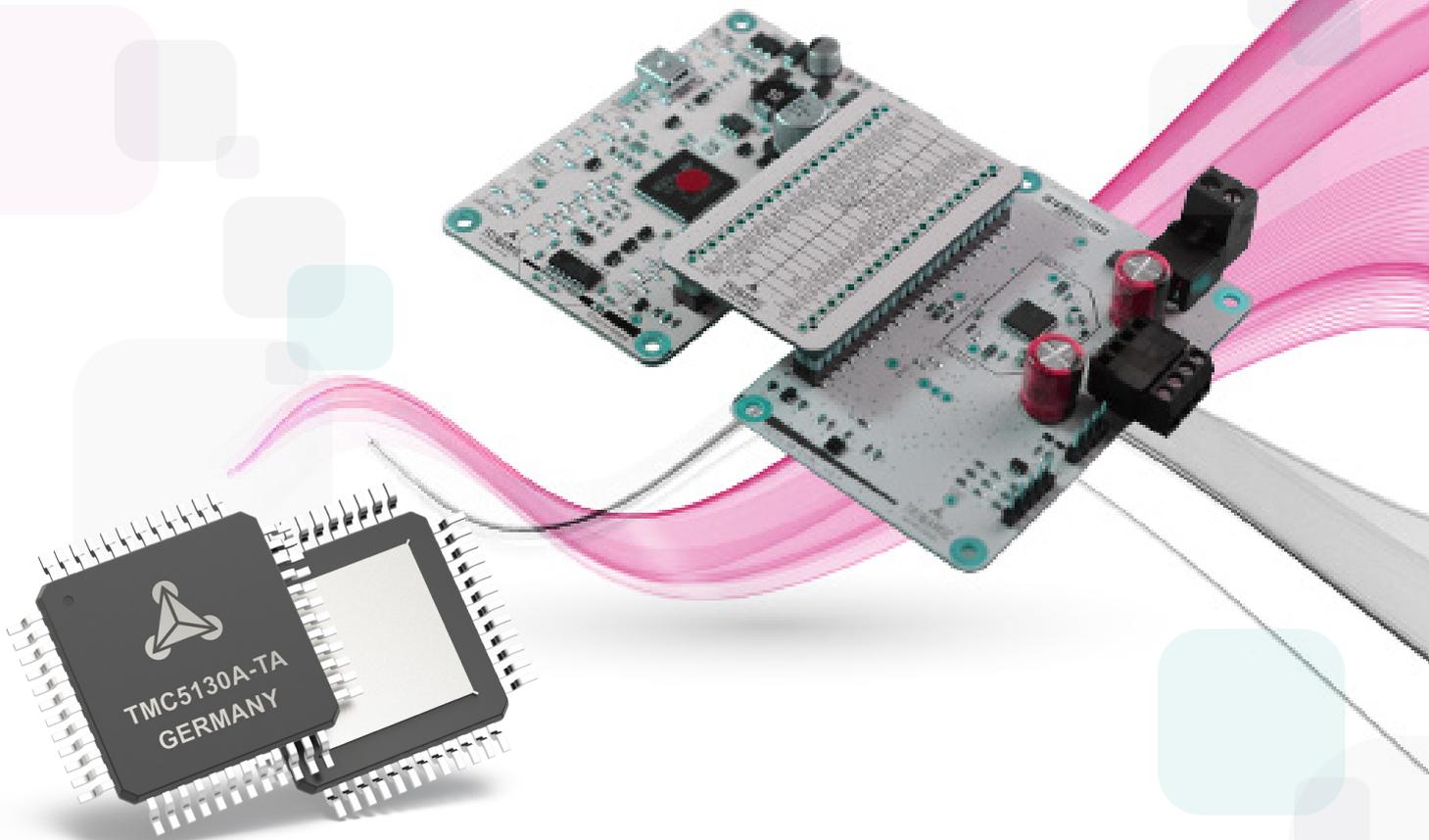


Whitepaper



Leitfaden zur Inbetriebnahme der modernen Funktionen eines aktuellen Schrittmotor-ICs

Am Beispiel eines Trinamic TMC5130-Eval-Kits
incl. Optimierungsanleitung der Parameter für stallGuard™,
coolStep™, dcStep™, spreadCycle™ & stealthChop™



”

Leitfaden zur Inbetriebnahme der modernen Funktionen eines aktuellen Schrittmotor-ICs

Inhalt

1	Einleitung	3
1.1	Einführung	4
2	Der Motion-Controller & Treiber TMC5130	5
2.1	Kombinierbarkeit der Funktionen	7
2.2	Das TMC5130-Eval-Kit	8
3	Inbetriebnahme	9
3.1	Vorbereitung des Evaluierungssystems	9
3.2	Bedienung der IDE	11
3.2.1	Stromeinstellung	13
3.2.2	Berechnung der physikalischen Einheiten	14
3.2.3	Motor konstant drehen	15
3.2.4	Motor positionieren	15
3.2.5	Direktmodus	16
3.2.6	Register Browser	17
4	Parametrierung der Funktionen	18
4.1	spreadCycle – Konstantstrom-Chopper	18
4.1.1	stallGuard2	21
4.1.2	coolStep	26
4.2	stealthChop – „No-Noise-Chopper“	29
4.3	dcStep - lastabhängige Geschwindigkeitsanpassung	34
4.3.1	Stall Detection mit dcStep	36
5	Registertabelle	32

1 Einleitung

Während früher beim Design einer Schrittmotor-Ansteuerung im wesentlichen nur der Strom und die Spannung richtig gewählt werden mussten, weisen moderne kombinierte Motion-Controller und Treiber eine Vielzahl von Zusatzfunktionen auf. So gibt es zum Beispiel Funktionen zur Steigerung der Energieeffizienz, zur Erhöhung der Betriebssicherheit, zur Reduzierung der Geräusentwicklung und zur Vereinfachung der Steuerung. Der Strom ist per Software skalierbar, die Anzahl der Mikroschritte ist ein- und automatisch umstellbar, die Mikroschritt-Wellenform ist veränderbar und es gibt einen Mikroschritt-Interpolator. Der Strom kann sowohl intern über die MOSFETs als auch über externe Shunt-Widerstände gemessen werden. Hinzu kommen Schutz- und Diagnosefunktionen. Der Motion-Controller besteht aus einem parametrierbaren Beschleunigungsrampen-Generator mit mehreren Stützstellen sowie einem Pulserzeuger für Mikroschritte. End- und Stoppschalter sowie inkrementelle Drehgeber zur Positionsüberwachung können direkt angeschlossen und ausgewertet werden. Um aus dieser Anzahl von Funktionen die passenden für die eigene Anwendung auszuwählen und die richtige Parametrierung zu finden, ist auf den ersten Blick eine große Hürde.

Dieses Whitepaper hilft als Leitfaden dabei die Funktionen zu verstehen und die passenden Parameterwerte zu finden. Ablaufdiagramme helfen bei der Optimierung.

So wird der Leser in die Lage versetzt seine Schrittmotoranwendung rasch mit den aktuell modernsten Funktionen auszustatten und deren Vorteile zu nutzen.

Motion-Seminar am 11.10.17 in Würzburg

Anmeldung und Infos unter:
www.mev-elektronik.com/events

1.1 Einführung

Das Trinamic TMC5130-Eval-Kit ist ein System zum Erproben von Schrittmotor-Anwendungen. Es besteht aus einem Evaluationsboard für den kombinierten Motion-Controller und Schrittmotortreiber TMC5130, einer Mikrocontroller-Platine (MCU-Bord) sowie der „Eselbrücke“, die beide Einheiten miteinander verbindet. Das MCU-Board wird über USB an einen Windows-PC angeschlossen, auf dem die Entwicklungsumgebung (TMCL-IDE) installiert ist. Die aktuelle Software kann bei Trinamic auf der Website kostenlos heruntergeladen werden. Über die IDE können alle Parameter des TMC5130 eingestellt und angezeigt werden.

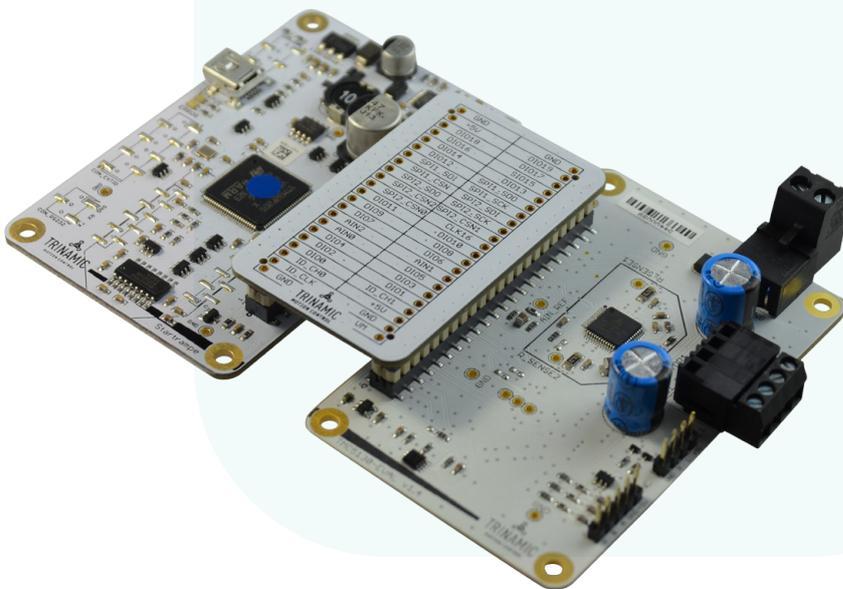


Bild: Evaluations-Kit TMC5130-EVAL-KIT

Dieser Leitfaden beantwortet hauptsächlich zwei Fragen:

- Wie ist ein Schrittmotor an einem TMC5130-Evaluationsboard in Betrieb zu nehmen?
- Wie sind die für den Betrieb mit den Funktionen stallGuard2, coolStep, dcStep, spreadCycle und stealthChop optimalen Parameter mit der TMCL-IDE zu ermitteln?

2 Der Motion-Controller & Treiber TMC5130

“

Der TMC5130 ist derzeit (Stand 08-2017) der Schrittmotorbaustein mit dem größten Funktionsumfang auf dem Markt.

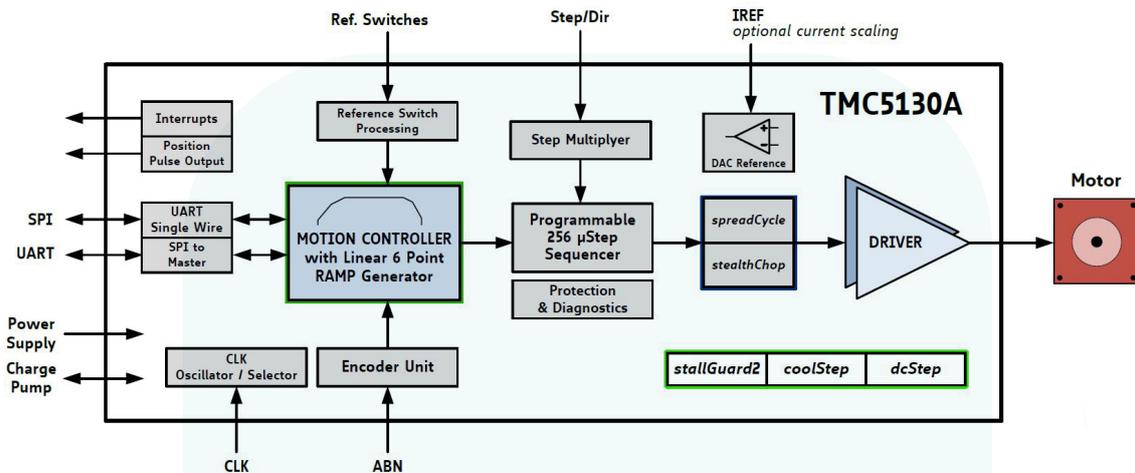


Bild: Blockschaltbild TMC5130

Der integrierte Motion-Controller besteht aus einem Takterzeuger mit Generator für komplexe Beschleunigungsrampen mit bis zu 6 Stützstellen (sixPoint).

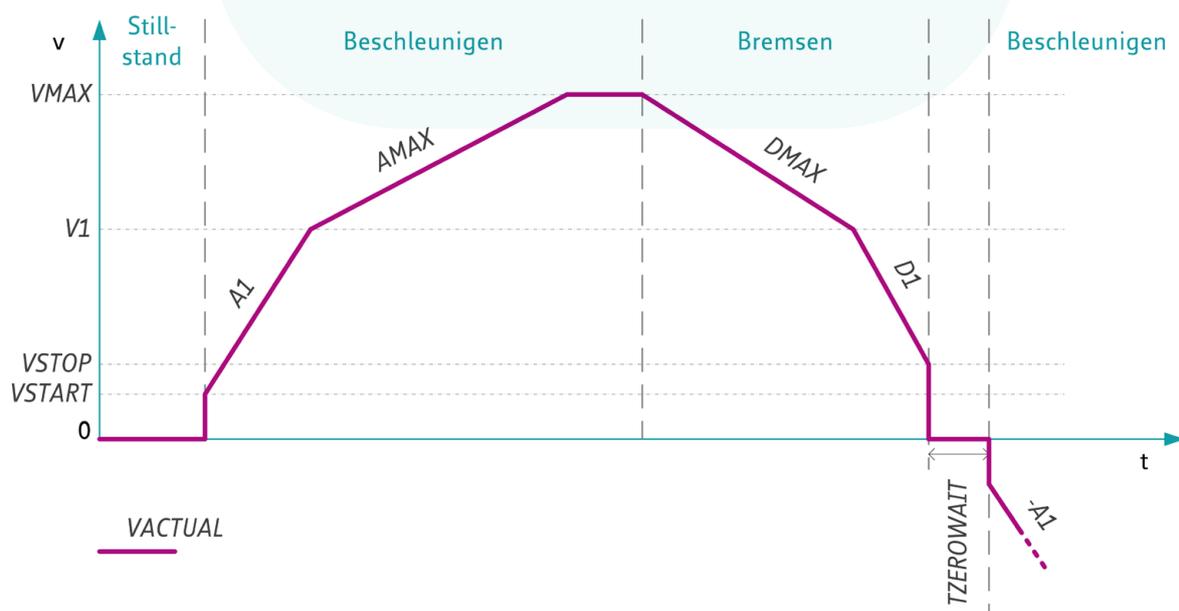


Bild: sixPoint-Rampe

End- und Referenzschalter können verbunden und ausgewertet werden. Zur Positionskontrolle kann ein inkrementeller Quadraturencoder mit AB-Signalen und optional Nullposition direkt überwacht werden.

Der integrierte Treiber kann bipolare 2-Phasen-Schrittmotoren mit Phasenströmen bis zu 1,4A bei max. 46V antreiben. Der maximale Strom sowie ein Ruhe-/Haltestrom sind programmierbar.

Über Takt- und Richtungs-Signale kann der Treiber ohne den integrierten Motion Controller angesteuert werden. Die maximale Mikroschrittauflösung beträgt 256 Mikroschritte pro Vollschritt. Um die Ansteuerfrequenz zu reduzieren und den Baustein in bestehende Lösungen einsetzen zu können, gibt es den Mikroschritt-Vervielfacher „microPlyer™“. Damit ist es möglich mit beliebigem Eingangstakt (Voll-/Halb-, 1/4-, 1/8- 1/16- und 1/32-Schritt) die Laufruhe von 1/256-Schritt zu erhalten.

Es stehen verschiedene Stromregler-Modi zur Auswahl: ein herkömmlicher Mixed-Decay Constant-TOFF Chopper, der von Trinamic patentierte spreadCycle Chopper mit Hysterese-Regelung, sowie der spannungsgesteuerte Betrieb stealthChop mit Stromnachführung für absolut lautlosen Betrieb.

Sensorlose Verfahren:

stallGuard²

Die sensorlose Lasterkennung stallGuard2 ermöglicht die Überwachung des Motors im laufenden Betrieb und eine Anschlags- und Überlasterkennung.



coolStep™

Die sensorlose Stromanpassung coolStep nutzt den stallGuard-Wert, um den Motorstrom an die aktuelle Belastung anzupassen.

dcStep™

Im dcStep Modus verhält sich der Motor ähnlich wie ein DC-Motor. Wenn die Last die eigentliche Lastgrenze überschreitet, verlangsamt sich der Motor soweit, sodass er das nötige Drehmoment aufbringen kann. Dabei wird die Eigenschaft des Schrittmotors ausgenutzt, bei geringerer Drehzahl ein höheres Drehmoment zu erzeugen.

2.1 Kombinierbarkeit der Funktionen

Die Nutzung eines Konstantstrom-Choppers wie spreadCycle ist Voraussetzung für stallGuard2 und das darauf aufbauende coolStep. Die Nutzung dieser Funktionen ist also mit einem spannungsgesteuerten PWM-Betrieb wie stealthChop nicht möglich. Ebenso ist im dcStep-Betrieb die Funktionalität von stallGuard2 nicht gegeben. Hier ist eine separate Stall-Erkennung nutzbar. Die möglichen Kombinationen von Chopper-Modi und Funktionen sind in der folgenden Grafik dargestellt:

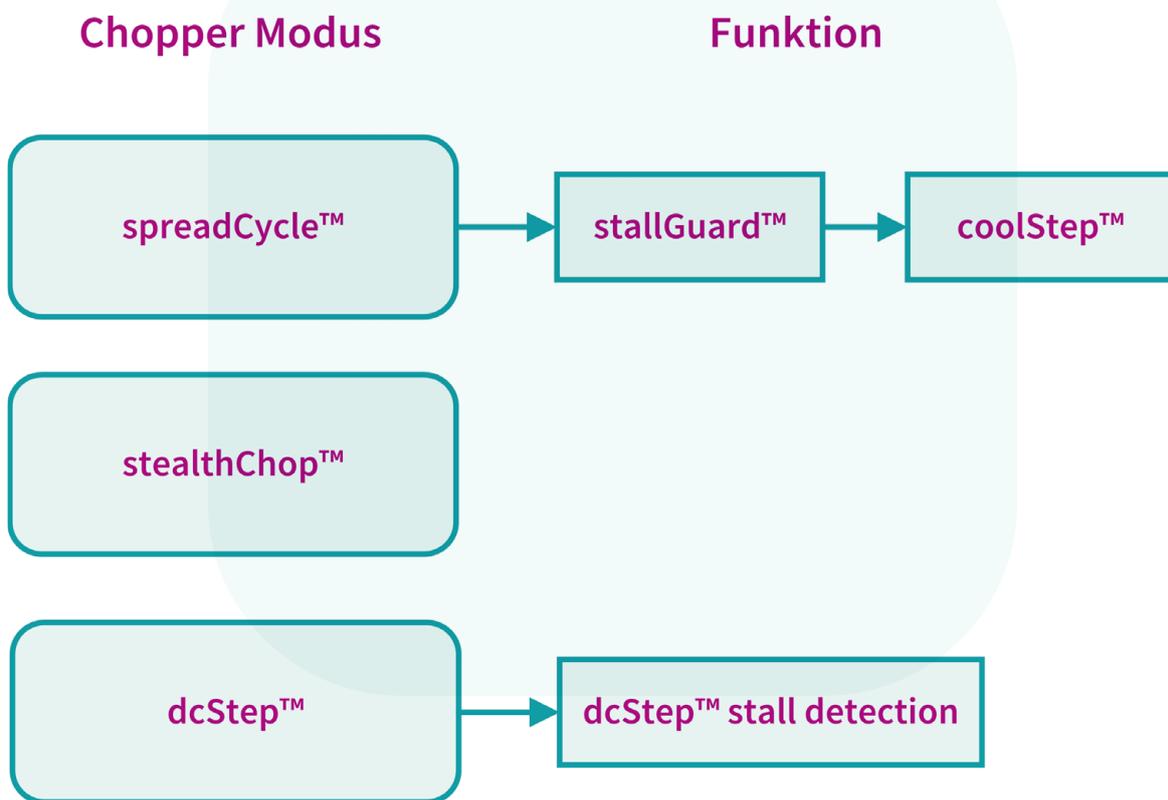


Bild: Mögliche Kombinationen von Chopper und Funktionen

2.2 Das TMC5130-EVAL-KIT

Das Evaluationskit besteht aus dem TMC5130-EVAL, der einen TMC5130 als kombinierten Motion Controller und Smart Driver enthält, sowie dem MCU-Board „Startrampe“ oder „Landungsbrücke“. Der Unterschied zwischen Landungsbrücke und Startrampe ist für den Nutzer hier nicht relevant. Die wesentliche Funktion des MCU-Boards ist das Übersetzen der TMCL-Befehle (Trinamic Motion Control Language) von der TMCL-IDE auf dem PC zu SPI-Telegrammen für den TMC5130. Die beiden Boards werden mit einer Brückenplatine (Eselsbrücke) miteinander verbunden. Auf der Brücke befinden sich beschriftete Testpunkte für alle Signale.

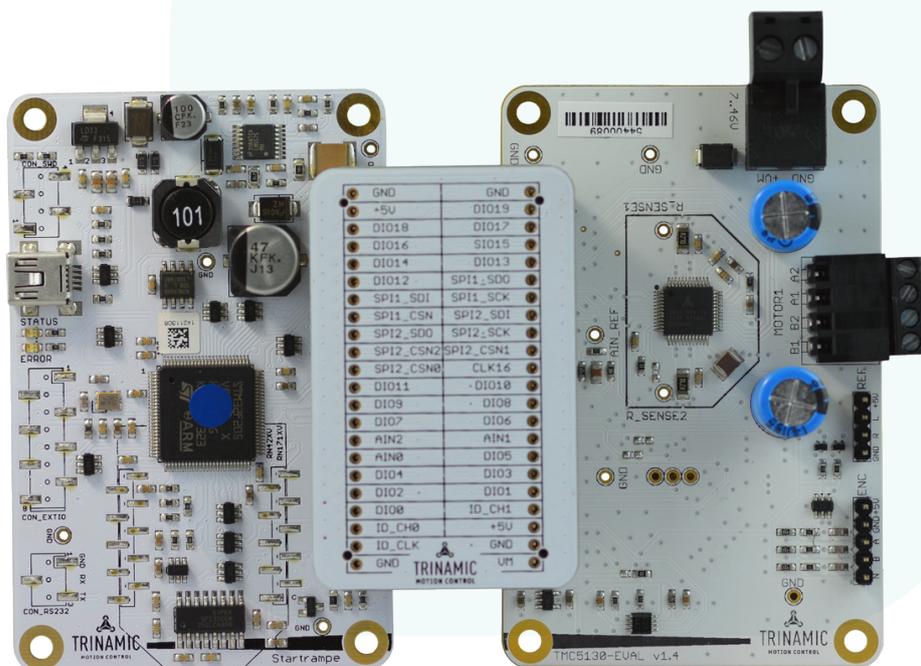


Bild: Startrampe, Eselsbrücke und TMC5130-Eval

3 Inbetriebnahme

“

3.1 Vorbereitung des Evaluierungssystems

Zunächst wird die Windows PC-Software TMCL-IDE benötigt. Diese kann in der aktuellsten Version kostenlos von der Trinamic Website heruntergeladen werden und ist nach einer kurzen Installation bereit: <https://www.trinamic.com/support/software/tmcl-ide/>

Um das Evaluationssystem vorzubereiten, werden das MCU-Board (Landungsbrücke oder Startrampe) und das TMC5130-EVAL mit der Eselsbrücke verbunden. Dabei ist darauf zu achten, dass die Trinamic-Logos und schwarzen Balken unten in dieselbe Richtung zeigen. Anschließend ist ein bipolarer Schrittmotor mit dem Motoranschluss des TMC5130-EVAL zu verbinden. Alle notwendigen Stecker liegen bei.

Nun kann eine Gleichspannungsquelle (7-46 V) angeschlossen werden. Nachdem diese eingeschaltet wurde, leuchtet auf dem MCU-Board eine grüne LED (neben der USB-Buchse).

Achtung: Bitte achten Sie darauf, dass der Motor nicht bei aktiviertem Treiber abgezogen werden darf, da dieser sonst zerstört werden kann.

Schalten sie zuerst die Versorgung ab oder deaktivieren Sie den Treiber über die Software.

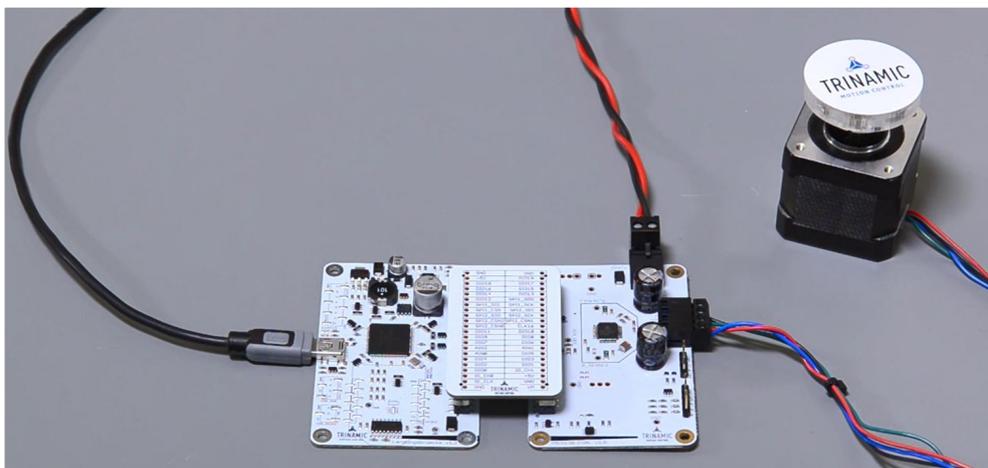


Bild: Angeschlossenes Evaluationssystem

Als letztes wird das Evaluationssystem per USB an den PC angeschlossen. Beim ersten Anschluss wird der USB-Treiber automatisch installiert und nach öffnen der TMCL-IDE wird das jeweilige MCU-Board angezeigt (Landungsbrücke oder Startrampe).

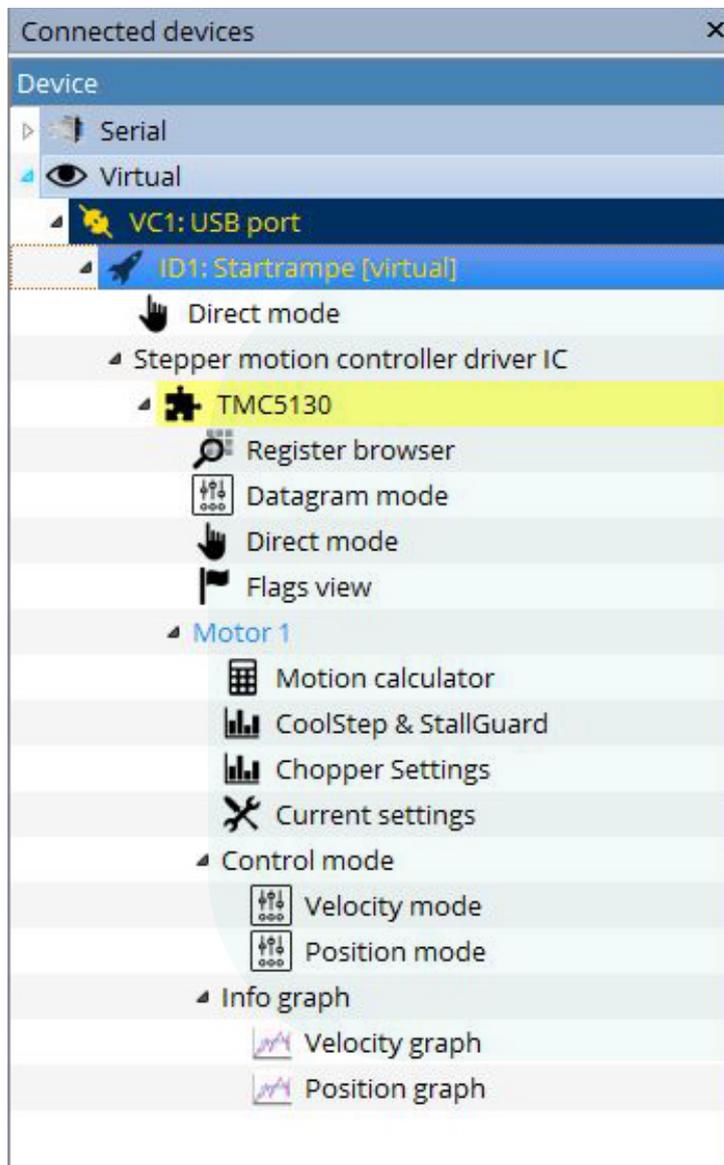


Bild: Funktionen des erkannten Evaluationssystem: Startrampe & TMC5130

3.2 Bedienung der IDE

Hier im Beispiel mit Version 3.0.18.0

Zur Bedienung der IDE gibt es drei Bereiche: Grundfunktionen der IDE können über die Drop-Down Menüs oben links angewählt werden **(1)**. Spezialfunktionen finden sich oben rechts **(2)**. Die Funktionen der Angeschlossenen Module (Devices) werden über die Menü-Struktur auf der linken Seite angewählt **(3)**.

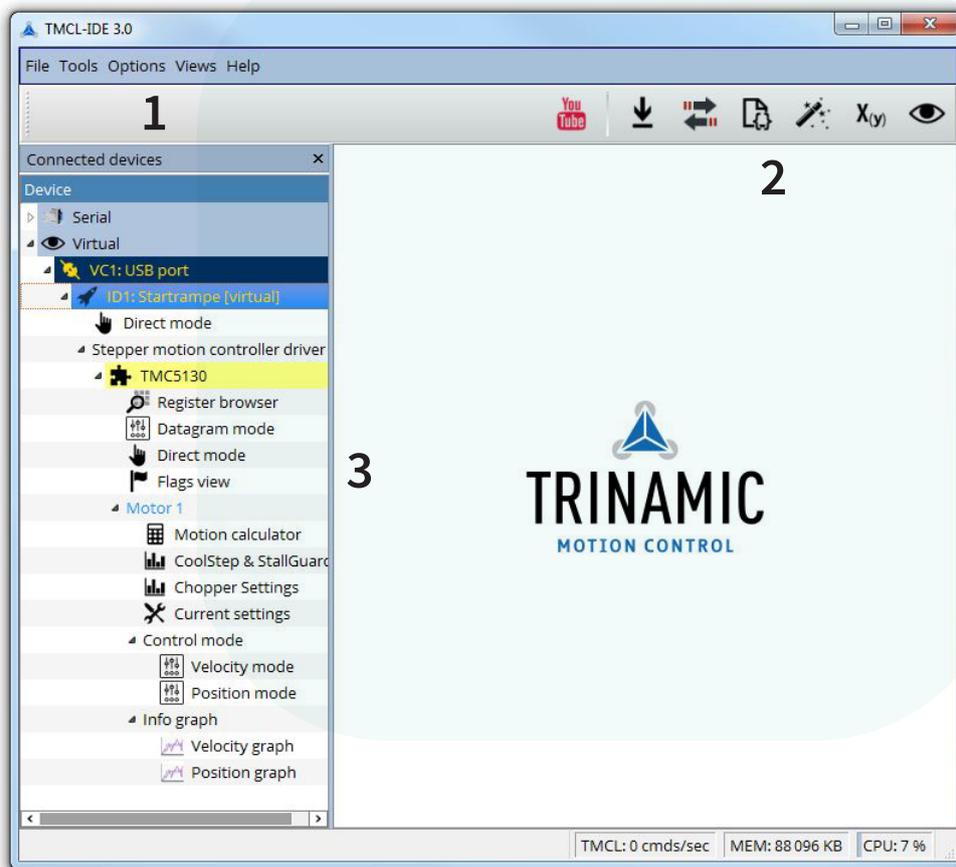


Bild: IDE mit TMC5130-Eval

Bedeutung der Symbole für Spezial-Funktionen (2):

	Download-Manager für Updates		YouTube-Kanal von TRINAMIC
	Firmware-Update		Im- und Export der Einstellungen
	TMCL/PC-Host		<u>Wizzard</u>
	XY-Plotter		Virtuelles Modul hinzufügen

Die Fenster zur Einstellung der Parameter zur Inbetriebnahme werden über die Device-Menue-Struktur (3) der TMCL-IDE aufgerufen.

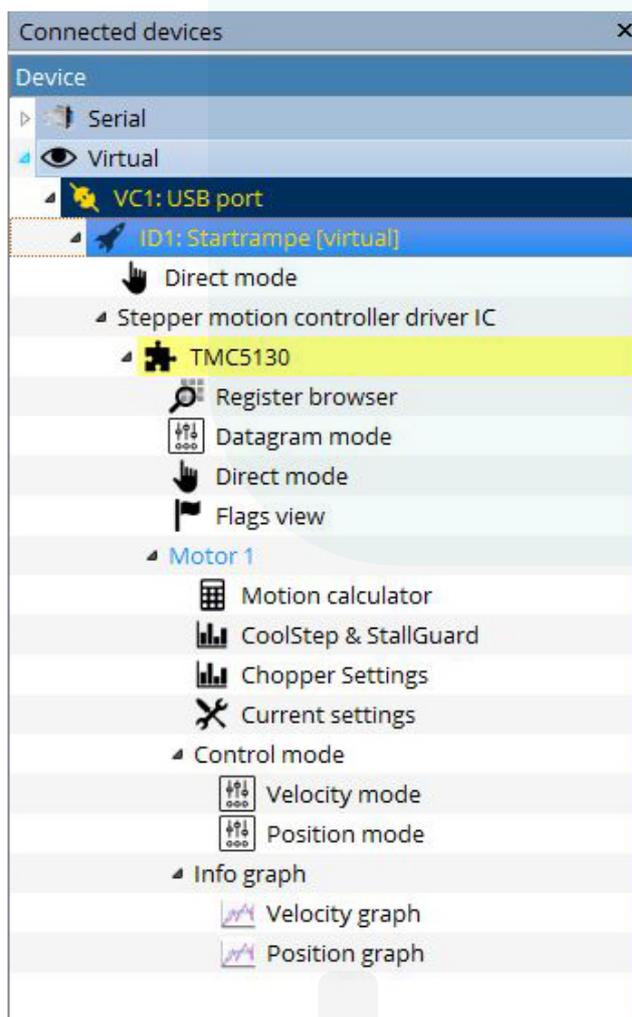


Bild: Device-Menü der IDE

3.2.1 Stromeinstellung

→ Fenster „Motor 1 / Current settings“

Als erstes erfolgt die Grundeinstellung des Motor-Stroms.

Der Arbeitsstrom wird mit dem Parameter I_{RUN} eingestellt. In der Betriebsart Konstantstrom-Chopper wird dieser Strom konstant in die Motorwicklungen eingeprägt. Da im Stillstand häufig nicht der volle Strom benötigt wird um das Haltemoment aufzubringen, kann der Strom reduziert werden, um die Motorerwärmung zu reduzieren. Der Haltestrom wird im Parameter I_{HOLD} eingestellt.

Mit I_{HOLD_DELAY} wird die Zeit eingestellt, die der Motor noch mit I_{RUN} bestromt wird, nachdem ein Stillstand ($v=0$) festgestellt wurde. Ist diese Zeit abgelaufen, wird der Strom Stufenweise von I_{RUN} auf I_{HOLD} abgesenkt. Die Länge der einzelnen Stufen wird in $T_{POWERDOWN}$ angegeben.

Bei allen Angaben bezieht sich die Stromangabe auf den Phasenstrom einer Motorphase als quadratischer Mittelwert (RMS).

Die Parameter I_{RUN} (Arbeitsstrom) und I_{HOLD} (Haltestrom) können in 32 Stufen eingestellt werden. Im IDE-Fenster wird auch der physikalische Stromwert in A (RMS) angezeigt.

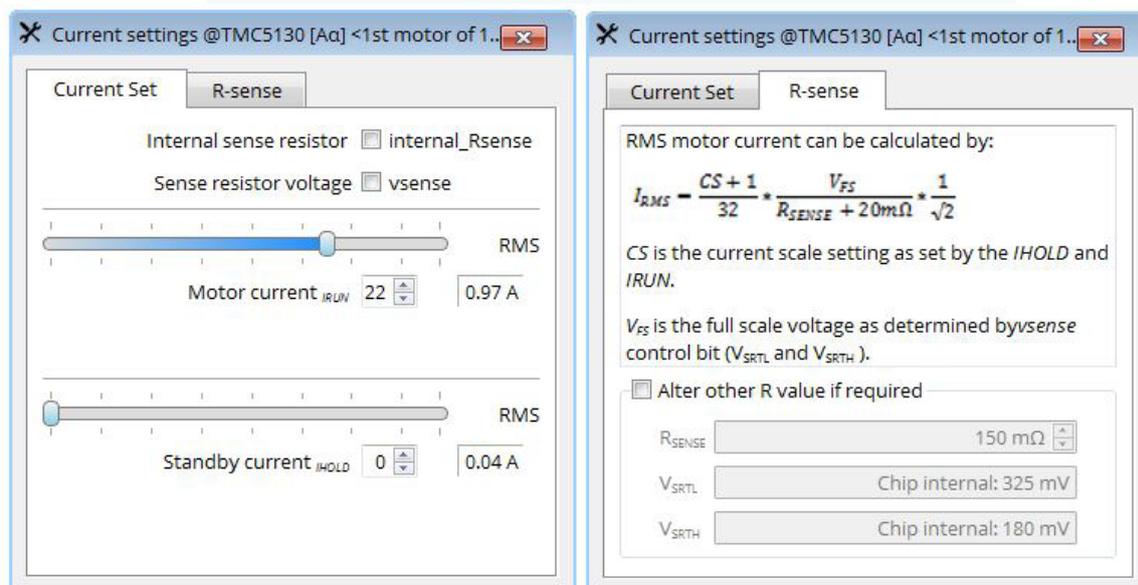


Bild: Fenster „Current settings“

Sollten die originalen Strommesswiderstände R_{SENSE} verändert worden sein, so kann im „R-sense“ Reiter der Punkt „Alter other R value if required“ aktiviert werden. Unter R_{SENSE} ist dann der aktuelle Widerstandswert einzugeben. Der Stromwert im Reiter „Current Set“ ändert sich dann entsprechend.

Alternativ können IRUN und IHOLD mit der im Fenster angegebenen Formel berechnet werden.

CS	=	Current Scaling-Wert (IRUN bzw. IHOLD)
I_{RMS}	=	gewünschter Phasenstrom in [A]
V_{FS}	=	Full Scale Voltage (VSRTL bzw. VSRTH) in [V]
R_{sense}	=	Rsense-Widerstand in [Ω]

3.2.2 Berechnung der physikalischen Einheiten

→ Fenster „Motor 1 / Motion calculator“

Die Parameter des TMC5130 werden nicht mit physikalischen Einheiten, sondern mit internen Registerwerten beschrieben. Das einfache Umrechnen in physikalische Einheiten ist mit dem Tool „Motion Calculator“ möglich.

Zunächst werden die Systemparameter und die Mikroschrittauflösung eingegeben, dann können aus Registerwerten physikalischen Einheiten angezeigt werden und umgekehrt.

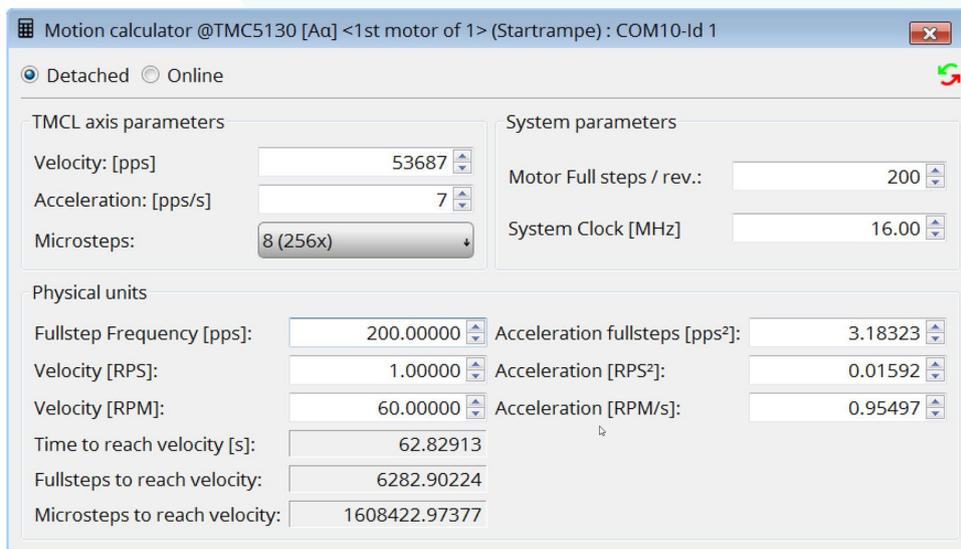


Bild: Motion Calculator zur Berechnung der Geschwindigkeitseinstellung

3.2.3 Motor konstant drehen

→ Fenster „Control mode / Velocity mode“

Als einfachste Anwendung kann der Motor mit konstanter Geschwindigkeit drehen. Diese Anwendung beschreibt der „Velocity mode“.

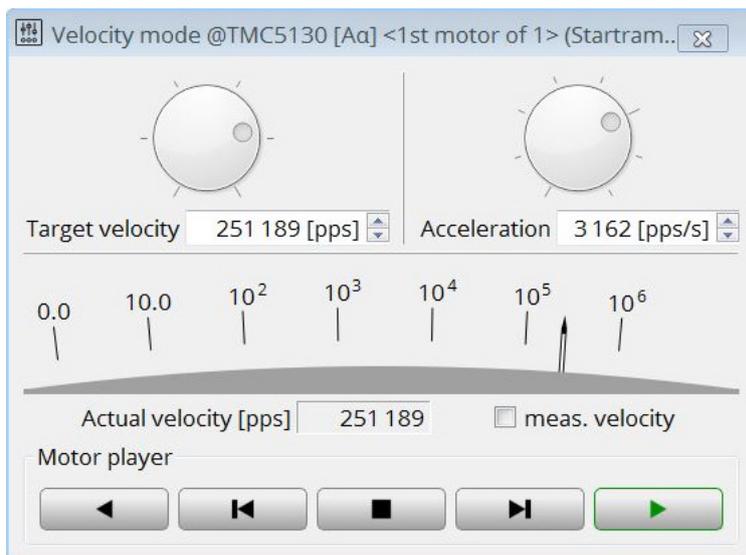


Bild: Geschwindigkeitseinstellung im Fenster Velocity mode

Im Fenster **Velocity mode** können die Geschwindigkeit und die Beschleunigung eingestellt werden: Target Velocity [pps] und Acceleration [pps/s]. Ein Klick auf den Pfeil nach rechts oder nach links setzt den Motor in Bewegung.

3.2.4 Motor positionieren

→ Fenster „Control mode / Position mode“

Im Fenster **Position mode** kann der Motor positioniert werden. Dabei kann entweder die Position oder die Anzahl der Schritte vorgegeben werden. Ebenso ist es möglich den Motor automatisch zwischen zwei Positionen hin- und herfahren zu lassen.

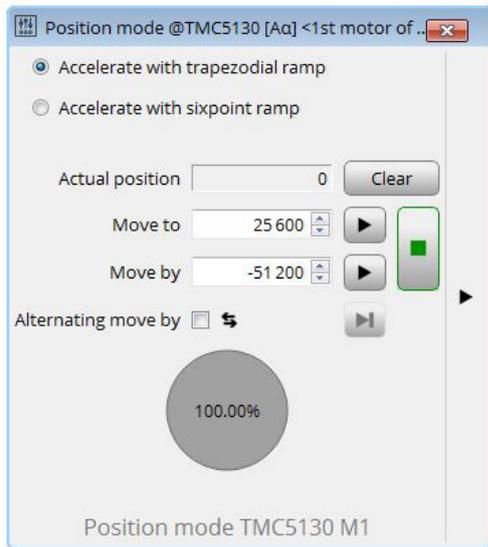


Bild: Positionsvorgaben im Fenster Position mode

3.2.5 Direktmodus

→ Fenster „TMC5130 / Direct mode“

Im **Direct mode** Fenster können sämtliche TMCL-Befehle direkt ausgewählt und ausgeführt werden.

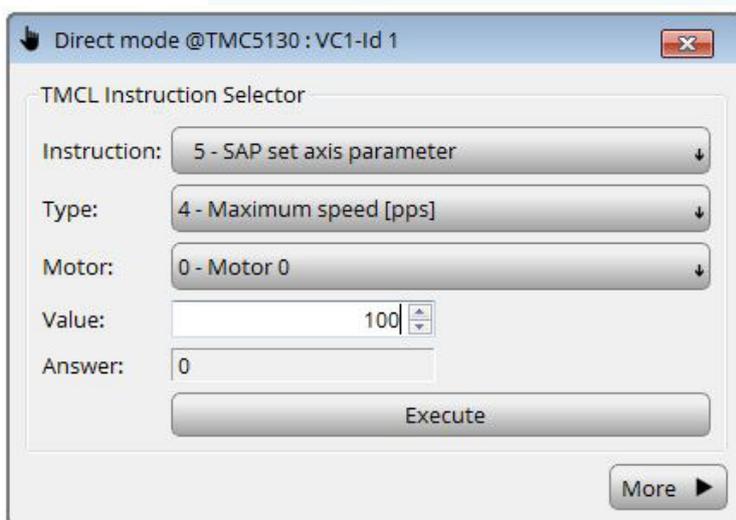


Bild: Direct mode

3.2.6 Register Browser

→ Fenster „TMC5130 / Register browser“

Im Fenster **Register browser** kann der Zustand sämtlicher Register eingesehen werden. Zusätzlich gibt es Informationen zu den einzelnen Bits.

Name	ADR	ACS	Size/Mask	Read value	To write value	Function	Description(s)
Active registers							
SLAVECONF	0x03	W	32	0x00000000	0x00000000		
CHOPCONF	0x6C	RW	32	0x00000000	0x00000000		
toff		RW	0000 000F	id: 0	Choice ↓	Dr...ble	off time and driver enable
HSTRT		RW	0000 0070	id: 0	Choice ↓	HS... 1	hysteresis sta...each 16 clocks
HEND		RW	0000 0780	id: 0	Choice ↓	-3	hysteresis low value
rndtf		RW	0000 2000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 13	Ch...FF	random TOFF time
chm		RW	0000 4000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 14	Sta...le	chopper mode
tbl		RW	0001 8000	id: 0	Choice ↓	16 ...les	blank time select
vsense		RW	0002 0000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 17	Lo...ge	sense resistor...urrent scaling
vhighfs		RW	0004 0000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 18	Hig...led	high velocity fullstep selection
vhighchm		RW	0008 0000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 19	Hig...led	high velocity chopper mode
sync		RW	00F0 0000	0	Unsigned ↓		PWM synchronization clock
mres		RW	0F00 0000	id: 0	Choice ↓	256...er.	Micro step res...ep resolution.
intpol		RW	1000 0000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 28	No ...ion	interpolation to 256 microsteps
dedge		RW	2000 0000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 29	Sin...ses	enable double ...ge step pulses
diss2g		RW	4000 0000	<input type="checkbox"/> low	<input type="checkbox"/> Set bit 30	Sho... on	short to GND protection disable

Bild: Register browser

4 Parametrierung der Funktionen

“

4.1 spreadCycle – Konstantstrom-Chopper

Der spreadCycle Konstantstrom-Chopper mit fester Ausschaltzeit und Hysterese-Regelung bietet ein deutlich verbessertes Stromregelverhalten gegenüber einem herkömmlichen Konstantstrom-Chopper mit fester Ausschaltzeit mit und ohne Mixed-Decay.



Bild: spreadCycle Konstantstrom-Chopper-Schema

Beim spreadCycle folgt auf die Einschaltphase erst Slow-Decay, dann Fast-Decay gefolgt von einer weiteren Slow-Decay-Phase. So ergibt sich ein symmetrischer Stromrippel. Der Durchschnittsstrom entspricht dem Zielstrom. Auf Grund der einstellbaren Hysterese-Punkte, optimiert sich der Fast-Decay-Anteil von selbst, so dass stets ein sauberer sinusförmiger Phasenstrom erreicht wird.

Weiterhin trägt der somit möglichst geringe Fast-Decay Anteil dazu bei, dass der Motor nicht unnötig durch Stromrippel erhitzt wird. Ein sauber regelnder Chopper ist die Voraussetzung für eine sinnvolle Verwendung von hohen Mikroschrittauflösungen. Versuche haben gezeigt, dass die Qualität des Choppers wichtiger für die Laufruhe ist als die Mikroschrittauflösung. Ein Motor, der mit 16 Mikroschritten und spreadCycle betrieben wird, ist bereits ruhiger und

vibrationsärmer als mit 256 Mikroschritten und einem nicht optimierten Mixed-Decay Chopper mit fester Ausschaltzeit.

Die optimale Konfigurierung von `spreadCycle` wird mit Hilfe einer Berechnung und eines Algorithmus erstellt. Die Einstellung der Werte erfolgt im Fenster „Register Browser“ unter `CHOPCONF (0x6C)`.

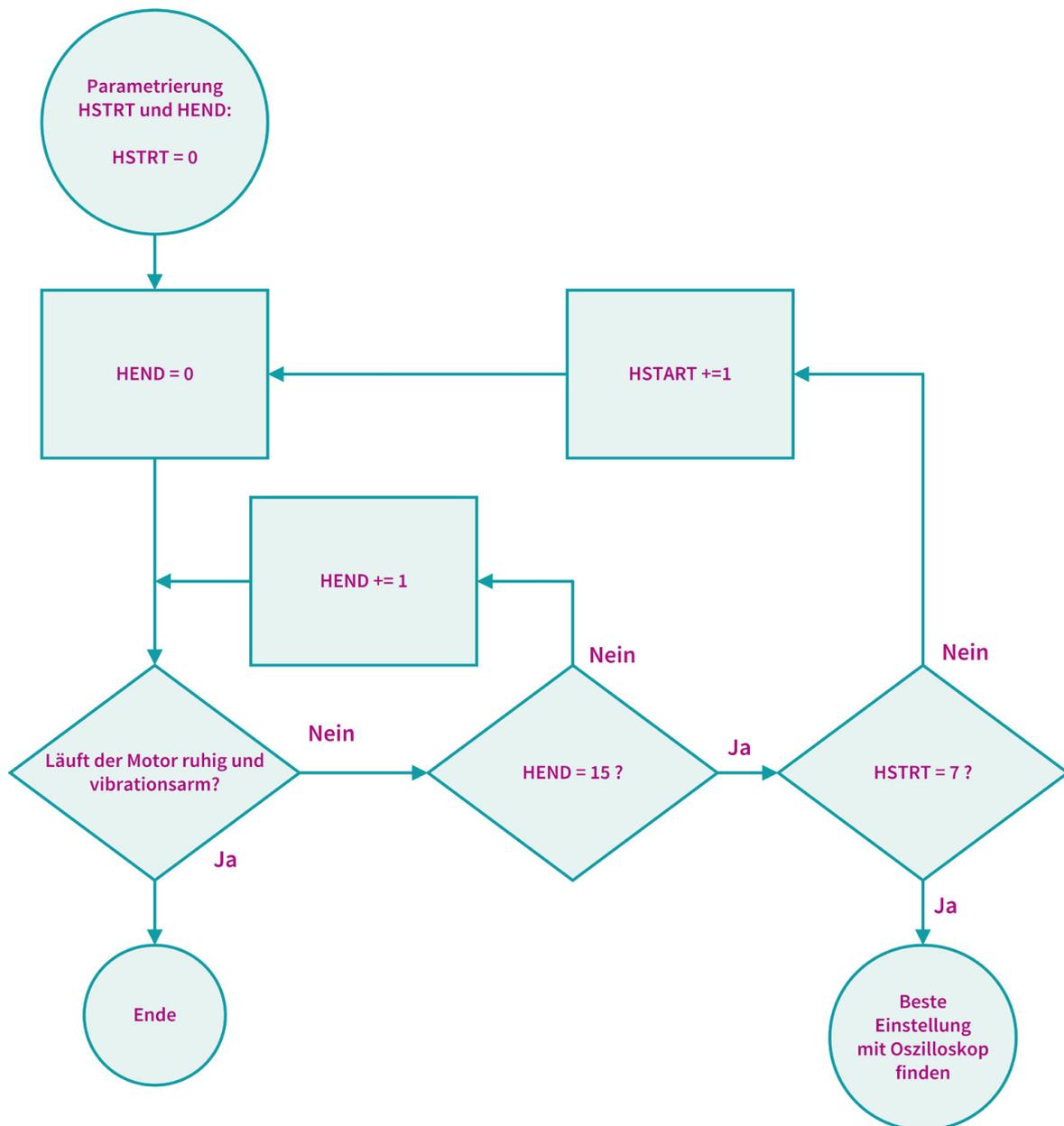
Nach der folgenden Formel ist der Registereintrag `TOFF` mit dem für die Slow-Decay-Zeit ausgewählten Wert zu errechnen:

$$TOFF = t_{off} * \frac{f_{clk}}{32} - \frac{12}{32}$$

<code>TOFF</code>	=	Beschleunigung in [Mikroschritte / Systemtakt ²]
<code>t_{off}</code>	=	Slow-Decay-Zeit
<code>f_{clk}</code>	=	Frequenz der System Clock in [Hz]

Von TRINAMIC wird eine Zeit von 5µs bis 20µs für `toff` empfohlen.

Anschließend sind die Start- und Endpunkte der Hysterese `HEND` und `HSTRT` einzustellen. Diese sind über den folgenden Algorithmus ermittelbar. Zu Beginn der Parametrierung sollte sichergestellt werden, dass `stealthChop` ausgeschaltet ist (`GCONF (0x00) -> en_pwm_mode = 0`) sowie `spreadCycle` aktiviert ist (`CHOPCONF (0x6C) -> chm = 0`).



Mit einem Oszilloskop kann der Phasen-Strom überprüft werden. Er sollte möglichst sinusförmig sein. Sollten sich in der Nähe des Nulldurchgangs Verzerrungen zeigen, kann die Blank Time des Choppers mit dem Register *TBL* angepasst werden (z.B. über den Registerbrowser). Die Blank Time sollte so gering wie möglich gehalten werden.

Beispielrechnung:

Bei der Kombination des TMC5130-EVAL-KITs ist der vorgegebene Systemtakt 16MHz. Für t_{off} wird gemäß der Empfehlung eine Zeit von $t_{off} = 10\mu s$ gewählt.

$$\begin{aligned}
 TOFF &= t_{off} * \frac{f_{clk}}{32} - \frac{12}{32} \\
 &= 10\mu s * \frac{16MHz}{32} - \frac{12}{32} \\
 &= 4,625 \\
 &\approx 5
 \end{aligned}$$

4.1.1 stallGuard2

→ Fenster „Motor 1 / coolStep“, Reiter „stallGuard2“

Zur Einstellung der sensorlosen Lasterkennung stallGuard2 wird im Fenster coolStep der stallGuard-Wert grafisch dargestellt. Dieser zeigt die Lastreserve des Motors. Mit dem Parameter *stall guard threshold* wird ein Offset ausgeglichen, der durch die Gegen-EMK entsteht. Ziel ist es, unter Last den stallGuard-Wert auf 0 sinken zu lassen, bevor ein Schrittverlust durch Überlastung („stall“) entsteht. Im laufenden Betrieb bei normaler Last sollte der Wert nach der Korrektur durch den *stall guard threshold* zwischen 0 und 1023 sein.

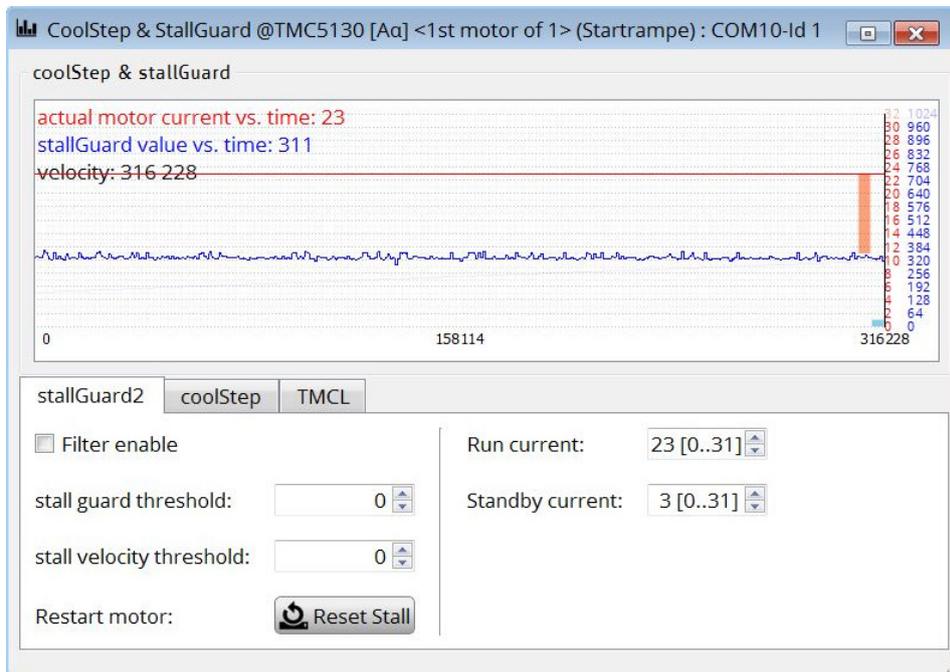
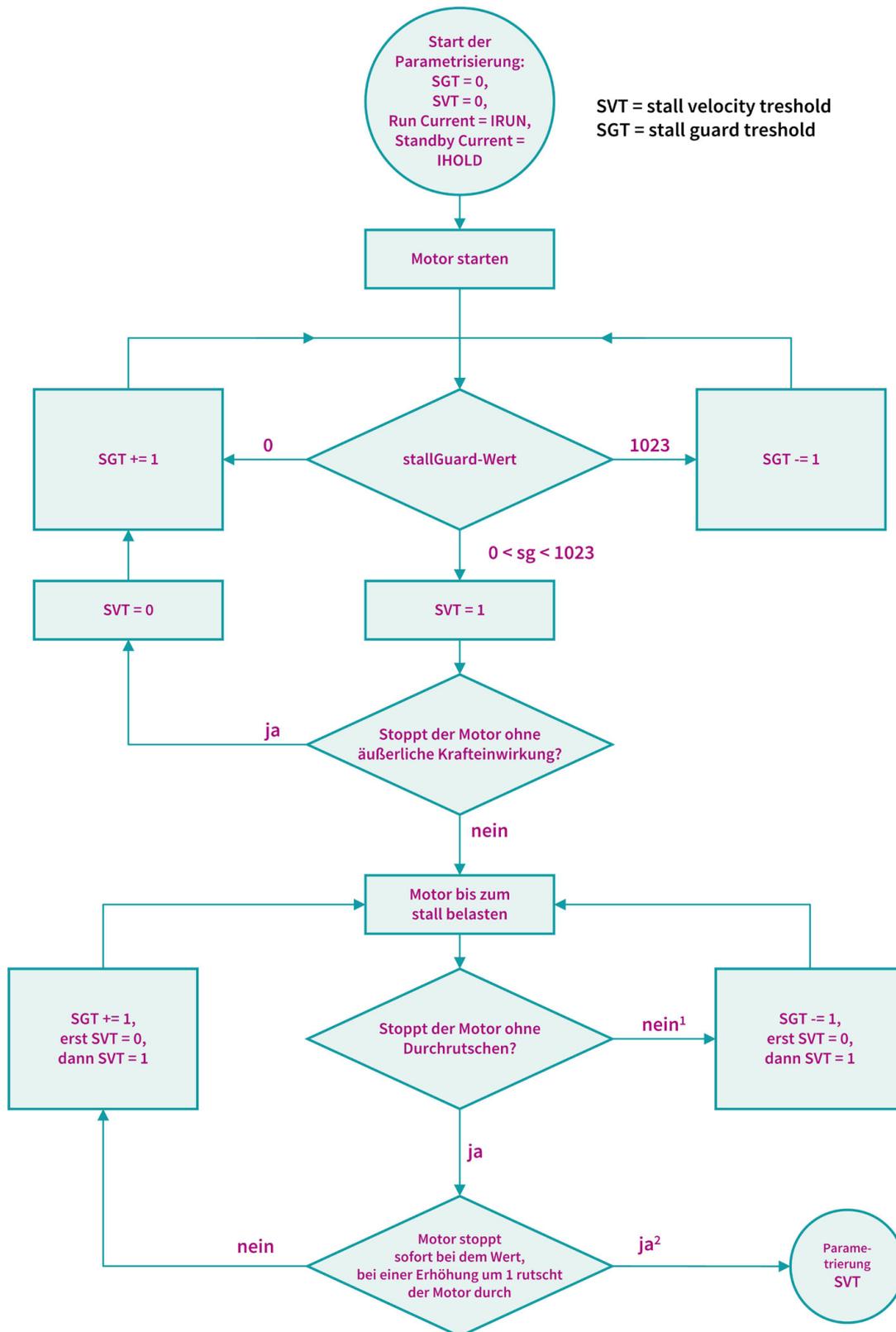


Bild: stallGuard-Einstellungen im Fenster coolStep

Im Reiter „stallGuard“ finden sich die Einstellungen für stallGuard. Auf der rechten Seite befinden sich erneut *IRUN* (Arbeitsstrom) und *IHOLD* (Haltestrom), auf der linken Seite ist oben der *Filter* aktivierbar, darunter sind *stall guard threshold* und *stall velocity threshold* einstellbar. Der Filter bewirkt, dass der stallGuard-Wert bei schwankenden Funktionen gefiltert wird und glättet die Schwankungen.

Die maximale Einstellung von *stall guard threshold* kann mit einem Algorithmus ermittelt werden. Zuvor sollten *IRUN* und *IHOLD* eingestellt sein.

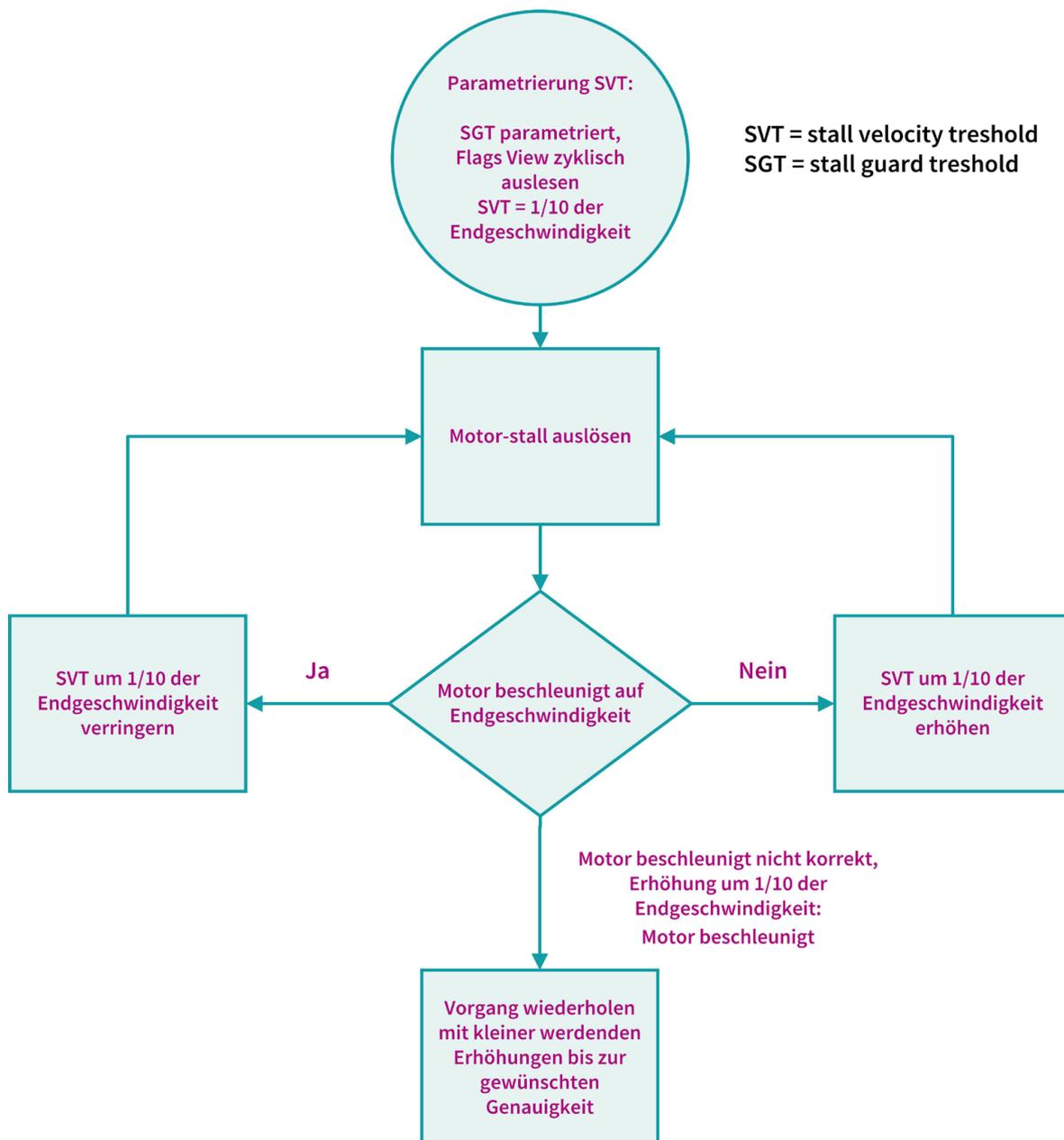
Tipp: Durch Montage eines Klemmflansches kann der Motor einfach mit der Hand belastet werden.



Anmerkungen zum stallGuard-Algorithmus:

1. Für manche Motoren lassen sich keine Einstellungen finden, bei denen der Motor ohne Durchrutschen stoppt. Dies kann z.B. durch eine geringe Magnetfeldrückwirkung des Motors begründet sein. In dem Fall ist stallGuard i.d.R. nicht einsetzbar.
2. Hier ist der Maximalwert gefunden, bei dem stallGuard seine Funktion erfüllt. Um eine höhere Sicherheit zu bekommen, kann der Wert nun wieder verringert werden. Eine Verringerung um 1-2 sollte in den meisten Fällen ausreichen.

Stall velocity threshold stellt hierbei den Schwellenwert ein, ab welcher Geschwindigkeit stallGuard aktiviert wird. Da bei der Beschleunigung der Wert auf 0 absinkt, würde dies als stall erkannt werden und der Motor sofort wieder stoppen. Daher sollte stallGuard erst eingeschaltet werden, wenn der Motor bereits eine Geschwindigkeit aufgenommen hat, bei welcher der stallGuard-Wert größer als 0 ist. Die Einstellung hierzu ist auch in einem Algorithmus zu ermitteln:

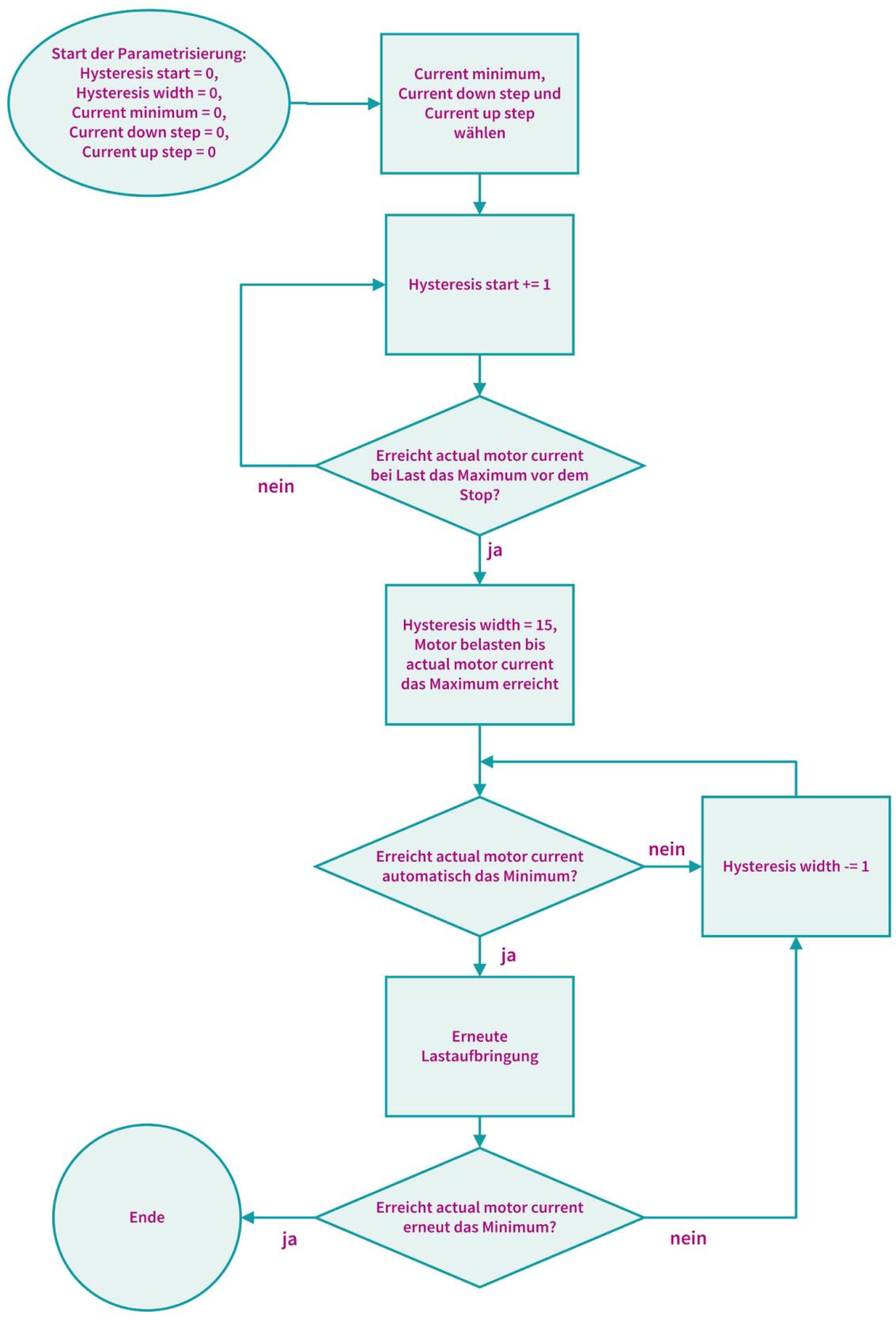


4.1.2 coolStep

Die lastabhängige Stromanpassung coolStep nutzt den stallGuard-Wert, um bei einer ausreichenden Lastreserve den Strom zu reduzieren. Dadurch wird sowohl Energie gespart, wodurch sich der Motor durch die geringeren Wärmeverluste nicht so stark erhitzt, sondern auch die Lautstärke des Motors reduziert. Sollte die Lastreserve absinken, erhält der Motor mehr Strom und kann damit mehr Drehmoment aufbringen.

Um coolStep einzustellen, sind mit *Hysteresis start* und *Hysteresis width* zwei Parameter so einzustellen, dass zum einen das Reduzieren als auch das bedarfsweise Inkrementieren ermöglicht wird. Die Parametrisierung erfolgt nach dem Algorithmus für coolStep:





Bei der Verwendung von coolStep und stallGuard können aus dem Verlaufsgraphen des stallGuard-Wertes Informationen zur Art der Motorlaständerung gezogen werden:

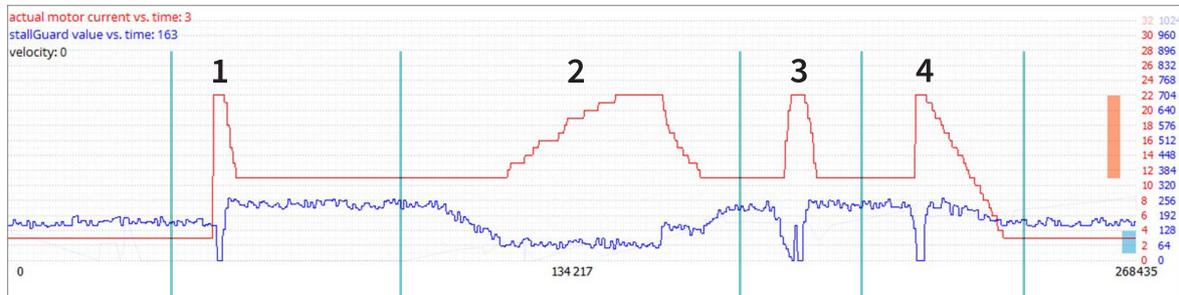


Bild: coolStep-Graph mit verschiedenen Events: Start, Belastung ohne Stall, Stall und Motorstopp

Die Events im oben gezeigten Bild sind zur Verdeutlichung des Verlaufes aufgezeichnet. Zunächst wird der Motor auf die gewünschte Geschwindigkeit beschleunigt **(1)**. Der Strom steigt von der Einstellung *I*HOLD auf *I*RUN an. Dabei sinkt der stallGuard-Wert auf 0 ab, woraus sich die Notwendigkeit der Einstellung von *stall velocity threshold* ergibt. Ist während der Beschleunigung stallGuard aktiv, würde der Motor sofort wieder gestoppt.

Als nächstes wurde der Motor belastet **(2)**. Dabei ist zu beobachten, dass der Betriebsstrom langsam ansteigt, während der stallGuard-Wert absinkt. Nach Entlastung steigt der stallGuard-Wert wieder, und der Strom wird wieder verringert, wodurch sich der stallGuard-Wert wieder erhöht.

Anschließend wurde der Motor schlagartig bis zum Stall belastet, also ein harter Stopp simuliert **(3)**. Dabei ist der schnelle Stromanstieg zu beobachten, der aus dem schnellen Absinken des stallGuard-Wertes resultiert. Bei diesem harten Stopp ist zusätzlich zu beobachten, dass der Motor wieder anläuft (bedingt durch das zyklische Auslesen der Status Flags) und nach Erreichen der Mindestgeschwindigkeit für stallGuard erneut stoppt, da der Motor noch immer belastet ist. Der zweite Anfahrversuch ist erfolgreich und der Motor erreicht wieder seine eingestellte Endgeschwindigkeit.

Zuletzt erhält der Motion Controller den Stop-Befehl **(4)**, woraufhin der Motor sofort stoppt und der Strom auf das Maximum ansteigt, da die Mindestgeschwindigkeit für coolStep unterschritten wird. Der Strom wird nach dem Stillstand langsam auf den Ruhestrom *I*HOLD abgesenkt. Zum Ende erreichen sowohl stallGuard-Wert als auch Strom den Ruhezustand, wie er auch zu Beginn des Diagramms herrscht.

Die Einstellungen von *Current up step* und *Current down step* sind ebenfalls empirisch zu ermitteln. Für die Anwendung des Motors ist hierbei wichtig, dass die Stufen nicht zu groß gewählt werden. Zu große Schritte können schnell zu einem unnötigen Stop-on-stall-Event führen, da die Erhöhung des Betriebsstromes zu einem Absinken des stallGuard-Wertes führt. Ein zu langsames Ansteigen kann zu einer Überlastung führen.

4.2 stealthChop – „No-Noise-Chopper“

Absolut lautlosen Betrieb bei niedrigen Geschwindigkeiten bietet der stealthChop-Chopper Modus. Es handelt sich hier um einen spannungsgesteuerten PWM-Betrieb bei dem sich der Strom über den Wicklungswiderstand einstellt. Um der Gegen-EMK bei steigender Geschwindigkeit entgegenzuwirken, kann bei stealthChop auch der Strom gemessen und von Zeit zu Zeit nachgeführt werden. Diese Technik ermöglicht das Erreichen des gleichen Drehmomentes wie beim Strom-Chopper, jedoch kann nicht ganz so schnell beschleunigt werden. Der stealthChop-Modus kann im Fenster „Register Browser“ unter *GCONF* (0x00) -> *en_pwm_mode* = 1 aktiviert werden. Zusätzlich gibt es einen Wizard zur automatischen Einstellung. Die Werte entsprechen den Registereinträgen im Register *PWMCONF* (0x70). Einzig *VPWMTHRS* ist im Register *TPWMTHRS* (0x13) gespeichert.

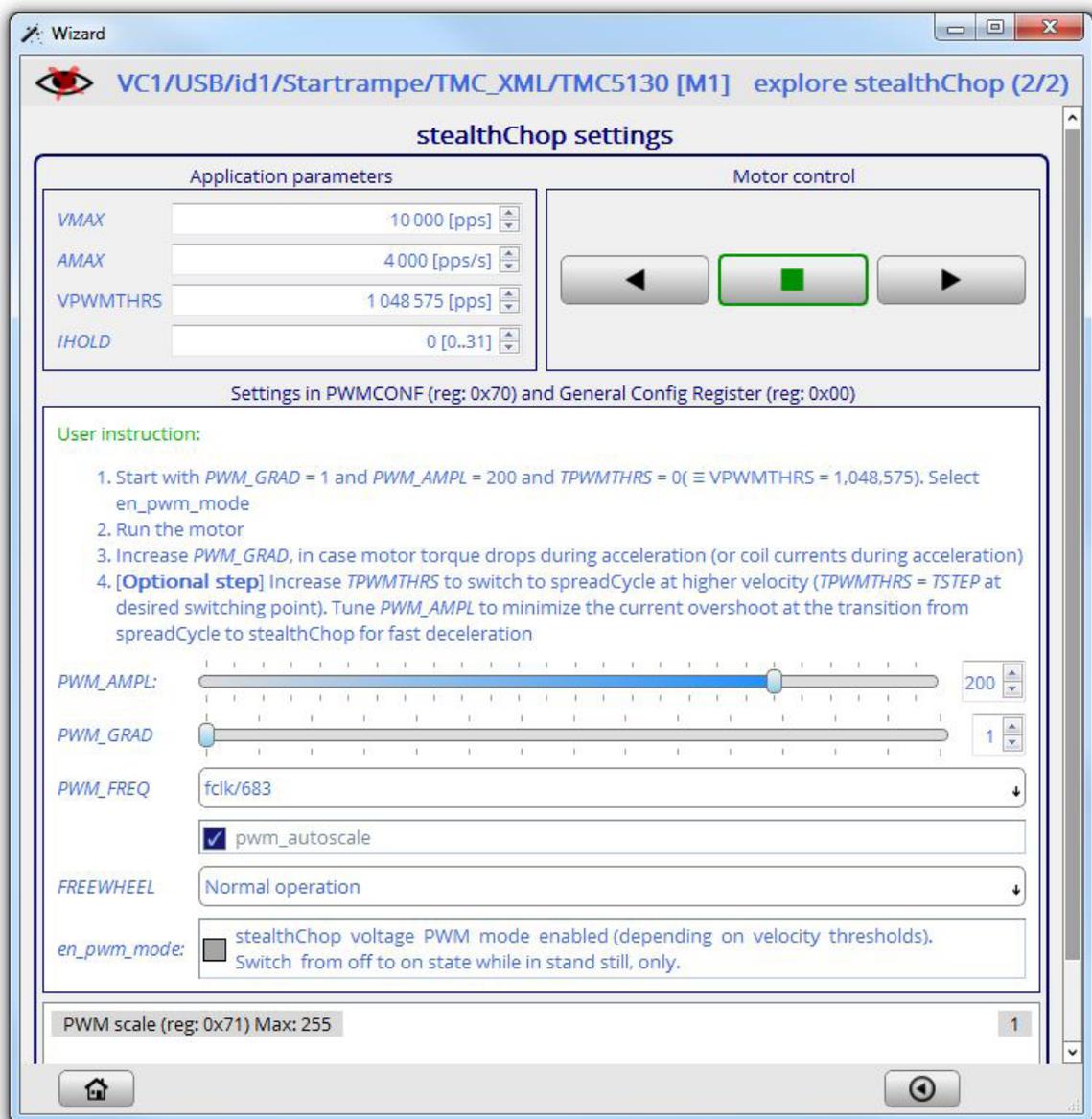


Bild: Wizzard für stealthChop

Als erstes ist die PWM-Frequenz zu ermitteln. Sie setzt sich aus dem Systemtakt (f_{clk}) und einem Teiler (einstellbar in PWM_FREQ) zusammen. Dabei darf die PWM-Periodendauer ($1/PWM$ -Frequenz) die elektrische Zeitkonstante des Motors nicht überschreiten, da sonst der entstehende Stromrippel das Drehmoment verringert und der Motor unruhiger läuft.

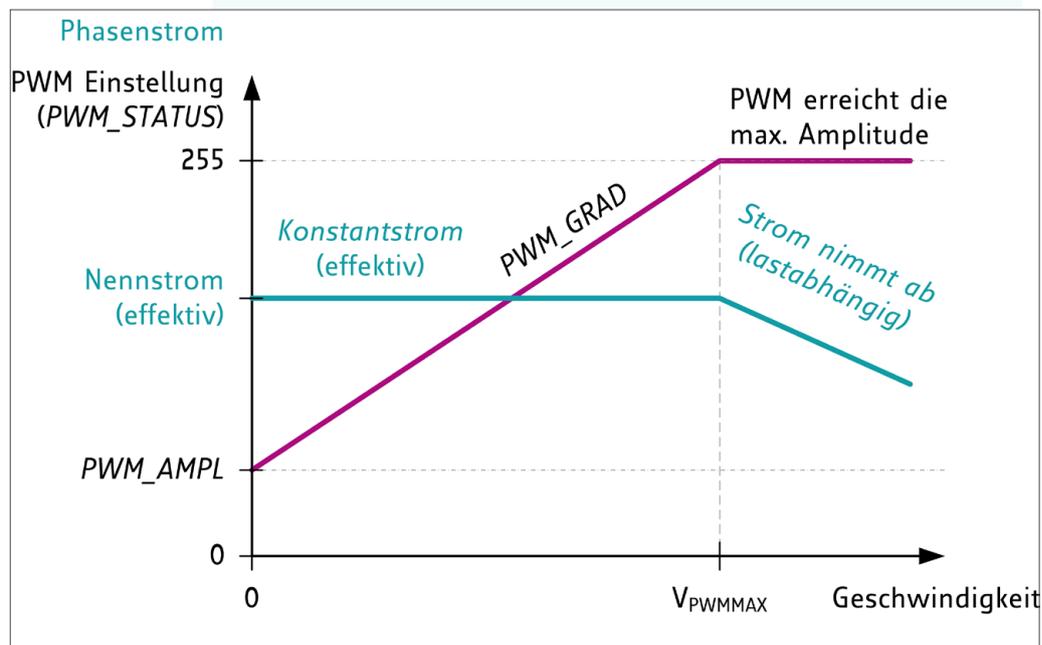
Um PWM_FREQ zu bestimmen, benötigt man zum einen die Frequenz des Systemtaktes (f_{clk}),

zum anderen aber die elektrische Zeitkonstante des Motors. Diese findet sich entweder im Datenblatt des Motors, oder kann aus der Formel

$$t_{mot} = \frac{L}{R}$$

t_{mot} = elektrische Zeitkonstante des Motors in [ms]
 L = Phaseninduktivität in [mH]
 R = Phasenwiderstand in [Ω]

errechnet werden. Das Evaluationsboard nutzt einen Systemtakt von $f_{clk} = 16$ MHz. Um die PWM-Zeitkonstante nicht zu überschreiten, ist bei dieser Motor-Evaluationsboard-Kombination ein Systemtaktteiler von mindestens 20 zu benutzen. Der TMC5130 bietet Systemtaktteiler von 410, 512, 683 und 1024 an. Bei einer effektiven Chopperfrequenz ($2 \cdot f_{CLK}/\text{PWM-Teiler}$) bis 15 kHz kann es zu unangenehmen Spulengeräuschen kommen. Die PWM-Amplitude (PWM_AMPL) stellt einen Offset für die PWM dar, bei der die Amplitude beginnt und bildet zusammen mit dem PWM-Gradient die resultierende Amplitude.



PWM_AMPL kann mit der Formel errechnet werden:

$$PWM_AMPL = \frac{374 * R_{COIL} * I_{COIL}}{U_M}$$

- R_{COIL} = Phasenwiderstand in [Ω]
 I_{COIL} = Phasenstrom in [A]
 U_M = Motorbetriebsspannung in [V]

Der PWM-Gradient (PWM_GRAD) stellt die Steigung des Motorstromes dar. Die gesamte Formel für den Strom beträgt:

$$PWM_GRAD = \frac{M_{Hold}}{2 * I_{COIL}} * 2\pi * \frac{f_{CLK} * 1,46}{U_M * MSPR}$$

- M_{Hold} = Haltemoment des Motors in [Nm]
 I_{COIL} = Phasenstrom in [A]
 f_{CLK} = Systemtakt in [Hz]
 U_M = Motorbetriebsspannung in [V]
 $MSPR$ = Mikroschritte pro Umdrehung

Das Setzen von `pwm_autoscale` ermöglicht es dem TMC5130A, den Phasenstrom nachzuregeln, um so das maximale Drehmoment mit stealthChop zu ermöglichen. Dabei wird der Strom einmal pro Vollschritt gemessen und innerhalb von etwa 10 Vollschritten nachgeregelt. Für eine erste Parametrisierung kann dieses Register auf 1 gesetzt werden. Der PWM-Wert kann sich während einer Periode ändern, das Aktivieren von `pwm_symmetric` erzwingt eine symmetrische PWM. Diese Funktion wird nur in Ausnahmen verwendet, daher wird dieser Wert bei 0 belassen.

Die Einstellung von *FREEWHEEL* umfasst die Optionen *Freewheeling*, *Normal Operation*, *Coil shorted using LS drivers* und *Coil shorted using HS drivers*. Damit wird das Verhalten der Treiberstufe im Stillstand definiert. *Freewheeling* ermöglicht das weitgehend freie Drehen des Motors, während *Coil shorted* jeweils passives Bremsverhalten erzeugen. Hierbei ist *Coil shorted using LS drivers* stärker, denn es werden kleinere Widerstände verwendet. Durch die Einstellung von *Normal operation* wird der Motor mit dem durch *I_{HOLD}* beschriebenen Strom gehalten.

Beispielrechnung:

Die Kennwerte eines Motors sind: Phaseninduktivität $L = 6,6\text{mH}$, ein Phasenwiderstand $R_{\text{COIL}} = 5,3\Omega$ sowie ein Phasenstrom $I_{\text{COIL}} = 1\text{A}$. Die Versorgungsspannung U_M beträgt 24V . Das Haltemoment beträgt $M_{\text{HOLD}} = 0,27\text{Nm}$, die Systemfrequenz beträgt $f_{\text{CLK}} = 16\text{MHz}$ und es sind 51200 Mikroschritte pro Umdrehung.

Die elektrische Zeitkonstante des Motors ist:

$$t_{\text{Mot}} = \frac{L}{R}$$

$$= \frac{6,6\text{mH}}{5,3\Omega} = 1,25\text{ms}$$

Bei einem Systemtaktteiler von 512 beträgt die effektive Chopperfrequenz 62500Hz und die PWM-Periodendauer $0,032\text{ms}$. Dieser Teiler kann verwendet werden, da die PWM-Periodendauer kleiner ist als die elektrische Zeitkonstante des Motors.

Die Einstellung für *PWM_AMPL* ist:

$$PWM_{\text{AMPL}_{\text{Mot}}} = \frac{374 * R_{\text{COIL}} * I_{\text{COIL}}}{U_M}$$

$$= \frac{374 * 5,3\Omega * 1\text{A}}{24\text{V}} = 82,59$$

$$\approx 8$$

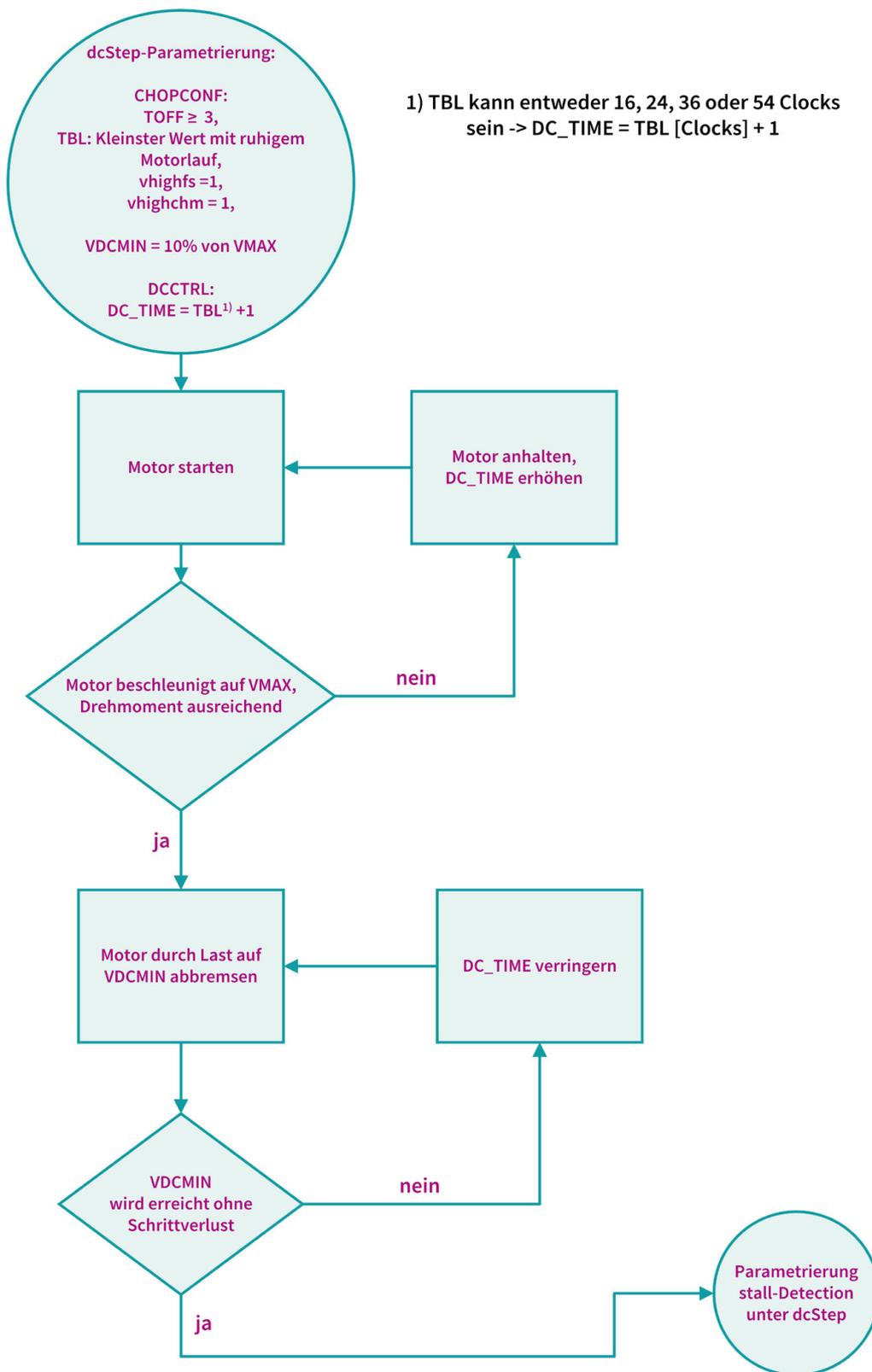
Die Einstellung für PWM_GRAD ist:

$$\begin{aligned}
 PWM_GRAD_{Mot} &= \frac{M_{Hold}}{2 * I_{COIL}} * 2\pi * \frac{f_{CLK} * 1,46}{U_M * MSPR} \\
 &= \frac{0,27Nm}{2 * 1A} * 2\pi * \frac{16 MHz * 1,46}{24V * 51200} \\
 &= 16,13 \\
 &\approx 16
 \end{aligned}$$

Da die Einstellung von PWM_GRAD nur bis 15 reicht, können die Phasenströme mit einem Oszilloskop überprüft werden. Die beste Einstellung befindet sich bei der am geringsten verzerrten Sinusform der Phasenströme.

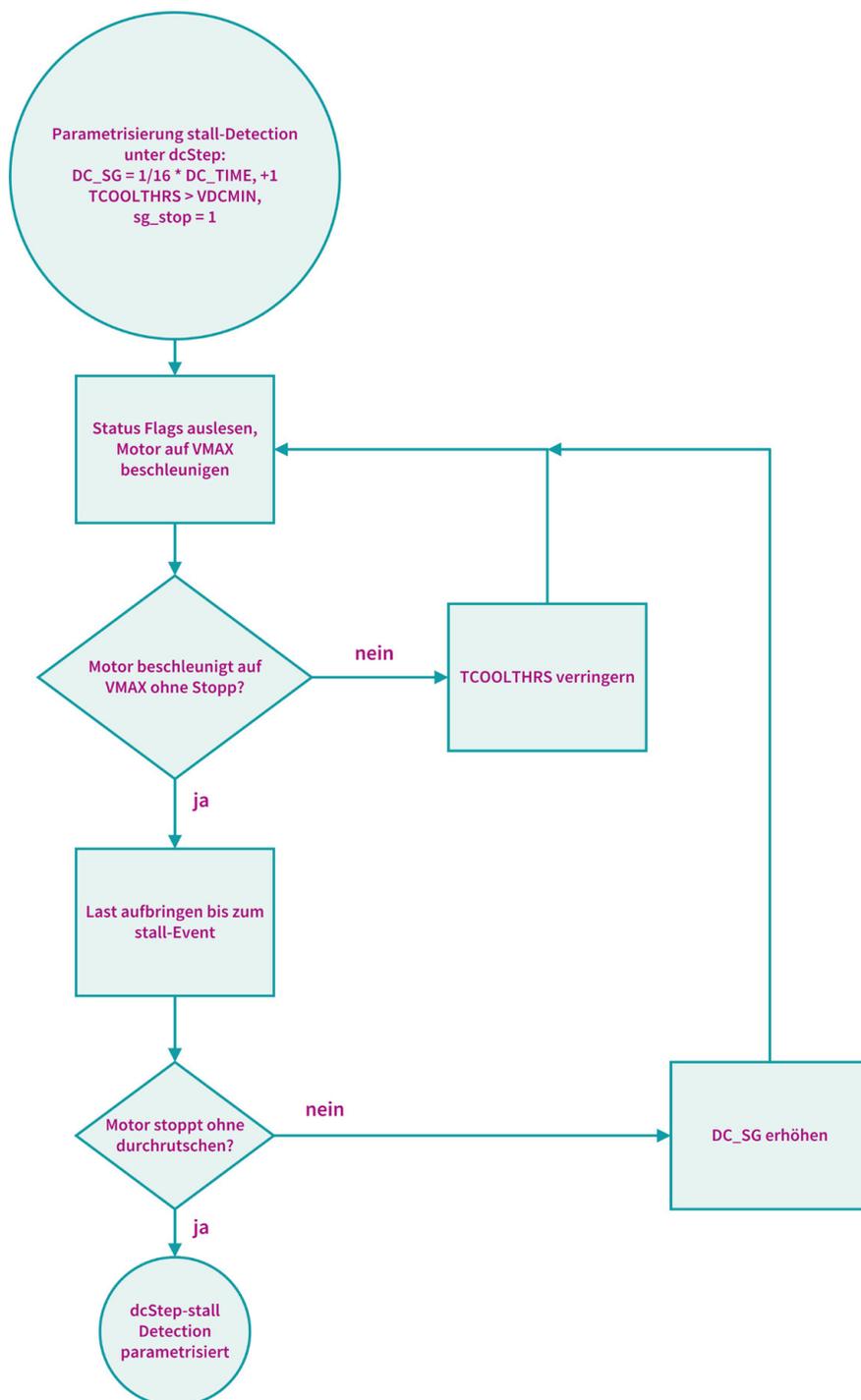
4.3 dcStep - lastabhängige Geschwindigkeitsanpassung

Das Parametrisieren der lastabhängigen Geschwindigkeitsanpassung dcStep ist auf Basis von Algorithmen möglich. Dabei kann die Parametrisierung in zwei Schritten erfolgen. Zuerst wird dcStep aktiviert. Die Parametrierung erfolgt wenn möglich mithilfe eines Oszilloskops, um so die kleinen Unterschiede im Stromverlauf erkennen zu können.



4.3.1 Stall Detection mit dcStep

Wenn für den Betrieb mit dcStep auch die Erkennung eines Motorstillstands (Stall) gewünscht ist, kann dies über den zusätzlichen Algorithmus eingestellt werden.



5 Registertabelle

Die in diesem Leitfaden genannten Register sind im Register Browser abzurufen. Dennoch sind bei einer Vielzahl von über 160 Registern diese nicht immer leicht zu finden. Daher sind die in diesem Leitfaden erwähnten Register hier noch einmal tabellarisch mit ihrem Ort aufgelistet.

Parameter	in Register	Adresse
AMAX	AMAX	38
DC_SG	DCCTRL	24
DC_TIME	DCCTRL	24
en_pwm_mode	GCONF	0
Flags	RAMP_STAT	53
FREEWHEEL	PWMCONF	112
HEND	CHOPCONF	108
HSTART	CHOPCONF	108
IHOLD	IHOLD_IRUN	16
IRUN	IHOLD_IRUN	16
PWM_AMPL	PWMCONF	112
PWM_FREQ	PWMCONF	112
PWM_GRAD	PWMCONF	112
sfilt	COOLCONF	109
sg_stop	sw_mode	52
SGT	COOLCONF	109
TBL	CHOPCONF	108
TCOOLTHRS	TCOOLTHRS	20
TOFF	CHOPCONF	108
VDCMIN	VDCMIN	51
vhighcm	CHOPCONF	108
vhighfs	CHOPCONF	108
VMAX	VMAX	39
VPWMTHRS	TPWMTHRS	19

Autoren



Dipl.-Ing. (FH) Guido Gandolfo hat mehr als 20 Jahre Erfahrung mit dem Ansteuern von Schrittmotoren. Er arbeitet seit 2012 als Produkt Line Manger für Motion Control bei der MEV Elektronik Service in Hilter. Davor hat er über 12 Jahre bei TRINAMIC im Vertrieb gearbeitet. Seit 1996 hat er als Entwickler für Bühnenscheinwerfer Erfahrungen mit Schrittmotoren gesammelt.



B. Sc. Andreas Kinzel beendete sein Mechatronik-Studium mit einer Bachelorarbeit über die Inbetriebnahme eines Schrittmotors am TMC5130 und erstellte dabei die Grundlage für diesen Leitfaden. Nach drei Monaten Training bei Trinamic arbeitet er jetzt bei der MEV Elektronik Service in Hilter als Field Application Engineer für das Fachgebiet Motion Control.

Warenzeichen

spreadCycle™, stealthChop™, stealthChop2™, stallGuard2™, coolStep™, microPlyer™ sind geschützte Warenzeichen der TRINAMIC Motion Control GmbH & Co.KG

Firmenprofil

MEV Elektronik Service GmbH

MEV ist ein Distributor/Stocking Rep. und Herstellerrepräsentant für elektronische Bauelemente, Module und Systeme. Bei der Betreuung der Kunden in Deutschland sowie in Zentral- und Osteuropa steht die optimale technische Unterstützung und Beratung durch 14 Ingenieure im Vordergrund. Dabei konzentriert sich die MEV auf „State Of The Art“ Applikationen in führenden Marktsegmenten. Den Kunden stehen dabei das hauseigene Labor und Demonstrationsräume zur Verfügung, in denen insbesondere in den Bereichen Power Management, Motion Control und Optoelektronik mit Hilfe der FAEs gemeinsam bestmögliche technische Lösungen erarbeitet werden können.

Neben der Design-In-orientierten Distribution fokussiert sich die MEV noch auf Logistik-Konzepte und Dienstleistungen, die exakt auf die jeweiligen Bedürfnisse der Kunden abgestimmt sind. Dabei ist die offene, ehrliche und zuverlässige Partnerschaft die Basis für die erfolgreiche Zusammenarbeit mit Kunden und Lieferanten, sowie innerhalb des Teams.



MEV Elektronik Service GmbH

Nordel 5A

D-49176 Hilter

info@mev-elektronik.com

www.mev-elektronik.com

Firmenprofil

TRINAMIC Motion Control GmbH & Co.KG

TRINAMIC, mit Firmensitz in Hamburg, liefert integrierte Schaltkreise und Module zur Motorsteuerung an Kunden weltweit. Der applikationsgetriebene Ansatz von Trinamic sowie ein tiefgehendes Verständnis der Anwendungen ermöglichen es Lösungen zu bieten, die die Design-Phase vereinfachen und verkürzen, was erhebliche Einsparungen beim Arbeits- und Kostenaufwand im Bereich der Entwicklung und der „total cost of ownership“ ermöglicht.

Die Kunden von TRINAMIC profitieren vom umfangreichen Know-how des Unternehmens im Bereich Motorphysik sowie dem Portfolio an Schutzrechten (IP), die aus jahrelanger Anwendungserfahrung entstanden sind. Die Produktentwicklung bei TRINAMIC fokussiert sich vollständig auf Miniaturisierung, Steigerung der Effizienz sowie Diagnose und Schutzfunktionen, welche die Zuverlässigkeit des Gesamtsystems sicherstellen.

Leitbild von TRINAMIC ist das Bereitstellen energieeffizienter Lösungen. Branchenführende Technologien, wie die patentierten coolStep™ Produkte, bieten neben Bedienerfreundlichkeit und der Präzision von Schrittmotoren zusätzlich vor allem Energieersparnis.

Die TRINAMIC Motion Control Language (TMCL) erleichtert die Entwicklung von Motorsteuerungs-Anwendungen und ermöglicht kürzere Entwicklungszyklen und Produkteinführungszeiten.

Mit seiner mehr als 20-jährigen Geschichte und der traditionell deutschen Eigentümerstruktur mit allen Gesellschaftsanteilen in privater Hand, kann TRINAMIC langfristige Verfügbarkeit gewährleisten.

Die TRINAMIC -Produkte werden über ein weltweites Vertriebsnetz verkauft.



TRINAMIC Motion Control GmbH & Co. KG

Waterloohain 5

D-22769 Hamburg

tmc_info@trinamic.com

www.trinamic.com