

Click [here](#) to ask an associate for production status of specific part numbers.

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、3.0A_{MAX} H桥以及非耗散集成电流传感器 (ICS)。

TRINAMIC 精密的stealthChop2™ 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学习曲线降至最低，同时提供一流的性能。

H 桥 FET 具有非常低的阻抗，从而导致高驱动效率和产生的热量最少。典型的总 R_{on} (高侧 + 低侧) 为 0.23Ω。

每个 H 桥的最大输出电流为 $I_{MAX}=5.0A_{MAX}$ ，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 $I_{RMS}=2.1A_{RMS}$ 。

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 $I_{FS}= 3.0A$ ，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 消除了笨重的外部功率电阻器，从而显著节省空间和功耗。

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、估计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 封装和经过热优化的 TSSOP38 9.7x4.4mm 外露焊盘。

应用

- Textile, Sewing Machines, Knitting Machines
- Lab & Factory Automation
- 3D Printers, ID printers / Card printers
- Liquid Handling, Medical
- Office Automation and Paper Handling
- POS, Gambling, Massage Chairs,
- ATM, Cash Recycler, Bill Validators, Cash Machines
- CCTV, Security
- Pumps and Valve Control
- HelioStat and Antenna Positioning

优点和特点

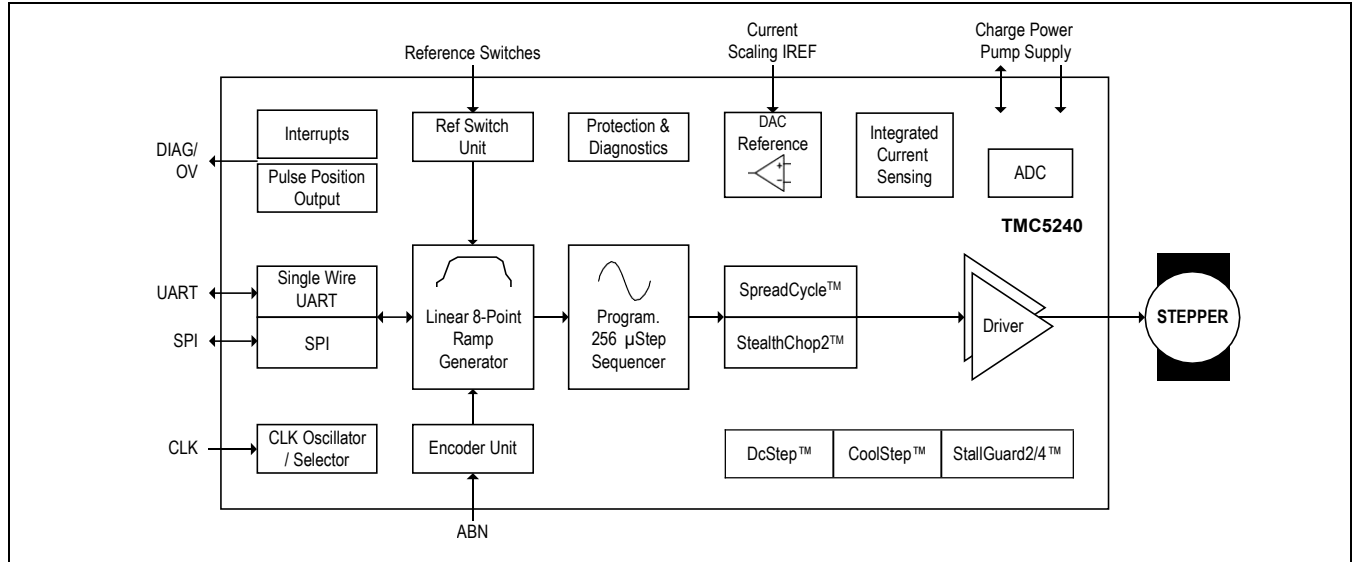
- 电压范围 4.5... 36V DC
- 低 R_{dson} (HS+LS): 典型值为 230 mΩ ($T_A=25C$)
- 每个 H 桥的额定电流 (25C 时的典型值):
 - $I_{MAX}=5.0A$ (bridge peak current)
 - $I_{RMS}=2.1A_{RMS}$ (3A sine wave peak)
- 完全集成的无损电流检测 (ICS)
- EightPoint™ 最小抖动的运动控制器
- SPI & 单线 UART
- 编码器接口和 2x 参考开关输入
- 最高分辨率 256 微步/整步
- 灵活的波形表 & 相序以匹配电机
- StealthChop2™ 静音电机操作
- SpreadCycle™ 高动态电机控制斩波器
- StealthChop 和 SpreadCycle 的无抖动组合
- 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

Ordering Information appears at end of data sheet.

© 2022 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.

One Analog Way, Wilmington, MA 01887 U.S.A. | Tel: 781.329.4700 | © 2022 Analog Devices, Inc. All rights reserved.

简化框图



PRELIMINARY CONFIDENTIAL

TABLE OF CONTENTS

General Description.....	1
Applications.....	1
Benefits and Features.....	1
Simplified Block Diagram.....	2
Absolute Maximum Ratings.....	9
Package Information.....	9
TQFN32 5x5.....	9
TSSOP38 9.7x4.4 EP.....	9
Electrical Characteristics.....	9
Pin Configurations.....	13
TMC5240 TQFN Pin Configuration.....	13
TMC5240 TSSOP Pin Configuration.....	14
Pin Description.....	15
Functional Diagrams.....	18
TMC5240.....	18
Detailed Description.....	19
Principles of Operation.....	19
Full Featured Motion Controller & Driver.....	19
Key Concepts.....	20
Control Interfaces.....	20
Integrated Motion Controller.....	20
Automatic Standstill Power Down.....	20
StealthChop2 & SpreadCycle Driver.....	21
StallGuard – Mechanical Load Sensing.....	21
CoolStep – Load Adaptive Current Control.....	21
DcStep – Load Dependent Speed Control.....	22
Encoder Interface.....	22
SPI Interface.....	22
SPI Datagram Structure.....	22
Selection of Write / Read (WRITE_notREAD).....	23
SPI Status Bits Transferred with Each Datagram Read Back.....	23
Data Alignment.....	24
SPI Signals.....	24
SPI Timing.....	24
UART Single Wire Interface.....	24
Datagram Structure.....	25
Write Access.....	25
Read Access.....	25
CRC Calculation.....	26

TABLE OF CONTENTS (CONTINUED)

C-Code Example for CRC calculation.....	26
UART Signals.....	27
Addressing Multiple Slaves.....	27
StealthChop2.....	28
Automatic Tuning.....	29
StealthChop Options.....	31
StealthChop Current Regulator.....	31
Lower Current Limit.....	33
Velocity Based Scaling.....	34
Combining StealthChop and SpreadCycle.....	35
Flags in StealthChop.....	37
Open Load Flags.....	37
PWM_SCALE_SUM Informs about the Motor State.....	37
Freewheeling and Passive Braking.....	37
Parameters related to StealthChop.....	38
SpreadCycle and Classic Chopper.....	39
SpreadCycle Chopper.....	40
Classic Constant Off Time Chopper.....	42
Integrated Current Sense (ICS).....	43
Setting the Motor Current.....	43
Setting the Full-Scale Current Range.....	44
Velocity Based Mode Control.....	44
Ramp Generator.....	46
Real Word Unit Conversion.....	47
Motion Profiles.....	47
Ramp Mode.....	47
Start and Stop Velocity.....	49
Eight Point Ramp.....	49
Velocity Mode.....	52
Early Ramp Termination.....	52
Application Example: Joystick Control.....	53
Velocity Thresholds.....	53
Reference Switches.....	54
Virtual Reference Switches.....	56
External STEP/DIR Driver.....	56
Position Compare Functions [adapt].....	56
StallGuard2 Load Measurement.....	57
Tuning StallGuard2 Threshold SGT.....	58
Variable Velocity Limits TCOOLTHRS and THIGH.....	59

TABLE OF CONTENTS (CONTINUED)

Small Motors with High Torque Ripple and Resonance.....	59
Temperature Dependence of Motor Coil Resistance.....	59
Accuracy and Reproducibility of StallGuard2 Measurement.....	60
StallGuard2 Update Rate and Filter.....	60
Detecting a Motor Stall.....	60
Homing with StallGuard.....	60
Limits of StallGuard2 Operation.....	60
StallGuard4 Load Measurement.....	60
StallGuard4 vs. StallGuard2.....	62
Tuning StallGuard4.....	62
StallGuard4 Update Rate.....	62
Detecting a Motor Stall.....	63
Limits of StallGuard4 Operation.....	63
CoolStep Operation.....	63
Setting up for CoolStep.....	63
Tuning CoolStep.....	64
Response Time.....	65
Low Velocity and Standby Operation.....	65
Diagnostic Outputs.....	65
DcStep.....	66
Designing-In DcStep.....	66
DcStep Integration with the Motion Controller.....	67
Stall Detection in DcStep Mode.....	68
Measuring Actual Motor Velocity in DcStep Operation.....	69
Sine Wave Lookup Table.....	69
Microstep Table.....	69
ABN Incremental Encoder Interface.....	71
Setting the Encoder to Match Motor Resolution.....	73
Reset, Disable/Stop and Power Down.....	73
Emergency Stop.....	73
External Reset and Sleep Mode.....	73
Protections and Driver Diagnostics.....	74
Over Current Protection.....	74
Thermal Protection and Shutdown.....	74
Temperature Measurement.....	74
Chip Temperature Measurement.....	75
Motor Temperature Measurement.....	75
Overvoltage Protection and Pin OV.....	75
Short to GND Protection.....	76

TABLE OF CONTENTS (CONTINUED)

Open Load Diagnostics.....	76
Under Voltage Lockout Protection.....	76
ESD Protection.....	76
Clock Oscillator and Clock Input.....	77
Using the Internal Clock.....	77
Using an External Clock.....	77
General Register Mapping and Register Information.....	77
Register Map.....	79
TMC5240.....	79
Register Details.....	84
Typical Application Circuits.....	142
Standard Application Circuit.....	142
High Motor Current.....	142
Driver Protection and EME Circuitry.....	143
Ordering Information.....	145
Revision History.....	146

PRELIMINARY CONFIDENTIAL

LIST OF FIGURES

Figure 1. TMC5240 block diagram.....	18
Figure 2. TMC5240 with typical external components.....	19
Figure 3. Automatic motor current control at standstill and ramp up.....	21
Figure 4. SPI Timing Diagram.....	24
Figure 5. UART daisy chaining example.....	28
Figure 6. Motor coil sine wave current with StealthChop (measured with current probe).....	29
Figure 7. StealthChop2 automatic tuning procedure.....	30
Figure 8. StealthChop2: good setting for PWM_REG.....	32
Figure 9. StealthChop2: too small setting for PWM_REG during AT#2.....	32
Figure 10. Successfully determined PWM_GRAD(_AUTO) and PWM_OFS(_AUTO).....	33
Figure 11. Velocity based PWM scaling (pwm_autoscale=0).....	35
Figure 12. TPWMTHRS for optional switching to SpreadCycle.....	36
Figure 13. Typical chopper decay phases.....	39
Figure 14. SpreadCycle chopper scheme showing coil current during a chopper cycle.....	41
Figure 15. Classic const. off time chopper with offset showing coil current.....	42
Figure 16. Zero crossing with classic chopper and correction using sine wave offset.....	43
Figure 17. Choice of velocity dependent modes.....	45
Figure 18. Ramp generator velocity trace showing consequent move in negative direction.....	48
Figure 19. Illustration of optimized motor torque usage with the ramp generator.....	49
Figure 20. 8-point-ramp with VMAX not reached due to too low displacement.....	50
Figure 21. V2 not reached and no AMAX & DMAX phase due to low displacement.....	50
Figure 22. V1 not reached due to low displacement.....	51
Figure 23. TVMAX not kept due to low displacement.....	51
Figure 24. 8-Point Ramp examples with on the fly target position change.....	52
Figure 25. Ramp generator velocity dependent motor control.....	54
Figure 26. Using reference switches (example).....	55
Figure 27. Function principle of StallGuard2.....	57
Figure 28. Example: optimum SGT setting and StallGuard2 reading with an example motor.....	59
Figure 29. StallGuard4 mode of operation.....	61
Figure 30. CoolStep adapts motor current to the load.....	64
Figure 31. DcStep extended application operation area.....	67
Figure 32. Velocity profile with impact by overload situation.....	68
Figure 33. LUT programming example.....	70
Figure 34. Shifting the cosine wave via OFFSET_SIN90.....	71
Figure 35. Outline of ABN signals of an incremental encoder.....	72
Figure 36. Brake chopper circuit example.....	76
Figure 37. TMC5240 standard application circuit.....	142
Figure 38. Simple ESD enhancement.....	143
Figure 39. Elaborate motor output protection.....	144

LIST OF TABLES

Table 1. SPI Datagram Structure.....	22
Table 2. SPI Read/Write Example Flow.....	23
Table 3. SPI_STATUS – status flags transmitted with each SPI access in bits 39 to 32.....	23
Table 4. UART Write access datagram structure.....	25
Table 5. UART Read access request datagram structure.....	25
Table 6. UART Read access reply datagram structure.....	26
Table 7. TMC5240 UART Interface Signals.....	27
Table 8. Constraints and requirements for StealthChop autotuning AT#1 and AT#2.....	29
Table 9. Choice of PWM frequency for StealthChop (bold = recommended).....	31
Table 10. Parameters related to StealthChop.....	38
Table 11. Parameters used for controlling SpreadCycle and Classic Chopper.....	40
Table 12. SpreadCycle mode parameters.....	41
Table 13. Parameters controlling constant off-time chopper mode.....	43
Table 14. Parameters controlling the motor current.....	43
Table 15. IFS full scale range settings (example for RREF = 12K Ω).....	44
Table 16. StallGuard2 related parameters.....	57
Table 17. StallGuard4 related parameters.....	61
Table 18. CoolStep critical parameters.....	63
Table 19. CoolStep additional parameters and status information.....	64
Table 20. Encoder example settings for a 200 fullstep motor with 256 microsteps.....	73
Table 21. Overview of Register Map.....	77

绝对最大额定值

VS to GND.....	V to 41V	IREF, AIN to GND.....	V to min(2.2, VDD+0.3)V
VDD to GND.....	V to min(2.2, VS+0.3)V	VCC_IO to GND.....	V to 5.5V
AGND to GND.....	V to +0.3V	Logic input/output voltage to GND.....	V to VCC_IO+0.3V
OUT1A, OUT2A, OUT1B, OUT2B.....	V to VS+0.3V	OV to GND.....	V to 6V
VCP to GND.....	VS-0.3V to min(44, VS+6)V	工作温度范围.....	-40°C to 125°C
CPO to GND.....	VS-0.3V to min(44, VS+6)V	结温.....	+160°C
CPI to GND.....	V to min(41, VS+0.3)V	储存温度范围.....	-65°C to +150°C
SLEEPN to GND.....	V to VS+0.3V	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。这只是额定最大值，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

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

TSSOP38 9.7x4.4 EP

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

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.

Electrical Characteristics

(VS = 4.5V to 36V, RREF = from 12k Ω to 24k Ω , Typical Values assume TA=25°C and VS=24V, Limits are 100% tested at TA = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
POWER SUPPLY						
Supply Voltage Range	VS		4.5		36	V
Sleep Mode Current consumption	IVS	V(SLEEPN)=0		4	18	μ A

Electrical Characteristics (continued)

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ, Typical Values assume TA=25°C and VS=24V, Limits are 100% tested at TA = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Quiescent Current Consumption	IVS	V(SLEEPN)=1, V(DRV_ENN)=1		2.3	4	mA
1.8V regulator Output Voltage	VVDD	VS=4.5V		1.8		V
VDD Current Limit	IV18LIM		20			mA
Charge Pump Voltage	VCP			VS+2.7		V
Logic I/O Supply Voltage Range	VCC_IO		2.2		5.5	V
Sleep Mode Current Consumption	IVCC	V(SLEEPN)=0		5	10	uA
Quiescent Current consumption	IVCC	V(SLEEPN)=1		35	60	uA
LOGIC LEVEL INPUTS-OUTPUTS						
Input Voltage Level - High	VIH			0.7 x VCC_IO		V
Input Voltage Level - Low	VIL				0.3 x VCC_IO	V
Input Hysteresis	VHYS			0.15 x VCC_IO		V
Pull-up / Pull-down Resistance	RpD	to GND or to VCC_IO	60	100	140	KΩ
Input Leakage	I _{nLeak}	Inputs without pull-up/pull-down resistance	-1		1	uA
Output Logic-low Voltage	VOL	ILOAD=5mA			0.4	V
Push-pull Output logic-high Voltage	VOH	ILOAD=5mA		VCC_IO - 400mV		
Open Drain Output logic high Leakage Current	IOH	V(PIN)=5.5V	-1		1	uA
SLEEPN Voltage Level High	VIHSLEEPN		0.9			V
SLEEPN Voltage Level Low	VILSLEEPN				0.6	V
SLEEPN Pull-down Input Resistance	RPDSLEEPN		0.8	1.5		MΩ
OUTPUT SPECIFICATIONS						
Output ON-Resistance Low Side	RonLS	Full scale bits = 10		0.11	0.2	Ω
		Full scale bits = 01		0.15	0.28	
Output ON-Resistance Low Side	RonLS	Full scale bits = 00		0.28	0.54	Ω
Output ON-Resistance High Side	RonHS			0.12	0.22	Ω
Output Leakage	ILEAK		-5		5	μA

Electrical Characteristics (continued)

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ, Typical Values assume TA=25°C and VS=24V, Limits are 100% tested at TA = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Slew Rate	SR	Slew-rate bits = 00		100		V/us
		Slew-rate bits = 01		200		
		Slew-rate bits = 10		400		
		Slew-rate bits = 11		800		
PROTECTION CIRCUITS						
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
UVLO Threshold on VS	UVLO	VS falling	3.75	3.9	4.05	V
UVLO Threshold on VS Hysteris	UVLOHYS			0.12		V
UVLO Threshold on VCC_IO	UVLO	VCC_IO falling	0.9	1.5	1.95	
VCC_IO UVLO Hysteresis	UVLOVCCCH			100		mV
热保护阈值温度	TSD			165		°C
热保护温度滞后				20		°C
CURRENT REGULATION						
IREF pin Resistor range	RREF		12		60	kΩ
IREF Output Voltage	VREF		0.882	0.9	0.918	V
Full Scale Current Constant	KIFS	IFS = 1A		11.75		A * kΩ
Full Scale Current Constant	KIFS	IFS = 2A		24		A * kΩ
Full Scale Current Constant	KIFS	IFS = 3A		36		A * kΩ
Current Trip Regulation Accuracy	DITRIP1	ITRIG from 7% to 100% FS, RREF=12kΩ	-5		5	%
FUNCTIONAL TIMINGS						
SLEEP time	tSLEEP	SLEEPN=0 to OUT_ tristate			50	us
Wake-up time from Sleep	TWAKE	SLEEPN=1 to normal operation			2.5	ms
Enable Time	TEN	Time from DRV_ENN pin falling edge to driver on			1.5	us
Disable Time	TEN	Time from DRV_ENN pin rising edge to driver off			6	us

Electrical Characteristics (continued)

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ, Typical Values assume TA=25°C and VS=24V, Limits are 100% tested at TA = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
CLOCK						
Internal Clock Frequency	fCLKOSC		11.9	12.5	13.2	MHz
External Clock Frequency	fCLK		12	16	20	MHz
External Clock Duty-Cycle	tCLKL		40		60	%
External clock detection in cycles			4		8	
External clock timeout detection in cycles of internal fCLKOSC			12		16	
External clock detection lower frequency threshold	fCLKLO		4			MHz
SPI TIMINGS						
SCK valid before or after change of CSN	tCC		T _{SCLK}			ns
CSN High Time	tCSH		4*T _{CLK}			ns
SCK Low Time	t _{CL}		20			ns
SCK High Time	t _{CH}		20			ns
SCK Frequency	f _{SCK}				10	MHz
SDI setup time before SCK rising edge	t _{DU}		10			ns
SDI hold time after SCK rising edge	t _{DH}		10			ns
Data out valid time after SCK falling edge	t _{DO}	VCCIO = 3.3V		27	40	ns
SDI, SCK, and CSN filter delay time	t _{FILT}	rising and falling edge		10		ns
ENCODER TIMING						
Encoder Counting Frequency	fCNT			<2/3 f _{CLK}	f _{CLK}	
A/B/N Input low time	t _{ABNL}		3t _{CLK} + 20			ns
A/B/N Input High Time	t _{ABNH}		3t _{CLK} + 20			ns
A/B/N spike filtering time	t _{FILTABN}	Rising and falling edge	3t _{CLK}			
ADC / Analog Input / Temperature						
ADC Resolution				13		Bit
Analog Input voltage range	VAIN		0		1.25	V
Analog Input Leakage	AINleak		-1		1	uA

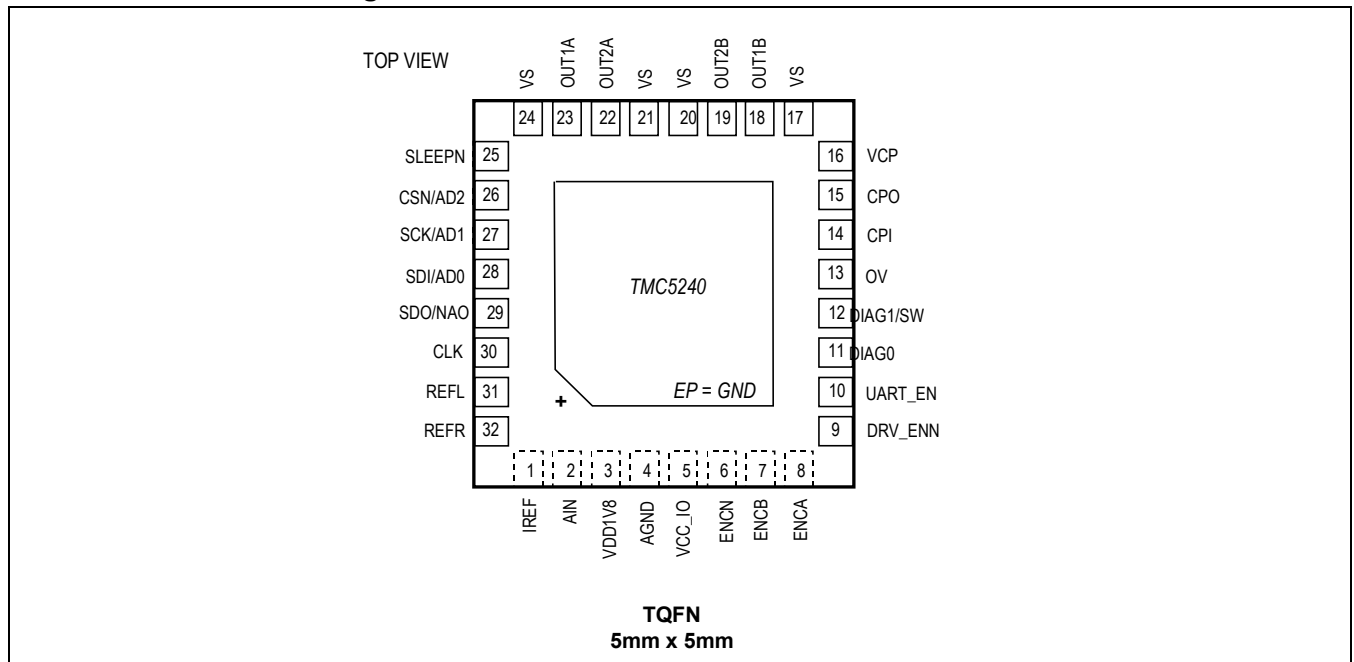
Electrical Characteristics (continued)

(VS = 4.5V to 36V, RREF = from 12kΩ to 24kΩ , Typical Values assume TA=25C and VS=24V, Limits are 100% tested at TA = +25°C. Limits over the operating temperature range and relevant supply voltage range are guaranteed by design and characterization. Specifications marked "GBD" are guaranteed by design and not production tested.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Analog Input Frequency	fAIN				70	kHz
Driver temperature accuracy	T _{driver}			+/-10		°C
Supply voltage measurement accuracy			-5		5	%
ADC Sample Rate	f _{sample, ADC}			f _{CLK} / 2048		

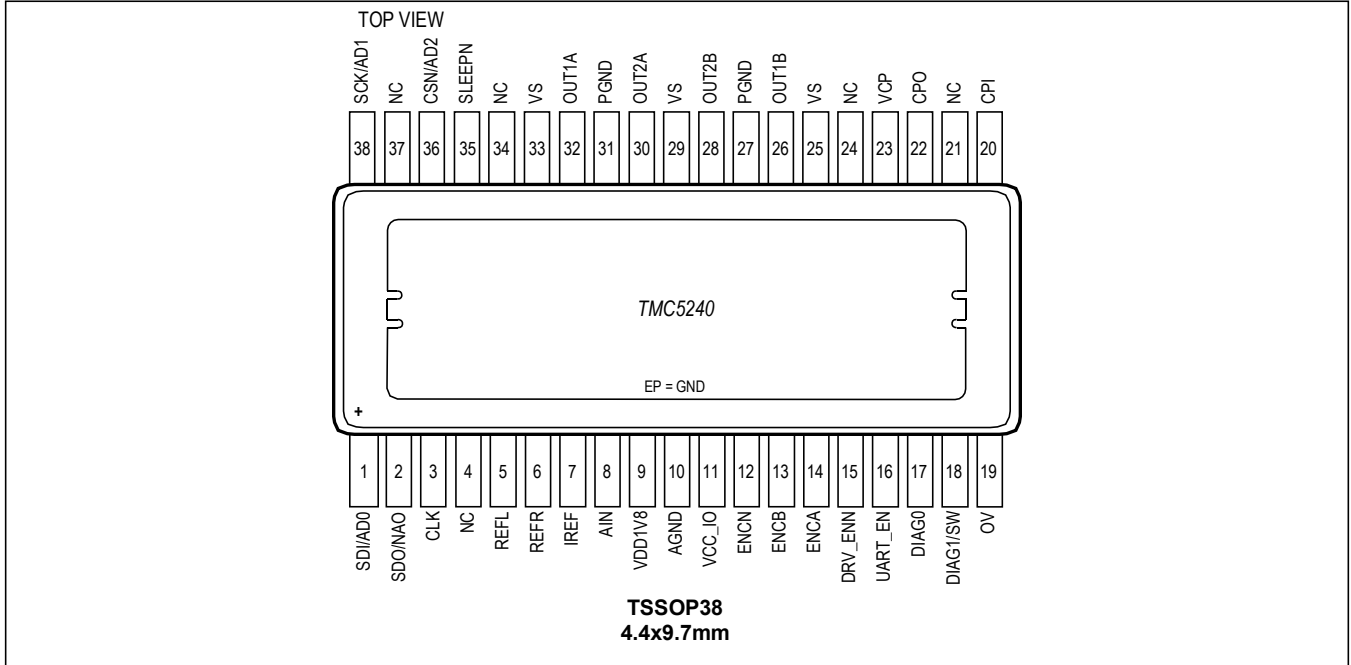
Pin Configurations

TMC5240 TQFN Pin Configuration



PRELIMINARY CONFIDENTIAL

TMC5240 TSSOP Pin Configuration



PRELIMINARY CONFIDENTIAL

Pin Description

PIN		NAME	FUNCTION	REF SUPPLY	TYPE
TQFN32	TSSOP38				
4	10	AGND	Analog Ground. Connect to Ground Plane.		GND
17, 20, 21, 24	25, 29, 33	VS	Motor supply voltage. Provide filtering capacity near pin with shortest loop to GND plane / exposed pad.		Supply
3	9	VDD1V8	Output of internal 1.8V regulator. Attach 2.2 μ F or larger ceramic capacitor to AGND near to pin for best performance.		Supply
16	23	VCP	Charge pump voltage. Tie to VS using 1.0 μ F capacitor. Connect positive end of capacitor close to VS pin to avoid inductive peaks.		Output
15	22	CPO	Charge pump capacitor output.		Output
14	20	CPI	Charge pump capacitor input. Tie to CPO using 22nF 50V capacitor.		Output
30	3	CLK	CLK input. Tie to GND using short wire for internal clock or supply external clock. Internal clock-fail over circuit protects against loss of external clock signal.	VCC_IO	DI
31	5	REFL	Left reference input for internal ramp generator	VCC_IO	DI
32	6	REFR	Right reference input for internal ramp generator	VCC_IO	DI
26	36	CSN/AD2	SPI chip select input (negative active) (UART_EN=0) or Address input 2 (+4) in UART mode (UART_EN = 1)	VCC_IO	DI (pd)
27	38	SCK/AD1	SPI serial clock input (UART_EN=0) or address input 1 (+2) in UART mode (UART_EN=1)	VCC_IO	DI (pd)
28	1	SDI/AD0	SPI data input (UART_EN=0) or address input 0 (+1) for single wire interface (UART_EN=1).	VCC_IO	DI (pd)
29	2	SDO/NAO	SPI data output (tristate) (UART_EN=0) or next address output (NAO) for single wire interface (UART_EN=1).	VCC_IO	DIO (pd)
1	7	IREF	Analog reference current for current scaling. Provide external resistor to GND.	VCC_IO	AI
10	16	UART_EN	Interface selection pin. When tied low, the SPI interface is enabled. When tied high, the UART interface is enabled. Integrated pull-down resistor.	VCC_IO	DI (pd)
7	13	ENCB	Encoder B-channel input	VCC_IO	DI
8	14	ENCA	Encoder A-channel input	VCC_IO	DI
6	12	ENCN	Encoder N-channel input	VCC_IO	DI
9	15	DRV_ENN	Enable input. The power stage becomes switched off (all motor outputs floating) when this pin becomes driven to a high level.	VCC_IO	DI (pu)

引脚说明 (续)

PIN		名称	功能	参考供电	TYPE
TQFN32	TSSOP38				
11	17	DIAG0	<p>诊断输出 DIAG0。</p> <p>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。</p> <p>在开漏模式下使用外部上拉电阻。</p> <p>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。</p>	VCC_IO	DO
12	18	DIAG1/SW	<p>诊断输出 DIAG1。</p> <p>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。</p> <p>在开漏模式下使用外部上拉电阻。</p> <p>UART 模式下的单线 I/O。</p>	VCC_IO	DIO
25	25	SLEEPN	<p>低电平有效掉电输入/复位输入。</p> <p>提供连续低电平使设备进入睡眠模式。</p> <p>SLEEPN 有一个内部上拉电阻。</p> <p>如果不使用连接到 VS 或 VCC_IO (这是一个高压引脚)。</p> <p>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。</p> <p>在重新配置 IC 时，建议仍然使用 DRV_ENN 禁用功率桥驱动器。</p> <p>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。</p> <p>不要在高电机速度下使用！</p>	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	<p>裸露的芯片焊盘。</p> <p>将裸露芯片焊盘连接到 GND 平面。</p> <p>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。</p>		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

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 $f_{\text{CLK}} / 2048$	VCC_IO	DO (OD)
2	8	AIN	通用模拟输入通过测量 $f_{\text{CLK}}/2048$ 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

功能图

TMC5240

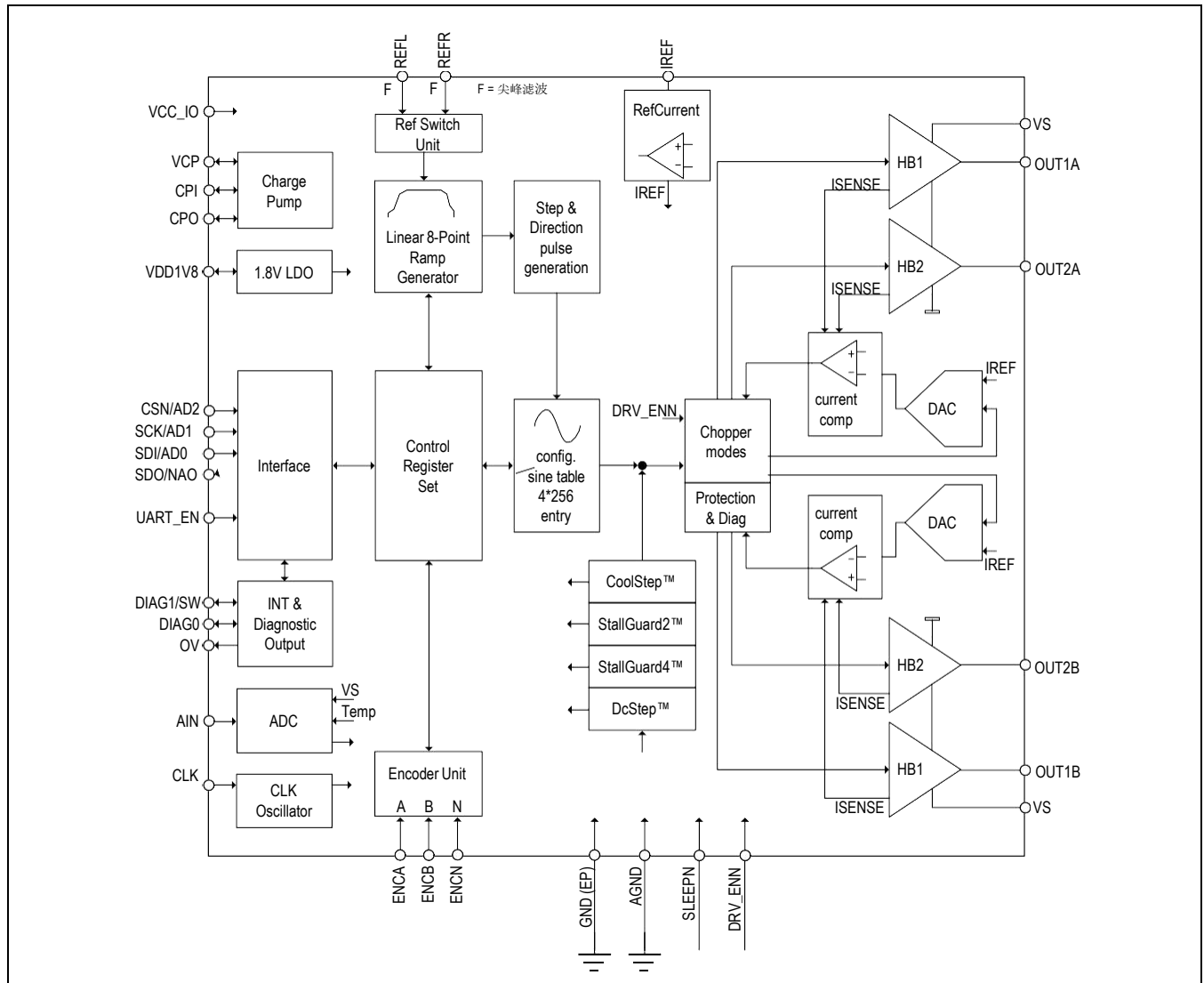


Figure 1. TMC5240 block diagram

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。

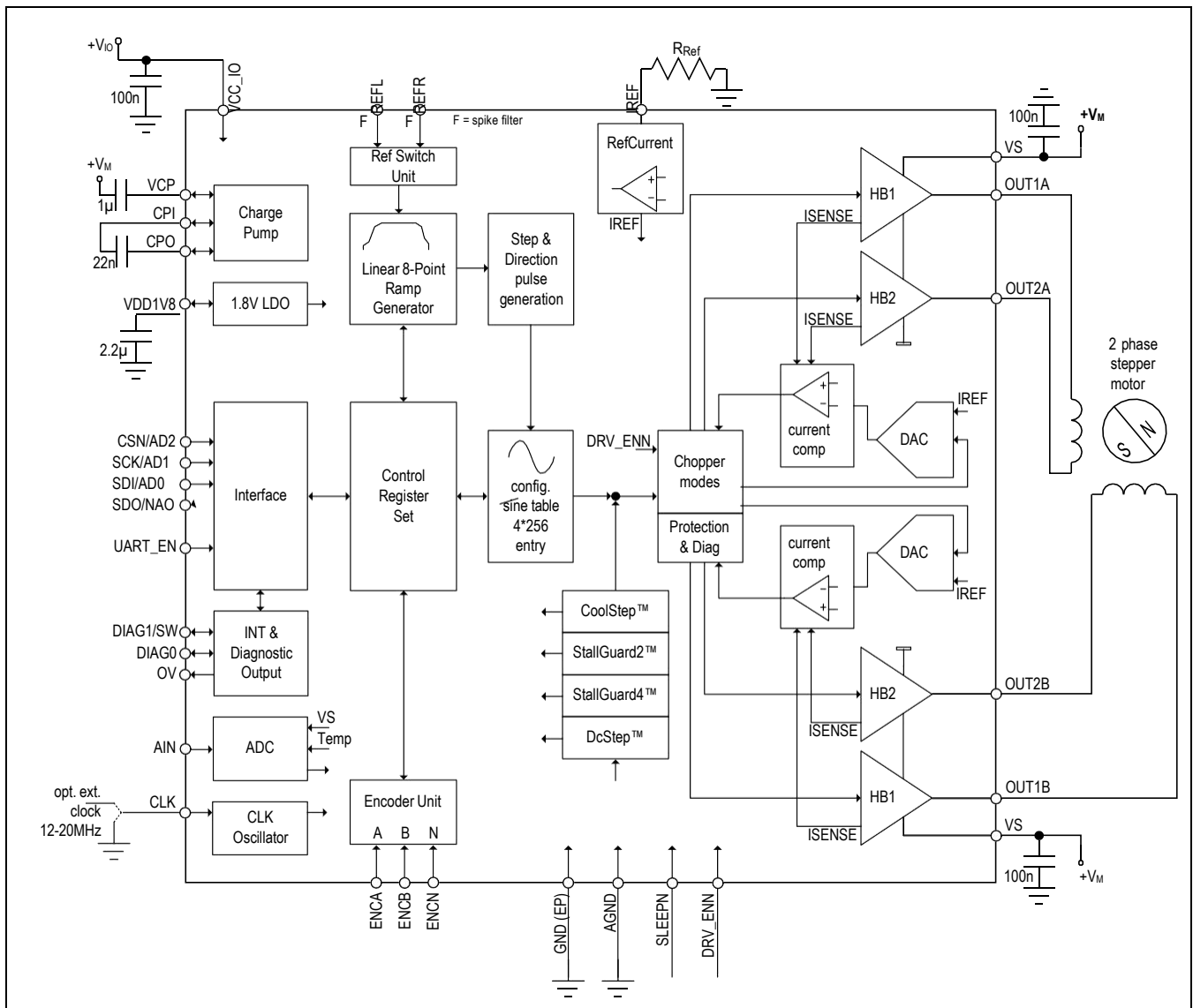


图2. 带有典型外部组件的 TMC5240

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著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！

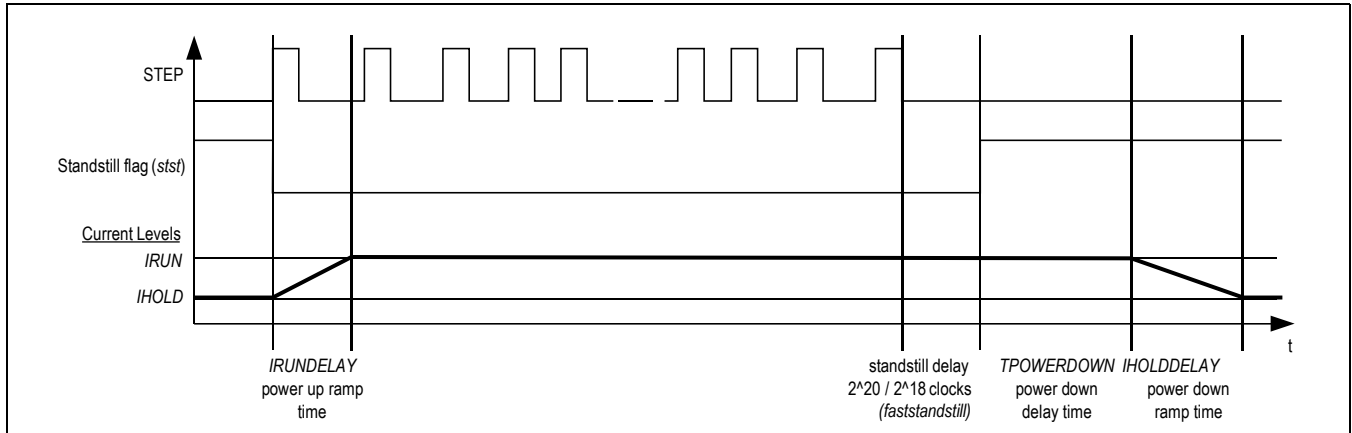


图 3. 静止和加速时的自动电机电流控制

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** 甚至可以在组合配置中使用，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：

- 使用低成本电机显著改善微步性能
- 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

- 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)
	39 ... 0	
写：8位地址 读取：8 位 SPI 状态		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	8 bit data		8 bit data		8 bit data		8 bit data	
读自 8位SPI状态	31 ... 24		23 ... 16		15 ... 8		7 ... 0	
39 / 38 ... 32	31 ... 28	27 ... 24	23 ... 20	19 ... 16	15 ... 12	11 ... 8	7 ... 4	3 ... 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 322

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)

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。

If more than 40 clocks are driven, the additional bits shifted into SDI are shifted out on SDO after a 40-clock delay through an internal shift register. This can be used for daisy chaining multiple chips.

CSN must be low during the whole bus transaction. When CSN goes high, the contents of the internal shift register are latched into the internal control register and recognized as a command from the master to the slave. If more than 40 bits are sent, only the last 40 bits received before the rising edge of CSN are recognized as the command.

SPI Timing

The SPI max frequency is at 10 MHz. SCK is independent from the clock frequency of the system while the only parameter depending on the clock frequency is the minimum CSN high time. 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.

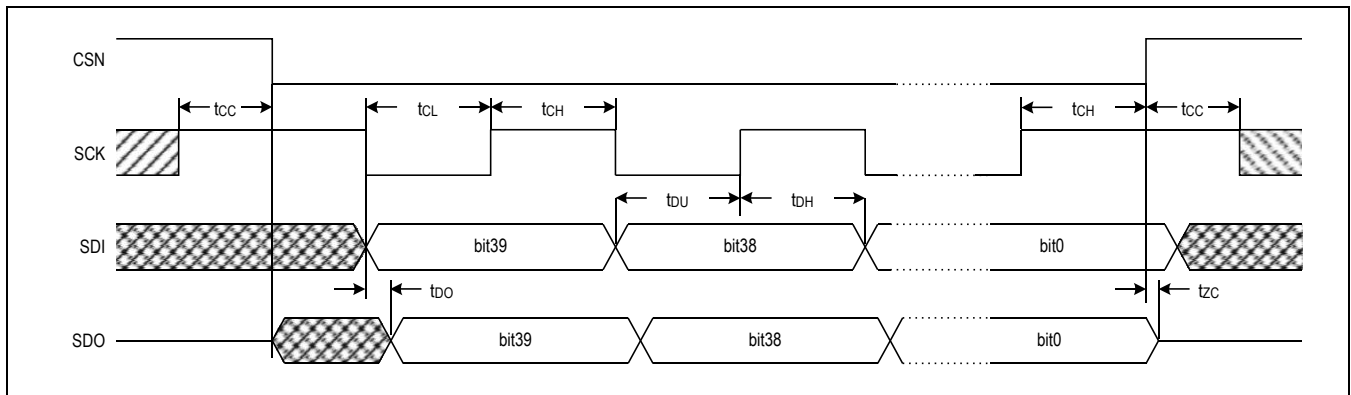


Figure 4. SPI Timing Diagram

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

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	

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 *SENDDDELAY* time setting (default=8 bit times) according to the needs of the master. In a multi-slave system, set *SENDDDELAY* 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);
            }
        }
    }
}
```

```

    }
    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.

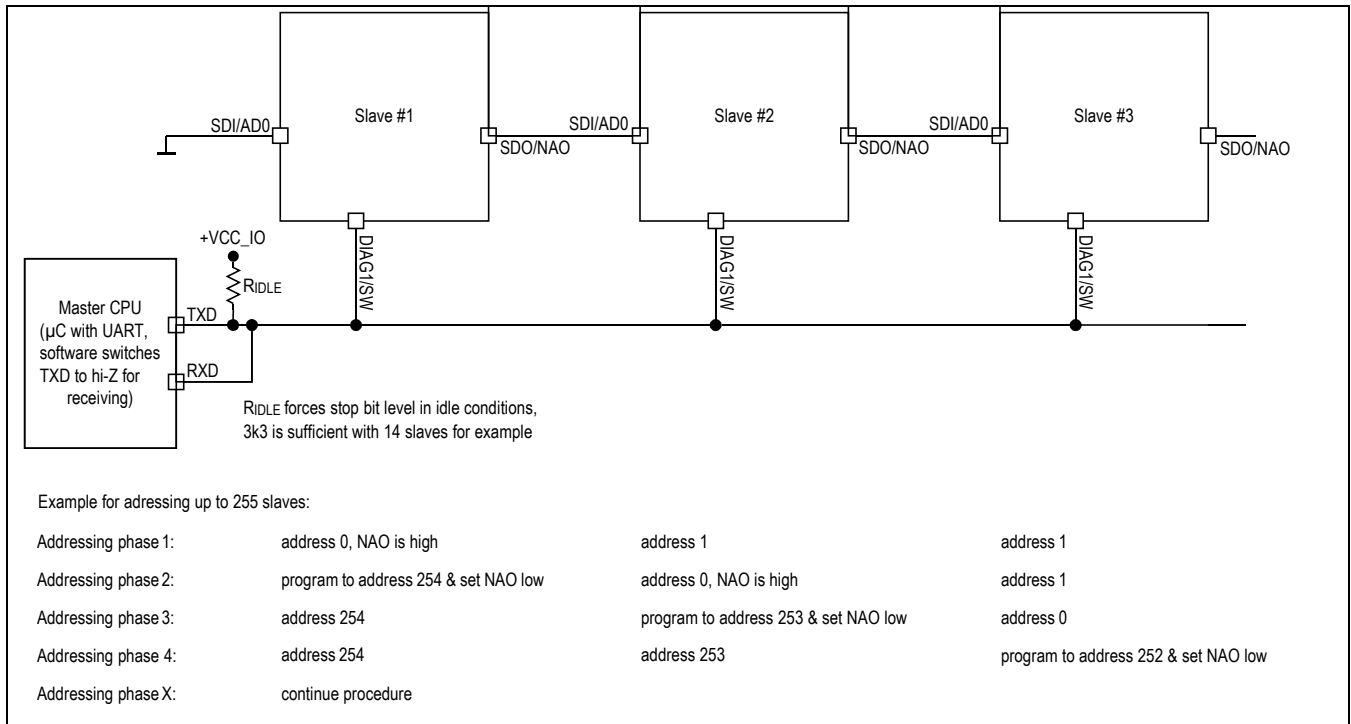


Figure 5. UART daisy chaining example

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

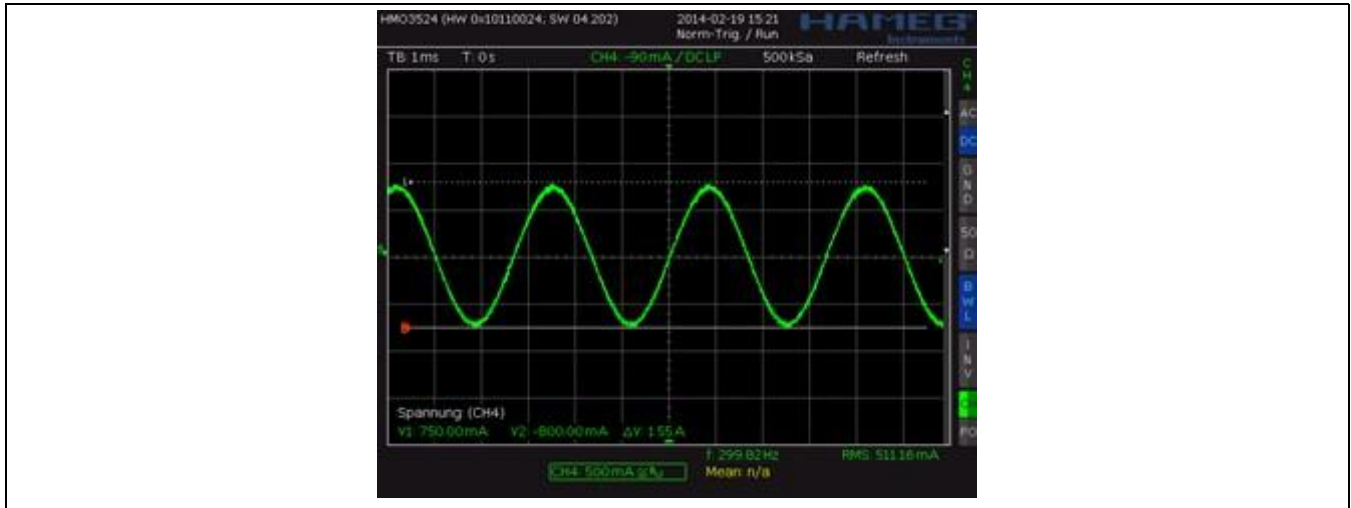


图 6. 采用 StealthChop 的电机线圈正弦波电流（用电流探头测量）

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	<ul style="list-style-type: none"> 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。 	$\leq 2^{20} + 2^{2 \cdot 18} t_{CLK}$ $\leq 130\text{ms}$ (使用内部时钟)
AT#2	PWM_GRAD_AUTO	<ul style="list-style-type: none"> 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： $1.5 \cdot PWM_OFS_AUTO \cdot (IRUN+1) / 32 < PWM_SCALE_SUM < 4 \cdot PWM_OFS_AUTO \cdot (IRUN+1) / 32$ $PWM_SCALE_SUM < 255$。 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:

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。

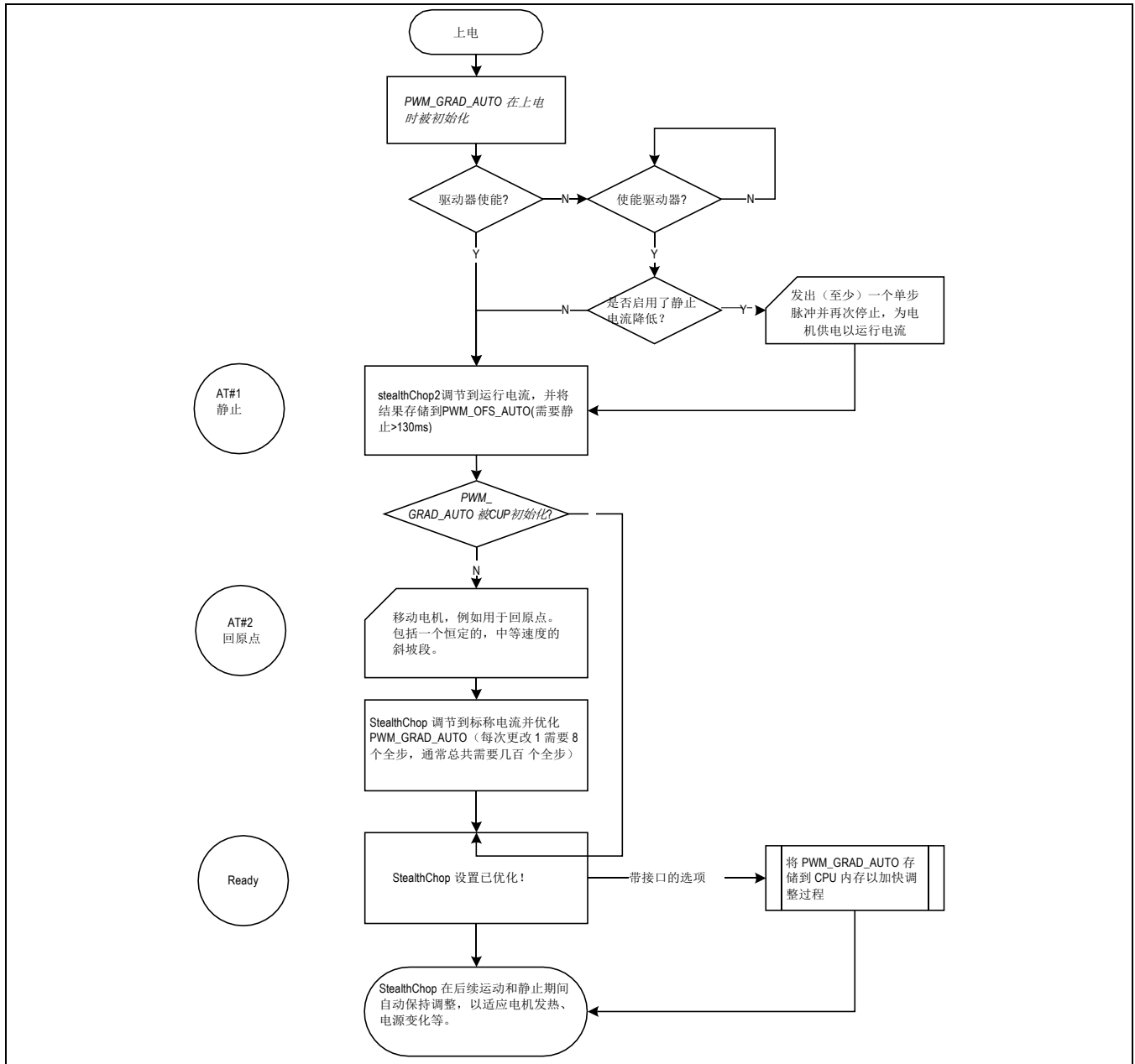


图 7. StealthChop2 自动调谐过程

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)。

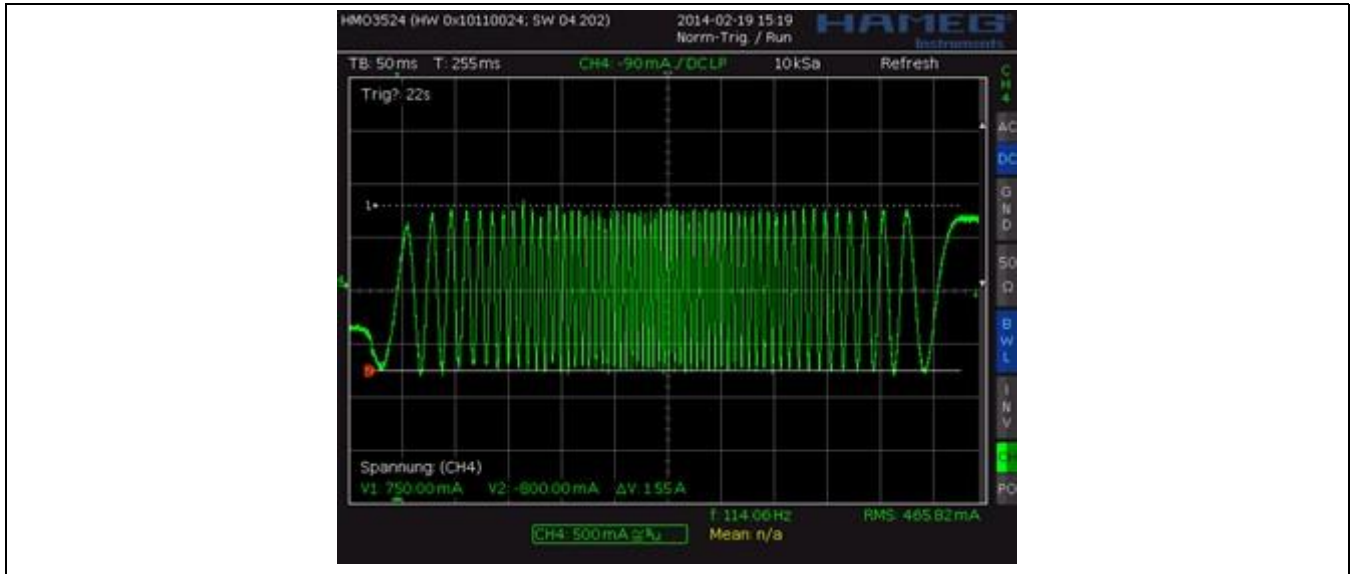


图 8. StealthChop2: PWM_REG 的良好设置

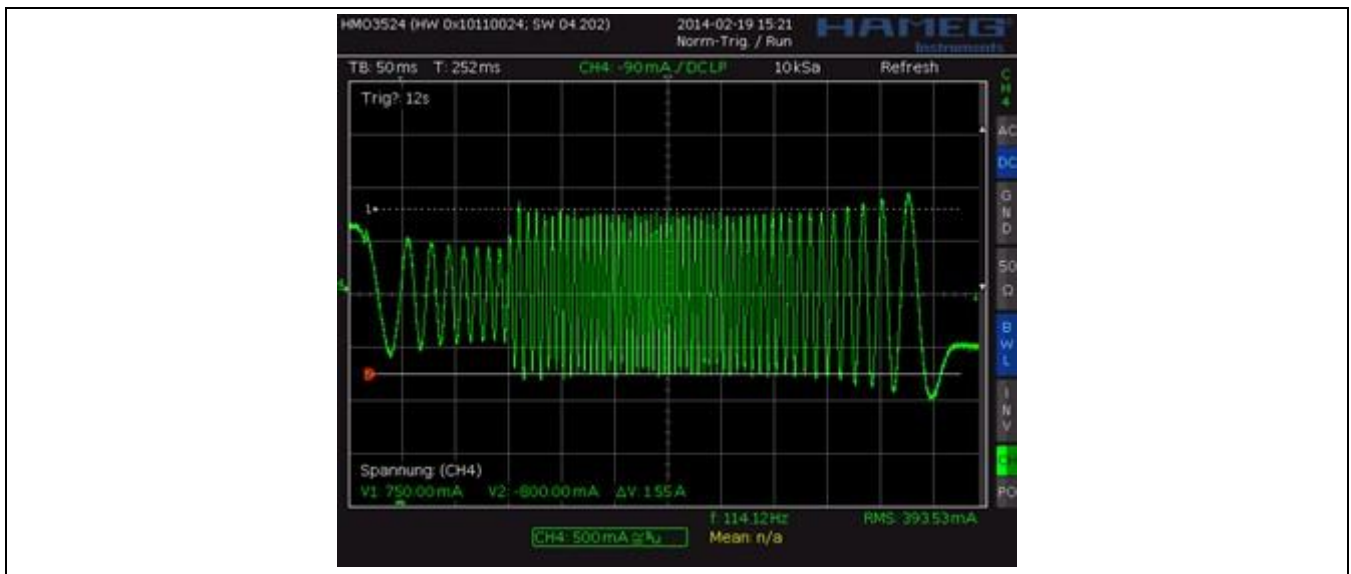


图 9. StealthChop2: 在 AT#2 期间 PWM_REG 的设置太小

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）。

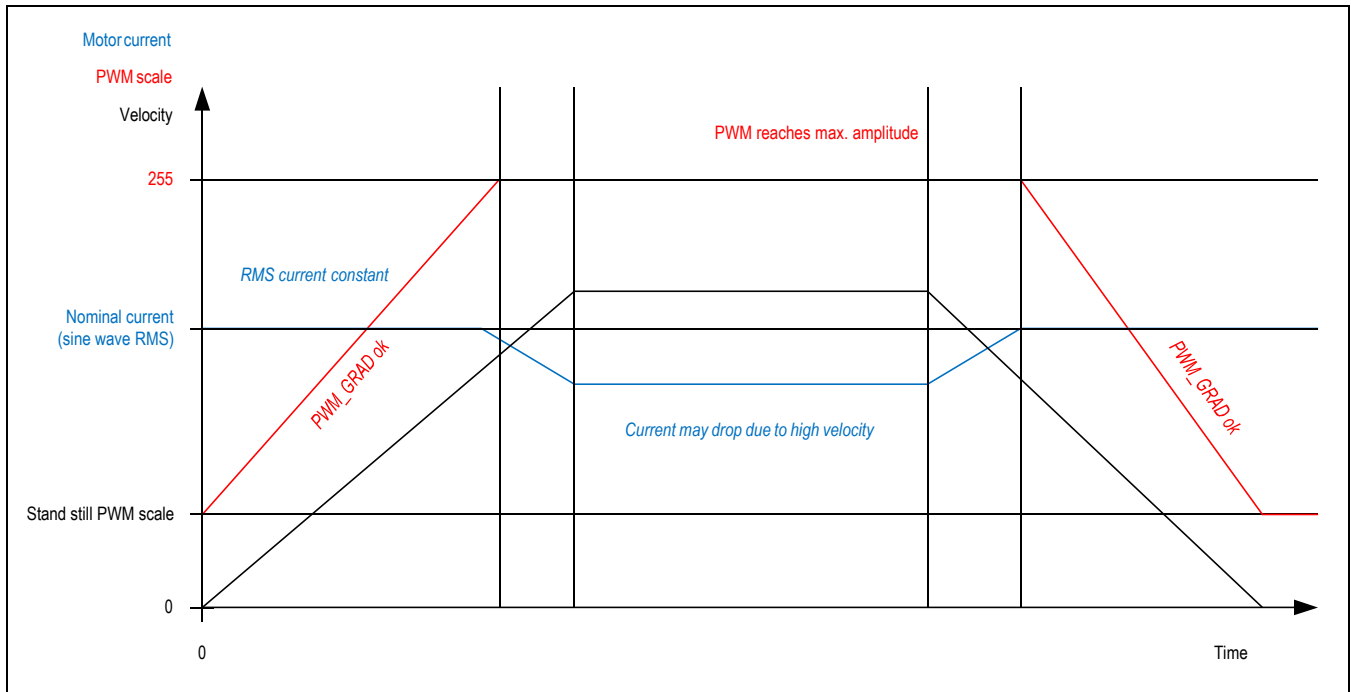


图 10. 成功确定 PWM_GRAD_AUTO 和 PWM_OFS_AUTO

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。较低的消息时间 (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。

$I_{LowerLimit}$ 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, t_{BLANK} 为 24 个时钟周期， f_{PWM} 为 $2/(1024 \text{ 个时钟周期})$ ：

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

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。

$IRUN \geq 8$: $IRUN$ 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 `ILOWER 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：

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

V_M 为电机电源电压， I_{COIL} 为目标 RMS 电流

考虑到 8 位分辨率和 248 个正弦波峰值，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`：

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

随着电机速度的提高，电机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：

$$PWM_SCALE_SUM = PWM_OFS + PWM_GRAD * 256 * \frac{f_{STEP}}{f_{CLK}}$$

f_{STEP} 是相当于 256 微步分辨率的微步频率， f_{CLK} 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：

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

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， $51200 = 256\mu\text{steps}$ 乘以 200 fullsteps。

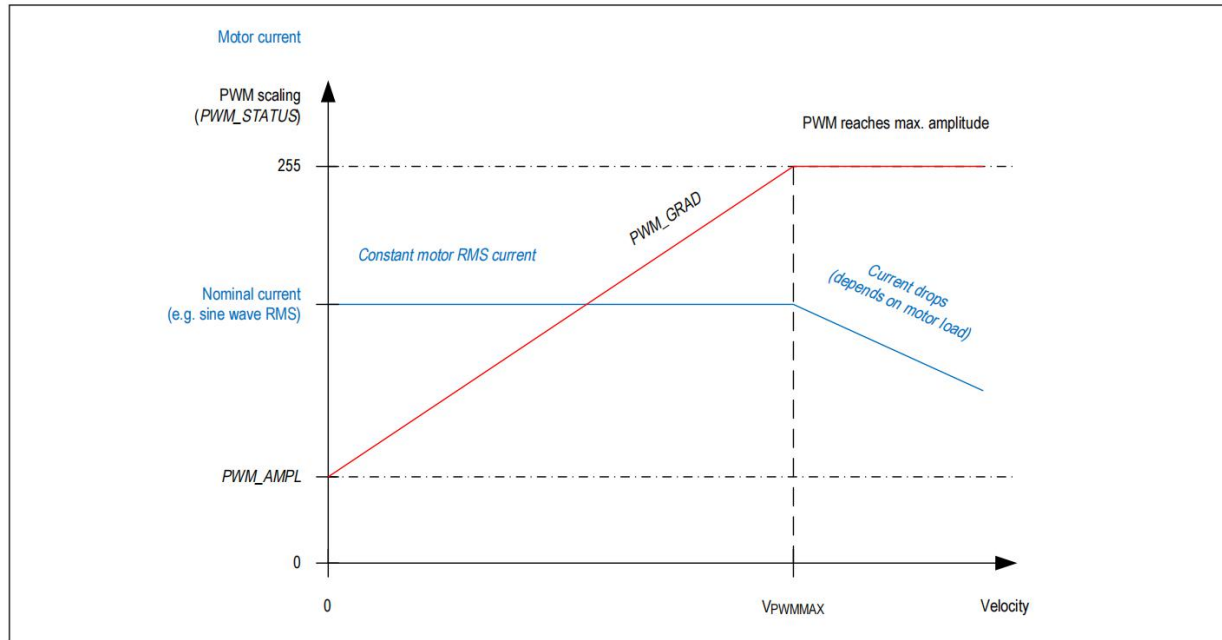


图 11. 基于速度的 PWM 标定 (pwm_autoscale=0)

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} 的数值与扭矩常数的数值相同。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。

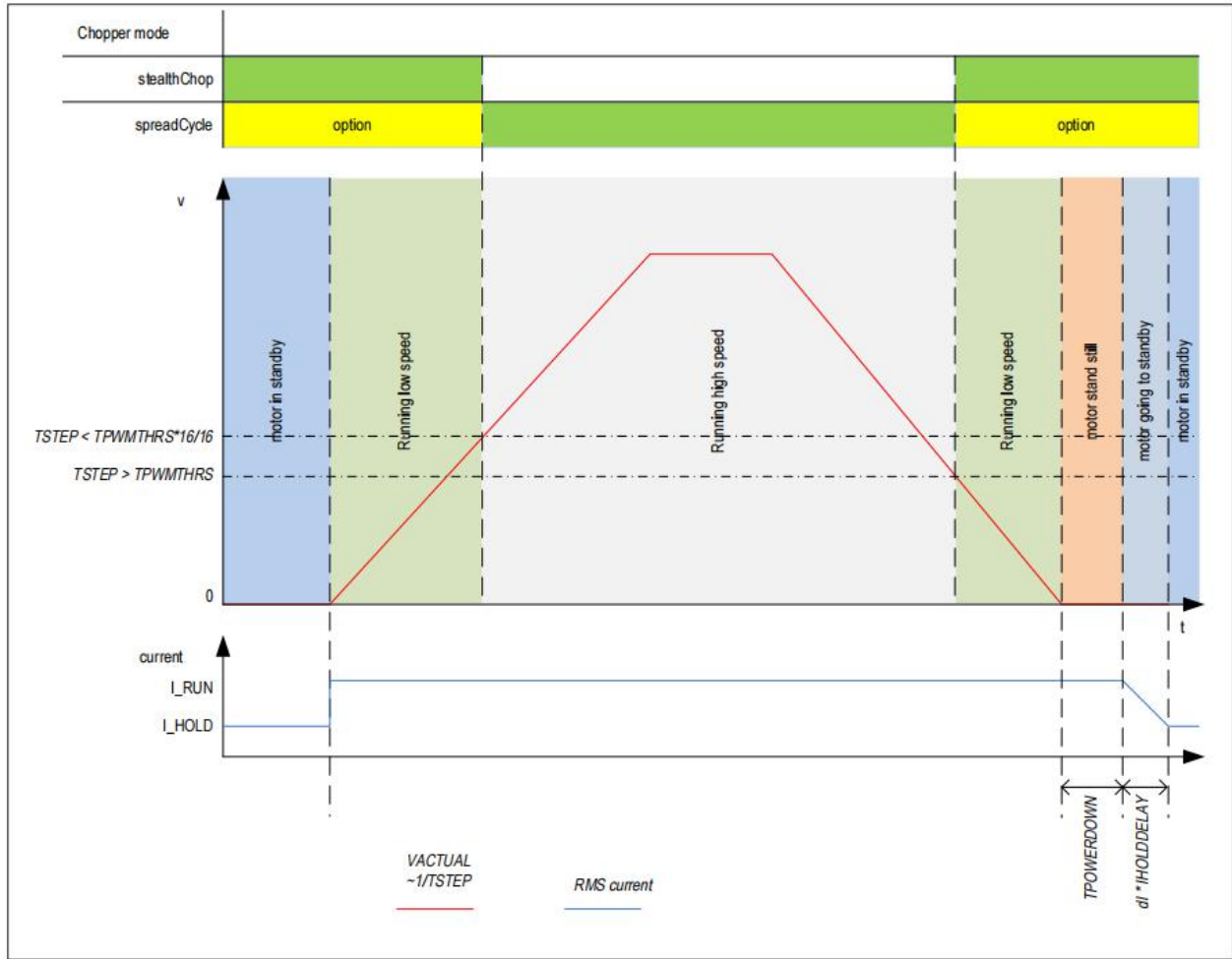


图 12. TPWMTHRS 用于可换到 SpreadCycle

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，并将该结果值设置到 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。

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。

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 fPWM 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_OFFSETS_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

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	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	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 tCLK
		1	24 tCLK
		2	36 tCLK
		3	54 tCLK

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。

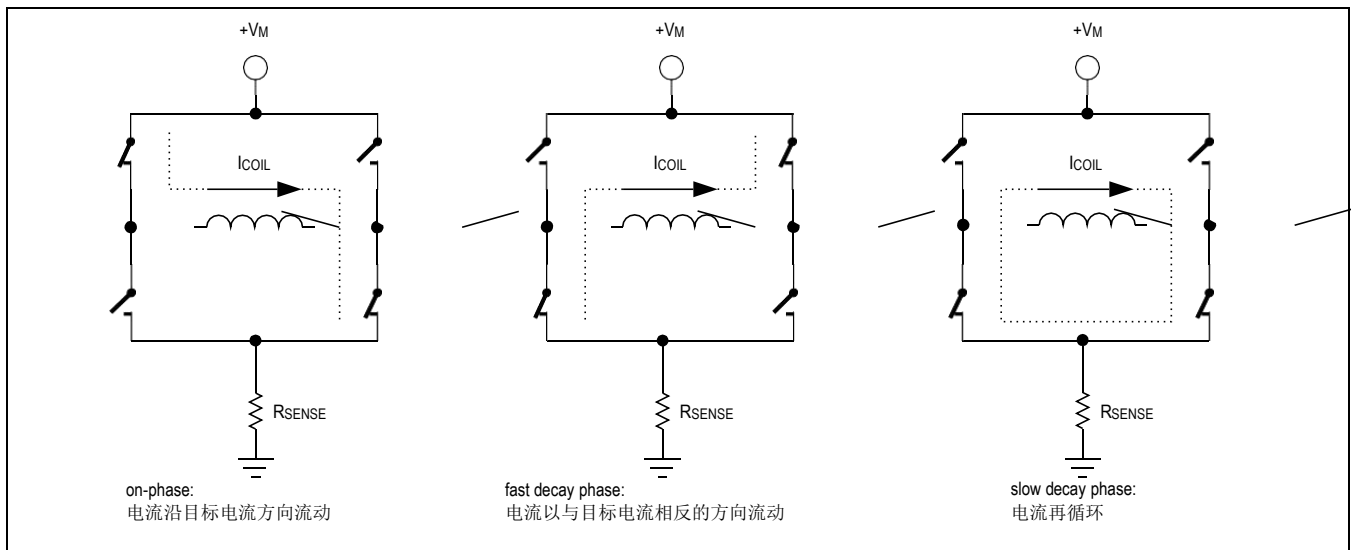


图 13. 典型斩波衰减阶段

尽管可以仅使用相位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。

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。恒定关断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，需要找到折衷方案。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

Parameter	Description	Setting	Comment
TOFF	设置慢衰减时间 (<i>off time</i>)。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内工作)
TBL	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个慢衰减阶段组成。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%
- $t_{OFF} = 1 / 25\text{kHz} * 50 / 100 * 1 / 2 = 10 \mu\text{s}$
- 对于 **TOFF** 设置，这意味着： $TOFF = (t_{OFF} * f_{CLK} - 12) / 32$
- 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。

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。

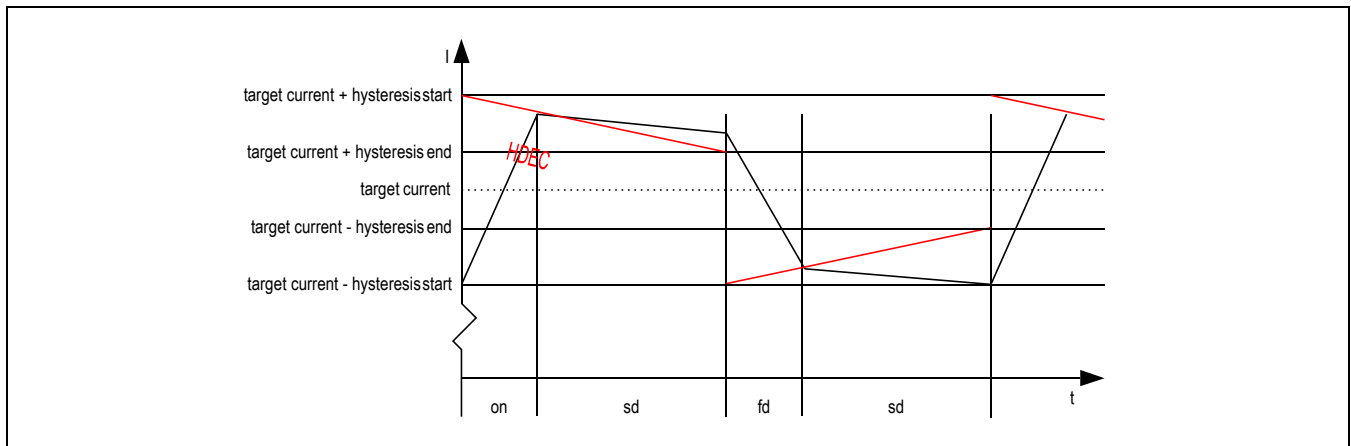


图 14. SpreadCycle 斩波器方案显示斩波器周期期间的线圈电流

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 $HEND$

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 $+7: 7-3=4$)

经典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。

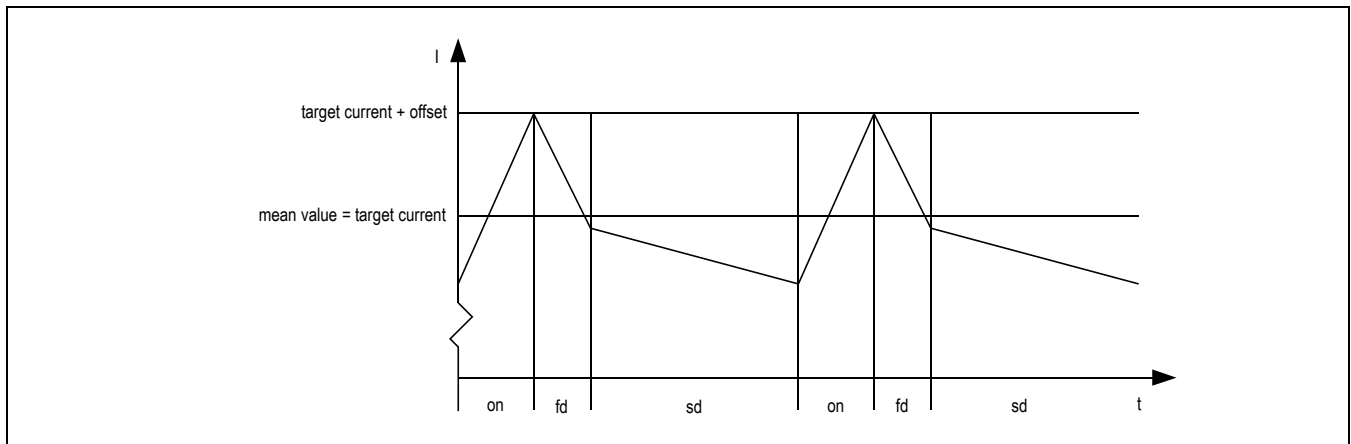


图15。经典的常量关断时间偏移显示线圈电流的关断时间斩波器

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。

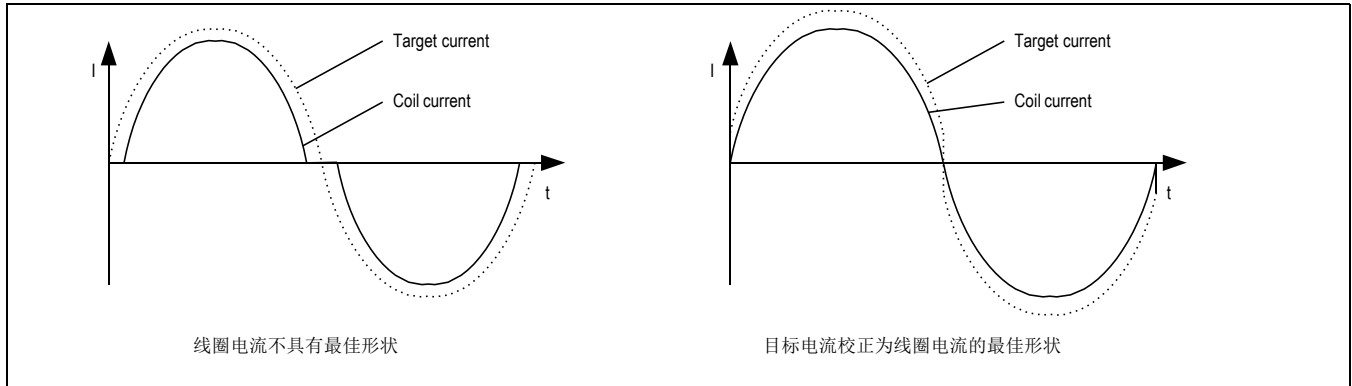


图 16. 使用经典斩波器的过零和使用正弦波偏移的校正

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 显著节省了空间和功耗。

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。		

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 TZERO WAIT 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 $4 \times 15 \times 2^{18}$ 个时钟周期，即在 16MHz 时大约为一秒。	0	瞬间下降至 IHOLD
		1 ... 15	$1 \times 2^{18} \dots 15 \times 2^{18}$ 每个电流递减的时钟
IRUNDELAY	Controls the number of clock cycles for motor power up after start is detected. Allows smooth current increment upon start of a motion from hold current (IHOLD) to run current (IRUN). While a quick power up is important to establish full motor torque, a small delay time helps to reduce acoustic noise and avoids a jerk on the power supply current.	0	instant power up to IRUN
		1...15	Delay per current increment step in multiple of IRUNDELAY * 512 clocks

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 K_{IFS} 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 12K Ω 和 60K Ω 之间。

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

表 15. IFS 满量程设置（RREF = 12K Ω 的示例）

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

基于速度的模式控制

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。

CoolStep 或 StallGuard2 之类的一些功能在有限的速度范围内很有用。许多速度阈值允许在需要宽速度范围的应用中组合不同的操作模式。

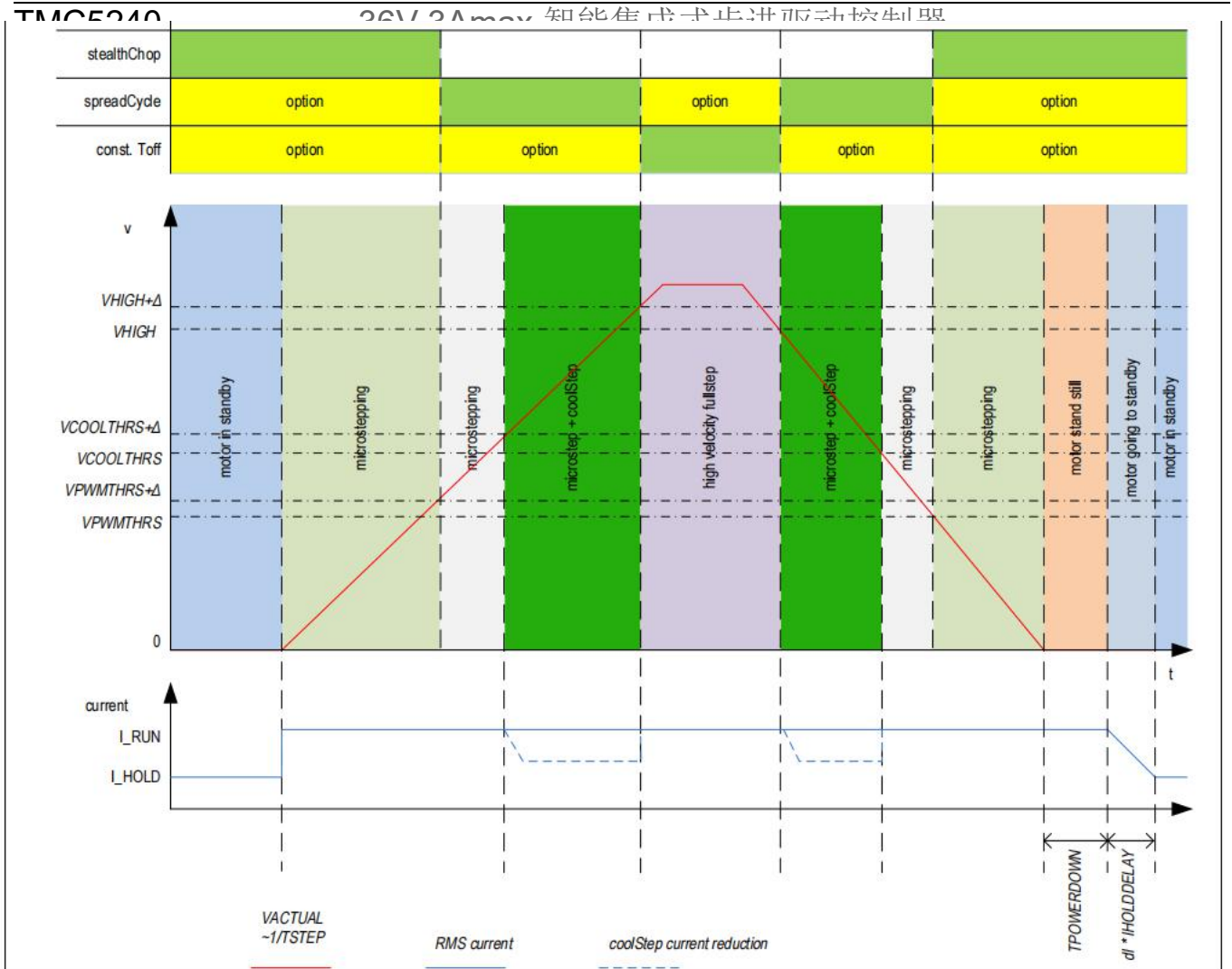


图17. 速度相关模式的选择

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
TPOWER DOWN	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	$TSTEP \geq TPWMTHRS$ <ul style="list-style-type: none"> 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	$TCOOLTHRS \geq TSTEP \geq THIGH$: <ul style="list-style-type: none"> CoolStep 已启用（如果已配置） StealthChop 电压 PWM 模式禁用 $TCOOLTHRS \geq TSTEP$ <ul style="list-style-type: none"> 启用失速输出信号（如果已配置）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	$TSTEP \leq THIGH$: <ul style="list-style-type: none"> 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，

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	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	
$\mu\text{step velocity } v$ [Hz]	$\mu\text{steps} / \text{s}$	$v[\text{Hz}] = v[\text{TMC5240}] * (f_{CLK}[\text{Hz}]/2 / 2^{23})$
$\mu\text{step acceleration } a$ [Hz/s]	$\mu\text{steps} / \text{s}^2$	$a[\text{Hz/s}] = a[\text{TMC5240}] * f_{CLK}[\text{Hz}]^2 / (512 * 256) / 2^{24}$
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	$v[\text{rps}] = v[\mu\text{steps/s}] / \text{USC} / \text{FSC}$ FSC: motor fullsteps per rotation, e.g. 200
rps acceleration a [rps/s ²]	rotations / s ²	$a[\text{rps/s}^2] = a[\mu\text{steps/s}^2] / \text{USC} / \text{FSC}$
ramp steps[μsteps] = rs	μsteps	$rs = (v[\text{TMC5240}])^2 / a[\text{TMC5240}] / 2^8$ 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 $\text{intpol}=1$ 和 $\text{MRES}=\%0001$ 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

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。

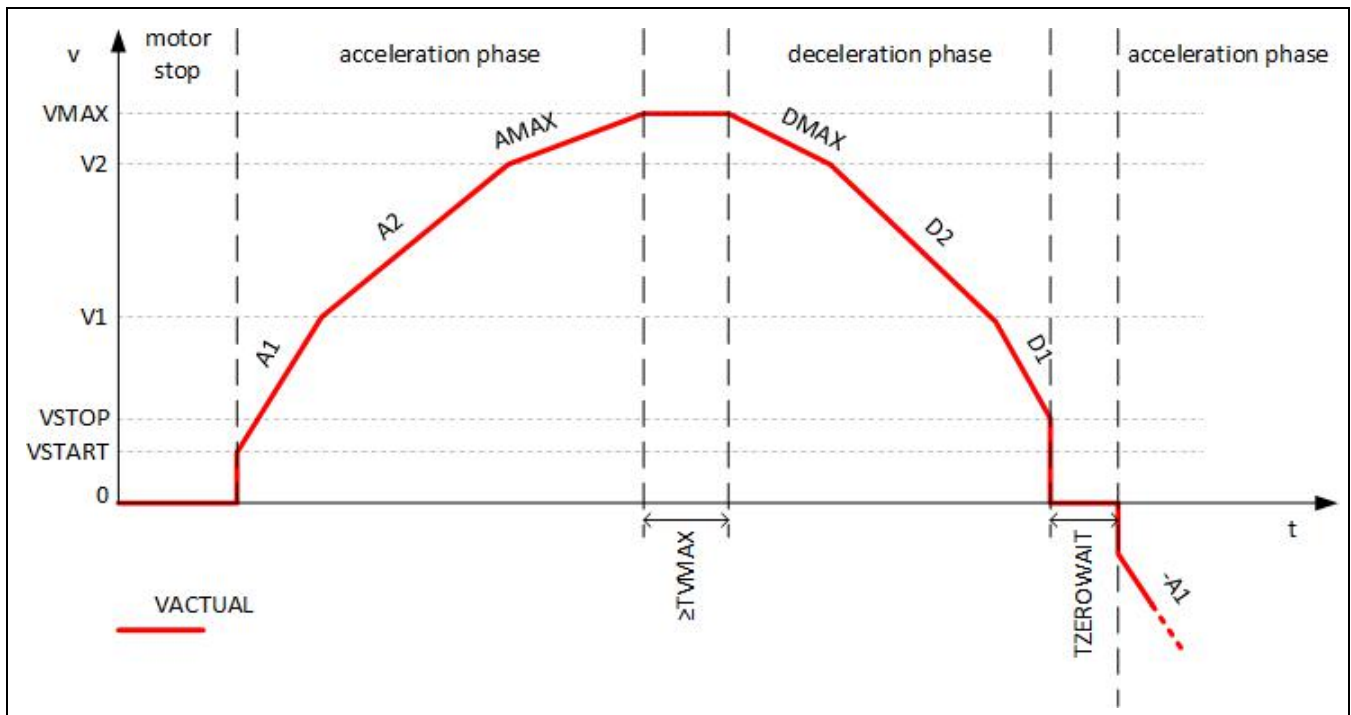


图 18. 斜坡发生器速度轨迹显示随后的负方向移动

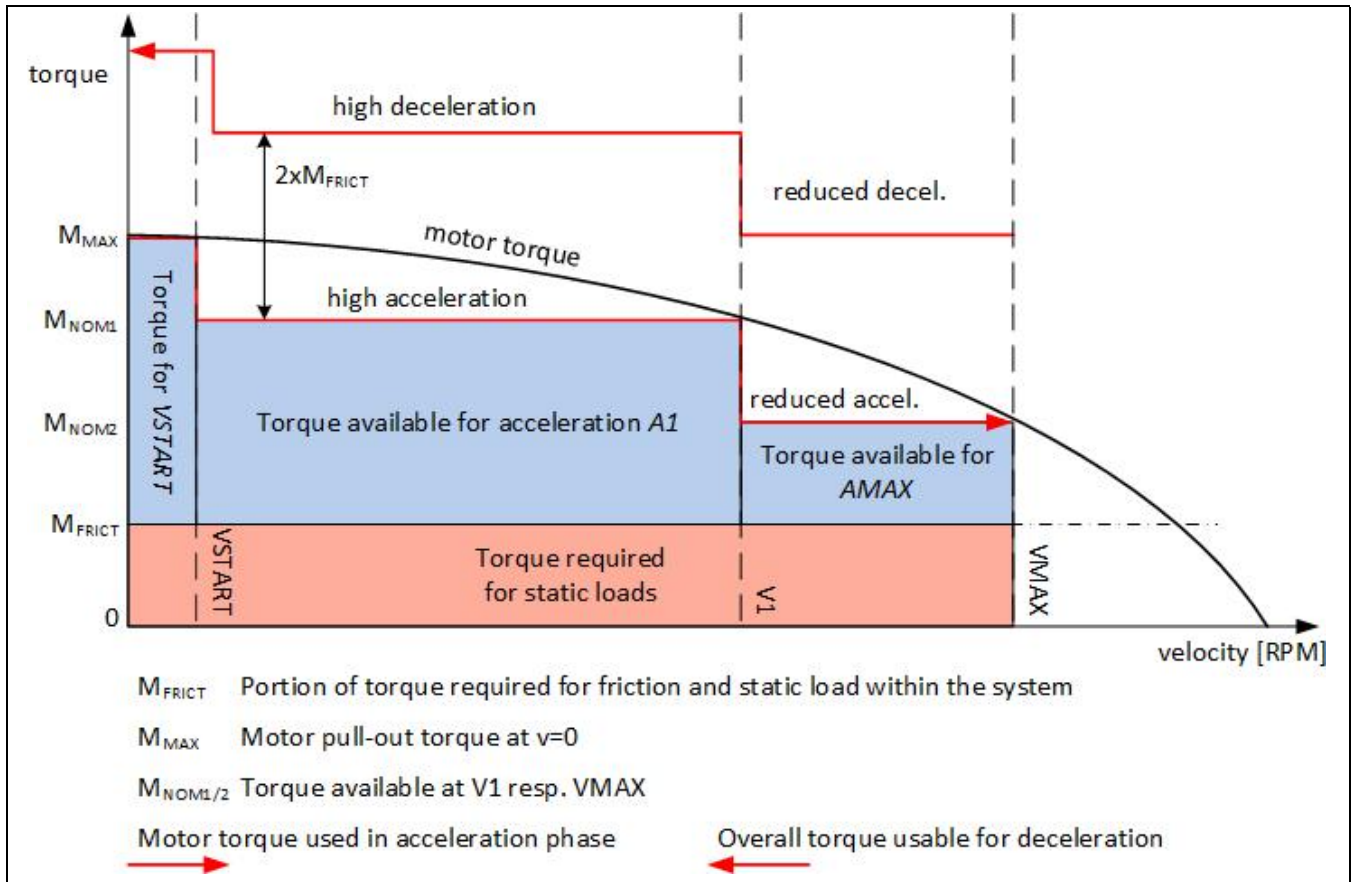


图 19. 使用斜坡发生器优化电机扭矩的图示

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 $V_{START}+V_{STOP}$ 相同的加加速度，而不仅仅是 V_{START} 。由于电机可能无法跟随这一点，您可以通过设置 $T_{ZEROWAIT}$ 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 (T_{VMAX}) 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 V_{MAX} (V_{MAX}') 并用于等速段。最小 V_{MAX}' 与 V_{STOP} 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。

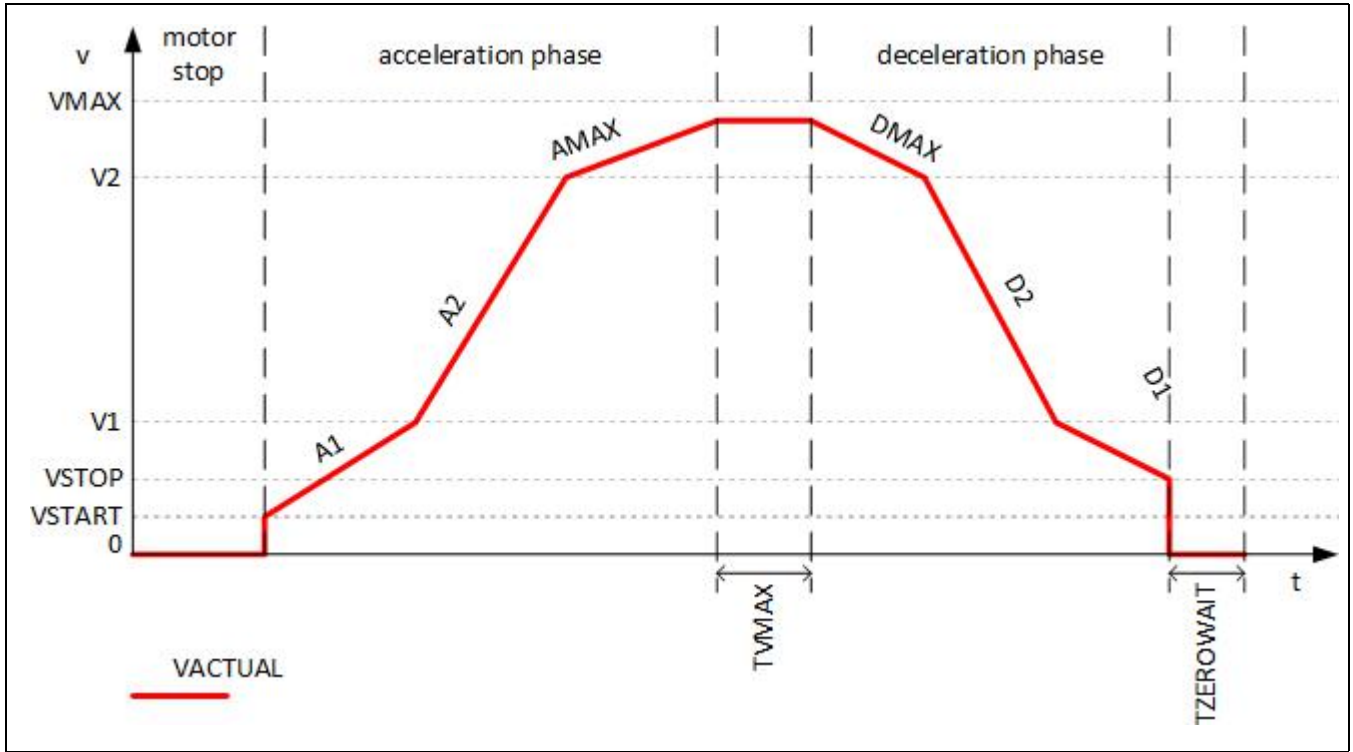


图 20. 由于距离过短而未达到 $VMAX$ 的 8 点斜坡

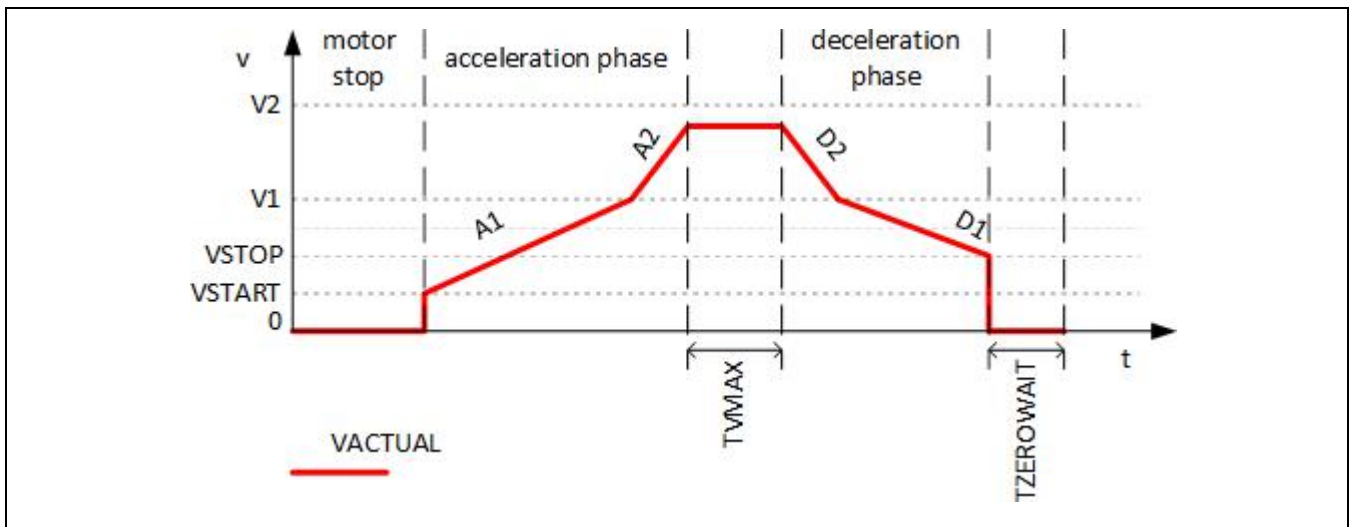


图 21. 由于距离过低，未达到 $V2$ 且无 $AMAX$ 和 $DMAX$ 阶段

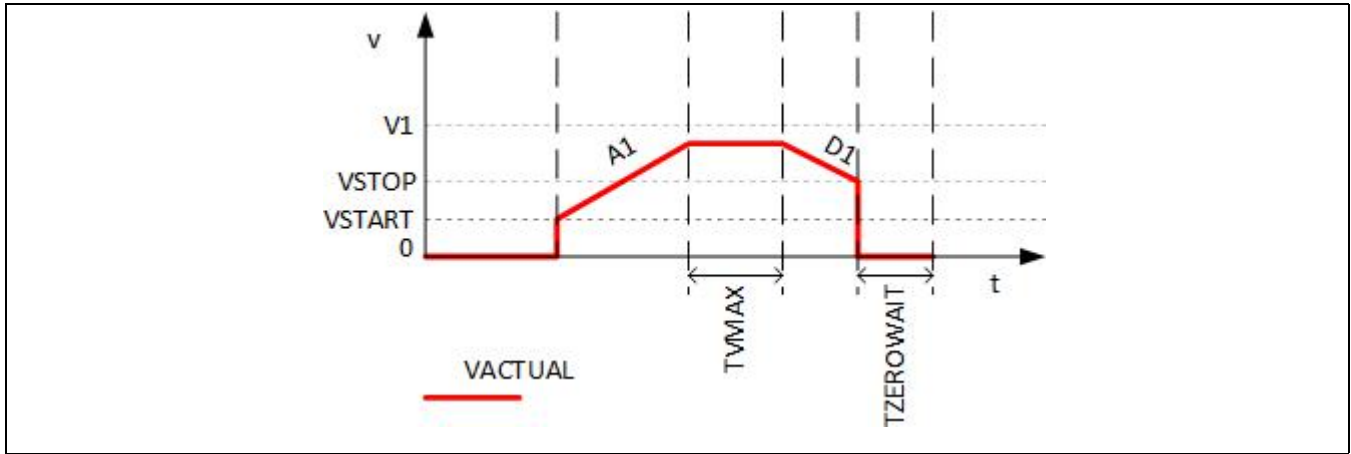


图 22. 由于距离过短而未达到 V1

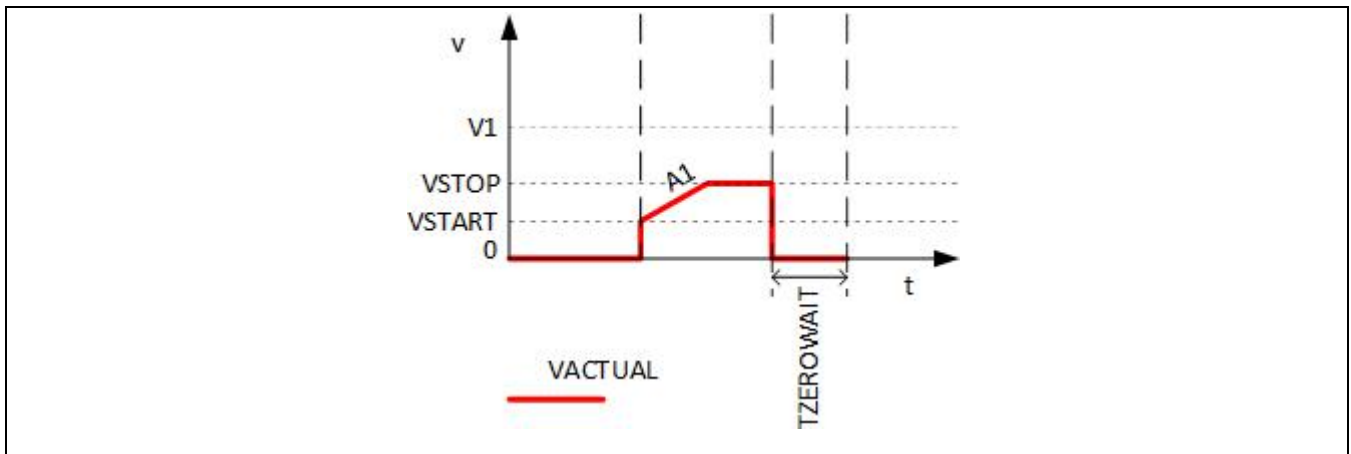


图 23. TVMAX 由于低位移而未保留

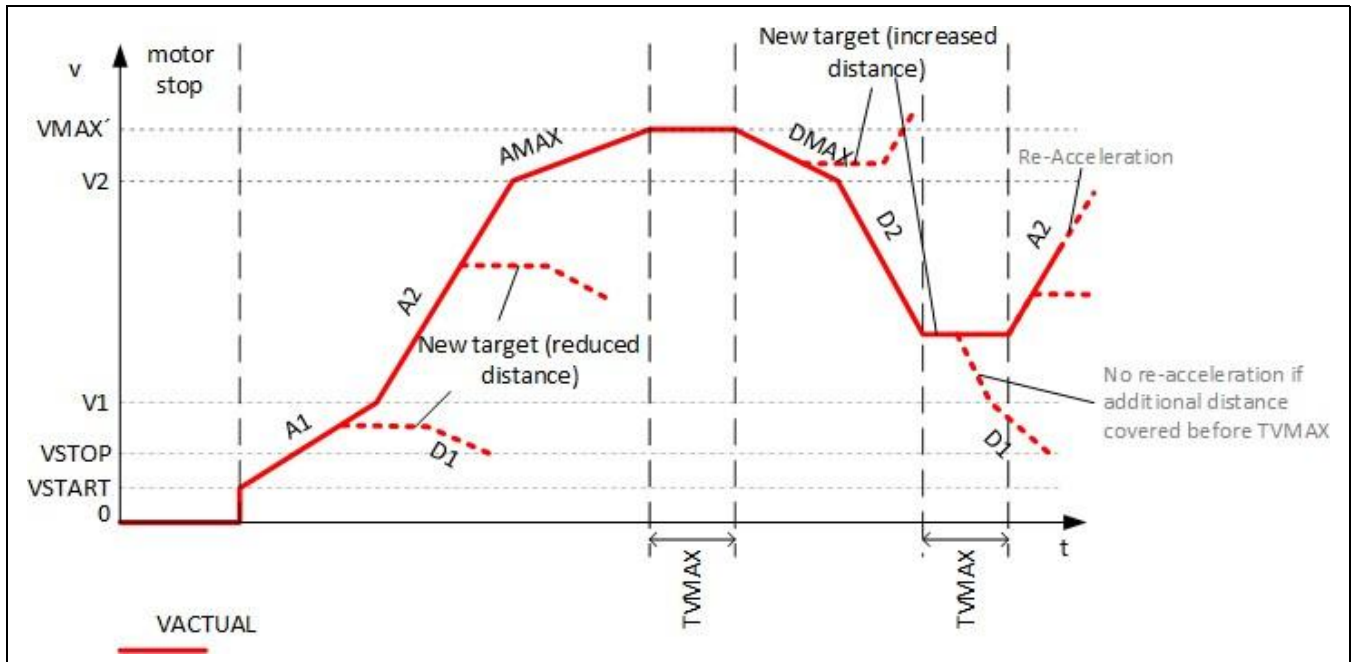


图 24. 带有动态目标位置变化的 8-Point Ramp 示例

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 V_{MAX} 和 A_{MAX} 。在这种模式下，斜坡发生器总是使用 A_{MAX} 加速或减速到 V_{MAX} 。

为了使电机减速停止，只需将 V_{MAX} 设置为零即可。标志 v_{zero} 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 $V_{MAX}=0$ 和 A_{MAX} 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 $V_{START}=0$ 和 $V_{MAX}=0$ 。在这种情况下不使用 V_{STOP} 。驱动器将使用 A_{MAX} 和 $A1$ （由 $V1$ 确定）以达到零速度。
2. 对于正使用 D_{MAX} 、 $D1$ 、 $D2$ 和 V_{STOP} 进行的停止，通过将 X_{ACTUAL} 复制到 X_{TARGET} 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。驱动器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。

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。

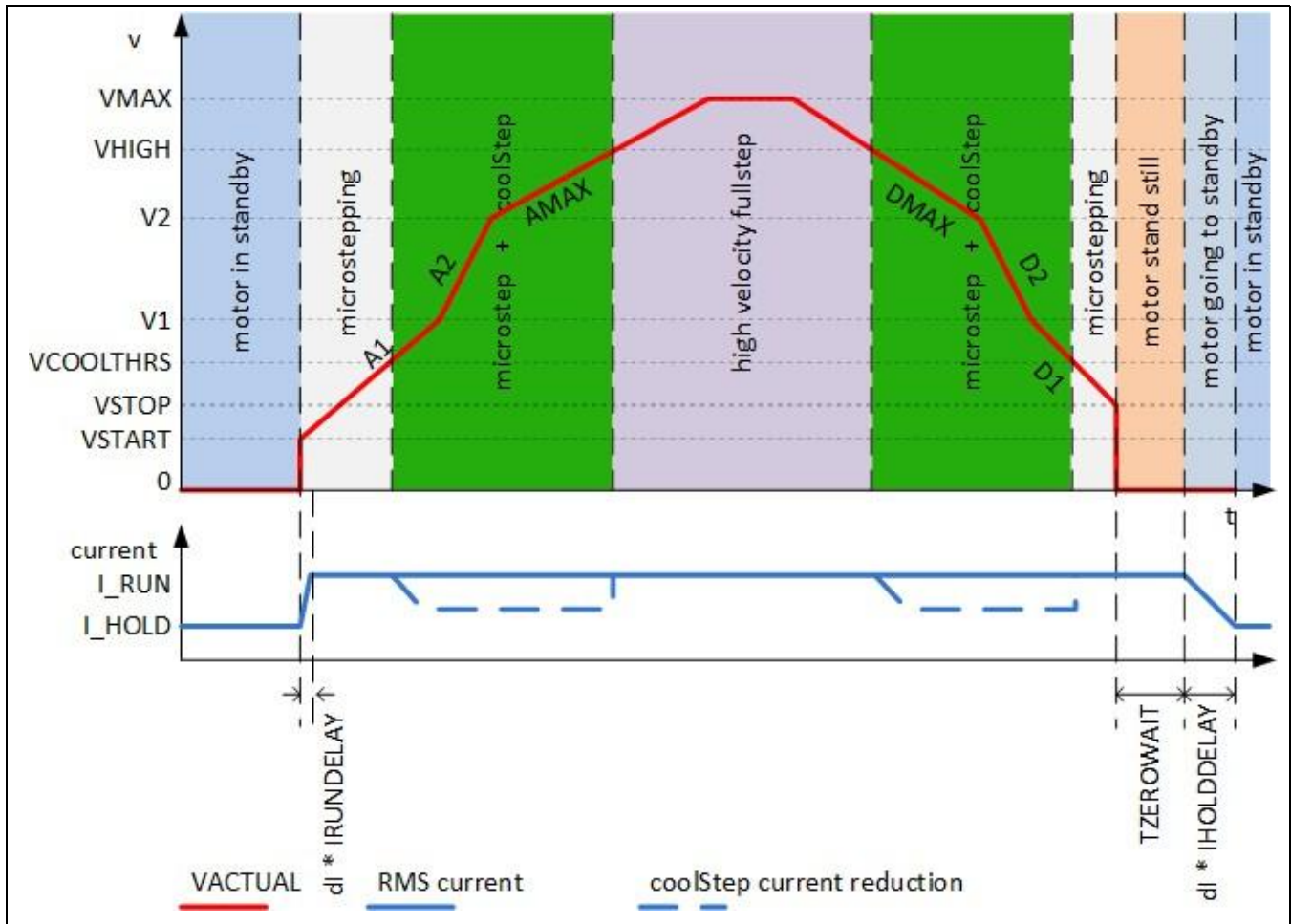


图25. 斜坡发生器速度依赖电机控制

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。

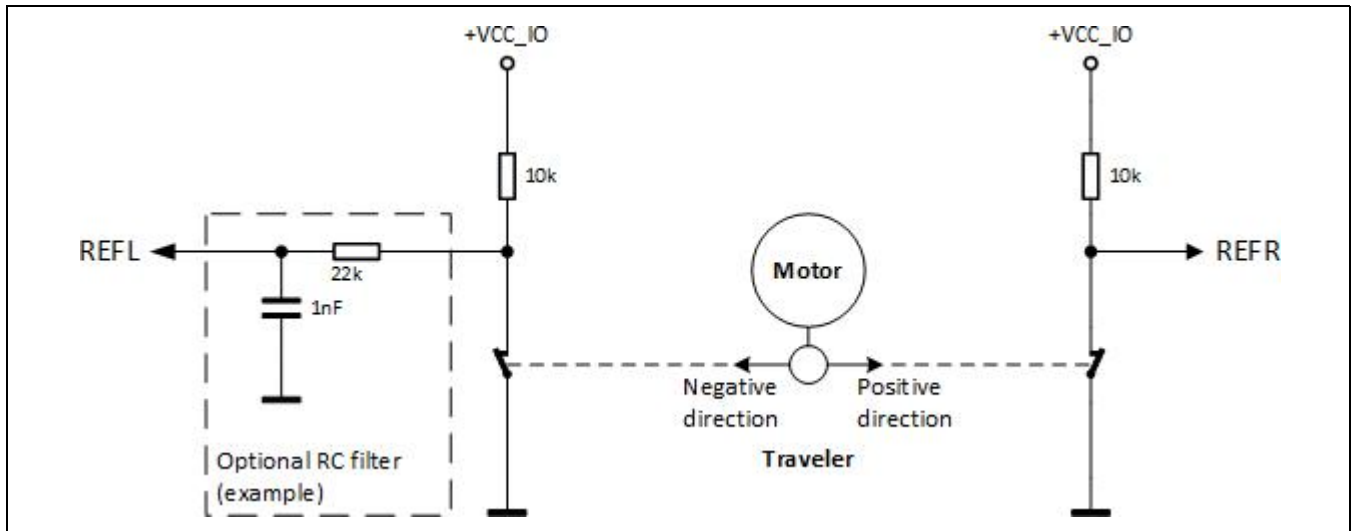


图 26. 使用参考开关 (示例)

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)。

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）。

- 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。
- 一旦开关被触发，读出 **XLATCH**。它给出了开关事件的位置。
- 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）

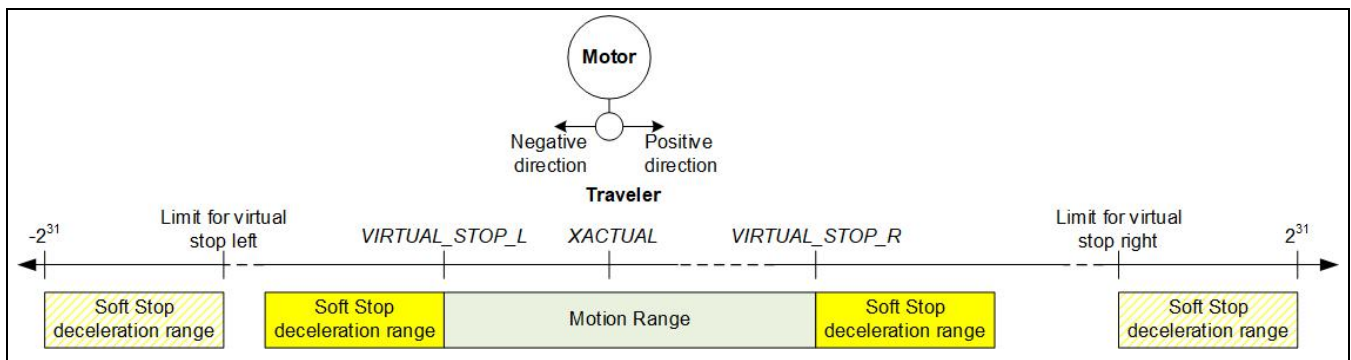
虚拟参考开关

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：

- 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。
- 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]

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！

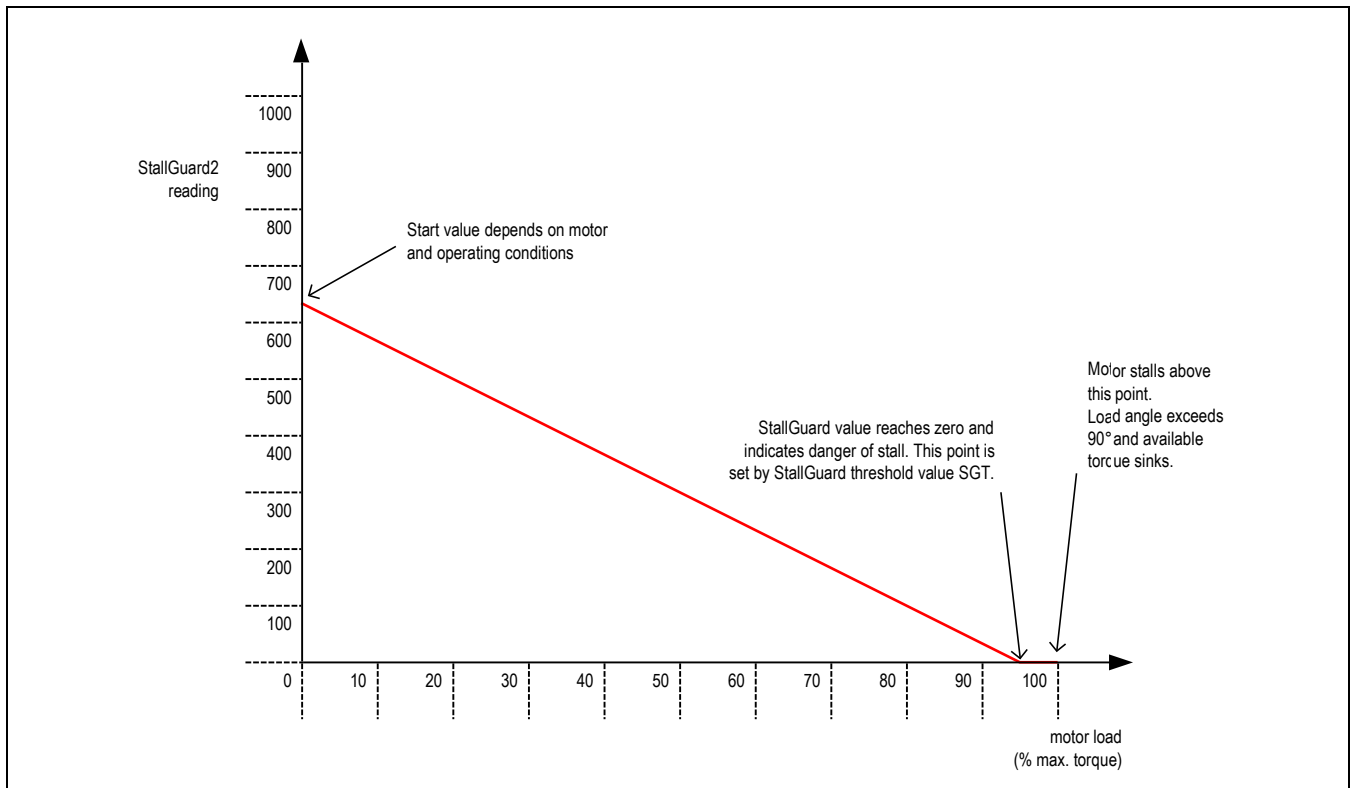


图 27 StallGuard2 功能原理

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

表 16. StallGuard2 相关参数（待续）

<i>sfilt</i>	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
<i>SG_RESULT</i>	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 <i>SG_RESULT</i> 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

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。

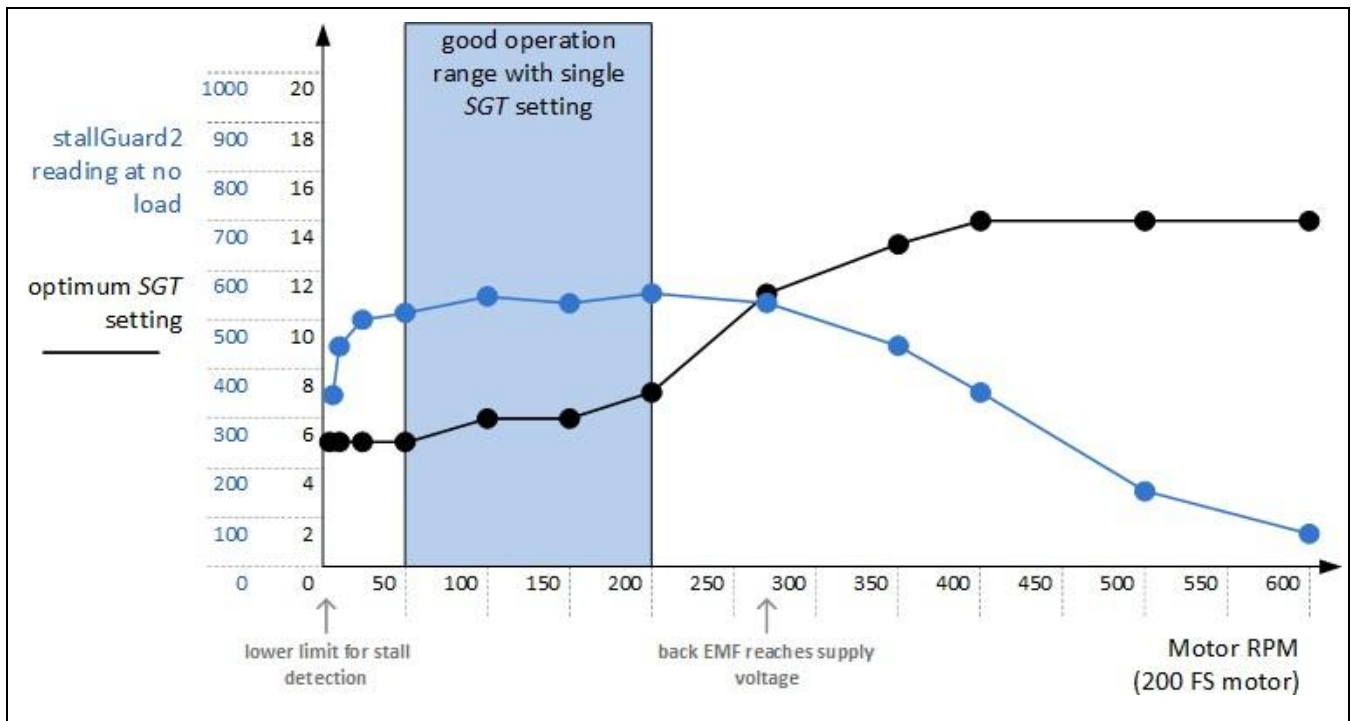


图28。例子:一个例子电机下的最佳SGT设置和StallGuard2读取

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。

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：

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

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 (sfil=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，电机将振动并显示更高的 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。

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。

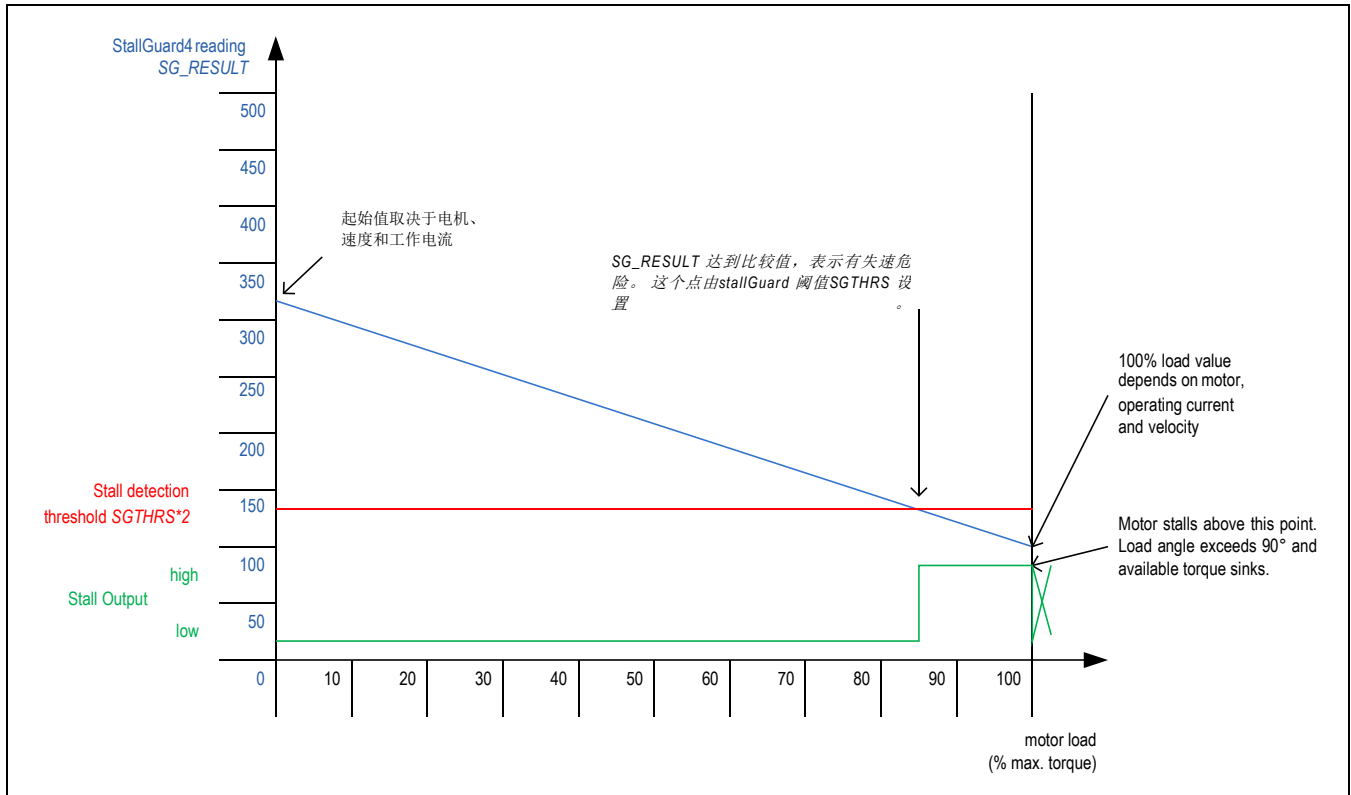


图 29. StallGuard4 操作模式

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	将该值的两倍与 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

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 $TCOOLTHRS \geq TSTEP > TPWMTHRS$ 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。

Setting up for CoolStep

CoolStep is controlled by several parameters, but two are critical for understanding how it works:

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。

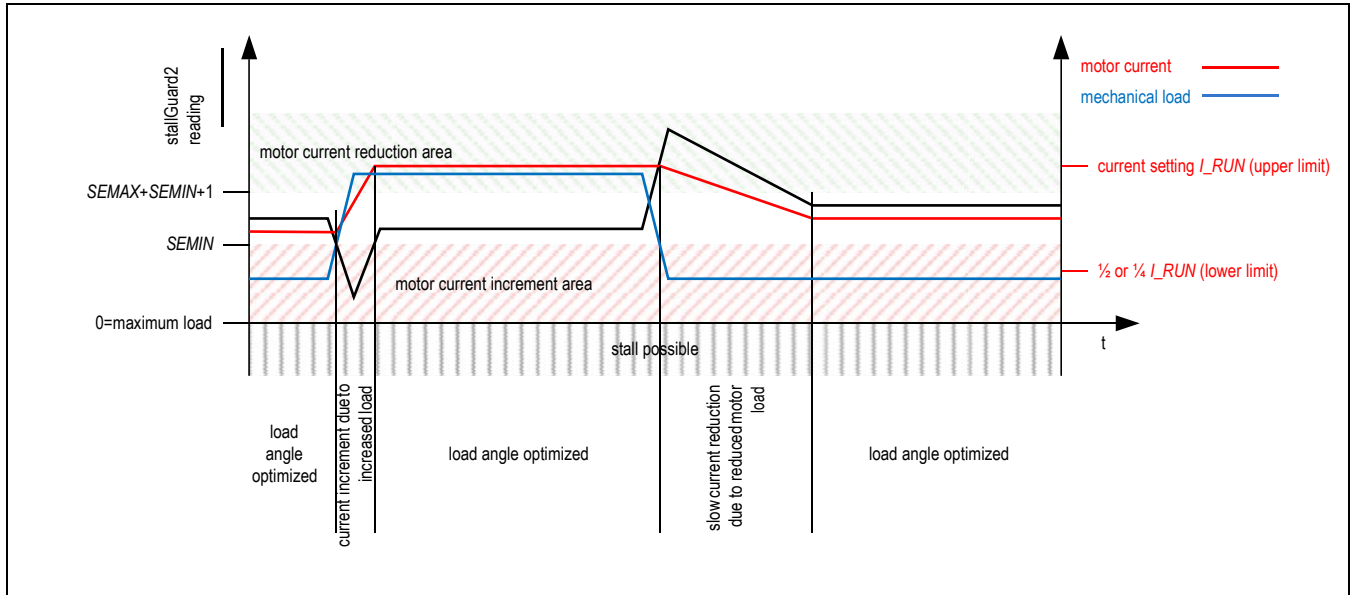


图 30. CoolStep 使电机电流适应负载

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，这会影响到负载测量值 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。

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。

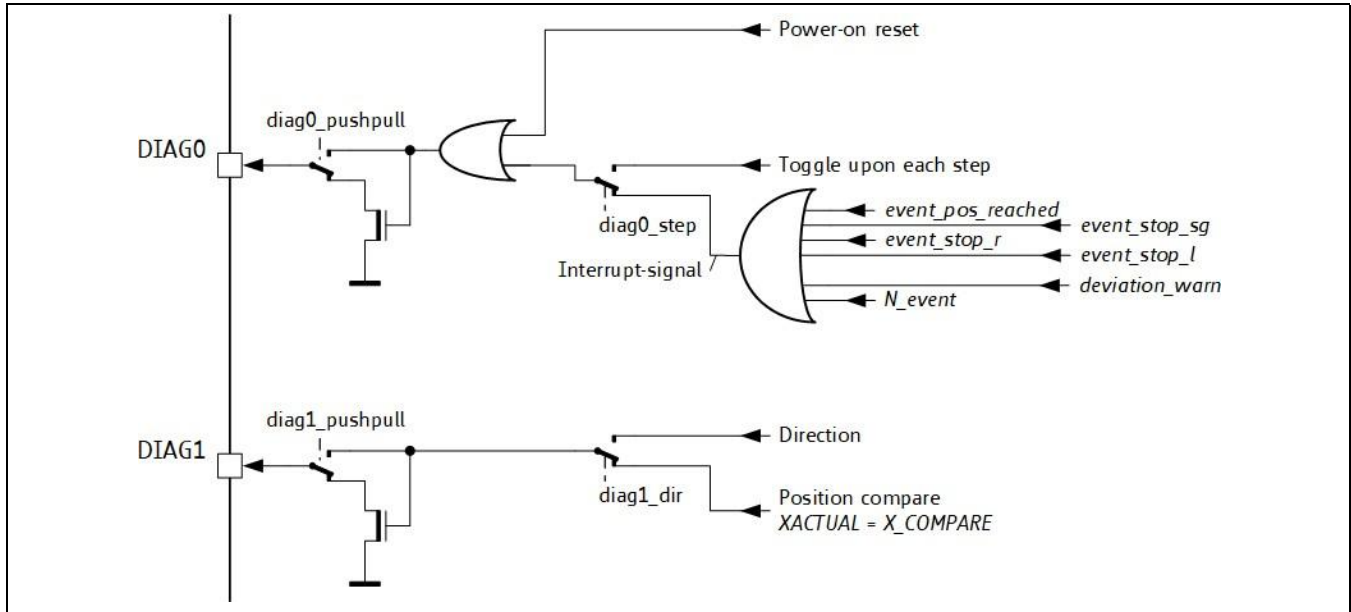
建议：最常见和最有益的用途是将 CoolStep 用于典型系统目标操作速度的控制，并据此设置速度阈值。由于加速和减速通常应该很快，它们将需要满电机电流，而由于它们的持续时间短，它们对整体功耗的影响很小。

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。



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。

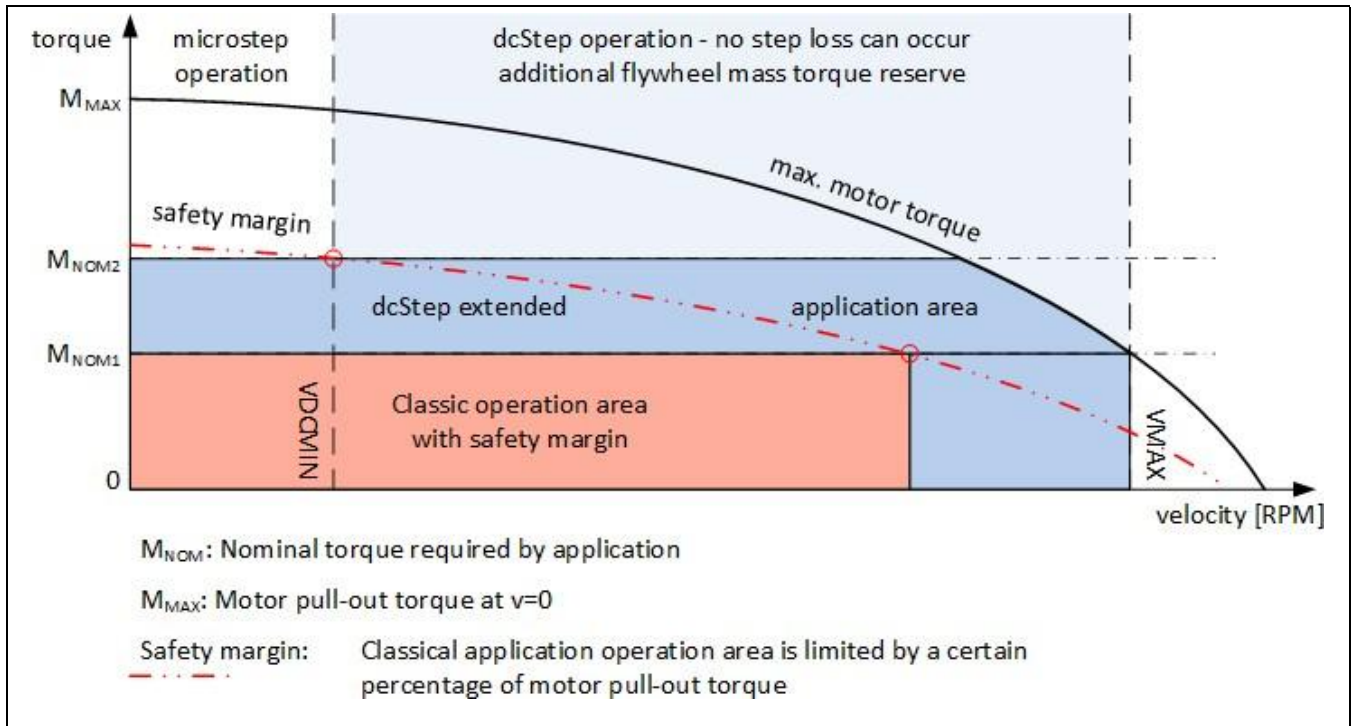


图 31. DcStep 扩展应用程序操作区域

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 V_{ACTUAL} 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。

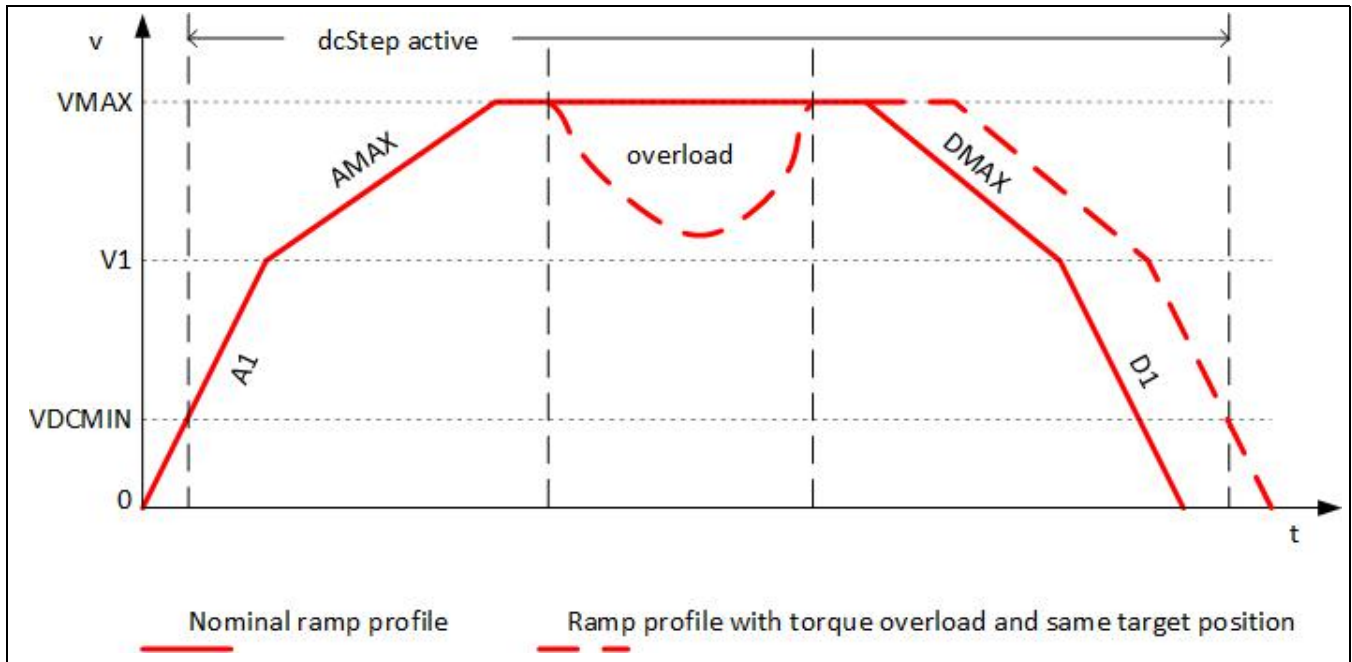


图 32. 受过载情况影响的速度曲线

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 & vhighcm	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。

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：

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

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好起点。将表重新编程为电机特定的波形可以显著改进微步，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”。

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 W_x 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。

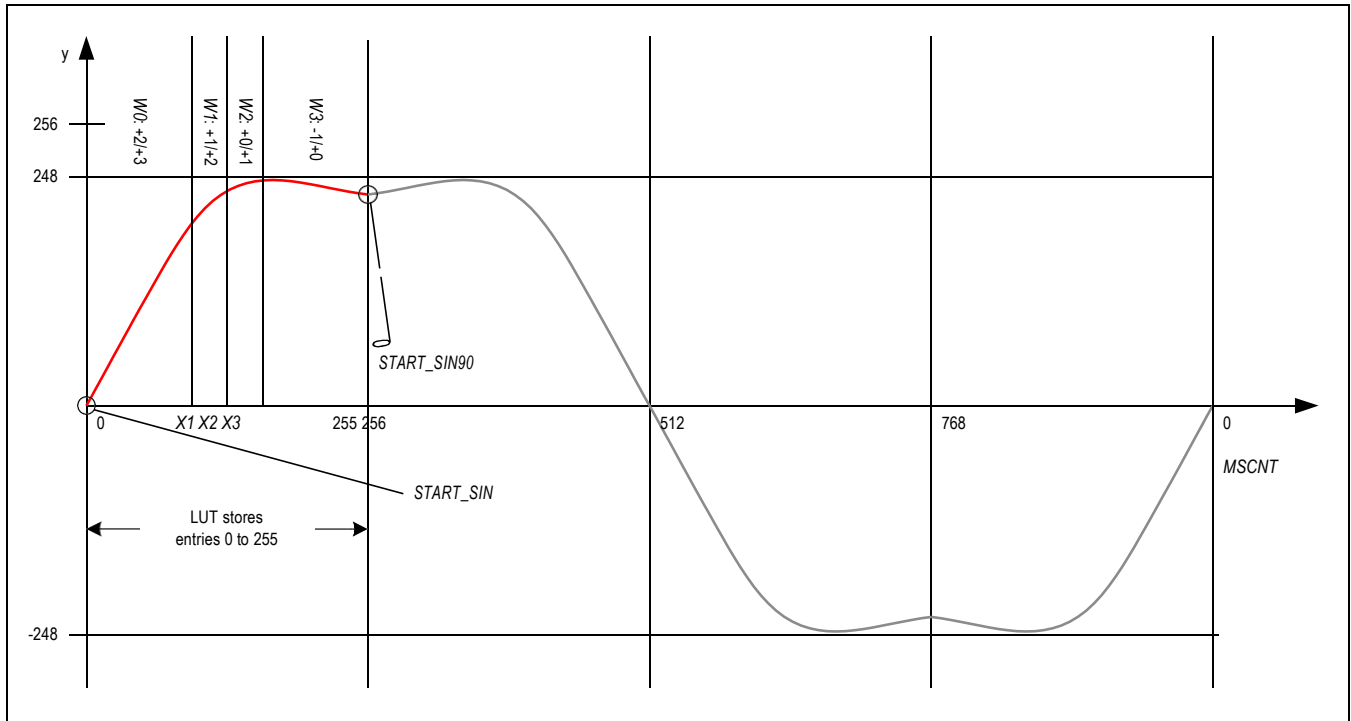


图 33. LUT 编程示例

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。

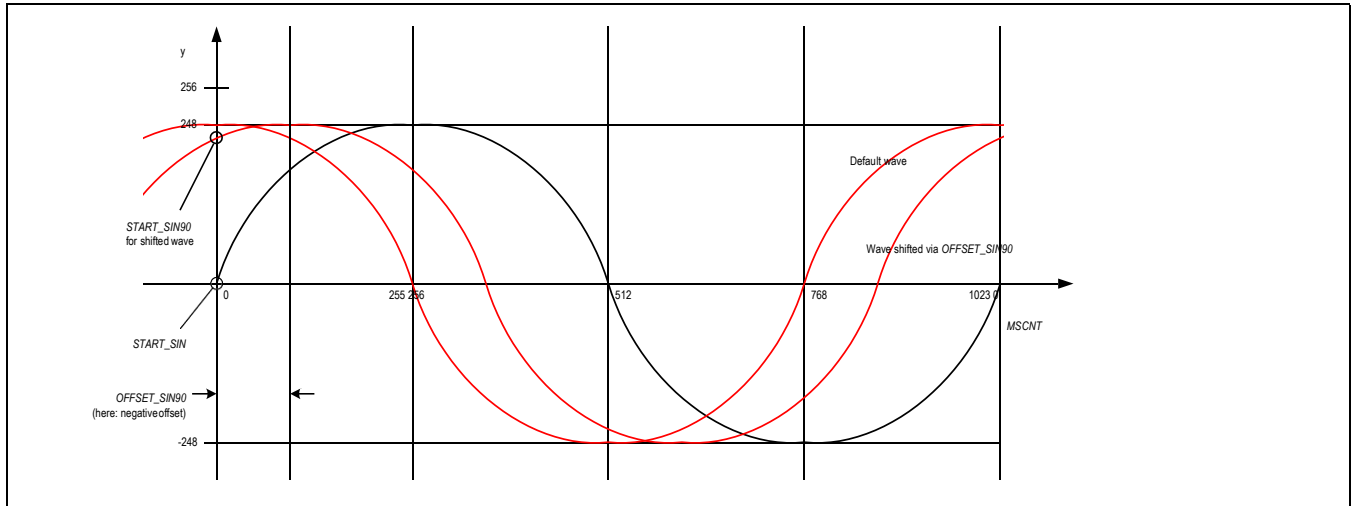


图 34. 通过 `OFFSET_SIN90` 移动余弦波

默认表是实现该表的一个很好的基础。这是一个复位默认微步表的初始化示例：

```
MSLUT[0]= %1010101010101010101010101010100 = 0xAAAAB554
MSLUT[1]= %01001010100101010101010010101010 = 0x4A9554AA
MSLUT[2]= %0010010001001001001001001001001 = 0x24492929
MSLUT[3]= %00010000000100000100001000100010 = 0x10104222
MSLUT[4]= %11111011111111111111111111111111 = 0xFBFFFFFF
MSLUT[5]= %101101011011101101101110110111101 = 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。

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`。

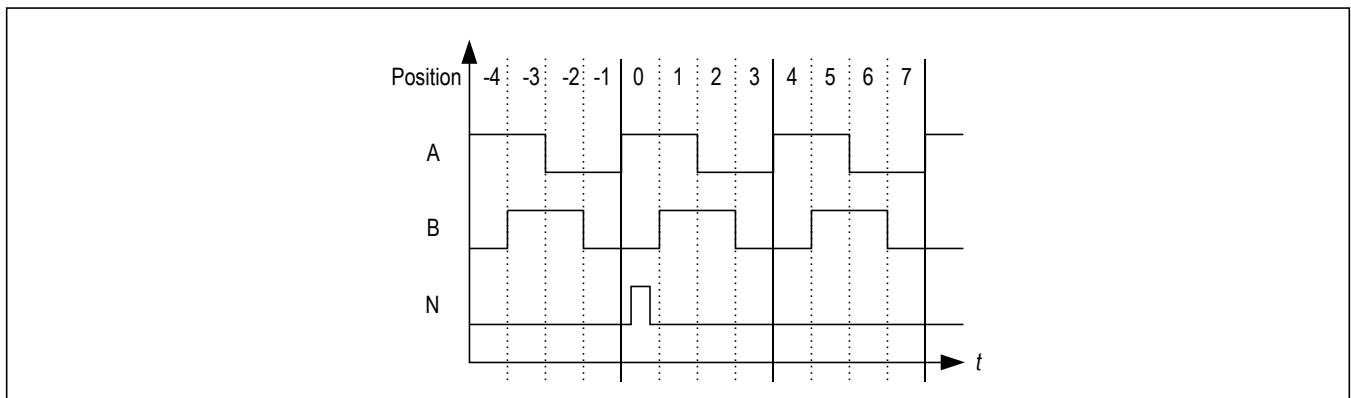


图 35. 增量编码器的 ABN 信号示意图

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 $(2^{16} - (FACTOR+1)) \cdot (2^{16} - FRACTION)$
- 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: $(2^{16} - (25+1)) \cdot (10000 - 6000) = (2^{16} - 26) \cdot (4000) = 0xFFE6.0x0FA0$
 - 使用以下等式计算负编码器常数: $(2^{16} - (FACTOR+1)) \cdot (10000 - DECIMALS)$

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。

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 μ steps, 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

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

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 >30 μ s 的短脉冲会导致芯片复位 (在诊断输出中也可见)。

<30 μ s 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。

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 OVERTEMPREWARNING_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。

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：

$$\text{ADC_TEMP} = 7.7 * \text{TEMP} + 2038$$

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

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。

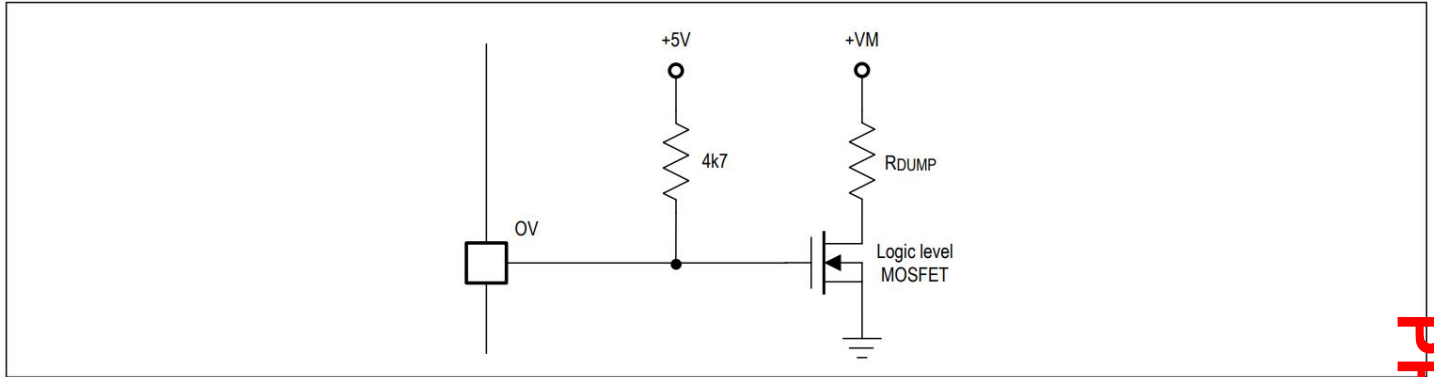


图 36. 制动斩波电路示例

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 `SpreadCycle` 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。

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！

Table 21. Overview of Register Map

Register	Description
General Configuration Registers	These registers contain <ul style="list-style-type: none"> • global configuration • global status flags • interface configuration • and I/O signal configuration
Ramp Generator Motion Control Register Set	This register set offers registers for <ul style="list-style-type: none"> • choosing a ramp mode • choosing velocities • homing • acceleration and deceleration • target positioning • reference switch and StallGuard2 event configuration • ramp and reference switch status
Velocity Dependent Driver Feature Control Register Set	This register set offers registers for <ul style="list-style-type: none"> • driver current control • setting thresholds for CoolStep operation • setting thresholds for different chopper modes • setting thresholds for DcStep operation
Direct Mode Registers	This register group offers registers used for the direct coil current control mode.
Encoder Register Set	The encoder register set offers all registers needed for proper ABN encoder operation.
ADC Registers	This register group offers registers to control and read the internal ADC.

Table 21. Overview of Register Map (continued)

Motor Driver Register Set	This register set offers registers for <ul style="list-style-type: none">• setting / reading out microstep table and counter• chopper and driver configuration• CoolStep and StallGuard configuration• DcStep configuration• reading out StallGuard values and driver error flags
---------------------------	---

Register Map

TMC5240

ADDRESS	NAME	MSB							LSB	
General Configuration Registers										
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_np_oscomp_dir	
	GCONF[7:0]	diag0_int_step	-	-	shaft	-	en_pwm_mode	fast_stall	-	
0x01	GSTAT[7:0]					uv_cp	drv_err	reset		
0x02	IFCNT[7:0]	IFCNT[7:0]								
0x03	SLAVECONF[15:8]					SENDDDELAY[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_EN	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]								
Velocity Dependent Configuration Registers										
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]								

ADDRESS	NAME	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]
Ramp Generator Registers								
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]

ADDRESS	NAME	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]
Ramp Generator Driver Feature Control Registers									
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_renable	stop_lenable
0x35	RAMP_STAT[15:8]	status_virtual_stop_r	status_virtual_stop_l	status_sing	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]
Encoder Registers									
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_A_B	pol_N	pol_B	pol_A	
0x39	X_ENC[31:24]								X_ENC[31:24]
	X_ENC[23:16]								X_ENC[23:16]
	X_ENC[15:8]								X_ENC[15:8]
	X_ENC[7:0]								X_ENC[7:0]
0x3A	ENC_CONST[31:24]								ENC_CONST[31:24]
	ENC_CONST[23:16]								ENC_CONST[23:16]
	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	

ADDRESS	NAME	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 Registers							
0x50	ADC_VSUPPLY_AIN[31:24]	ADC_AIN[12:8]					
	ADC_VSUPPLY_AIN[23:16]	ADC_AIN[7:0]					
	ADC_VSUPPLY_AIN[15:8]	-	-	-	ADC_VSUPPLY[12:8]		
	ADC_VSUPPLY_AIN[7:0]	ADC_VSUPPLY[7:0]					
0x51	ADC_TEMP[31:24]	RESERVED[12:8]					
	ADC_TEMP[23:16]	RESERVED[7:0]					
	ADC_TEMP[15:8]	-	-	-	ADC_TEMP[12:8]		
	ADC_TEMP[7:0]	ADC_TEMP[7:0]					
0x52	OTW_OV_VTH[31:24]	OVERTEMPPREWARNING_VTH[12:8]					
	OTW_OV_VTH[23:16]	OVERTEMPPREWARNING_VTH[7:0]					
	OTW_OV_VTH[15:8]	-	-	-	OVERVOLTAGE_VTH[12:8]		
	OTW_OV_VTH[7:0]	OVERVOLTAGE_VTH[7:0]					
Motor Driver Registers							
0x60	MSLUT_0[31:24]	MSLUT_0[31:24]					
	MSLUT_0[23:16]	MSLUT_0[23:16]					
	MSLUT_0[15:8]	MSLUT_0[15:8]					
	MSLUT_0[7:0]	MSLUT_0[7:0]					
0x61	MSLUT_1[31:24]	MSLUT_1[31:24]					
	MSLUT_1[23:16]	MSLUT_1[23:16]					

ADDRESS	NAME	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]	

ADDRESS	NAME	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 \(0x0\)](#)

Global Configuration Flags

PRELIMINARY CONFIDENTIAL

BIT				20	19	18	17	16
Field	length_step_pulse[3:0]							direct_mode
Reset	0x0							0x0
Access Type	Write, Read							Write, Read
BIT	15	14	13	12	11	10	9	8
Field	stop_enable	small_hysteresis	diag1_poscomp_pushpull	diag0_int_pushpull	–	–	–	diag1_nposcomp_dir
Reset	0x0	0x0	0x0	0x0	–	–	–	0x0
Access Type	Write, Read	Write, Read	Write, Read	Write, Read	–	–	–	Write, Read
BIT	7	6	5	4	3	2	1	0
Field	diag0_nint_step	–	–	shaft	–	en_pwm_mode	fast_standstill	–
Reset	0x0	–	–	0x0	–	0x0	0x0	–
Access Type	Write, Read	–	–	Write, Read	–	Write, Read	Write, Read	–
BITFIELD	BITS	DESCRIPTION			DECODE			
length_step_pulse	20:17	cDriver only: length_step_pulse = 0: STEP output toggles upon each step; length_step_pulse = 1...15: STEP pin high time in number of clock cycles						
direct_mode	16	Enable direct motor phase current control via serial interface.			0x0: Normal operation 0x1: Motor coil currents and polarity directly programmed via serial interface: Register XTARGET (0x2D) specifies signed coil A current (bits 8..0) and coil B current (bits 24..16). In this mode, the current is scaled by I _{HOLD} setting. Velocity based current regulation of StealthChop is not available in this mode. The automatic StealthChop current regulation will work only for low stepper motor velocities.			
stop_enable	15	Motor hard stop function enable.			0x0: Normal operation 0x1: Emergency stop: ENCA stops the sequencer when tied high (no steps become executed by the sequencer, motor goes to standstill state).			
small_hysteresis	14				0x0: Hysteresis for step frequency comparison is 1/16 0x1: Hysteresis for step frequency comparison is 1/32			
diag1_poscomp_pushpull	13	DIAG1 output type configuration.			0x0: DIAG1 is open collector output (active low) 0x1: Enable DIAG1 push pull output (active high)			
diag0_int_pushpull	12	DIAG0 output type configuration.			0x0: DIAG0_SW is open collector output (active low) 0x1: Enable DIAG0_SW push pull output (active high)			
diag1_nposcomp_dir	8	DIAG1 output configuration, when not using UART.			0x0: DIAG1 outputs position compare signal 0x1: Enable DIAG1 as DIR output for external STEP/DIR driver			

BITFIELD	BITS	DESCRIPTION	DECODE
diag0_nint_step	7	DIAG0 output configuration.	0x0: DIAG0 outputs interrupt signal 0x1: Enable DIAG0 as STEP output (half frequency, dual edge triggered or frequency when length_step_pulse != 0) for external STEP/DIR driver
shaft	4	Change motor direction / direction sign	0x0: Default motor direction 0x1: Inverse motor direction
en_pwm_mode	2	Enable the stealthChop(TM) mode	0x0: no stealthChop 0x1: StealthChop voltage PWM mode enabled (depending on velocity thresholds). Switch from off to on state while in stand-still and at IHOLD=nominal IRUN current, only.
fast_standstill	1	Timeout for step execution until standstill detection	0x0: Normal time: 2 ²⁰ clocks 0x1: Short time: 2 ¹⁸ clocks

GSTAT (0x1)

Global Status Flags

(Re-Write with '1' bit to clear respective flags)

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 undervoltage has occurred since last reset.	
register_reset	3		0x0: normal operation 0x1: Indicates that the registermap has been reset. All registers have been cleared to reset values.
uv_cp	2	Charge pump undervoltage condition flag	0x0: normal operation 0x1: Indicates an undervoltage on the charge pump. The driver is disabled during undervoltage. This flag is latched for information.
drv_err	1	Driver error flag	0x0: normal operation 0x1: Indicates, that the driver has been shut down due to overtemperature or short circuit detection. Read DRV_STATUS for details. The flag can only be cleared when the temperature is below the limit again
reset	0	Reset flag	0x0: normal operation 0x1: Indicates that the IC has been reset.

IFCNT (0x2)

Interface transmission counter.

This register becomes incremented with each successful UART interface write access. It can be read out to check the serial transmission for lost data. Read accesses do not change the content. Disabled in SPI operation. The counter wraps around from 255 to 0.

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	Interface transmission counter. This register becomes incremented with each successful UART interface write access. It can be read out to check the serial transmission for lost data. Read accesses do not change the content. Disabled in SPI operation. The counter wraps around from 255 to 0.

SLAVECONF (0x3)

BIT	11	10	9	8
Field	SENDDDELAY[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
SENDDDELAY	11:8	SWUART Slave Configuration	0x0: 8 bit times (not allowed with multiple slaves) 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: These eight bits set the address of unit for the UART interface. The address becomes incremented by one up to seven as defined by SDI, SCK, CSN. CSN, SCK, SDI 000: +0 001: +1 010: +2 011: +3 100: +4 101: +5 110: +6 111: +7 Range: 0-254 (do not increment beyond 254)	

IOIN (0x4)

Reads the state of all input pins available and returns IC revision in highest byte

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 = first version of the IC Identical numbers mean full digital compatibility.
SILICON_RV	18:16	Silicon revision number
ADC_ERR	15	1: Signals that the ADC is not working correctly. Do not utilize ADC-features.
EXT_CLK	14	0: The internal oscillator is used for generating the clock-signal (12.5 MHz). 1: The external oscillator is used for generating the clock-signal.
EXT_RES_DET	13	1: External resistor between REF and GND 0: No external resistor detected
OUTPUT	12	Output polarity of SDO pin when UART is enabled via pin UART_EN. Its main purpose it to use SDO as NAO next address output signal for chain addressing of multiple ICs. Attention: Reset Value is 1 for use as NAO to next IC in single wire chain
COMP_B1_B2	11	COMP_B1_B2 (StallGuard4 comparator B, for IC test)
COMP_A1_A2	10	COMP_A1_A2 (StallGuard4 comparator A, for IC test)
COMP_B	9	COMP_B (chopper comparator B, for IC test)
COMP_A	8	COMP_A (chopper comparator A, for IC test)
reserved	7	
UART_EN	6	1 = UART interface is enabled
ENCN	5	N-channel state
DRV_ENN	4	Driver disabled/enabled state.
ENCA	3	A-channel state
ENCB	2	B-channel state
REFR	1	

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	<p>Position comparison register for motion controller position strobe. X_COMPARE is an absolute position. The Position pulse is available on output SWP_DIAG1.</p> <p><i>XACTUAL</i> = X_COMPARE: Output signal PP (position pulse) becomes high. It returns to a low state, if the positions mismatch.</p> <p>If X_COMPARE_REPEAT is >1, X_COMPARE is the position reference for the periodic position strobe trigger output.</p>

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							

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		<p>This register defines a relative distance in microsteps (based on MRES configuration).</p> <p>If set to >1, the position compare pulse is raised every time a multiple of X_PDISTANCE μsteps have been made.</p> <p>Thereby, the X_COMPARE register defines the base position for the modulo calculation of X_PDISTANCE steps have been made into positive or negative direction.</p>					

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			0x0: 100V/ μ s 0x1: 200V/ μ s 0x2: 400V/ μ s 0x3: 800V/ μ s		
CURRENT_RANGE	1:0		This setting allows a basic adaptation of the drivers RDSon current sensing to the motor current range. Select the lowest fitting range for best current precision. The value is the peak current setting.			0x0: 1A 0x1: 2A 0x2: 3A 0x3: 3A		

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	<p>Global scaling of Motor current. This value is multiplied to the current scaling in order to adapt a drive to a certain motor type. This value should be chosen before tuning other settings, because it also influences chopper hysteresis. This value is just intended for finetuning the motor current.</p> <p>0: Full Scale (or write 256) 1 ... 31: Not allowed for operation 32 ... 255: 32/256 ... 255/256 of maximum current.</p> <p><i>Hint:</i> Values >128 recommended for best results</p>

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		<p>Controls the number of clock cycles for motor power up after start is detected. 0: instant power up 1..15: Delay per current increment step in multiple of IRUNDELAY * 512 clocks</p>					

BITFIELD	BITS	DESCRIPTION
IHOLDDELAY	19:16	Controls the number of clock cycles for motor power down after a motion as soon as standstill is detected (<i>stst</i> =1) and <i>TPOWERDOWN</i> has expired. The smooth transition avoids a motor jerk upon power down. 0: instant power down 1..15: Delay per current reduction step in multiple of 2^{18} clocks
IRUN	12:8	Motor run current (0=1/32...31=32/32) <i>Hint:</i> Use a setting between 16 to 31 for best microstep performance.
IHOLD	4:0	Standstill current (0=1/32...31=32/32) In combination with StealthChop mode, setting <i>IHOLD</i> =0 allows to choose freewheeling or coil short circuit for motor stand still.

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	<i>TPOWERDOWN</i> sets the delay time after stand still (<i>stst</i>) of the motor to motor current power down. Time range is about 0 to 4 seconds. <i>Attention:</i> A minimum setting of 2 is required to allow automatic tuning of StealthChop PWM_OFFS_AUTO. Reset Default = 10 $0 \dots ((2^8) - 1) * 2^{18} t_{CLK}$

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							

BITFIELD	BITS	DESCRIPTION
TSTEP	19:0	<p>Actual measured time between two 1/256 microsteps derived from the step input frequency in units of 1/fCLK. Measured value is $(2^{20})-1$ in case of overflow or stand still.</p> <p>All TSTEP related thresholds use a hysteresis of 1/16 of the compare value to compensate for jitter in the clock or the step frequency. The flag <i>small_hysteresis</i> modifies the hysteresis to a smaller value of 1/32. $(T_{xxx} \times 15/16) - 1$ or $(T_{xxx} \times 31/32) - 1$ is used as a second compare value for each comparison value.</p> <p>This means, that the lower switching velocity equals the calculated setting, but the upper switching velocity is higher as defined by the hysteresis setting.</p> <p>When working with the motion controller, the measured TSTEP for a given velocity V is in the range $(2^{24} / V) \leq TSTEP \leq 2^{24} / V - 1$.</p> <p>In DcStep mode TSTEP will not show the mean velocity of the motor, but the velocities for each microstep, which may not be stable and thus does not represent the real motor velocity in case it runs slower than the target velocity.</p>

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	<p>This is the upper velocity for StealthChop voltage PWM mode.</p> <p>$TSTEP \geq TPWMTHRS$</p> <ul style="list-style-type: none"> StealthChop PWM mode is enabled, if configured DcStep is disabled 						

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	<p>This is the lower threshold velocity for switching on smart energy CoolStep and StallGuard feature. (unsigned)</p> <p>Set this parameter to disable CoolStep at low speeds, where it cannot work reliably. The stop on stall function (enable with <i>sg_stop</i> when using internal motion controller) and the stall output signal become enabled when exceeding this velocity. In non-DcStep mode, it becomes disabled again once the velocity falls below this threshold.</p> <p><i>TCOOLTHRS</i> ≥ <i>TSTEP</i> ≥ <i>THIGH</i>:</p> <ul style="list-style-type: none"> CoolStep is enabled, if configured <p><i>TCOOLTHRS</i> ≥ <i>TSTEP</i></p> <ul style="list-style-type: none"> Stop on stall is enabled, if configured Stall output signal (DIAG0/1) is enabled, if configured

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							

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	<p>This velocity setting allows velocity dependent switching into a different chopper mode and fullstepping to maximize torque. (unsigned) The stall detection feature becomes switched off for 2-3 electrical periods whenever passing <i>THIGH</i> threshold to compensate for the effect of switching modes.</p> <p><i>TSTEP</i> ≤ <i>THIGH</i>:</p> <ul style="list-style-type: none"> • CoolStep is disabled (motor runs with normal current scale) • StealthChop voltage PWM mode is disabled • If <i>vhighchm</i> is set, the chopper switches to <i>chm</i>=1 with <i>TFD</i>=0 (constant off time with slow decay, only). • If <i>vhighfs</i> is set, the motor operates in fullstep mode and the stall detection becomes switched over to DcStep stall detection.

RAMPMODE (0x20)

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

BITFIELD	BITS	DESCRIPTION	DECODE
RAMPMODE	1:0	Motion Controller ramping mode	0x0: Positioning mode (using all A, D and V parameters) 0x1: Velocity mode to positive <i>VMAX</i> (using <i>AMAX</i> acceleration) 0x2: Velocity mode to negative <i>VMAX</i> (using <i>AMAX</i> acceleration) 0x3: Hold mode (velocity remains unchanged, unless stop event occurs)

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							

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		Actual motor position (signed) <i>Hint:</i> This value normally should only be modified, when homing the drive. In positioning mode, modifying the register content will start a motion.					

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		Actual motor velocity from ramp generator (signed) The sign matches the motion direction. A negative sign means motion to lower <i>XACTUAL</i> . $\pm(2^{23})-1$ $[\mu\text{steps} / \text{t}]$					

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		Motor start velocity (unsigned) For universal use, set $VSTOP \geq VSTART$. This is not required if the motion distance is sufficient to ensure deceleration from $VSTART$ to $VSTOP$. $0 \dots (2^{18})-1$ $[\mu\text{steps} / t]$					

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		First acceleration between $VSTART$ and $V1$ (unsigned) $0 \dots (2^{18})-1$ $[\mu\text{steps} / \text{ta}^2]$					

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		First acceleration / deceleration phase threshold velocity (unsigned) 0: Disables A1 and D1 phase, use AMAX, DMAX only 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		Second acceleration between V1 and VMAX (unsigned) This is the acceleration and deceleration value for velocity mode. 0...(2 ¹⁸)-1 [μsteps / ta ²]					

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		Motion ramp target velocity (for positioning ensure $VMAX \geq VSTART$) (unsigned) This is the target velocity in velocity mode. It can be changed any time during a motion. $0 \dots (2^{23}) - 512$ [μ steps / 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		Deceleration between $VMAX$ and $V1$ (unsigned) $0 \dots (2^{18}) - 1$ [μ steps / ta^2]					

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	<p>Minimum time for constant velocity segments in multiple of 512 clocks.</p> <p>0: Disables minimum duration setting for constant velocity phase >0: A minimum duration of constant velocity is inserted in between of any change from acceleration to deceleration or vice versa to reduce jerk</p> <p>$0 \dots (2^{16}) - 1 * 512 t_{CLK}$</p> <p>Note: Configure this register after setting VMAX when in Positionmode and standstill. Set TVMAX=0 during velocity mode to avoid triggering the TVMAX delay when switching back to ramp mode.</p>

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	<p>Deceleration between V1 and VSTOP (unsigned)</p> <p>Attention: Do not set 0 in positioning mode, even if V1=0!</p> <p>$1 \dots (2^{16}) - 1$ $[\mu\text{steps} / \text{ta}^2]$ Reset Default = 10</p>

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	Motor stop velocity (unsigned) <i>Hint:</i> Set VSTOP ≥ VSTART to allow positioning for short distances <i>Attention: Do not set 0 in positioning mode, minimum 10 recommend!</i> $1 \dots (2^{18}) - 1$ [μsteps / t] Reset Default=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	Defines the waiting time after ramping down to zero velocity before next movement or direction inversion can start. Time range is about 0 to 2 seconds. This setting avoids excess acceleration e.g. from VSTOP to -VSTART. $0 \dots (2^{16}) - 1 * 512 t_{CLK}$

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		<p>Target position for ramp mode (signed). Write a new target position to this register in order to activate the ramp generator positioning in <i>RAMPMODE</i>=0. Initialize all velocity, acceleration and deceleration parameters before.</p> <p><i>Hint:</i> The position is allowed to wrap around, thus, <i>XTARGET</i> value optionally can be treated as an unsigned number.</p> <p><i>Hint:</i> The maximum possible displacement is $\pm((2^{31})-1)$.</p> <p><i>Hint:</i> When increasing <i>V1</i>, <i>D1</i> or <i>DMAX</i> during a motion, rewrite <i>XTARGET</i> afterwards in order to trigger a second acceleration phase, if desired.</p> <p>-2³¹... +(2³¹)-1</p>					

V2 (0x2E)

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

BIT	15	14	13	12	11	10	9	8
Field	V2[15:8]							
Reset	0x0							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	V2[7:0]							
Reset	0x0							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
V2	19:0		Velocity difference from VMAX for activation of acceleration segments with AMAX/2 and DMAX/2. 0: Disables AMAX/2 and DMAX/2 phase, use AMAX, DMAX only 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		Acceleration between V1 and V2 (unsigned) 0...(2 ¹⁸)-1 [μsteps / ta ²]					

D2 (0x30)

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

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	Deceleration between V2 and V1 (unsigned) <i>Attention: Do not set 0 in positioning mode, even if V2=0!</i> 1...(2 ¹⁸)-1 [μsteps / ta ²] Reset Default = 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	Automatic commutation DcStep becomes enabled above velocity <i>VDCMIN</i> (unsigned) In this mode, the actual position is determined by the sensorless motor commutation and becomes fed back to <i>XACTUAL</i> . In case the motor becomes heavily loaded, <i>VDCMIN</i> also is used as the minimum step velocity. Activate stop on stall (<i>sg_stop</i>) to detect step loss. 0: Disable, DcStep off VACT ≥ VDCMIN ≥ 256: <ul style="list-style-type: none"> • Triggers the same actions as exceeding <i>THIGH</i> setting. • Switches on automatic commutation DcStep <i>Hint: Also set DCCTRL parameters in order to operate DcStep.</i> (Only bits 22... 8 are used for value and for comparison)
reserved	7:0	reads always 0

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	Source for virtual stop (VIRTUAL_STOP_L and VIRTUAL_STOP_R) 0: Virtual stop relates to ramp generator position <i>XACTUAL</i> 1: Virtual stop relates to encoder position <i>X_ENC</i>	
en_virtual_stop_r	13	1: Enables automatic motor stop during active right virtual stop condition	
en_virtual_stop_l	12	1: Enables automatic motor stop during active left virtual stop condition	
en_softstop	11	0: Hard stop 1: Soft stop The soft stop mode always uses the deceleration ramp settings <i>DMAX</i> , <i>V1</i> , <i>D1</i> , <i>V2</i> , <i>D2</i> , <i>VSTOP</i> and <i>TZEROWAIT</i> for stopping the motor. A stop occurs when the velocity sign matches the reference switch position (<i>REFL</i> , <i>VIRTUAL_STOP_L</i> for negative velocities, <i>REFR</i> , <i>VIRTUAL_STOP_R</i> for positive velocities) and the respective switch stop function is enabled. A hard stop also uses <i>TZEROWAIT</i> before the motor becomes released. <i>Attention: Do not use soft stop in combination with StallGuard2. Use soft stop for StealthChop operation at high velocity. In this case, hard stop must be avoided, as it could result in severe overcurrent.</i>	0x0: Hard stop 0x1: Soft stop

BITFIELD	BITS	DESCRIPTION	DECODE
sg_stop	10	Enable stop by StallGuard2 (also available in DcStep mode). Disable to release motor after stop event. Program <i>TCOOLTHRS</i> for velocity threshold. <i>Hint:</i> Do not enable during motor spin-up, wait until the motor velocity exceeds a certain value, where StallGuard2 delivers a stable result. This velocity threshold should be programmed using <i>TCOOLTHRS</i> .	0x0: disabled 0x1: enabled
en_latch_encoder	9	1: Latch encoder position to <i>ENC_LATCH</i> upon reference switch event.	
latch_r_inactive	8	1: Activates latching of the position to <i>XLATCH</i> upon an inactive going edge on the right reference switch input REFR. The active level is defined by <i>pol_stop_r</i> .	
latch_r_active	7	1: Activates latching of the position to <i>XLATCH</i> upon an active going edge on the right reference switch input REFR. <i>Hint:</i> Activate <i>latch_r_active</i> to detect any spurious stop event by reading <i>status_latch_r</i> .	
latch_l_inactive	6	1: Activates latching of the position to <i>XLATCH</i> upon an inactive going edge on the left reference switch input REFL. The active level is defined by <i>pol_stop_l</i> .	
latch_l_active	5	1: Activates latching of the position to <i>XLATCH</i> upon an active going edge on the left reference switch input REFL. <i>Hint:</i> Activate <i>latch_l_active</i> to detect any spurious stop event by reading <i>status_latch_l</i> .	
swap_lr	4	1: Swap the left and the right reference switch input REFL and REFR	
pol_stop_r	3	Sets the active polarity of the right reference switch input 0=non-inverted, high active: a high level on REFR stops the motor 1=inverted, low active: a low level on REFR stops the motor	
pol_stop_l	2	Sets the active polarity of the left reference switch input 0=non-inverted, high active: a high level on REFL stops the motor 1=inverted, low active: a low level on REFL stops the motor	
stop_r_enable	1	1: Enables automatic motor stop during active right reference switch input <i>Hint:</i> The motor restarts in case the stop switch becomes released.	

BITFIELD	BITS	DESCRIPTION	DECODE
stop_l_enable	0	1: Enables automatic motor stop during active left reference switch input <i>Hint: The motor restarts in case the stop switch becomes released.</i>	

RAMP_STAT (0x35)

Ramp status and switch event status

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	Virtual reference switch right status (1=active)
status_virtual_stop_l	14	Virtual reference switch left status (1=active)
status_sg	13	1: Signals an active StallGuard2 input from the CoolStep driver or from the DcStep unit, if enabled. <i>Hint: When polling this flag, stall events may be missed – activate sg_stop to be sure not to miss the stall event.</i>
second_move	12	1: Signals that the automatic ramp required moving back in the opposite direction, e.g. due to on-the-fly parameter change (Write '1' to clear)
t_zerowait_active	11	1: Signals, that <i>TZEROWAIT</i> is active after a motor stop. During this time, the motor is in standstill.
vzero	10	1: Signals, that the actual velocity is 0.
position_reached	9	1: Signals, that the target position is reached. This flag becomes set while <i>XACTUAL</i> and <i>XTARGET</i> match.
velocity_reached	8	1: Signals, that the target velocity is reached. This flag becomes set while <i>VACTUAL</i> and <i>VMAX</i> match.
event_pos_reached	7	1: Signals, that the target position has been reached (<i>position_reached</i> becoming active). (Write '1' to clear flag and interrupt condition) This bit is ORed to the <i>interrupt output</i> signal.
event_stop_sg	6	1: Signals an active StallGuard2 stop event. Resetting the register will clear the stall condition and the motor may re-start motion, unless the motion controller has been stopped. (Write '1' to clear flag and interrupt condition) This bit is ORed to the <i>interrupt output</i> signal.

BITFIELD	BITS	DESCRIPTION
event_stop_r	5	1: Active stop right condition due to stop switch or virtual stop. The stop condition and the interrupt condition can be removed by setting <i>RAMP_MODE</i> to hold mode or by commanding a move to the opposite direction. In <i>soft_stop</i> mode, the condition will remain active until the motor has stopped motion into the direction of the stop switch. Disabling the stop switch or the stop function also clears the flag, but the motor will continue motion. This bit is ORed to the <i>interrupt output</i> signal.
event_stop_l	4	1: Active stop left condition due to stop switch or virtual stop. The stop condition and the interrupt condition can be removed by setting <i>RAMP_MODE</i> to hold mode or by commanding a move to the opposite direction. In <i>soft_stop</i> mode, the condition will remain active until the motor has stopped motion into the direction of the stop switch. Disabling the stop switch or the stop function also clears the flag, but the motor will continue motion. This bit is ORed to the <i>interrupt output</i> signal.
status_latch_r	3	1: Latch right ready (enable position latching using <i>SW_MODE</i> settings <i>latch_r_active</i> or <i>latch_r_inactive</i>) (Write '1' to clear)
status_latch_l	2	1: Latch left ready (enable position latching using <i>SW_MODE</i> settings <i>latch_l_active</i> or <i>latch_l_inactive</i>) (Write '1' to clear)
status_stop_r	1	Reference switch right status (1=active)
status_stop_l	0	Reference switch left status (1=active)

XLATCH (0x36)

Ramp generator latch position

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							

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

BITFIELD	BITS	DESCRIPTION
XLATCH	31:0	Ramp generator latch position, latches <i>XACTUAL</i> upon a programmable switch event (see <i>SW_MODE</i>). <i>Hint:</i> The encoder position can be latched to <i>ENC_LATCH</i> together with <i>XLATCH</i> to allow consistency checks.

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	Encoder prescaler mode selection	0x0: Encoder prescaler divisor binary mode: Counts <i>ENC_CONST(fractional part) / 65536</i> 0x1: Encoder prescaler divisor decimal mode: Counts in <i>ENC_CONST(fractional part) / 10000</i>
latch_x_act	9	Position latch configuration	0x0: disabled 0x1: Also latch <i>XACTUAL</i> position together with <i>X_ENC</i> . Allows latching the ramp generator position upon an N channel event as selected by <i>pos_edge</i> and <i>neg_edge</i> .
clr_enc_x	8	Encoder latch configuration	0x0: Upon N event, <i>X_ENC</i> becomes latched to <i>ENC_LATCH</i> only 0x1: Latch and additionally clear encoder counter <i>X_ENC</i> at N-event
pos_neg_edge	7:6	N channel event sensitivity	0x0: N channel event is active during an active N event level 0x1: N channel is valid upon active going N event 0x2: N channel is valid upon inactive going N event 0x3: N channel is valid upon active going and inactive going N event
clr_once	5	Position latch configuration	0x0: disabled 0x1: Latch or latch and clear <i>X_ENC</i> on the next N event following the write access

BITFIELD	BITS	DESCRIPTION	DECODE
clr_cont	4	Position latch configuration	0x0: disabled 0x1: Always latch or latch and clear <i>X_ENC</i> upon an N event (once per revolution, it is recommended to combine this setting with edge sensitive N event)
ignore_AB	3	N event configuration	0x0: An N event occurs only when polarities given by <i>pol_N</i> , <i>pol_A</i> and <i>pol_B</i> match. 0x1: Ignore A and B polarity for N channel event
pol_N	2	Defines active polarity of N	0x0: low active 0x1: high active
pol_B	1	Required B polarity for an N channel event	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)						

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
ENC_CONST	31:0	<p>Accumulation constant (signed) 16 bit integer part, 16 bit fractional part</p> <p>X_ENC accumulates $\pm ENC_CONST / (2^{16} * X_ENC)$ (binary) or $\pm ENC_CONST / (10^4 * X_ENC)$ (decimal)</p> <p><i>ENCMODE</i> bit <i>enc_sel_decimal</i> switches between decimal and binary setting. Use the sign, to match rotation direction!</p> <p>binary: $\pm [\mu\text{steps}/2^{16}]$ $\pm(0 \dots 32767.999847)$ decimal: $\pm(0.0 \dots 32767.9999)$ <i>reset default</i> = 1.0 (=65536)</p>

ENC_STATUS (0x3B)

Encoder status information

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: no warning 0x1: deviation_warn cannot be cleared while a warning still persists. Set <i>ENC_DEVIATION</i> to zero to disable.
n_event	0		0x0: no event 0x1: Event detected. To clear the status bit, write with a 1 bit at the corresponding position.

ENC LATCH (0x3C)

BIT	31	30	29	28	27	26	25	24
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	Encoder position X_ENC latched on N event

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		Maximum number of steps deviation between encoder counter and XACTUAL for deviation warning Result in flag ENC_STATUS.deviation_warn 0=Function is off.					

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							

BITFIELD	BITS	DESCRIPTION
VIRTUAL_STOP_L	31:0	Virtual stop switch based on encoder or ramp position. A stop is raised, based on the signed comparison <i>virtual_stop_enc</i> = 1: $X_ENC \leq VIRTUAL_STOP_L$ <i>virtual_stop_enc</i> = 0: $XACTUAL \leq VIRTUAL_STOP_L$ $-2^{31} \dots$ $+(2^{31})-1$

VIRTUAL STOP R (0x3F)

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

BITFIELD	BITS	DESCRIPTION
VIRTUAL_STOP_R	31:0	Virtual Stop Switch based on Encoder. A stop is raised, based on the signed comparison <i>virtual_stop_enc</i> = 1: $X_ENC \geq VIRTUAL_STOP_R$ <i>virtual_stop_enc</i> = 0: $XACTUAL \geq VIRTUAL_STOP_R$ $-2^{31} \dots$ $+(2^{31})-1$

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			Value of voltage on ADC_AIN pin in integer				
ADC_VSUPPLY	12:0			Actual Value of voltage on VS (filtered with low pass filter), update rate: each 2048 clocks				

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				

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		Actual Temperature(filtered with low pass filter), update rate: each 2048 clocks					

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		Overtemperature warning threshold register: ADC_TEMP >= OVERTEMPPREWARNING_VTH Overtemperatureprewarning will be triggered (Reset: 0xB92 equals 120°C)					
OVERVOLTAGE_VTH	12:0		Overvoltage threshold for output OV. Default: 38V, 36 V equals 1.125 V at ADC inputs					

MSLUT 0 (0x60)

Microstep table entries 0...31

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

MSLUT 1 (0x61)

Microstep table entries 32...63

BIT	31	30	29	28	27	26	25	24
Field	MSLUT_1[31:24]							
Reset	0x4A954AA							
Access Type	Write, Read							

BIT	23	22	21	20	19	18	17	16
Field	MSLUT_1[23:16]							
Reset	0x4A9554AA							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	MSLUT_1[15:8]							
Reset	0x4A9554AA							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	MSLUT_1[7:0]							
Reset	0x4A9554AA							
Access Type	Write, Read							
BITFIELD	BITS		DESCRIPTION					
MSLUT_1	31:0		<p>Each bit gives the difference between entry x and entry x+1 when combined with the corresponding <i>MSLUTSEL W</i> bits:</p> <p>0: <i>W</i>= %00: -1 %01: +0 %10: +1 %11: +2</p> <p>1: <i>W</i>= %00: +0 %01: +1 %10: +2 %11: +3</p> <p>This is the differential coding for the first quarter of a wave. Start values for <i>CUR_A</i> and <i>CUR_B</i> are stored for <i>MSCNT</i> position 0 in <i>START_SIN</i> and <i>START_SIN90</i>. <i>ofs31, ofs30, ..., ofs01, ofs00</i> ... <i>ofs255, ofs254, ..., ofs225, ofs224</i></p> <p><i>reset default= sine wave table</i></p>					

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							

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

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							

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

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							

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

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							

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

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

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

MSLUTSEL (0x68)

BIT	31	30	29	28	27	26	25	24
Field	X3[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BIT	23	22	21	20	19	18	17	16
Field	X2[7:0]							
Reset	0xFF							
Access Type	Write, Read							
BIT	15	14	13	12	11	10	9	8
Field	X1[7:0]							
Reset	0x80							
Access Type	Write, Read							
BIT	7	6	5	4	3	2	1	0
Field	W3[1:0]		W2[1:0]		W1[1:0]		W0[1:0]	
Reset	0x1		0x1		0x1		0x2	
Access Type	Write, Read		Write, Read		Write, Read		Write, Read	

BITFIELD	BITS	DESCRIPTION
X3	31:24	<p>LUT segment 1 start</p> <p>The sine wave look up table can be divided into up to four segments using an individual step width control entry W_x. The segment borders are selected by $X1$, $X2$ and $X3$.</p> <p>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.</p> <p>For defined response the values shall satisfy: $0 < X1 < X2 < X3$</p>
X2	23:16	<p>LUT segment 1 start</p> <p>The sine wave look up table can be divided into up to four segments using an individual step width control entry W_x. The segment borders are selected by $X1$, $X2$ and $X3$.</p> <p>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.</p> <p>For defined response the values shall satisfy: $0 < X1 < X2 < X3$</p>
X1	15:8	<p>LUT segment 1 start</p> <p>The sine wave look up table can be divided into up to four segments using an individual step width control entry W_x. The segment borders are selected by $X1$, $X2$ and $X3$.</p> <p>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.</p> <p>For defined response the values shall satisfy: $0 < X1 < X2 < X3$</p>
W3	7:6	<p>LUT width select from $ofs(X3)$ to $ofs255$</p> <p>Width control bit coding $W0...W3$:</p> <p>%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</p>
W2	5:4	<p>LUT width select from $ofs(X2)$ to $ofs(X3-1)$</p> <p>Width control bit coding $W0...W3$:</p> <p>%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</p>

BITFIELD	BITS	DESCRIPTION
W1	3:2	LUT width select from <i>ofs(X1)</i> to <i>ofs(X2-1)</i> Width control bit coding <i>W0...W3</i> : %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 <i>ofs00</i> to <i>ofs(X1-1)</i> Width control bit coding <i>W0...W3</i> : %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 <i>START_SIN90</i> to match the microstep wave table at position <i>MSCNT=0</i>						
START_SIN90	23:16	<i>START_SIN90</i> gives the absolute value for cosine wave microstep table entry at <i>MSCNT=0</i> (table position $256 + \text{OFFSET_SIN90}$).						
START_SIN	7:0	<i>START_SIN</i> gives the absolute value at microstep table entry 0.						

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		Microstep counter. Indicates actual position in the microstep table for <i>CUR_A</i> . <i>CUR_B</i> uses an offset of 256 (2 phase motor). <i>Hint: Move to a position where MSCNT is zero before re-initializing MSLUTSTART or MSLUT and MSLUTSEL.</i>						

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		Actual microstep current for motor phase A (co-sine wave) as read from MSLUT (not scaled by current)						
CUR_B	8:0		Actual microstep current for motor phase B (sine wave) as read from MSLUT (not scaled by current)						

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	short to supply protection disable			0x0: Short to VS protection is on 0x1: Short to VS protection is disabled			
diss2g	30	short to GND protection disable			0x0: Short to GND protection is on 0x1: Short to GND protection is disabled			
dedge	29	enable double edge step pulses			0x0: disabled 0x1: Enable step impulse at each step edge to reduce step frequency requirement.			
intpol	28	interpolation to 256 microsteps			0x0: no interpolation 0x1: The actual microstep resolution (<i>MRES</i>) becomes extrapolated to 256 microsteps for smoothest motor operation (useful for STEP/DIR operation, only)			

BITFIELD	BITS	DESCRIPTION	DECODE
MRES	27:24	<p>micro step resolution selection</p> <p>%0000: Native 256 microstep setting. Normally use this setting with the internal motion controller.</p> <p>%0001 ... %1000: 128, 64, 32, 16, 8, 4, 2, FULLSTEP Reduced microstep resolution esp. for STEP/DIR operation. The resolution gives the number of microstep entries per sine quarter wave. The driver automatically uses microstep positions which result in a symmetrical wave, when choosing a lower microstep resolution. step width=2^MRES [microsteps]</p>	
TPFD	23:20	<p>passive fast decay time</p> <p>TPFD allows dampening of motor mid-range resonances. Passive fast decay time setting controls duration of the fast decay phase inserted after bridge polarity change NCLK= 128*TPFD %0000: Disable %0001 ... %1111: 1 ... 15</p>	
vhighchm	19	<p>high velocity chopper mode</p> <p>This bit enables switching to <i>chm</i>=1 and <i>fd</i>=0, when <i>VHIGH</i> is exceeded. This way, a higher velocity can be achieved. Can be combined with <i>vhighfs</i>=1. If set, the <i>TOFF</i> setting automatically becomes doubled during high velocity operation in order to avoid doubling of the chopper frequency.</p>	
vhighfs	18	<p>high velocity fullstep selection</p> <p>This bit enables switching to fullstep, when <i>VHIGH</i> is exceeded. Switching takes place only at 45° position. The fullstep target current uses the current value from the microstep table at the 45° position.</p>	
TBL	16:15	<p>TBL / blank time select</p> <p>%00 ... %11: Set comparator blank time to 16, 24, 36 or 54 clocks <i>Hint</i>: %01 or %10 is recommended for most applications (Reset Default: OTP %01 or %10)</p>	
chm	14	<p>chopper mode</p>	<p>0x0: Standard mode (SpreadCycle) 0x1: Constant off time with fast decay time. Fast decay time is also terminated when the negative nominal current is reached. Fast decay is after on time.</p>

BITFIELD	BITS	DESCRIPTION	DECODE						
disfdcc	12	fast decay mode with <i>chm</i> =1: <i>disfdcc</i> =1 disables current comparator usage for termination of the fast decay cycle							
fd3	11	TFD[3] with <i>chm</i> =1: MSB of fast decay time setting <i>TFD</i>							
HEND_OFFSET	10:7	with <i>chm</i> =0: HEND = hysteresis low value %0000 ... %1111: Hysteresis is -3, -2, -1, 0, 1, ..., 12 (1/512 of this setting adds to current setting) This is the hysteresis value which becomes used for the hysteresis chopper. with <i>chm</i> =1: OFFSET = sine wave offset %0000 ... %1111: Offset is -3, -2, -1, 0, 1, ..., 12 This is the sine wave offset and 1/512 of the value becomes added to the absolute value of each sine wave entry.							
HSTRT_TFD 210	6:4	<table border="1"> <tr> <td>with <i>chm</i>=0</td> <td>%000 ... %111: Add 1, 2, ..., 8 to hysteresis low value <i>HEND</i> (1/512 of this setting adds to current setting)</td> </tr> <tr> <td><i>HSTRT</i> hysteresis start value added to <i>HEND</i></td> <td><i>Attention: Effective</i> $HEND + HSTRT \leq 16$. <i>Hint:</i> Hysteresis decrement is done each 16 clocks</td> </tr> <tr> <td>with <i>chm</i>=1</td> <td>Fast decay time setting (MSB: <i>fd3</i>): %0000 ... %1111: <i>TFD</i> [2..0] Fast decay time setting <i>TFD</i> with NCLK = 32 * <i>TFD</i> (%0000: slow decay only)</td> </tr> </table>	with <i>chm</i> =0	%000 ... %111: Add 1, 2, ..., 8 to hysteresis low value <i>HEND</i> (1/512 of this setting adds to current setting)	<i>HSTRT</i> hysteresis start value added to <i>HEND</i>	<i>Attention: Effective</i> $HEND + HSTRT \leq 16$. <i>Hint:</i> Hysteresis decrement is done each 16 clocks	with <i>chm</i> =1	Fast decay time setting (MSB: <i>fd3</i>): %0000 ... %1111: <i>TFD</i> [2..0] Fast decay time setting <i>TFD</i> with NCLK = 32 * <i>TFD</i> (%0000: slow decay only)	
with <i>chm</i> =0	%000 ... %111: Add 1, 2, ..., 8 to hysteresis low value <i>HEND</i> (1/512 of this setting adds to current setting)								
<i>HSTRT</i> hysteresis start value added to <i>HEND</i>	<i>Attention: Effective</i> $HEND + HSTRT \leq 16$. <i>Hint:</i> Hysteresis decrement is done each 16 clocks								
with <i>chm</i> =1	Fast decay time setting (MSB: <i>fd3</i>): %0000 ... %1111: <i>TFD</i> [2..0] Fast decay time setting <i>TFD</i> with NCLK = 32 * <i>TFD</i> (%0000: slow decay only)								
TOFF	3:0	<i>TOFF</i> off time and driver enable Off time setting controls duration of slow decay phase $NCLK = 24 + 32 * TOFF$ %0000: Driver disable, all bridges off %0001: 1 – use only with $TBL \geq 2$ %0010 ... %1111: 2 ... 15							

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 and StallGuard4 filter enable			0x0: Standard mode, high time resolution for StallGuard 0x1: Filtered mode, StallGuard signal updated for each four fullsteps only to compensate for motor pole tolerances			
sgt	22:16	StallGuard2 threshold value This signed value controls StallGuard2 level for stall output and sets the optimum measurement range for readout. A lower value gives a higher sensitivity. Zero is the starting value working with most motors. -64 to +63: A higher value makes StallGuard2 less sensitive and requires more torque to indicate a stall.						
seimin	15	minimum current for smart current control			0x0: 1/2 of current setting (<i>IRUN</i>) 0x1: 1/4 of current setting (<i>IRUN</i>)			
sedn	14:13	current down step speed %00: For each 32 StallGuard2 values decrease by one %01: For each 8 StallGuard2 values decrease by one %10: For each 2 StallGuard2 values decrease by one %11: For each StallGuard2 value decrease by one						

BITFIELD	BITS	DESCRIPTION	DECODE
semax	11:8	StallGuard2 hysteresis value for smart current control If the StallGuard2 result is equal to or above $(SEMIN+SEMAX+1)*32$, the motor current becomes decreased to save energy. %0000 ... %1111: 0 ... 15	
seup	6:5	current up step width Current increment steps per measured StallGuard2 value %00 ... %11: 1, 2, 4, 8	
semin	3:0	minimum StallGuard2 value for smart current control and smart current enable If the StallGuard2 result falls below $SEMIN*32$, the motor current becomes increased to reduce motor load angle. %0000: smart current control CoolStep off %0001 ... %1111: 1 ... 15	

DCCTRL (0x6E)

DcStep (DC) automatic commutation configuration register (enable via pin DCEN or via $VDCMIN$)

Hint: Using a higher microstep resolution or interpolated operation, DcStep delivers a better StallGuard signal.

DC_SG is also available above VHIGH if vhighfs is activated. For best result also set vhighcm.

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	Max. PWM on time for step loss detection using DcStep StallGuard2 in DcStep mode. $(DC_SG * 16/f_{CLK})$ Set slightly higher than $DC_TIME/16$ 0=disable
DC_TIME	9:0	Upper PWM on time limit for commutation $(DC_TIME * 1/f_{CLK})$. Set slightly above effective blank time TBL .

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	standstill indicator This flag indicates motor stand still in each operation mode. This occurs 2 ²⁰ clocks after the last step pulse.	
olb	30	open load indicator phase B	0x0: normal operation 0x1: Open load detected on phase B. <i>Hint:</i> This is just an informative flag. The driver takes no action upon it. False detection may occur in fast motion and standstill. Check during slow motion, only.
ola	29	open load indicator phase A	0x0: normal operation 0x1: Open load detected on phase A. <i>Hint:</i> This is just an informative flag. The driver takes no action upon it. False detection may occur in fast motion and standstill. Check during slow motion, only.
s2gb	28	short to ground indicator phase B	0x0: normal operation 0x1: Short to GND detected on phase B. The driver becomes disabled. The flags stay active, until the driver is disabled by software (<i>TOFF</i> =0) or by the ENN input.
s2ga	27	short to ground indicator phase A	0x0: normal operation 0x1: Short to GND detected on phase A. The driver becomes disabled. The flags stay active, until the driver is disabled by software (<i>TOFF</i> =0) or by the ENN input.

BITFIELD	BITS	DESCRIPTION	DECODE
otpw	26	overtemperature pre-warning flag	0x0: normal operation 0x1: Overtemperature pre-warning threshold is exceeded. The overtemperature pre-warning flag is common for both bridges.
ot	25	overtemperature flag	0x0: normal operation 0x1: Overtemperature limit has been reached. Drivers become disabled until <i>otpw</i> is also cleared due to cooling down of the IC. The overtemperature flag is common for both bridges.
stallguard	24	StallGuard2/StallGuard4 status	0x0: normal operation 0x1: Motor stall detected by StallGuard2 (in SpreadCycle operation) resp. by StallGuard4 (in StealthChop operation) or DcStep stall (in DcStep mode).
CS_ACTUAL	20:16	actual motor current / smart energy current Actual current control scaling, for monitoring smart energy current scaling controlled via settings in register <i>COOLCONF</i> , or for monitoring the function of the automatic current scaling	
fsactive	15	full step active indicator	0x0: microstepping active 0x1: Indicates that the driver has switched to fullstep as defined by chopper mode settings and velocity thresholds
stealth	14	StealthChop indicator	0x0: stealthChop not active 0x1: Driver operates in StealthChop mode
s2vsb	13	short to supply indicator phase B	0x0: no error 0x1: Short to supply detected on phase B. The driver becomes disabled. The flags stay active, until the driver is disabled by software (<i>TOFF=0</i>) or by the ENN input.
s2vsa	12	short to supply indicator phase A	0x0: no error 0x1: Short to supply detected on phase A. The driver becomes disabled. The flags stay active, until the driver is disabled by software (<i>TOFF=0</i>) or by the ENN input.

BITFIELD	BITS	DESCRIPTION	DECODE
SG_RESULT	9:0	<p>StallGuard2 result respectively StallGuard4 result (depending on actual chopper mode) resp. PWM on time for coil A in stand still with SpreadCycle for motor temperature detection.</p> <p>Mechanical load measurement: The StallGuard2/4 result gives a means to measure mechanical motor load. A higher value means lower mechanical load. For StallGuard2, a value of 0 signals highest load. With optimum <i>SGT</i> setting, this is an indicator for a motor stall. The stall detection compares <i>SG_RESULT</i> to 0 in order to detect a stall. <i>SG_RESULT</i> is used as a base for CoolStep operation, by comparing it to a programmable upper and a lower limit. It is not applicable in StealthChop mode. StallGuard2 works best with microstep operation or DcStep.</p> <p>Temperature measurement during SpreadCycle mode: In standstill, no StallGuard2 result can be obtained. <i>SG_RESULT</i> shows the chopper on-time for motor coil A instead. Move the motor to a determined microstep position at a certain current setting to get a rough estimation of motor temperature by a reading the chopper on-time. As the motor heats up, its coil resistance rises and the chopper on-time increases. For StallGuard4 specifics, please refer <i>SG4_RESULT</i>.</p>	

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_re g_stst	pwm_meas _sd_enable	FREEWHEEL[1:0]		pwm_autogr ad	pwm_autos cale	PWM_FREQ[1:0]	
Reset	0x0	0x0	0x0		0x1	0x1	0x0	
Access Type	Write, Read	Write, Read	Write, Read		Write, Read	Write, Read	Write, Read	
BIT	15	14	13	12	11	10	9	8
Field	PWM_GRAD[7:0]							
Reset	0x0							
Access Type	Write, Read							

BIT	7	6	5	4	3	2	1	0
Field	PWM_OFS[7:0]							
Reset	0x1D							
Access Type	Write, Read							
BITFIELD	BITS	DESCRIPTION	DECODE					
PWM_LIM	31:28	<p>PWM automatic scale amplitude limit when switching on</p> <p>Limit for <i>PWM_SCALE_AUTO</i> when switching back from SpreadCycle to StealthChop. This value defines the upper limit for bits 7 to 4 of the automatic current control when switching back. It can be set to reduce the current jerk during mode change back to StealthChop. It does not limit <i>PWM_GRAD</i> or <i>PWM_GRAD_AUTO</i> offset. (Default = 12)</p>						
PWM_REG	27:24	<p>Regulation loop gradient</p> <p>User defined maximum PWM amplitude change per half wave when using <i>pwm_autoscale=1</i>. (1...15):</p> <p>1: 0.5 increments (slowest regulation)</p> <p>2: 1 increment</p> <p>3: 1.5 increments</p> <p>4: 2 increments (<i>Reset default</i>)</p> <p>...</p> <p>8: 4 increments</p> <p>...</p> <p>15: 7.5 increments (fastest regulation)</p>						
pwm_dis_reg_stst	23	1= Disable current regulation when motor is in standstill and current is reduced (less than IRUN). This option eliminates any regulation noise during standstill.						
pwm_meas_sd_enable	22	Default=0; 1: Uses slow decay phases on low side to measure the motor current to reduce the lower current limit.						
FREEWHEEL	21:20	<p>Allows different standstill modes</p> <p>Stand still option when motor current setting is zero (<i>I_HOLD=0</i>).</p> <p>%00: Normal operation</p> <p>%01: Freewheeling</p> <p>%10: Coil shorted using LS drivers</p> <p>%11: Coil shorted using HS drivers</p>						

BITFIELD	BITS	DESCRIPTION	DECODE
pwm_autograd	19	PWM automatic gradient adaptation	<p>0x0: Fixed value for <i>PWM_GRAD</i> (<i>PWM_GRAD_AUTO</i> = <i>PWM_GRAD</i>)</p> <p>0x1: Automatic tuning (only with <i>pwm_autoscale</i>=1) (<i>Reset default</i>) <i>PWM_GRAD_AUTO</i> is initialized with <i>PWM_GRAD</i> while <i>pwm_autograd</i>=0 and becomes optimized automatically during motion.</p> <p><u>Preconditions</u></p> <ol style="list-style-type: none"> <i>PWM_OFS_AUTO</i> has been automatically initialized. This requires standstill at <i>IRUN</i> for >130ms in order to a) detect standstill b) wait > 128 chopper cycles at <i>IRUN</i> and c) regulate <i>PWM_OFS_AUTO</i> so that $-1 < PWM_SCALE_AUTO < 1$ Motor running and $1.5 * PWM_OFS_AUTO * (IRUN+1)/32 < PWM_SCALE_SUM < 4 * PWM_OFS_AUTO * (IRUN+1)/32$ and $PWM_SCALE_SUM < 255$. <p><u>Time required for tuning <i>PWM_GRAD_AUTO</i></u> About 8 fullsteps per change of +/-1.</p> <p>Also enables use of reduced chopper frequency for tuning <i>PWM_OFS_AUTO</i>.</p>
pwm_autoscale	18	PWM automatic amplitude scaling	<p>0x0: User defined feed forward PWM amplitude. The current settings <i>IRUN</i> and <i>IHOLD</i> have no influence! The resulting PWM amplitude (limited to 0...255) is: $PWM_OFS * ((CS_ACTUAL+1) / 32) + PWM_GRAD * 256 / TSTEP$</p> <p>0x1: Enable automatic current control (<i>Reset default</i>)</p>
PWM_FREQ	17:16	<p>PWM frequency selection:</p> <p>%00: $f_{PWM}=2/1024 f_{CLK}$ (<i>Reset default</i>)</p> <p>%01: $f_{PWM}=2/683 f_{CLK}$</p> <p>%10: $f_{PWM}=2/512 f_{CLK}$</p> <p>%11: $f_{PWM}=2/410 f_{CLK}$</p>	
PWM_GRAD	15:8	<p>Velocity dependent gradient for PWM amplitude: $PWM_GRAD * 256 / TSTEP$ This value is added to <i>PWM_OFS</i> to compensate for the velocity-dependent motor back-EMF.</p> <p>Use <i>PWM_GRAD</i> as initial value for automatic scaling to speed up the automatic tuning process. To do this, set <i>PWM_GRAD</i> to the determined, application specific value, with <i>pwm_autoscale</i>=0. Only afterwards, set <i>pwm_autoscale</i>=1. Enable StealthChop when finished.</p> <p><i>Hint:</i> After initial tuning, the required initial value can be read out from <i>PWM_GRAD_AUTO</i>.</p>	

BITFIELD	BITS	DESCRIPTION	DECODE
PWM_OFS	7:0	<p>User defined PWM amplitude offset (0-255) related to full motor current (<i>CS_ACTUAL</i>=31) in stand still. (<i>Reset default</i>=30)</p> <p>Use <i>PWM_OFS</i> as initial value for automatic scaling to speed up the automatic tuning process. To do this, set <i>PWM_OFS</i> to the determined, application specific value, with <i>pwm_autoscale</i>=0. Only afterwards, set <i>pwm_autoscale</i>=1. Enable StealthChop when finished.</p> <p><i>PWM_OFS</i> = 0 will disable scaling down motor current below a motor specific lower measurement threshold. This setting should only be used under certain conditions, i.e. when the power supply voltage can vary up and down by a factor of two or more. It prevents the motor going out of regulation, but it also prevents power down below the regulation limit.</p> <p><i>PWM_OFS</i> > 0 allows automatic scaling to low PWM duty cycles even below the lower regulation threshold. This allows low (standstill) current settings based on the actual (hold) current scale (register <i>IHOLD_IRUN</i>).</p>	

PWM_SCALE (0x71)

Results of StealthChop amplitude regulator. These values can be used to monitor automatic PWM amplitude scaling (255=max. voltage).

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								

BITFIELD	BITS	DESCRIPTION
PWM_SCALE_AUTO	24:16	
PWM_SCALE_SUM	9:0	Bits: 9...0: [0...1023]PWM_SCALE_SUM: Actual PWM duty cycle. This value is used for scaling the values CUR_A and CUR_B read from the sine wave table. 1023: maximum duty cycle. This value is extended by two bits [1,0] for higher precision of duty cycle read out. Bits 9..2 correspond to the 8 bit values in other PWM duty cycle related registers.

PWM_AUTO (0x72)

These automatically generated values can be read out in order to determine a default / power up setting for *PWM_GRAD* and *PWM_OFS*.

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	Automatically determined gradient value
PWM_OFS_AUTO	7:0	Automatically determined offset value

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: Automatic phase shift compensation based on StallGuard4, when switching from StealthChop to SpreadCycle controlled via TPWMTHRS
sg4_filt_en	8	1: enable SG4 filter, 0: disable SG4 filter
SG4_THRS	7:0	Detection threshold for stall. The StallGuard4 value <i>SG4_RESULT</i> becomes compared to the double of this threshold. A stall is signaled with $SG_RESULT \leq SG4_THRS * 2$

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	StallGuard result for StallGuard4, only. SG4_RESULT becomes updated with each fullstep, independent of TCOOLTHRS and SG4THRS. A higher value signals a lower motor load and more torque headroom. Intended for StealthChop mode, only. Bits 9 and 0 will always show 0. Scaling to 10 bit is for compatibility to StallGuard2.

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	When SG4_filt_en = 1: Displays SG4 measurement 3 used as filter input

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

Typical Application Circuits

Standard Application Circuit

The standard application circuit uses a minimum set of additional components. Use low ESR capacitors for filtering the power supply. The capacitors need to cope with the current ripple cause by chopper operation. A minimum capacity of 100 μ F near the driver is recommended for best performance. Current ripple in the supply capacitors also depends on the power supply internal resistance and cable length. VCC_IO must be supplied from an external source, e.g., a low drop 3.3V regulator.

Place all filter capacitors as close as possible to the related IC pins. Use a solid common GND for all GND connections. Connect VDD1V8 filtering capacitor directly to VDD1V8 pin. Low ESR electrolytic capacitors are recommended for VS filtering.

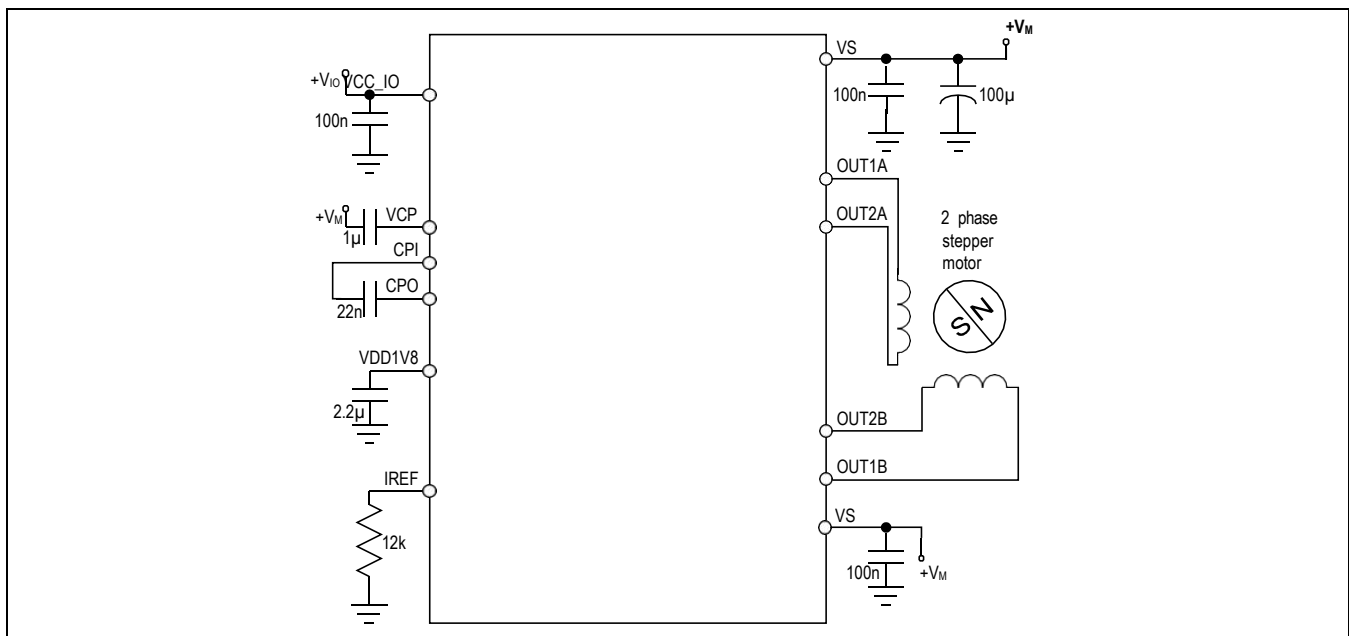


Figure 37. TMC5240 standard application circuit

High Motor Current

When operating at a high motor current, the driver power dissipation due to MOSFET switch on-resistance significantly heats up the driver. This power dissipation will heat up the PCB cooling infrastructure also, if operated at an increased duty cycle. This in turn leads to a further increase of driver temperature. An increase of temperature by about 100°C increases MOSFET resistance by roughly 50%. This is a typical behavior of MOSFET switches. Therefore, under high duty cycle, high load conditions, thermal characteristics have to be carefully taken into account, especially when increased environment temperatures are to be supported. Refer the thermal characteristics and the layout examples as well.

As a thumb rule, thermal properties of the PCB design may become critical at or above 1.5A RMS motor current for increased periods of time. Keep in mind that the resistive power dissipation raises with the square of the motor current. On the other hand, this means that a small reduction of motor current significantly saves heat dissipation and energy.

Typical Application Circuits (continued)

Driver Protection and EME Circuitry

Some applications have to cope with ESD events caused by motor operation or external influence. Despite ESD circuitry within the driver chips, ESD events occurring during operation can cause a reset or even a destruction of the motor driver, depending on their energy. Especially plastic housings and belt drive systems tend to cause ESD events of several kV. It is best practice to avoid ESD events by attaching all conductive parts, especially the motors themselves to PCB ground, or to apply electrically conductive plastic parts. In addition, the driver can be protected up to a certain degree against ESD events or live plugging / pulling the motor, which also causes high voltages and high currents into the motor connector terminals. A simple scheme uses capacitors at the driver outputs to reduce the dV/dt caused by ESD events. Larger capacitors will bring more benefit concerning ESD suppression, but cause additional current flow in each chopper cycle, and thus increase driver power dissipation, especially at high supply voltages. The values shown are example values – they might be varied between 100pF and 1nF. The capacitors also dampen high frequency noise injected from digital parts of the application PCB circuitry and thus reduce electromagnetic emission. A more elaborate scheme uses LC filters to de-couple the driver outputs from the motor connector. Varistors in between of the coil terminals eliminate coil overvoltage caused by live plugging. Optionally protect all outputs by a varistor against ESD voltage.

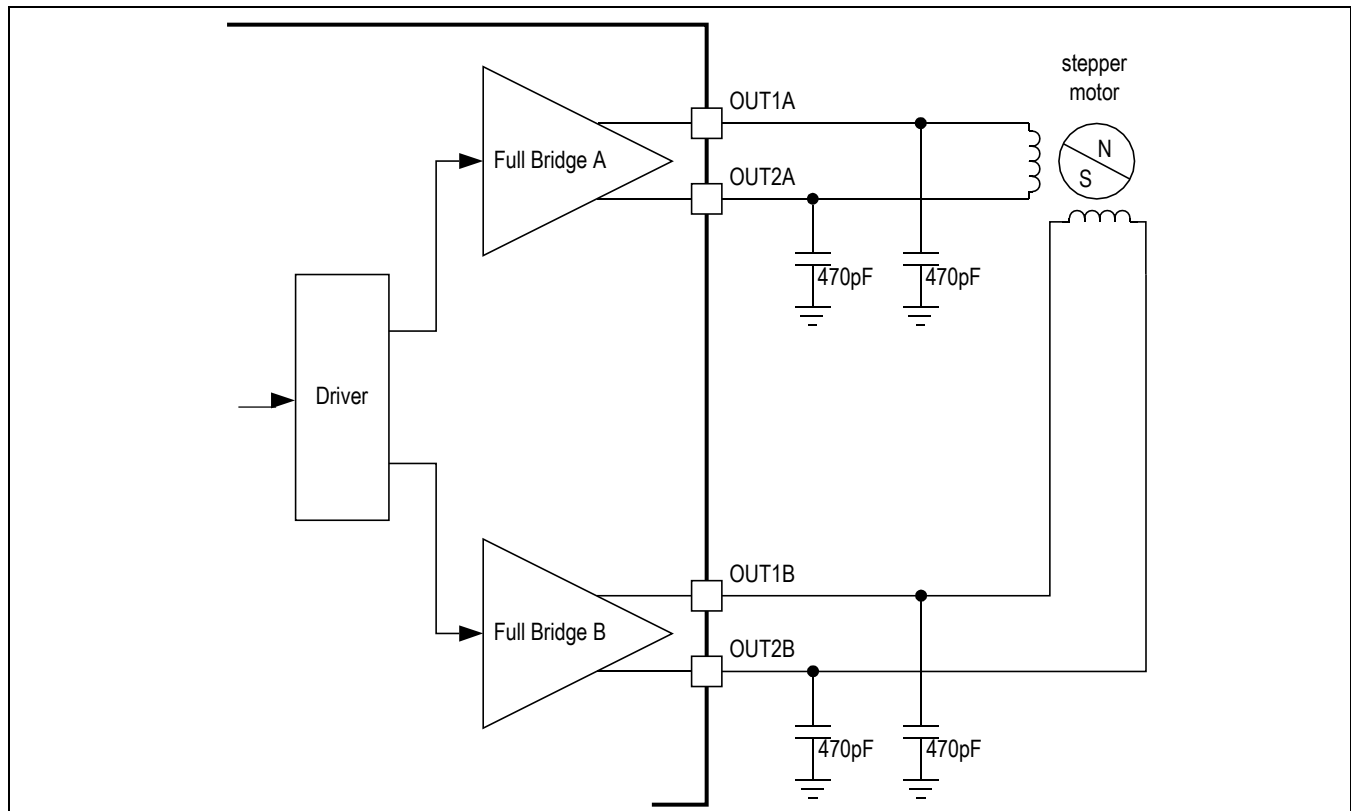


Figure 38. Simple ESD enhancement

Typical Application Circuits (continued)

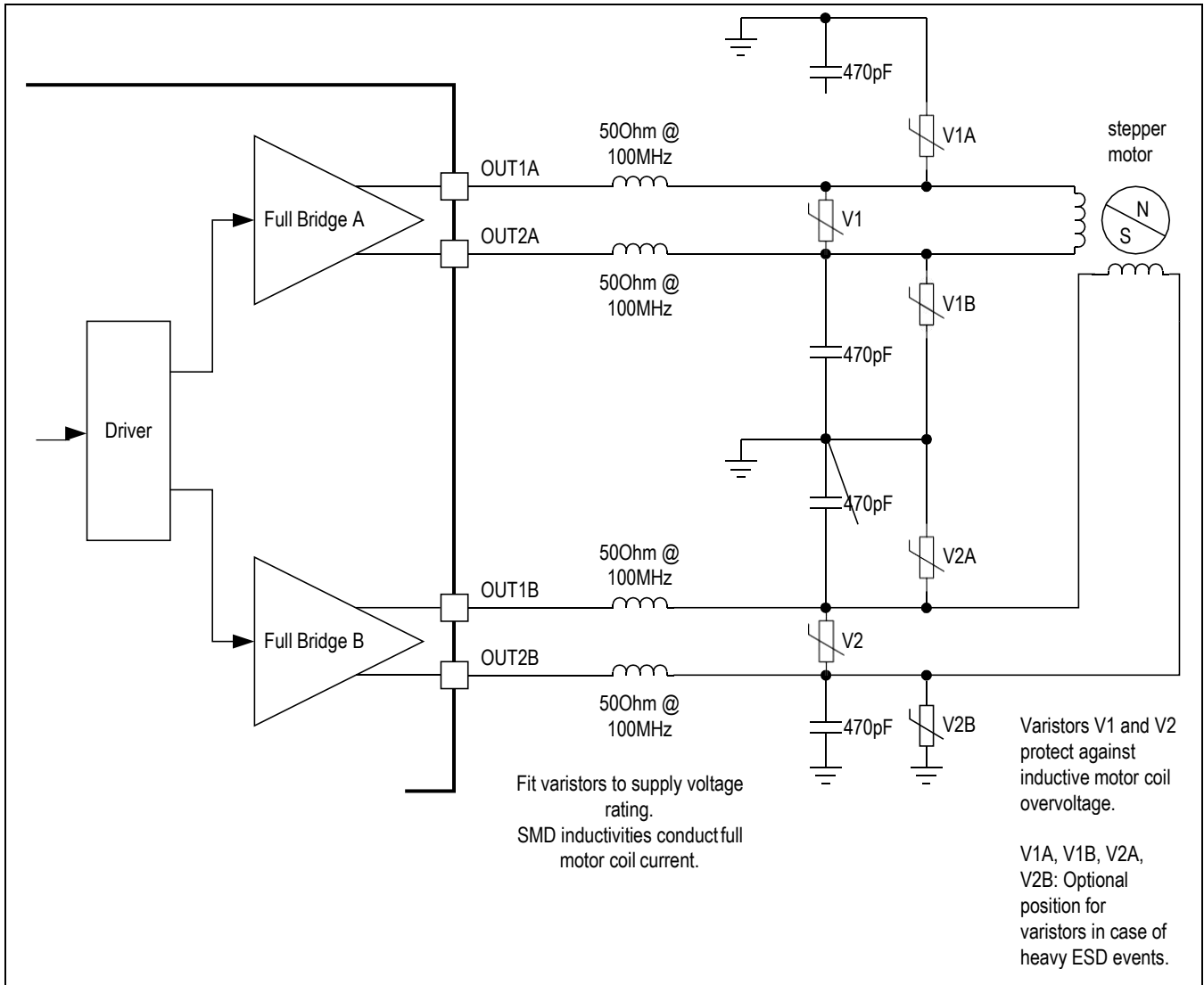


Figure 39. Elaborate motor output protection

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+ including 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.

Revision History

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

PRELIMINARY CONFIDENTIAL

