD	BITS	DESCRIPTION	DECODE
PWM_LIM	31:28	开启时PWM自动缩放幅度限制 从 SpreadCycle 切换回 StealthChop 时的 PWM_SCALE_AUTO 限制。该值定义切换回时自动电流控制的位 7 至 4 的上限。 可以将其设置为在模式更改回 StealthChop 期间减少电流的抖动。 它不会限制 PWM_GRAD 或 PWM_GRAD_AUTO 偏移量。 （默认 = 12）	
PWM_REG	27:24	调节回路梯度 当使用 pwm_autoscale=1 时，用户定义的最大 PWM 幅度变化每半波。 （1…15）： 1： 0.5增量（最慢调节） 2: 1 增量 3： 1.5 增量 4： 2 增量（重置默认值）） … 8: 4 增量 ... 15：7.5 增量（最快调节）	
pwm_dis_reg_stst	23	1= 当电机静止且电流减小（小于 IRUN）时禁用电流调节。 此选项可消除静止期间的任何调节噪音。	
pwm_meas_sd_enable	22	默认=0； 1：在低端使用缓慢衰减相位来测量电机电流以降低电流下限。	
FREEWHEEL	21:20	允许不同的静止模式 电机电流设置为零 (I_HOLD=0) 时的静止选项。 %00：正常运行 %01: 自由运转 %10: 线圈使用 LS 驱动器短路 %11: 线圈使用 HS 驱动器短路	

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
pwm_autograd	19	PWM自动梯度适应	0x0: PWM_GRAD 的固定值（PWM_GRAD_AUTO = PWM_GRAD） 0x1: 自动调整（仅当 pwm_autoscale=1 时）（重置默认值）*PWM_GRAD_AUTO 在 pwm_autograd=0 时使用 PWM_GRAD 进行初始化，并在运动过程中自动优化。* 前提条件 1. *PWM_OFS_AUTO 已自动初始化.这需要以等于IRun的停止电流持续>130ms,目的是* a) 检测到停顿 b) 在为IRUN电流下等待> 128斩波周期和 c) 调节 *PWM_OFS_AUTO* 使得 -1 < *PWM_SCALE_AUTO* < 1 2. 电机运行且 1.5 * PWM_OFS_AUTO * (IRUN+1)/32 < PWM_SCALE_SUM < 4* PWM_OFS_AUTO * (IRUN+1)/32 和 PWM_SCALE_SUM < 255。 调节 PWM_GRAD_AUTO 所需时间 每变化 +/-1 大约 8 个全步。 还可以使用降低的斩波频率来调整 PWM_OFS_AUTO。
pwm_autoscale	18	PWM 自动幅度缩放	0x0: 用户定义的前馈 PWM 幅度。电流设置 IRUN 和 IHOLD 没有任何影响！ 产生的 PWM 幅度（限制为 0…255）为： *PWM_OFS * ((CS_ACTUAL+1) / 32)* *+ PWM_GRAD* * 256 / *TSTEP* 0x1: 启用自动电流控制（重置默认值）
PWM_FREQ	17:16	PWM 频率选择: %00: f_{PWM}=2/1024 f_{CLK} *(Reset default)* %01: f_{PWM}=2/683 f_{CLK} %10: f_{PWM}=2/512 f_{CLK} %11: f_{PWM}=2/410 f_{CLK}	
PWM_GRAD	15:8	PWM振幅的速度相关梯度: *PWM_GRAD* * 256 / *TSTEP* 该值被添加到 PWM_OFS 以补偿与速度相关的电机反电动势. 使用 PWM_GRAD 作为自动缩放的初始值，以加快自动调整过程。为此，请将 PWM_GRAD 设置为确定的、特定于应用程序的值，其中 pwm_autoscale=0。 只有在之后，设置 pwm_autoscale=1。 完成后启用 StealthChop。 *提示：* *初始调整后，可以从 PWM_GRAD_AUTO 中读出所需的初始值。*	

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION	DECODE
PWM_OFS	7:0	用户定义的 PWM 幅度偏移 (0-255) 与静止状态下的全电机电流 (CS_ACTUAL=31) 相关。（重置默认值=30） 使用 PWM_OFS 作为自动缩放的初始值，以加快自动调谐过程。 为此，请将 PWM_OFS 设置为确定的、特定于应用程序的值，其中 pwm_autoscale=0。只有在之后，设置 pwm_autoscale=1。 完成后启StealthChop。 *PWM_OFS = 0 将禁止将电机电流按比例缩小至低于电机特定的下测量阈值.* 此设置应仅在特定条件下使用，即当电源电压可以上下波动两倍或更多时。它可以防止电机超出调节范围，但也可以防止功率低于调节限制。 *PWM_OFS > 0 允许自动调整到低 PWM 占空比，甚至低于较低的调节阈值。* 这允许基于实际（保持）电流标度（寄存器 IHOLD_IRUN）进行低（静止）电流设置。	

PWM_SCALE (0x71)

StealthChop 调幅器的结果。 这些值可用于监控自动 PWM 幅度缩放（255=最大电压）.

BIT								16
Field								PWM_SCALE_AUTO[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							

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
PWM_SCALE_AUTO	24:16	
PWM_SCALE_SUM	9:0	Bits: 9...0: [0...1023]PWM_SCALE_SUM: 实际 PWM 占空比。 该值用于缩放从正弦波表中读取的 CUR_A 和 CUR_B 值。1023：最大占空比。 该值扩展了两位 [1,0]，以实现更高的占空比读数精度。 位 9..2 对应于其他 PWM 占空比相关寄存器中的 8 位值。

PWM_AUTO (0x72)

可以读出这些自动生成的值以确定 PWM_GRAD 和 PWM_OFS 的默认/上电设置。

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							

BITFIELD	BITS	DESCRIPTION
PWM_GRAD_AUTO	23:16	自动确定的梯度值
PWM_OFS_AUTO	7:0	自动确定的偏移值

SG4_THRS (0x74)

BIT	7	6	5	4	3	2	1	0
Field	SG4_THRS[7:0]							
Reset	0x0							
Access Type	Write, Read							

BITFIELD	BITS	DESCRIPTION
sg_angle_offset	9	1: 当从 StealthChop 切换到通过 TPWMTHRS 控制的 SpreadCycle 时，基于 StallGuard4 的自动相移补偿
sg4_filt_en	8	1: 使能SG4 滤波, 0: 禁用 SG4 滤波
SG4_THRS	7:0	失速检测阈值. StallGuard4 值 SG4_RESULT 与该阈值的两倍进行比较。当SG_RESULT ≤ SG4_THRS*2时会触发失速

PRELIMINARY CONFIDENTIAL

SG4_RESULT (0x75)

BIT							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							

BITFIELD	BITS	DESCRIPTION
SG4_RESULT	9:0	仅限 StallGuard4 的 StallGuard 结果. *SG4_RESULT 随每个完整步更新，独立于 TCOOLTHRS 和 SG4THRS*。 较高的值表示较低的电机负载和较大的扭矩裕量。 仅适用于 StealthChop 模式。 第 9 位和第 0 位将始终显示 0。缩放为 10 位是为了与 StallGuard2 兼容。

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							

BITFIELD	BITS	DESCRIPTION
SG4_IND_3	31:24	当SG4_filt_en = 1: Displays SG4 measurement 3 used as filter input

PRELIMINARY CONFIDENTIAL

BITFIELD	BITS	DESCRIPTION
SG4_IND_2	23:16	When SG4_filt_en = 1: Displays SG4 measurement 2 used as filter input
SG4_IND_1	15:8	When SG4_filt_en = 1: Displays SG4 measurement 1 used as filter input
SG4_IND_0	7:0	displays SG4 measurement When SG4_filt_en = 1: Displays SG4 measurement 0 used as filter input

PRELIMINARY CONFIDENTIAL

典型应用电路

标准应用电路

标准应用电路使用最少的附加组件。使用低 ESR 电容对电源进行滤波。电容需要应对由斩波器操作引起的电流纹波。建议在驱动器附近使用 最小100µF 的电容，以获得最佳性能。电源电容器中的电流纹波还取决于电源内部电阻和电缆长度。 VCC_IO 必须由外部电源供电，例如低压降 3.3V 稳压器。

将所有滤波电容尽可能靠近相关 IC 引脚放置。为所有 GND 连接使用一个固定的公共 GND。将 VDD1V8 滤波电容直接连接到 VDD1V8 引脚。VS 滤波推荐使用低 ESR 电解电容。

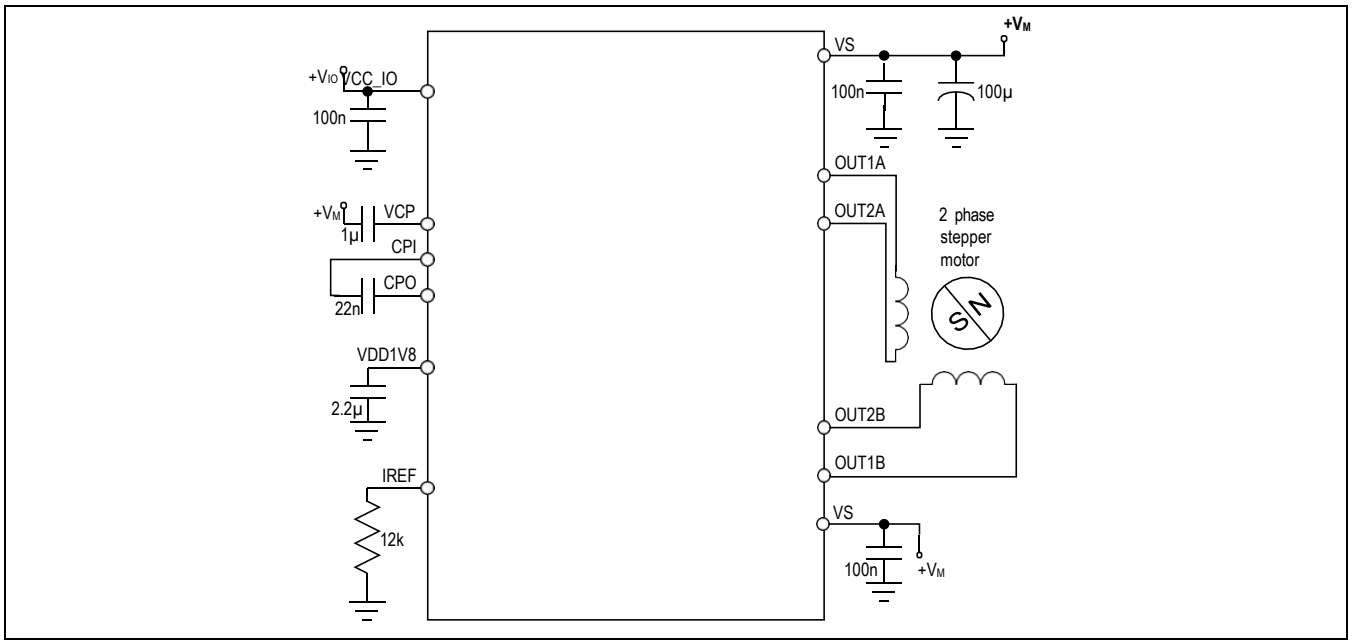


图 37. TMC5240 标准应用电路

大的电机电流

当在高电机电流下运行时，由于 MOSFET 开关导通电阻导致的驱动器功率耗散会显着是芯片温度升高。如果在增加的占空比下运行，这种功耗也会使 PCB 板部分温度升高。这又导致驱动器温度进一步升高。温度升高约 100°C 会使 MOSFET 电阻增加约 50%。这是 MOSFET 开关的典型特性。因此，在高占空比、高负载条件下，必须仔细考虑热特性，特别是在要支持增加的环境温度时。另请参阅热特性和布局示例。

根据经验法则，PCB设计的热特性可能在1.5A或以上的RMS电机电流时变得至关重要，从而导致时间延长。请记住，电阻功耗随着电机电流的平方而增加。 另一方面，这意味着电机电流的小幅降低可显着节省散热和能源。

PRELIMINARY CONFIDENTIAL

典型应用电路（待续）

驱动器保护和 EME 电路

一些应用必须应对由电机运行或外部影响引起的 ESD 事件. 尽管驱动器芯片内有 ESD 电路，但在运行期间发生的 ESD 事件可能会导致电机驱动器复位甚至损坏，具体取决于它们的能量。尤其是塑料外壳和皮带传动系统往往会引起数 kV 的 ESD 事件。 最好的做法是通过将所有导电部件（尤其是电机本身）连接到 PCB 地线或使用导电塑料部件来避免 ESD 事件。此外，驱动器可以在一定程度上受到保护，防止 ESD 事件或带电插拔电机，这也会导致高电压和高电流进入电机连接器端子。一个简单的方案是在驱动器输出端使用电容来降低 ESD 事件引起的 dV/dt。较大的电容器将带来更多关于 ESD 抑制的好处，但会在每个斩波周期中产生额外的电流，从而增加驱动器功耗，尤其是在高电源电压下。 显示的值是示例值——它们可能在 100pF 和 1nF 之间变化。 这些电容器还可以抑制从应用 PCB 电路的数字部分注入的高频噪声，从而减少电磁辐射。 更复杂的方案使用 LC 滤波器将驱动器输出与电机连接器分离。线圈端子之间的压敏电阻消除了带电插拔引起的线圈过电压。 可选地通过压敏电阻保护所有输出免受 ESD 电压的影响。

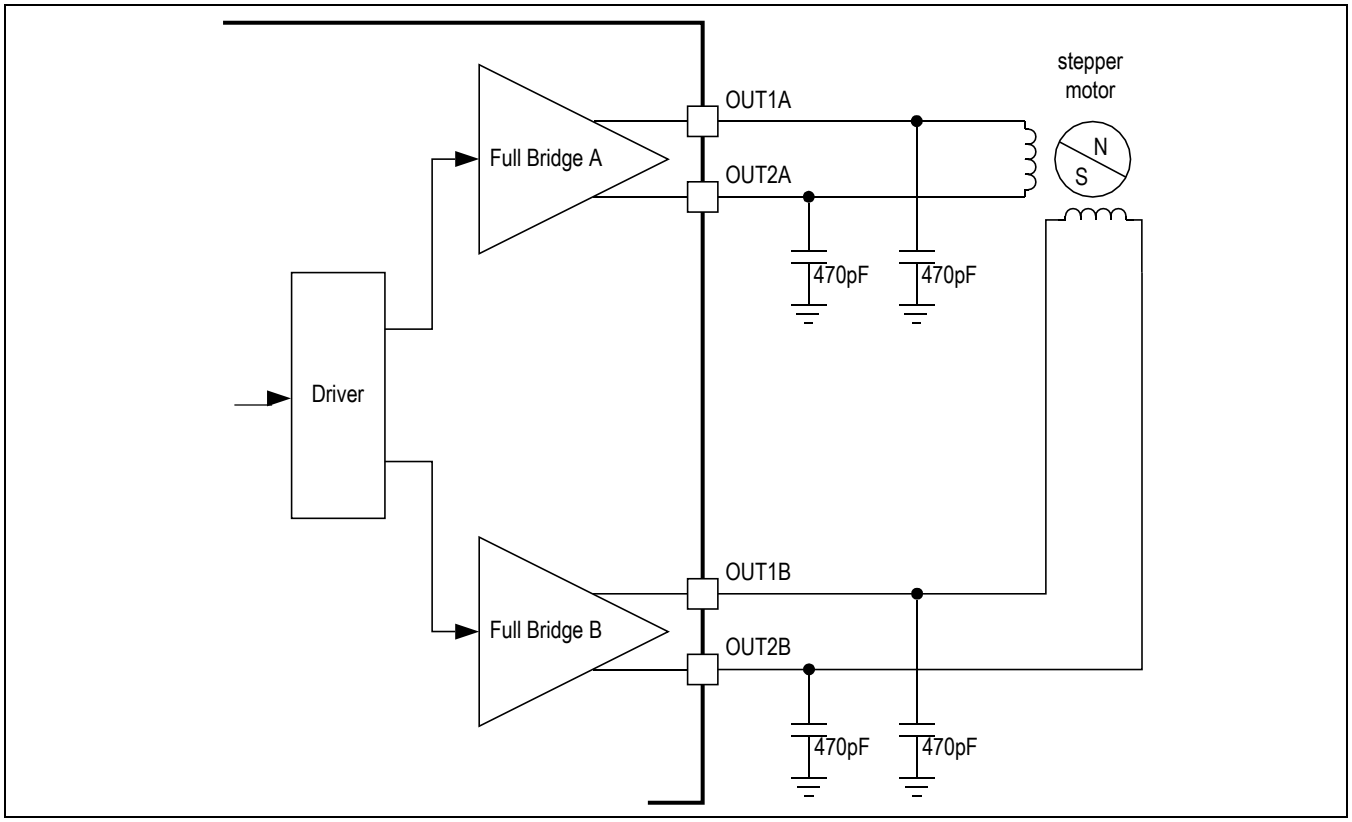


图 38. 简单的 ESD 增强功能

PRELIMINARY CONFIDENTIAL

典型应用电路（待续）

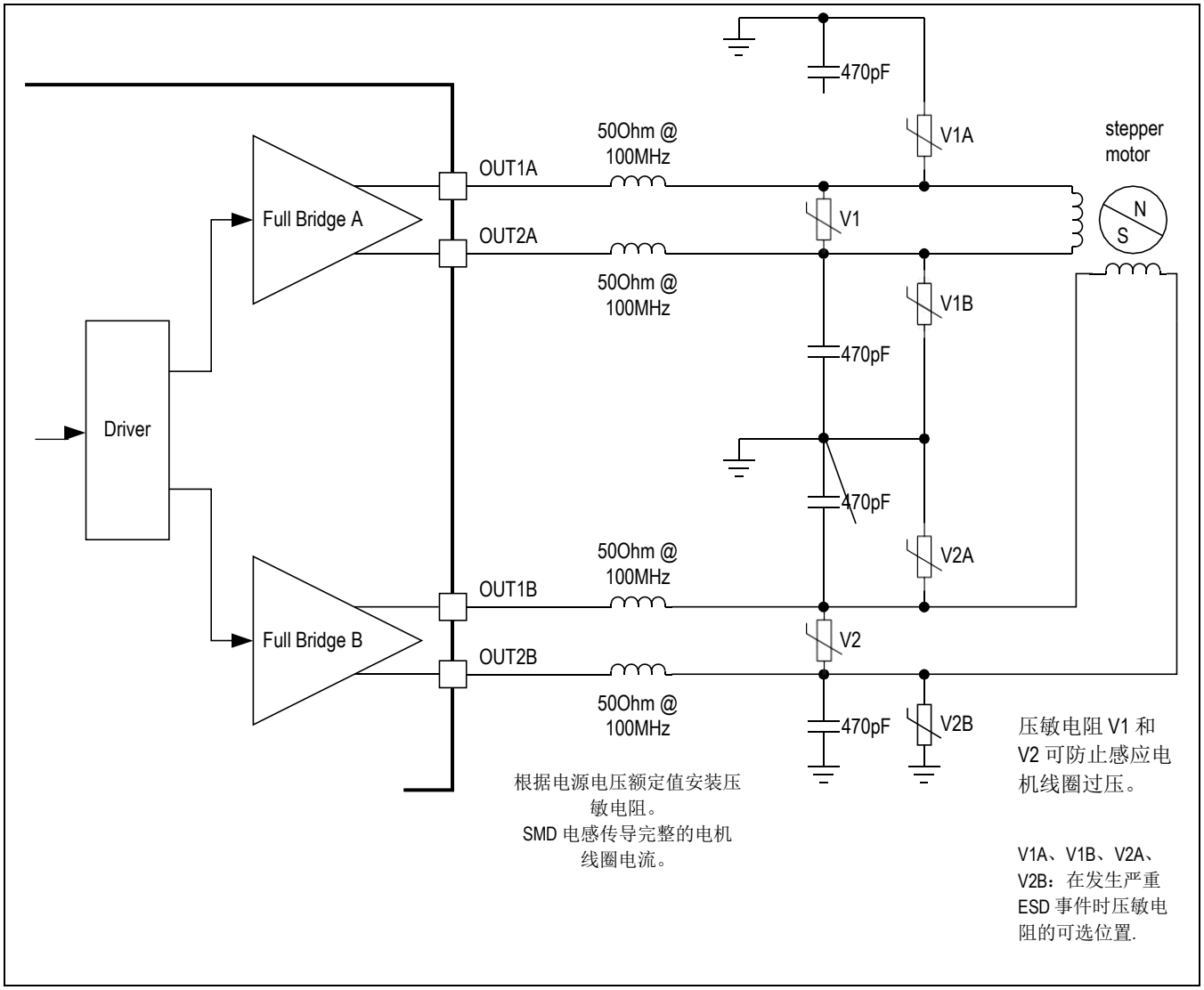


图 39. 精细的电机输出保护

PRELIMINARY CONFIDENTIAL

Ordering Information

PART NUMBER	TEMPERATURE RANGE	PIN-PACKAGE
TMC5240ATJ+	-40°C to +125°C	32 TQFN - 5x5mm
TMC5240ATJ+T	-40°C to +125°C	32 TQFN - 5x5mm
TMC5240AUU+*	-40°C to +125°C	38 TSSOP-EP 4.4x9.7mm
TMC5240AUU+T*	-40°C to +125°C	38 TSSOP-EP 4.4x9.7mm
TMC5240-EVAL	Evaluation board for TMC5240ATJ+	85mmx55mm
TMC5240-EVAL-KIT	Evaluation board kit for TMC5240ATJ+ ncluding Landungsbruecke interface and Eselsbruecke connector board	ca. 85mmx150mm
TMC5240-BOB*	Breakout board for TMC5240AUU+	ca. 1"x1"

** Future product—contact factory for availability.*

+ Denotes a lead(Pb)-free/RoHS-compliant package.

T Denotes tape-and-reel.

PRELIMINARY CONFIDENTIAL

Revision History

REVISION NUMBER	REVISION DATE	DESCRIPTION	PAGES CHANGED
0	02/22	Initial Release	—

扫描如需更多支持

PRELIMINARY CONFIDENTIAL