



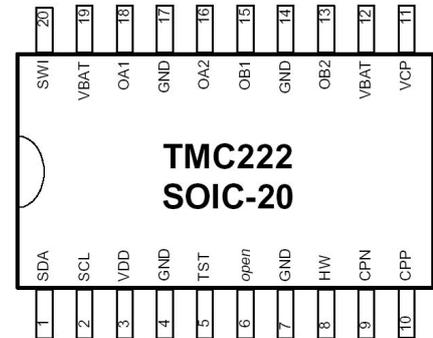
TRINAMIC® Microchips GmbH
Deelbögenkamp 4c
D – 22297 Hamburg
GERMANY

P +49 - (0) 40 - 51 48 06 - 0
F +49 - (0) 40 - 51 48 06 - 60

www.trinamic.com
info@trinamic.com

TMC222 – DATASHEET

Microschritt Schrittmotor Controller / Treiber
mit serieller Schnittstelle



1. Merkmale

Der TMC222 ist ein kombinierter Microschritt Schrittmotor Bewegungscontroller und Treiber mit RAM und OTP Speicher. Der RAM oder OTP Speicher wird verwendet, um Motorparameter und Konfigurationseinstellungen zu speichern. Der TMC222 ermöglicht Microschritte mit einer Auflösung von bis zu vier Bit und einen Spulenstrom von bis zu 800 mA. Nach der Initialisierung führt er automatisch alle zeitkritischen Aufgaben aus, die auf Zielposition und Geschwindigkeitsparametern basieren. Die Kommunikation zu einem Teilnehmer geschieht über eine serielle I²C Schnittstelle mit 2 Leitungen. Zusammen mit einem preisgünstigen Mikrocontroller implementiert der TMC222 ein vollständiges Bewegungssteuersystem.

Der Hauptnutzen des TMC222s sind:

- **Motortreiber**
 - Kontrolliert einen Schrittmotor mit einer Microschrittauflösung von 4Bit
 - Programmierbarer Spulenstrom bis zu 800mA
 - Versorgungsspannungsbereich von 8 ... 29V
 - Pulsweitenmodulierte Stromkontrolle fester Frequenz mit automatischer Selektion des schnellen und langsamen Abfallmodus
 - Vollschrittfrequenzen bis zu 1kHz
 - Diagnose über zu hohe Temperatur, Leitungsunterbrechung, Kurzschluß, Überstrom und Unterspannung
- **Bewegungscontroller**
 - Interner 16Bit breiter Positionszähler
 - Konfigurierbare Geschwindigkeits- und Beschleunigungseinstellungen
 - Eingebauter Rampengenerator für autonome Geschwindigkeits- und Positionskontrolle
 - Änderung der Zielposition während einer Bewegung
 - Referenzschaltereingang kann ausgelesen werden
- **Serielle I²C Schnittstelle**
 - Übertragungsrate bis zu 350 kbps
 - Zugreifbar auf Diagnose- und Statusinformationen sowie der Bewegungsparameter
 - Feldprogrammierbare Teilnehmeradresse (32)

Lebenserhaltungsversicherung

TRINAMIC Mikrochips GmbH ist nicht berechtigt oder garantiert nicht einige seiner Produkte zum Gebrauch in Lebenserhaltungssystemen, ohne die bestimmte schriftliche Zustimmung von TRINAMIC Mikrochips GmbH

Lebenserhaltungssysteme sind Geräte, mit denen beabsichtigt wird, Leben zu unterstützen oder aufrecht zu erhalten und dessen Unterlassen, wenn man es ohne Übereinstimmung mit den Anweisungen benutzt, zu Verletzungen oder ganz zum Tode führen kann.

Block Diagramm

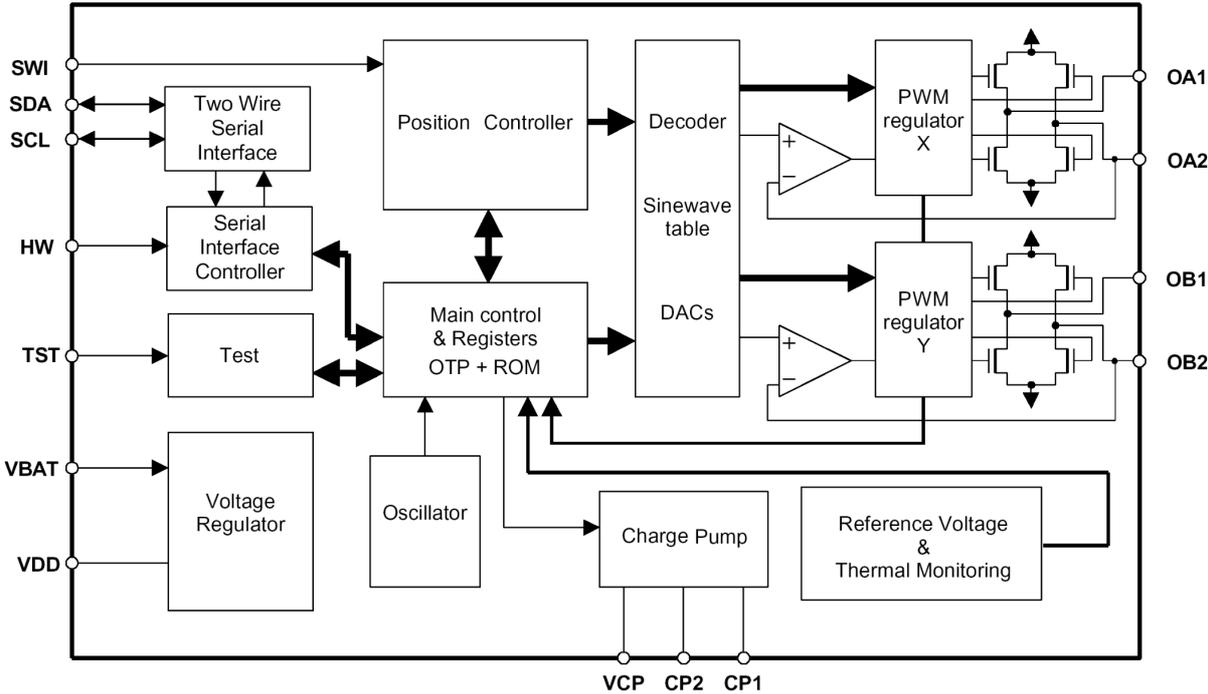


Bild 1: Block Diagramm

2 Allgemeine Beschreibung

2.1 Positioncontroller / Hauptsteuerung

Motorparameter, z.B. Beschleunigung, Geschwindigkeit und Positionsparameter werden an die Hauptsteuerung mittels der seriellen Schnittstelle übergeben. Diese Informationen werden im internen RAM oder OTP Memory gespeichert und stehen dem Positioncontroller zum Abruf bereit. Er übernimmt jeder Zeit die kritische Aufgabe, den Schrittmotor, unter Berücksichtigung der gewünschten Bewegungsparameter, zur gewünschten Position zu fahren.

Die Hauptsteuerung bekommt Rückmeldung vom Schrittmotortreiber und ist in der Lage, im Falle auftretender Probleme zu arrangieren. Diagnoseninformation über Probleme und Fehler werden zur seriellen Schnittstelle weitergeleitet.

2.2 Schrittmotortreiber

Zwei H-Brücken werden verwendet, um beide Windungen von einem bipolaren Schrittmotor zu treiben. Die internen Transistoren ausgelegt für einen Ausgangsstrom von bis zu 800 mA. Das PWM Prinzip wird verwendet, um den Strom durch die Spulen zu begrenzen. Die Regelung führt einen Vergleich zwischen dem wahrgenommenen Ausgangsstrom und der interne Referenz aus. Die PWM Signale, welche die Leistungstransistoren treiben, sind vom Ausgang des Strom-Comperators abgeleitet.

2.2.1 Serielle I2C Schnittstelle

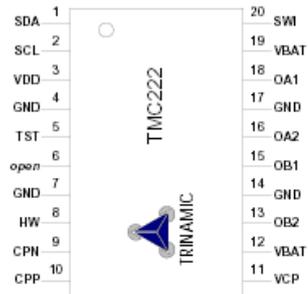
Kommunikation zwischen einem Host und den TMC222 wird über eine zweidratigen bidirektionalen Schnittstelle hergestellt. Bewegungsanweisungen und Diagnoseninformation werden zu oder von der Master geliefert. Es ist möglich, bis zu 32 Teilnehmer auf demselben Bus zu verbinden. Slave Adressen sind programmierbar über den OTP Memory oder einen externen PIN.

2.3 Verschiedenes

Außer der Hauptsteuerung enthält der TMC222 folgendes:

- Eine interne Ladepumpe wird verwendet, um die Nebentransistoren zu treiben.
- Ein interner Oszillator, der bei 4 MHz +/-10% läuft, taktet die LIN Protokoll Routine, die serielle I2C Schnittstelle, die Positioniereinheit und die Hauptsteuerung
- Interne Spannungsreferenz für genaue Referenzen
- Ein 5 Volt-Spannungsregler, der die digitale Logik versorgt
- Schutzmaßnahmen zur thermische Abschaltung, Power-On-Reset usw..

2.4 PIN – Belegung (vorläufig)



Name	SOIC20	QFN32	Beschreibung
SDA	1		SDA Serieller Datenleitungs Eingang / Ausgang
SCL	2		SCK Serielle Taktleitung
VDD	3		Interne Versorgungsspannung (benötigt externen Entkopplungskondensator)
GND	4,7,14,17		Masse, Kühlblech
TST	5		Test PIN (im Normalbetrieb mit Masse verbunden)
Open	6		Muss übrig bleiben
HW	8		Von außen programmierbares Schnittstellen-Adress-Bit (HW)
CPN	9		Negative Verbindung des Pumpenkondensators (Ladungspumpe)
CPP	10		Positive Verbindung des Pumpenkondensators (Ladungspumpe)
VCP	11		Filterkondensator für Ladungspumpe
VBAT	12,19		Spannungsversorgung
OB2	13		Negatives Spulenende Phase B
OB1	15		Positives Spulenende Phase B
OA2	16		Negatives Spulenende Phase A
OA1	18		Positives Spulenende Phase A
SWI	20		Eingang Referenzschalter

Tabelle1: TMC222 (vorläufige) PIN Ausgänge

3. Anwender Umgebung – Konfiguration

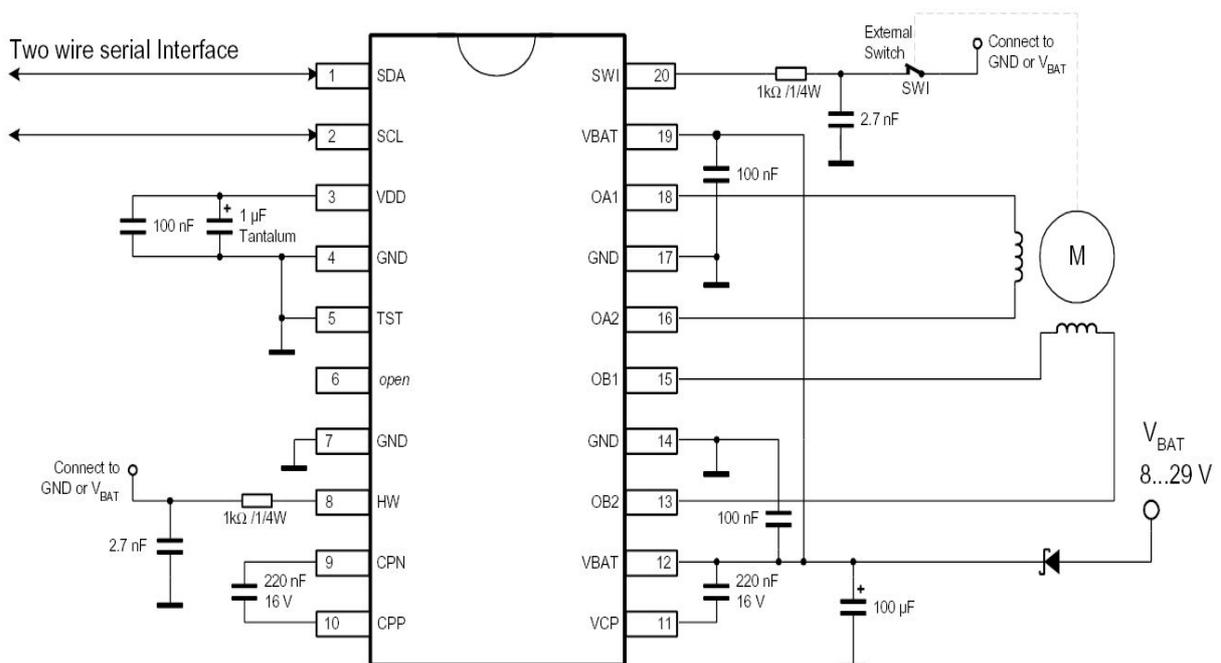


Bild 2: TMC222 Anwender Umgebung

6. funktionelle Beschreibung

6.1 Positionssteuerung und Hauptsteuerung

6.1.1 Schritarten

Das TMC222 unterstützt bis zu 16 Microschritte pro Vollschritt, welche zu glatter und niedriger Welligkeit des Drehmoments der Schrittmotorbewegung führt. Vier Schritarten (Microschrittauflösung) sind durch den Benutzer wählbar: Siehe auch 6.3 Schrittmotortreiber auf Seite 21.

- Halbschrittbetrieb
- 1/4 Microschritt
- 1/8 Microschritt
- 1/16 Microschritt

6.1.2 Geschwindigkeitsrampe

Eine allgemeine Geschwindigkeitsrampe, wo der Motor zu einer gewünschten Position fährt, wird in Abbildung 5 gezeigt: Typisch Geschwindigkeitsrampe auf Seite 10. Die Bewegung besteht aus einer Beschleunigungsphase, einer Phase konstanter Geschwindigkeit und wenigstens eine Verzögerungsphase. Sowohl die Beschleunigung als auch die Verzögerung sind symmetrisch. Der Beschleunigungsfaktor kann von einer Tabelle mit 16 Eintragungen gewählt werden. (Tabelle 6: Acc Parameter auf Seite 13). Eine typische Bewegung beginnt mit einer Anfangsgeschwindigkeit (V_{min}). Während Beschleunigungsphase nimmt die Geschwindigkeit, bis (V_{max}) erreicht ist zu. Nach der Beschleunigungsphase wird die Bewegung mit der Geschwindigkeit von V_{max} fortgesetzt bis die Geschwindigkeit vermindert werden muß, um an der gewünschten Zielposition stehen zu bleiben. Beide Geschwindigkeitsparameter V_{min} und V_{max} sind programmierbar, während V_{min} ein programmierbares Verhältnis von V_{max} ist. (Sehen Sie Tabelle 3: V_{max} Parameter auf Seite 12 und Tabelle 5: V_{min} Parameter auf Seite 13). Der Benutzer hat Rechnung zu tragen, das V_{min} nicht geändert werden darf, während einer Bewegung. V_{max} wird nur erlaubt, sich unter speziellen Umständen zu ändern. (siehe 6.1.4 V_{max} Parameter auf Seite 12). Die Höhe des Stromes durch jede Spule des Schrittmotors, ist von einer Tabelle mit 16 möglichen Werten wählbar. Unterscheiden wird er zwischen dem Laufstrom I_{run} und Haltestrom I_{hold} . Während einer Bewegung werden die Spulen des Schrittmotors mit I_{run} versorgt, wobei I_{hold} der Strom ist, der im Stillstand des Schrittmotors vor oder nach einer Bewegung fließt. Mehr Details über I_{run} und I_{hold} sind in 6.3.2 zu finden. Der Übergang von I_{run} zu I_{hold} auf Seite 23. Die Tabellen mit den aktuellen Einstellungen für I_{run} und I_{hold} befinden sich in Tabelle 14: I_{run} Einstellungen auf Seite 23 und Tabelle 15: I_{hold} Einstellungen auf Seite 24. Mehr Details wie I_{run} auf I_{hold} umschaltet, ist in Kapitel 6.3.2 zu finden. Der Übergang I_{run} zu I_{hold} auf Seite 23. Zugriff auf Geschwindigkeits- bzw. Beschleunigungsparameter kann über die LIN Schnittstelle erfolgen. Diese Parameter können über den **SetMotorParam** Befehl (siehe Seite 33) geschrieben und über den **GetFullStatus1** Befehl (siehe Seite 29) gelesen werden.

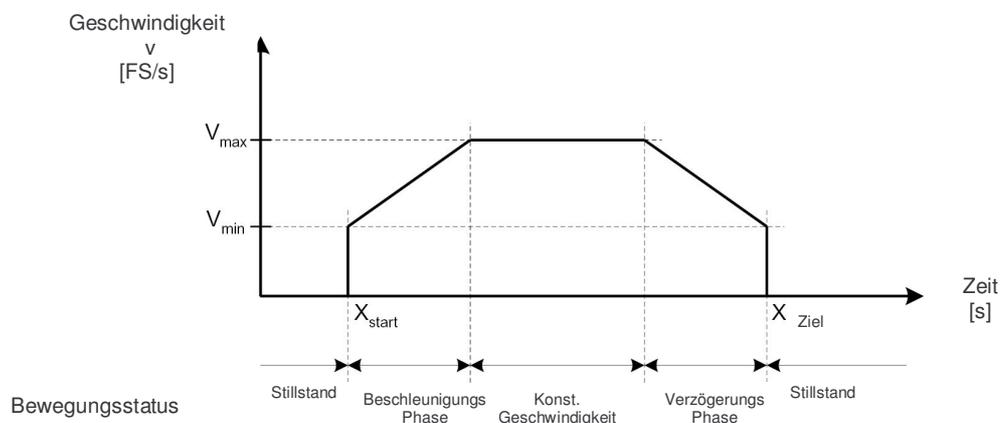


Bild 5: Typische Geschwindigkeitsrampe

Notizen:

- Tolleranz der Widerstände + - 5%
- Kondensator 2,7nF: minimum 2,7nF, maximal 10nF
- Der Kondensator 1uF und 100uF müssen einen niedrigen ESR Wert haben
- Der Kondensator 100nF sollte nah an den Pins von Vss und Vdd sein
- Der Kondensator sollte so nah wie möglich an den Pins CPN, CPP, V_{CP} und V_{BB} sein um die EMC Strahlung zu verringern

4. Bestellinformationen

Bezeichnung	Gehäuse	maximaler Strom	Temperaturbereich
TMC222-SI	SOIC-20	800mA	-40°C..85°C

Tabelle 2: Bestellinformationen

4.1 Innere Wärmebelastung

4.1.1 SOIC -20 Gehäuse

Der Wärmewiderstand des Gehäuses ist 28° C/W. Abgeführt an die Umgebung, über den GND Anschlüssen des IC's beträgt der Wärmewiderstand 63° C/W, wenn die Kupferfläche auf der Leiterplatte folgende Größe hat:

- PCB Dicke = 1,6mm
- Einseitig
- Kupferschicht = 35um

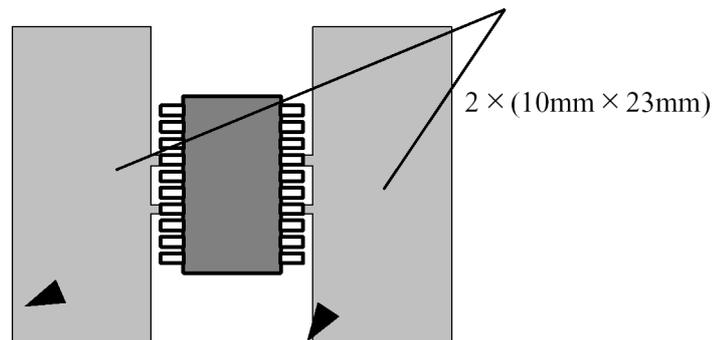


Bild 3: Ansicht des Layouts

5. Ausgangspins

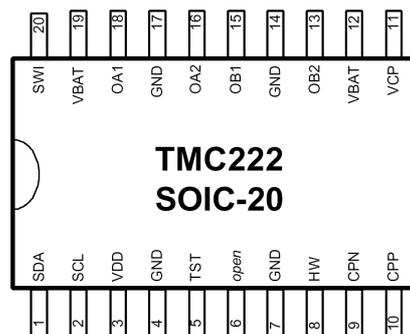


Bild 4: Pinbelegung SOIC-20 Gehäuse

6.1.3 Beispiele für verschiedene Geschwindigkeitsrampen

Die folgenden Abbildungen zeigen einige Beispiele für typische Bewegungen unter verschiedenen Bedingungen an:

6.1.3.1 Bewegung mit Änderung der Zielposition

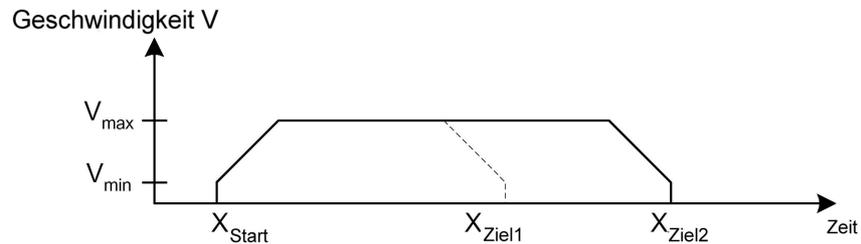


Bild 6: Bewegung mit Änderung der Zielposition

6.1.3.2 Bewegung mit Änderung der Zielposition während der Verzögerungsphase

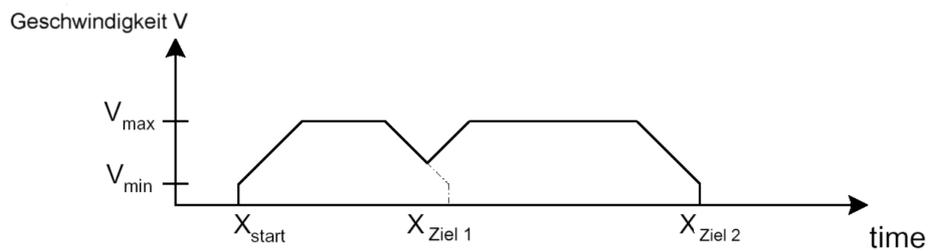


Bild 7: Bewegung mit Änderung der Zielposition während der Verzögerungsphase

6.1.3.3 Kurze Bewegung V_{\max} wird nicht erreicht

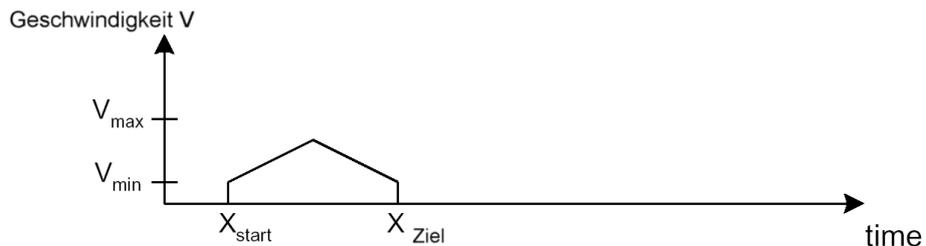


Bild 8: Kurze Bewegung V_{\max} wird nicht erreicht

6.1.3.4 Bewegung mit Änderung der Zielposition in entgegengesetzter Richtung

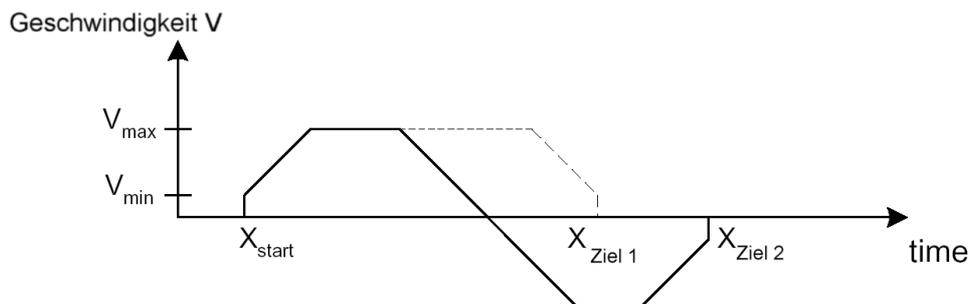


Bild 9: Bewegung mit Änderung der Zielposition in entgegengesetzter Richtung

Der Motor durchfährt den Nullpunkt mit einer linearen Geschwindigkeit. Die Geschwindigkeit kann kleiner als der programmierte Vmin Wert während des Nulldurchgangs sein. Ein linearer Nulldurchgang liefert dem Schrittmotor sehr niedrige Drehmomentwelligkeit während des durchfahrens.

6.1.4 Vmax Parameter

Die gewünschte Geschwindigkeit Vmax kann von der entsprechenden "Tabelle 3: Vmax Parameter" gewählt werden, welche 16 verschiedene Einstellungen für Vmax enthält.

Vmax index	Vmax [FS/s]	Stepping Mode			
		Half-Step Mode [half-steps/s]	1/4 micro stepping [micro-steps/s]	1/8microstepping [micro-steps/s]	1/16 micro stepping [micro-steps/s]
0	99	197	395	790	1579
1	136	273	546	1091	2182
2	167	334	668	1335	2670
3	197	395	790	1579	3159
4	213	425	851	1701	3403
5	228	456	912	1823	3647
6	243	486	973	1945	3891
7	273	546	1091	2182	4364
8	303	607	1213	2426	4852
9	334	668	1335	2670	5341
10	364	729	1457	2914	5829
11	395	790	1579	3159	6317
12	456	912	1823	3647	7294
13	546	1091	2182	4364	8728
14	729	1457	2914	5829	11658
15	973	1945	3891	7782	15564

Tabelle 3: Vmax Parameter

Unter speziellen Umständen ist es möglich, die Vmax Parameter zu ändern, während eine Bewegung läuft. Alle 16 Einträge für die Vmax Parameter sind in vier Gruppen eingeteilt (Tabelle 4: Vmax Gruppen, Seite 12).



Wenn Sie Vmax während einer Bewegung ändern, geben Sie Acht, daß die neuen Vmax Parameter innerhalb derselben Gruppe sind.

Gruppe	A	B	C	D
Vmax Index	0	1, 2, 3, 4, 5, 6	7, 8, 9, 10, 11, 12	13, 14, 15

Tabelle 4: Vmax Gruppen

6.1.5 Vmin Parameter

Dieser Geschwindigkeitsparameter ist ein programmierbares Verhältnis zwischen 1/32 und 15/32 von Vmax. Es ist auch möglich, durch Setzen des Vmin Index auf "0", Vmin auf dieselbe Geschwindigkeit wie Vmax zu setzen.

Die nächste Tabelle "Tabelle 5: Vmin Parameter" zeigt die möglichen Werte.

Vmin index	Vmax factor	Vmax [Full-step/s]															
		106	137	167	198	213	228	243	274	304	334	365	395	456	547	730	973
0	1	106	137	167	198	213	228	243	274	304	334	365	395	456	547	730	973
1	1/32	3	4	5	6	7	7	8	9	10	10	11	12	14	17	23	30
2	2/32	7	9	10	12	13	14	15	17	19	21	23	25	29	34	46	61
3	3/32	10	13	16	19	20	21	23	26	29	31	34	37	43	51	68	91
4	4/32	13	17	21	25	27	29	30	34	38	42	46	49	57	68	91	122
5	5/32	17	21	26	31	33	36	38	43	48	52	57	62	71	85	114	152
6	6/32	20	26	31	37	40	43	46	51	57	63	68	74	86	103	137	182
7	7/32	23	30	37	43	47	50	53	60	67	73	80	86	100	120	160	213
8	8/32	27	34	42	50	53	57	61	69	76	84	91	99	114	137	183	243
9	9/32	30	39	47	56	60	64	68	77	86	94	103	111	128	154	205	274
10	10/32	33	43	52	62	67	71	76	86	95	104	114	123	143	171	228	304
11	11/32	36	47	57	68	73	78	84	94	105	115	125	136	157	188	251	334
12	12/32	40	51	63	74	80	86	91	103	114	125	137	148	171	205	274	365
13	13/32	43	56	68	80	87	93	99	111	124	136	148	160	185	211	297	395
14	14/32	46	60	73	87	93	100	106	120	133	146	160	173	200	239	319	426
15	15/32	50	64	78	93	100	107	114	128	143	157	171	185	214	256	342	456

Tabelle 5: Vmax Parameter

6.1.6 Beschleunigungsparameter

Der Beschleunigungsparameter kann in einem breiten Wertebereich, wie in der nächsten Tabelle beschrieben gewählt werden, "Tabelle 6: Acc Parameter".

 **Bitte beachten Sie, daß der Beschleunigungsparameter, während einer Bewegung, nicht geändert darf.**

Acc index	Acceleration Values in [FS/s ²] dependent on Vmax															
	Vmax [FS/s]															
	99	136	167	197	213	228	243	273	303	334	364	395	456	546	729	973
0	49	49	49	49	49	49	49	106	106	106	106	106	106	473	473	473
1	218	218	218	218	218	218	218	218	218	218	218	218	218	735	735	735
2	1004	1004	1004	1004	1004	1004	1004	1004	1004	1004	1004	1004	1004	1004	1004	1004
3	3609	3609	3609	3609	3609	3609	3609	3609	3609	3609	3609	3609	3609	3609	3609	3609
4	6228	6228	6228	6228	6228	6228	6228	6228	6228	6228	6228	6228	6228	6228	6228	6228
5	8848	8848	8848	8848	8848	8848	8848	8848	8848	8848	8848	8848	8848	8848	8848	8848
6	11409	11409	11409	11409	11409	11409	11409	11409	11409	11409	11409	11409	11409	11409	11409	11409
7	13970	13970	13970	13970	13970	13970	13970	13970	13970	13970	13970	13970	13970	13970	13970	13970
8	16531	16531	16531	16531	16531	16531	16531	16531	16531	16531	16531	16531	16531	16531	16531	16531
9	14785	19092	19092	19092	19092	19092	19092	19092	19092	19092	19092	19092	19092	19092	19092	19092
10	14785	21886	21886	21886	21886	21886	21886	21886	21886	21886	21886	21886	21886	21886	21886	21886
11	14785	24447	24447	24447	24447	24447	24447	24447	24447	24447	24447	24447	24447	24447	24447	24447
12	14785	27008	27008	27008	27008	27008	27008	27008	27008	27008	27008	27008	27008	27008	27008	27008
13	14785	29570	29570	29570	29570	29570	29570	29570	29570	29570	29570	29570	29570	29570	29570	29570
14	14785	29570	29570	29570	29570	29570	29570	34925	34925	34925	34925	34925	34925	34925	34925	34925
15	14785	29570	29570	29570	29570	29570	29570	40047	40047	40047	40047	40047	40047	40047	40047	40047

Tabelle 6: Acc Parameter

Die Anzahl an äquivalenten Vollsritten während einer Beschleunigungsphase kann mit der folgenden Gleichung berechnet werden:

$$N_{step} = \frac{V_{max}^2 - V_{min}^2}{2 \cdot Acc}$$

Gleichung 1

6.1.7 Positionsbereiche

Die Positionsinformation wird durch Verwenden des Zweierkomplementformats codiert. Abhängig vom Schrittmodus (Bild 6.1.1.) ändert sich der Positionsbereich wie in folgender Tabelle aufgeführt:

Schritt-Modus	Positions-Bereich	Schrittzahl für Ganzen Bereich
Half-stepping	-4096...+4095 ($-2^{12} \dots +2^{12}-1$)	8192 half-steps 2^{13}
1/4 micro-stepping	-8192...+8191 ($-2^{13} \dots +2^{13}-1$)	16384 micro-steps 2^{14}
1/8 micro-stepping	-16384...+16383 ($-2^{14} \dots +2^{14}-1$)	32768 micro-steps 2^{15}
1/16 micro-stepping	-32768...+32767 ($-2^{15} \dots +2^{15}-1$)	65536 micro-steps 2^{16}

Tabelle 7: Positionsbereiche

6.1.8 Sichere Position

Der Befehl GotoSecPos fährt den Motor zu einer vorprogrammierten sicheren Position. (Siehe Kapitel 7.1.9.4 GotoSecurePosition, Seite 32). Die sichere Position ist durch den Benutzer programmierbar. Die sichere Position wird mit 11 Bits codiert, weshalb die Auflösung niedriger als bei den normalen Positionierbefehlen ist, wie es in der folgenden Tabelle gezeigt wird.

Schritt-Modus	Sichere Positionsauflösung
Half-stepping	4 half steps
1/4 micro stepping	8 micro steps ($1/4^{th}$)
1/8 micro stepping	16 micro steps ($1/8^{th}$)
1/16 micro stepping	32 micro steps ($1/16^{th}$)

Tabelle 8: Sichere Positoionsauflösung

6.1.9 Externer Schalter

Pin SW1 (20) (zu sehen in Bild 2 auf Seite 8) kontrolliert den Stromfluss des externen Schalters von/zur Quelle. Dies soll überprüfen, ob der externe Schalter geöffnet oder geschlossen ist, bzw. ob der Pin an Masse oder Vbat angeschlossen ist. Der Status des Schalters kann durch Verwenden des Befehls GetFullStatus1 ausgelesen werden. Solange der Schalter geöffnet ist, ist das <ESW> Flag auf "1" gesetzt.

6.1.10 Management der Motorabschaltung

Der TMC222 wird in den Motorabschalt-Modus gebracht, sobald eine der folgenden Bedingungen auftritt:

- Die Schaltkreistemperatur steigt über die thermische Abschaltsschwelle T_{sd} an. Sehen Sie 6.1.12 Temperatur Management auf Seite 17
- Die Batteriespannung fällt unter U_{V2s} ab siehe 6.1.13 Batteriespannungsverwaltung auf Seite 18.
- Ein elektrisches Problem trat auf, z.B. Kurzschluß, offener Schaltkreis, usw. im Falle von solchen Problemen, wird das < EIDef > Flag gesetzt
- Bei Chargepump Ausfall wird das < CPFail > Flag gesetzt

Während Motorabschaltung werden die folgenden Aktionen vom Hauptcontroller ausgeführt:

- H-Brücken werden in hohen Impedanzmodus eingesetzt
- Das Zielpositionsregister TagPos ist mit dem Inhalt des aktuellen Positionsregisters geladen ActPos

Das LIN Interface bzw. die serielle Schnittstelle bleibt während der Motorabschaltung aktiv.

- Bedingungen, die zu einem Motorabschalten führten, sind nicht mehr aktiv
- Ein GetFullStatus Kommando ist über serielle Schnittstelle ausgegeben worden

Folgendes führt zum verlassen des Motorabschaltzustands

- H Brücken in Ihold Modus
- Messen für die Motorsteuerung digitale Schaltkreise sind aktiviert
- Die Ladepumpe ist wieder aktiv

Jetzt ist das TMC222 bereit, jeden Positions- Befehl auszuführen.



Wichtige Notiz:

Zuerst muß der Befehl **GetFullStatus** nach dem Einschalten ausgeführt werden, um den TMC222 zu aktivieren.

6.1.11 Referenzsuche / Positionsinialisierung (Runinit)

Der Schrittmotor liefert keine Information über die aktuelle Position des Motors. Deshalb ist es empfehlenswert, eine Referenzsuche nach dem Einschalten, oder wenn eine Motorabschaltung aufgrund eines Problems auftrat, auszuführen. Der Befehl **Runinit** initialisiert eine Referenzsuche. Dieser Befehl besteht aus dem Vmin, dem Vmax Parameter und der Positionsinformation über das Ende der ersten und der zweiten Bewegung. (siehe Seite 33)

Eine Referenzsuche besteht aus 2 Bewegungen (Bild 10: **Runinit**). Die erste Bewegung fährt den Motor zu einer Blockierposition oder zum Referenzschalter. Diese Bewegung wird unter Einhaltung der gewählten Parameter Vmin, Vmax und Beschleunigung ausgeführt, die im RAM definiert sind. Die zweite Bewegung hat eine rechteckige Form, ohne eine Beschleunigungsphase und fährt den Motor aus der Blockierposition. Die Maximalgeschwindigkeit der zweiten Bewegung entspricht Vmin. Nach dieser Bewegung wird das aktuelle Positionsregister auf „0“ gesetzt.

Sobald der Befehl **Runinit** gestartet wurde, kann er von keinem anderen Befehl unterbrochen werden. Außer es tritt ein Ereignis auf, daß zu einer Motorabschaltung geführt hat oder der Befehl **HardStop** wurde erhalten. Weiterhin muß der Master überprüfen, ob die aktuelle Position des Schrittmotors **nicht** der Zielposition der ersten Bewegung entspricht. Dies ist sehr wichtig, sonst blockiert sich der IC selbst. Befindet sich der IC erst einmal in diesem Zustand, kann dieser nur noch mit dem Befehl **HardStop** gefolgt von dem Befehl **GetFullStatus** verlassen werden.

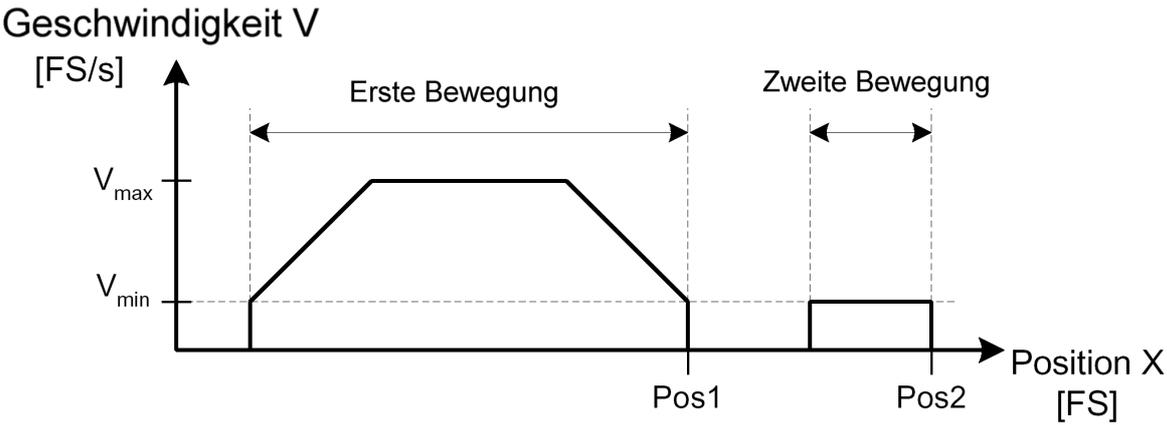


Bild 10: Runinit

6.1.12 Temperaturverwaltung

Der TMC222 hat eine interne Temperaturüberwachung. Der Schaltkreis geht in den Shutdown-Modus wenn die Temperatur T_{sd} übersteigt, weiterhin sind 2 Temperaturschwellwerte implementiert, um eine Temperaturvorwarnung zu erzeugen. Bild 11: Temperaturverwaltung auf Seite 17.

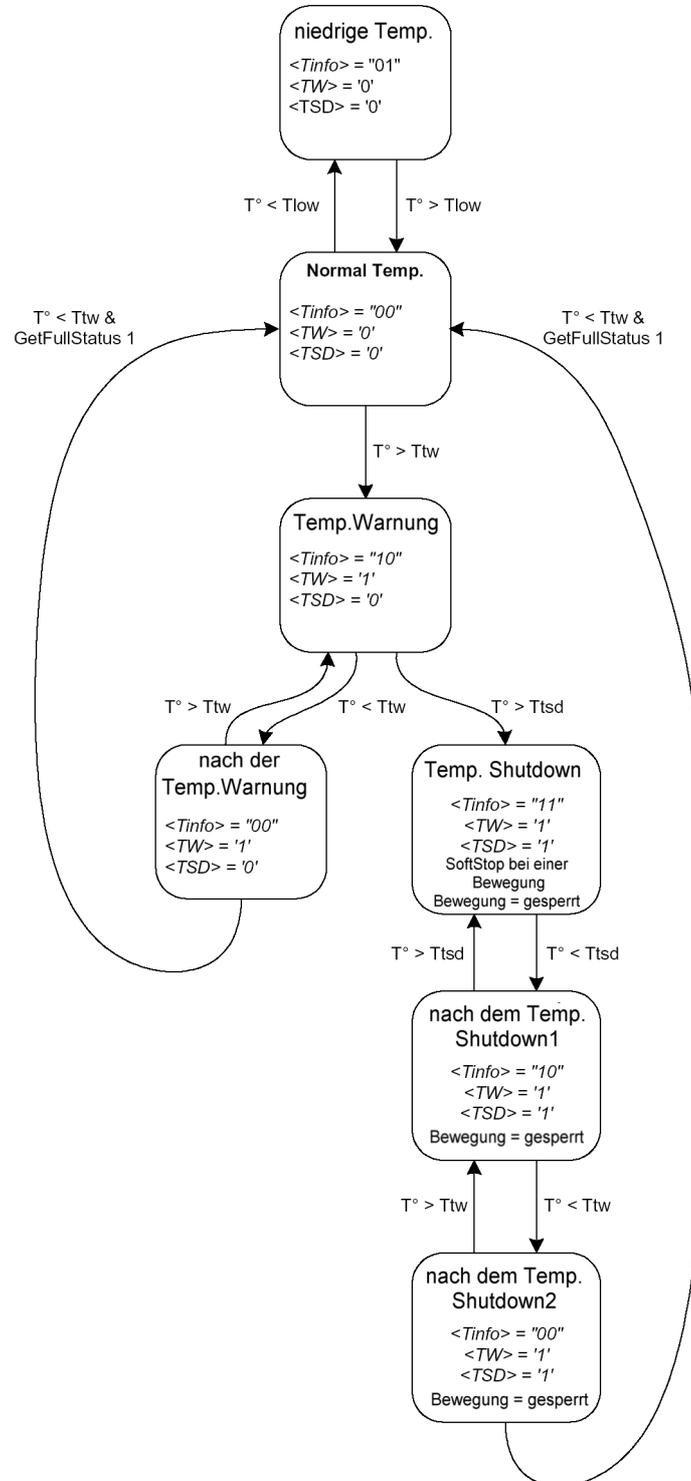


Bild 11: Temperaturverwaltung

6.1.13 Batteriespannungsverwaltung

Der TMC222 hat eine interne Batteriespannungsüberwachung. Der Schaltkreis geht in Shutdown-Modus, wenn die Batteriespannung unterhalb der Schwelle UV2 fällt, eine Schwelle UV1 wird weiterhin implementiert, um eine Niedrigspannungswarnung zu erzeugen. Bild 12: Batterie Spannungsverwaltung auf Seite 18.

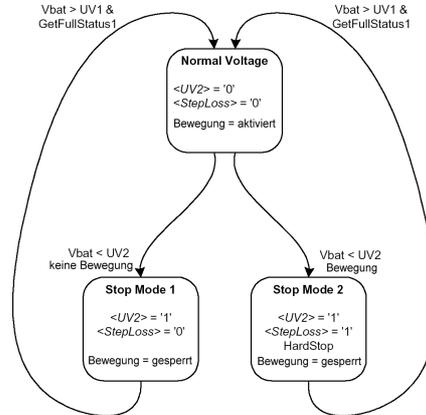


Bild 12: Batterie Spannungsmanagement

6.1.14 Interne Handhabung von Befehlen und Flags

Aufgrund des Schlafmodus unterscheidet sich die interne Handhabung von Befehlen und Flags. Befehle werden mit verschiedenen Prioritäten behandelt, die vom aktuellen Zustand und dem aktuellen Status der internen Flags abhängig sind. Der seriellen Schnittstelle betreffend sehen Sie in Bild 13: Interne Handhabung von Befehlen und Flags auf Seite 18. Befehle SetPosition oder GotoSecurePosition sind ignoriert, solange das <steploss> Flag gesetzt ist.

Notiz:

Der Befehl HardStop wird intern im Falle von einem elektrischen Fehler oder Übertemperatur vom Master gesandt oder ausgelöst.

Eine Beschreibung der verfügbaren Befehle kann in Kapitel 7.1.9 Befehlsbeschreibung auf Seite 29 gefunden werden. Ein Überblick über die internen Flags kann in Tabelle 10 gefunden werden: Status-Flags auf Seite 20. Wenn der Schaltkreis den Motor zu seiner programmierten Zielposition fährt, wird dieser Zustand im Flag "GotoPos" signalisiert. Es gibt drei Möglichkeiten/Befehle, diesen Status zu verlassen: HardStop, SoftStop oder TargetPosition sind möglich. Der Befehl HardStop hat die höchste Priorität bekommen und überschreibt die Alternativen Möglichkeiten. Beendete Bewegung (erreichte Zielposition) hat die geringste Priorität bekommen.

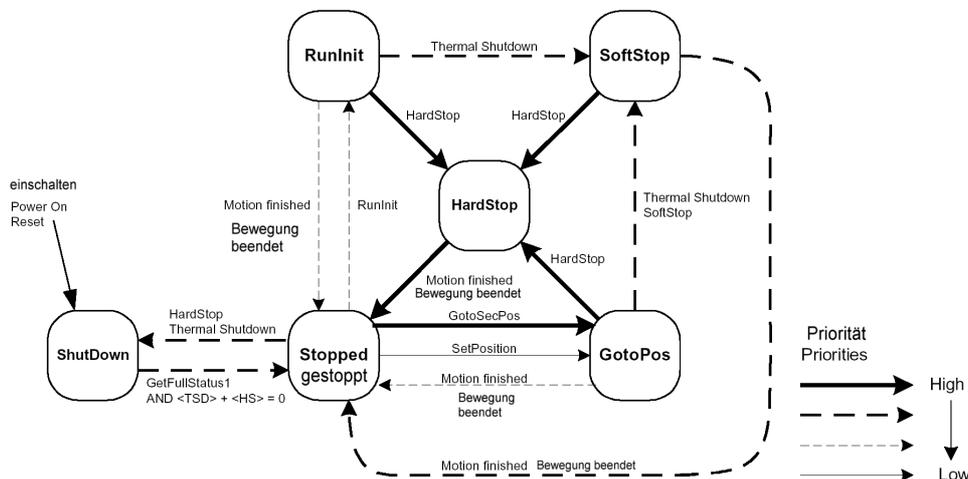


Bild 13: Interne Handhabung von Befehlen und Flags

6.2 RAM und OTP Speicher

6.2.1 RAM Register

Register	Mnemonic	Länge (bit)	verwandte Befehle	Kommentar	Reset State
Actual Position Aktuelle Position	ActPos	16	GetFullStatus 2 ResetPosition	Aktuelle Position des Schrittmotors 16Bit Auflösung	0x0000
Target Position Ziel Position	TagPos	16	SetPosition GetFullStatus 2 ResetPosition	Endposition des Schrittmotors 16Bit Auflösung	
Acceleration Shape Beschleunigungs Form	AccShape	1	GetFullStatus 1 SetMotorParam ResetToDefault	0 = Beschleunigung mit Beschleunigungsparameter 1 = Geschwindigkeit mit Vmin ohne Beschleunigung	
Coil Peak Current max. Spulenstrom	Irun	4	GetFullStatus 1 SetMotorParam ResetToDefault	Spulenstrom während einer Bewegung (Tabelle 14: Irun Einstellungen)	OTP Memory
Coil Hold Current Spulenstrom bei Stillstand	Ihold	4	GetFullStatus 1 SetMotorParam ResetToDefault	Spulenstrom während der Motor steht (Tabelle 15: IHold Einstellungen)	
Minimum Velocity minimale Geschwindigkeit	Vmin	4	GetFullStatus 1 SetMotorParam ResetToDefault	Startgeschwindigkeit des Schrittmotors (Tabelle 5: Vmin Parameter)	
Maximum Velocity maximale Geschwindigkeit	Vmax	4	GetFullStatus 1 SetMotorParam ResetToDefault	Endgeschwindigkeit des Schrittmotors (Tabelle 3: Vmax Parameter)	
Shaft Welligkeit/Stil	Shaft	1	GetFullStatus 1 SetMotorParam ResetToDefault	Bewegungsrichtung	
Acceleration / Deceleration <small>Beschleunigung/ Verzögerung</small>	Acc	4	GetFullStatus 1 SetMotorParam ResetToDefault	Parameter für Beschleunigung (Tabelle 6: Acc Parameter)	
Secure Position sichere Position	SecPos	11	GetFullStatus 2 ResetToDefault	Endposition für GotoSecurePosition Befehl (7.1.9.4 GotoSecurePosition)	
Stepping Mode Schrittmodus	StepMode	2	GetFullStatus 1 GetFullStatus 2 ResetToDefault	Microschritt Modus (6.1.1 Schrittmodus)	

Tabelle 9: RAM Register

Status Flags

Tabelle 10:

Status Flag zeigt die Flags an, auf die durch die serielle Schnittstelle zugegriffen werden kann, um Information über den internen Status des TMC222s zu erhalten.

Flag	Mnemonic	Länge (bit)	an diesem Befehl gebunden	Kommentar	Reset Status
Digital supply Reset	VddReset	1	GetFullStatus1	Ist nach den Einschalten oder nach einem Spannungstief auf "1" gesetzt. Warnt damit, daß der Inhalt des RAM's verlorengegangen sein kann. Es wird nach einem GetFullStatus1 Befehl auf "0" gesetzt.	'1'
Stromüberschreitung in Phase1	OVC1	1	GetFullStatus1	Ist auf "1" gesetzt, wenn eine Stromüberschreitung in Phase1 wahrgenommen wurde. Es wird nach einem GetFullStatus Befehl1 auf "0" gesetzt.	'0'
Stromüberschreitung in Phase2	OVC2	1	GetFullStatus1	Ist auf "1" gesetzt, wenn eine Stromüberschreitung in Phase2 wahrgenommen wurde. Es wird nach einem GetFullStatus Befehl auf "0" gesetzt.	'0'
Schrittverlust	StepLoss	1	GetFullStatus1	Ist auf "1" gesetzt, wenn eine Spannungs-, Strom- oder Temperaturüberschreitung aufgetreten ist. Es wird nach einem GetFullStatus Befehl auf "0" gesetzt. Die Befehle SetPosition und GotoSecurePos werden ignoriert, wenn <StepLoss> = 1 ist	'0'
Secure position aktiviert (nur LIN)	SecEn	1	intern verwendet	Ist "0" wenn SecPos = "100 0000 0000" ist Ansonsten ist es "1"	n.a.
Schaltung im Schlafmodus (nur LIN)	Sleep	1	intern verwendet	"1" = Schlaf Modus Wird von einem LIN Befehl gelöscht	'0'
elektrischer Defekt	EIDef	1	GetFullStatus1	Ist auf "1" gesetzt, wenn ein Leerlauf oder ein Kurzschluß aufgetreten ist (OVC1 oder OVC2). Wird nach einem GetFullStatus Befehl gelöscht.	'0'
Temperatur Information	Tinfo	2	GetFullStatus1	Zeigt die Chiptemperatur an: "0 0" = Normale Temperatur "0 1" = Warnung zu niedrige Temperatur "1 0" = Warnung zu hohe Temperatur "1 1" = Motor wurde abgeschalten	"00"
Temperatur Warnung	TW	1	GetFullStatus1	Wird auf "1" gesetzt, wenn die Temperatur über 145°C steigt. Wird nach einem GetFullStatus Befehl gelöscht.	'0'
Temperatur Abschaltung	TSD	1	GetFullStatus1	Wird auf "1" gesetzt, wenn die Temperatur über 155°C steigt. Wird nach einem GetFullStatus Befehl gelöscht.	'0'
Status einer Bewegung	Motion	3	GetFullStatus1	Zeigt das aktuelle Verhalten des Positioncontrollers an "000" : Aktuelle Position = Endposition; Geschwindigkeit = 0 "001" : Positive Beschleunigung; Geschwindigkeit > 0 "010" : Negative Beschleunigung; Geschwindigkeit > 0 "011" : Beschleunigung = 0 ; Geschwindigkeit = max. positive Geschwindigkeit "100" : Aktuelle Position <=> Endposition; Geschwindigkeit = 0 "101" : Positive Beschleunigung; Geschwindigkeit < 0 "110" : Negative Beschleunigung; Geschwindigkeit < 0 "111" : Beschleunigung = 0 ; Geschwindigkeit = max. negative Geschwindigkeit	"000"
Status des externen Schalters	ESW	1	GetFullStatus1	Zeigt den Status des externen Schalters an: "0" = geöffnet "1" = geschlossen	'0'
Ladepumpen Fehler	CPFail	1	GetFullStatus1	"0" = Ladungspumpe ist Okay "1" = Fehler in der Ladungspumpe	'0'
Fehler in der Elektrik	HS	1		<CPFail> oder <UV2> oder <EIDef>	'0'

Tabelle 10: Status Flags

6.2.2 OTP Speicherstruktur

Die Tabelle unten zeigt, wo sich die im OTP Speicher zu speichernden Parameter befinden.

Notiz: Wenn der OTP Speicher nicht programmiert worden ist, oder wenn das RAM nicht vom Befehl SetMotorParam programmiert werden konnte oder wenn <VddReset> = '1' ist, wird jeder Befehl ignoriert, um aufgrund unnützen RAM Inhalts die Folgen zu vermeiden.

Bitte überprüfen Sie, ob die richtige Versorgungsspannung für den Schaltkreis anliegt, bevor Sie das OTP "überschreiben". (Siehe Tabelle 39: Gleichstromparameter und Spannungsregler auf Seite 38) sonst wird der Schaltkreis zerstört.

OTP Address	OTP Bit Order							
	7	6	5	4	3	2	1	0
0x00	OSC3	OSC2	OSC1	OSC0	IREF3	IREF2	IREF1	IREF0
0x01	EnableLIN	TSD2	TSD1	TSD0	BG3	BG2	BG1	BG0
0x02	ADM	N.A.	N.A.	N.A.	AD3	AD2	AD1	AD0
0x03	Irun3	Irun2	Irun1	Irun0	Ihold3	Ihold2	Ihold1	Ihold0
0x04	Vmax3	Vmax2	Vmax1	Vmax0	Vmin3	Vmin2	Vmin1	Vmin0
0x05	N.A.	StepMode1	StepMode0	Shaft	Acc3	Acc2	Acc1	Acc0
0x06	SecPos7	SecPos6	SecPos5	SecPos4	SecPos3	SecPos2	SecPos1	SecPos0
0x07	N.A.	N.A.	N.A.	SecPos10	SecPos9	SecPos8	LOCKBT	LOCKBG

Tabelle 11: OTP Speicher Aufbau

Die unter der Adresse 0x00 und 0x01 gespeicherten Parameter und das LOCKBT Bit sind schon im OTP Speicher ab Werk programmiert. Sie entsprechen den Schalteinstellungen und werden hier genauso als ein Hinweis dokumentiert. Jedes OTP Bit ist "0" wenn es nicht überschrieben wurde. Ein gesetztes Bit entspricht „1“. Auf diese Weise müssen nur Bits, die eine '1' haben sollen, überschrieben werden. Das Überschreiben eines Bits das schon "1" ist, wird nicht möglich sein, um jeden Schaden der Zener Diode zu vermeiden. Es ist wichtig, zu beachten, daß nur ein einzelnes OTP Byte zu derselben Zeit programmiert werden kann. (Siehe Befehl SetOTPparam). Sobald die OTP Programmierung beendet ist, kann Bit LOCKBG überschrieben werden, um weitere Überschreibungen zu sperren. Ansonsten kann jedes OTP Bit was eine "0" hat weiterhin überschrieben werden.

Lock bit	Protected byte
LOCKBT (zapped before delivery)	0x00 to 0x01
LOCKBG	0x02 to 0x07

Tabelle 12: OTP Lock Bits

Der Befehl, der verwendet wird, um die Anwendungsparameter über den LIN Bus in das RAM vor einem OTP Speicher Programmierung zu laden, ist SetMotorParam.

Dies erlaubt eine funktionelle Prüfung bevor der Befehl SetOTPparam, der zum programmieren und löschen eines einzelnen OTP Speicherbytes benutzt wird. Der GetOTPparam Befehl, der nach jedem SetOTPparam Befehl ausgegeben wird, erlaubt eine Prüfung der korrekten Byte Überschreibung.

6.3 Schrittmotortreiber

Der StepMode Parameter im Befehl SetMotorParam (7.1.9.9 SetMotorParameter auf Seite 33) wird verwendet, um zwischen verschiedenen Schrittmodi auszuwählen. Folgende Modi sind verfügbar:

StepMode parameter	Mode
00	Half Stepping
01	1/4 µStepping
10	1/8 µStepping
11	1/16 µStepping

Tabelle 13: Schrittmodi

6.3.1 Form des Spulenstromes

Die nächsten vier Abbildungen zeigen die Form des Stroms, der durch jede Spule des Motors in den verschiedenen Betriebsmodi fließt.

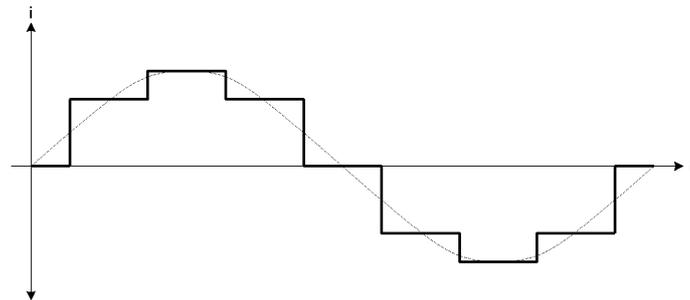


Bild 14: Spulenstrom für Halbschritt Modus

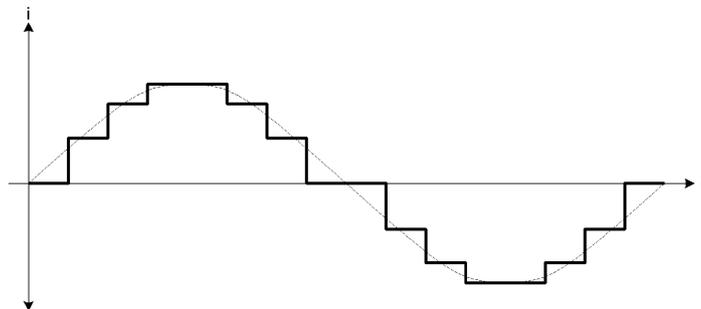


Bild 15: Spulenstrom für 1/4 Schrittmodus

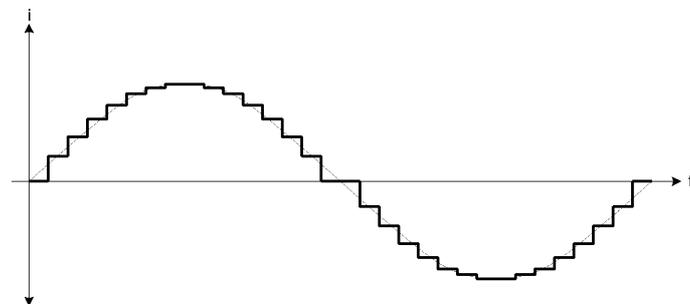


Bild 17: Spulenstrom für 1/8 Schrittmodus

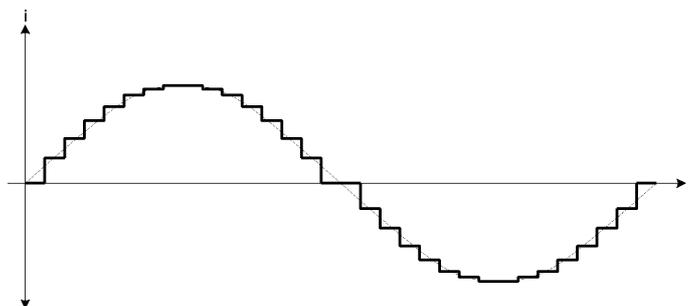


Bild 17: Spulenstrom für 1/16 Schrittmodus

6.3.2 Übergang von Irun zu Ihold

Am Ende einer Motor Bewegung werden die aktuellen Spulenströme "**Irun**" mit minimaler Geschwindigkeit für eine viertel Periode in den Spulen auf ihren aktuellen Gleichstrom gehalten. (zwei Halbschritte) Im Anschluß daran werden die Ströme dann auf ihren Hold Wert "**Ihold**" gehalten. Bild 18 zeigt den Übergang von **Irun** zu **Ihold**.

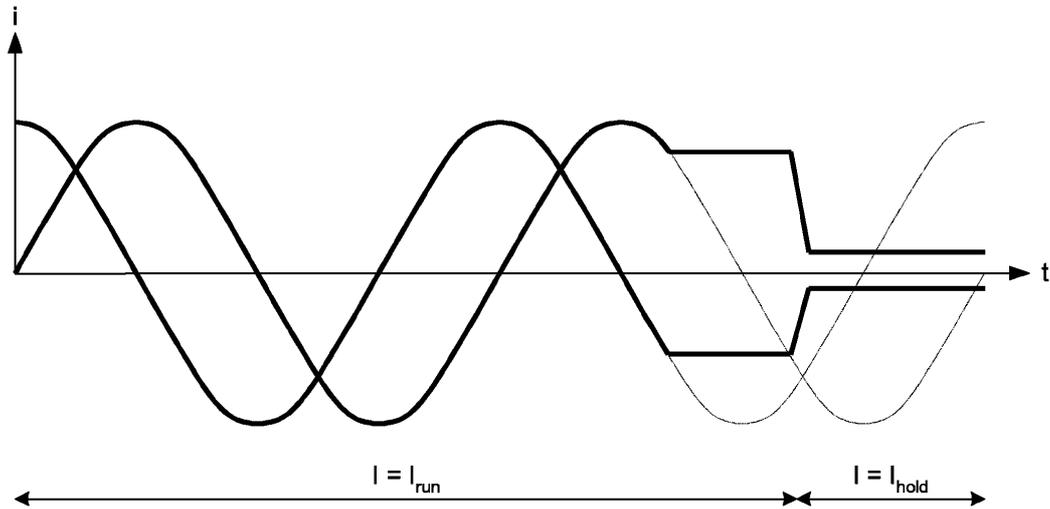


Bild 18: Übergang von Irun zu Ihold

Beide Ströme **Irun** und **Ihold** sind über das entsprechende serielle Datagramm parametrisierbar. Es sind 16 Stromwerte für Irun und Ihold verfügbar. Die nächsten Tabellen (Tabelle 14 und Tabelle 15) zeigen die entsprechenden Stromwerte.

Hexadecimal	Irun				Peak Current [mA]
	Binary				
0	0	0	0	0	59
1	0	0	0	1	71
2	0	0	1	0	84
3	0	0	1	1	100
4	0	1	0	0	119
5	0	1	0	1	141
6	0	1	1	0	168
7	0	1	1	1	200
8	1	0	0	0	238
9	1	0	0	1	283
A	1	0	1	0	336
B	1	0	1	1	400
C	1	1	0	0	476
D	1	1	0	1	566
E	1	1	1	0	673
F	1	1	1	1	800

Tabelle 14: Irun Einstellungen

Hexadecimal	Ihold				Peak Current [mA]
	Binary				
0	0	0	0	0	59
1	0	0	0	1	71
2	0	0	1	0	84
3	0	0	1	1	100
4	0	1	0	0	119
5	0	1	0	1	141
6	0	1	1	0	168
7	0	1	1	1	200
8	1	0	0	0	238
9	1	0	0	1	283
A	1	0	1	0	336
B	1	0	1	1	400
C	1	1	0	0	476
D	1	1	0	1	566
E	1	1	1	0	673
F	1	1	1	1	800

Tabelle 15: Ihold Einstellungen

6.3.3 Chopper Mechanismus

Die Chopper Frequenz ist angelegt an die Frequenz welche in Kapitel 8.5: Wechselstromparameter auf Seite 39 angegeben ist. Der TMC222 verwendet einen intelligenten Chopperalgorithmus, um einen reibungslosen Betrieb niedriger Resonanz zu schaffen. Der TMC222 verwendet interne Messungen, um den Stromfluß in den Spulen abzuleiten. Wenn der Strom weniger als der gewünschte Strom ist, schaltet das TMC222 eine H Brücke so, daß der Strom zunimmt. Andernfalls wird die H Brücke so umgeschaltet, daß der Strom sinkt, wenn der Strom zu hoch ist. Für das Vermindern sind die Modi Langsamer Stromabfall und Schneller Stromabfall verfügbar, während der Modus Schneller Stromabfall den Strom schneller verringert als der Modus Langsamer Stromabfall. Bild 19: Verschiedene Chopperzyklen mit schnellem und langsamem Stromabfall zeigen auf Seite 24 das Chopperverhalten.

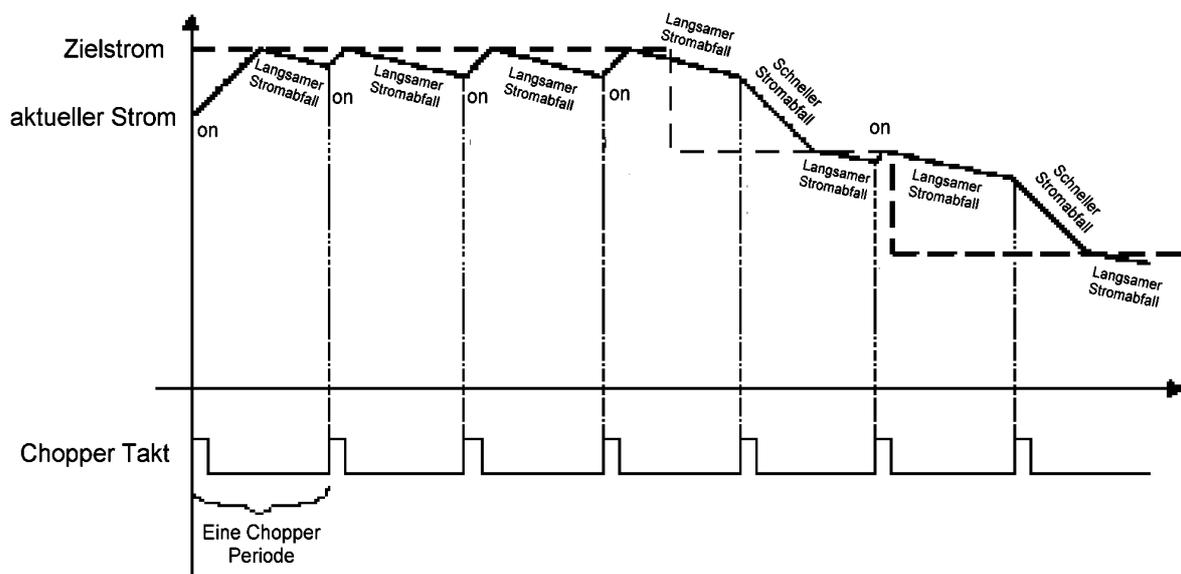


Bild 19: Unterschiedliche Chopperzyklen mit schnellem und langsamem Stromabfall

7. Die Serielle Schnittstelle (I²C Bus)

7.1.1 Physische Lage

Die SDA als auch die SCL Leitungen sind über einem Pull-up Widerstand an die positive Versorgungsspannung angeschlossen (Abbildung 20: Physische Lage). Wenn es keine Kommunikation auf dem Bus gibt, sind beide Leitungen High. Analoge Filter sind implementiert, um Spitzen mit einer Länge von bis zu 50ns zu unterdrücken.

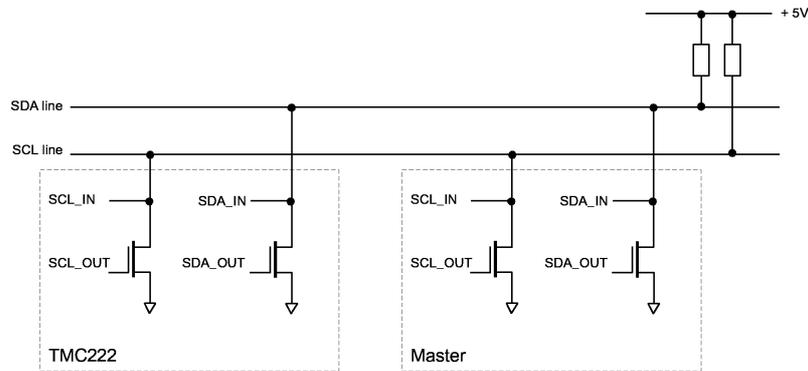
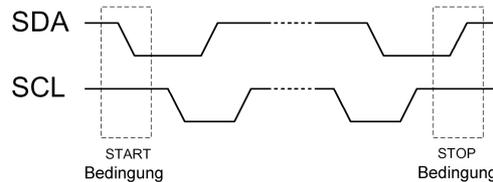


Abbildung 20: Physische Lage

7.1.2 Kommunikation über die serielle I²C Schnittstelle

Jede Datenübertragung beginnt mit einer Startbedingung und endet mit einer Stoppbedingung. Beide Bedingungen sind eindeutig und können nicht verwechselt werden mit den Daten. Ein Wechsel von High nach Low (fallende Flanke) auf der SDA Leitung während die SCL Leitung High ist, zeigt eine Startbedingung an. Ein Wechsel von Low zu High (steigende Flanke) auf der SDA Leitung während die SCL Leitung High ist, definiert eine Stoppbedingung. (Abbildung 21: Start-Stopp-Bedingungen)



(Abbildung 21: Start-Stopp-Bedingungen)

Der SCL Takt wird immer vom Master generiert. Bei jeder steigenden Flanke der SCL Leitung sind die Daten über SDA zulässig. Daten auf der SDA Leitung können sich nur ändern, solange SCL Low ist. (Abbildung 22: Bit Übertragung über die serielle I²C Schnittstelle)

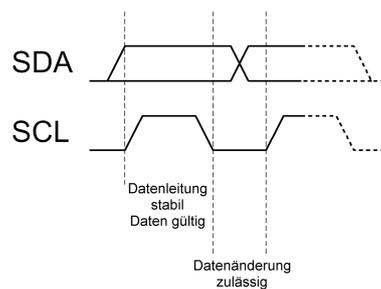
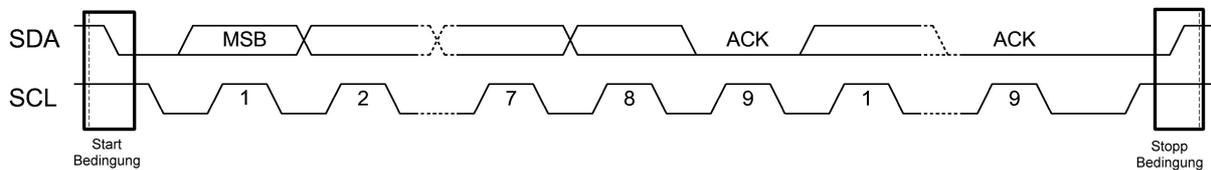


Abbildung 22: Bit Übertragung über die serielle I²C Schnittstelle

Jedes auf die SDA Leitung gelegte Byte muß eine Länge von 8 Bits haben, wo das höherwertige Bit (MSB) zuerst übertragen wird. Die Anzahl von Bytes, die an das TCM222 gesendet werden können, ist auf 8 Byte eingeschränkt. Jedes Byte wird gefolgt von einem Acknowledge Bit, welches vom Empfänger (Receiver) ausgegeben wird. (Abbildung 23: Daten Übertragung über die serielle I²C Schnittstelle)



(Abbildung 23: Daten Übertragung über die serielle I²C Schnittstelle)

7.1.3 Physische Adresse des Schaltkreises

Dem Schaltkreis muß eine physische Adresse gegeben werden, um diesen Schaltkreis von anderen auf dem seriellen Bus (Pin SDA und SCK) zu unterscheiden. Diese Adresse ist auf 7 Bits codiert (2 Bits sind hardwaremäßig immer '1'), wodurch theoretisch die Möglichkeit auf 32 verschiedene Schaltkreise auf demselben Bus anzusprechen besteht. Es ist eine Kombination von 4 OTP Speicherbits und dem einen hardwareseitigem Adreßbit (pin HW[2]). Bild 24: Die physische Adresse der seriellen Schnittstelle HW [2] muß entweder an Masse oder an +5 V angeschlossen sein. Wenn HW [2] nicht angeschlossen ist und die übrigen freigegeben sind, ist richtige Funktionalität des seriellen Schnittstellenteils nicht garantiert.

Das TCM222 unterstützt eine "allgemeine Aufruf" Adresse. Deshalb ist der Schaltkreis als physisches Slave adressierbar oder mit der Adresse "000 0000".

AD6	AD5	AD4	AD3	AD2	AD1	AD0	Physische Adresse
'1'	'1'	OTP_AD3	OTP_AD2	OTP_AD1	OTP_AD0		OTP Speicher
						HW2	Hardwired Bit (mit Masse oder +Vbat verbunden)

Bild 24: Physische Adresse der seriellen Schnittstelle

7.1.4 Daten zum TMC222 schreiben

Eine vollständige Datenübertragung besteht aus folgendes: eine Startbedingung, die Slave Adresse (7 Bit), ein Schreib/Lese-Bit ('0' = schreiben, '1' = lesen) und einem Acknowledge Bit. Jedes weitere Datenbyte wird von einem Acknowledge Bit gefolgt. Das Acknowledge Bit wird verwendet, um den Sender den korrekten Empfang des Datenbytes zu signalisieren. In diesem Fall zieht der TMC222 die SDA Leitung, auf Null. (Bild 25: Die serielle I²C Schnittstelle schreibt Daten zum Slave)

Der TMC222 liest die eingehenden Daten an SDA mit jeder ansteigenden Flanke auf der SCL Leitung. Um den Transfer zu beenden, muß der Master eine Stoppbedingung senden. Einige Befehle des TMC222 unterstützen acht Byte Daten, andere Befehle senden zwei Byte Daten.

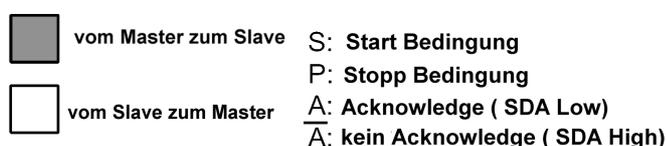
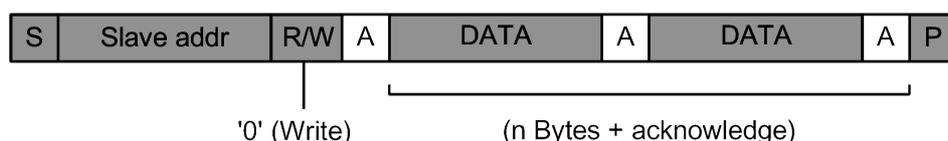


Bild 25: Die serielle I²C Schnittstelle schreibt Daten zum Slave

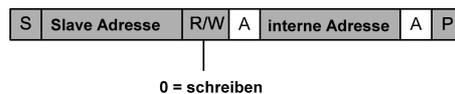
7.1.5 Daten aus TMC222 lesen

Wenn sie Daten aus einem Slave lesen, sind zwei Datenübertragungen erforderlich. Die erste Datenübertragung besteht aus zwei Byte Daten. Das Erst Byte besteht aus der Slave Adresse und dem Schreib Bit. Das Zweite Byte besteht aus der Adresse eines internen Registers des TMC222. (Bild 26: Daten vom Slave über die serielle Schnittstelle lesen)

Die interne Registeradresse ist in dem RAM des TMC222 gespeichert.

Die zweite Datenübertragung besteht aus der Slave Adresse und dem Lese Bit. Dann kann der Master die Datenbits auf der SDA Leitung mit jeder ansteigenden Flanke auf der SCL Leitung lesen. Nach jedem Byte Daten, muß der Master den Datenempfang durch herunterziehen der SDA Leitung auf Null, bestätigen. Das letzte Byte wird vom Master nicht bestätigt, so daß der Slave das Ende des Transfers erkennt. (Bild 26: Daten vom Slave über die serielle Schnittstelle lesen)

Lage der internen Adresse des Slave's



Daten lesen vom Slave

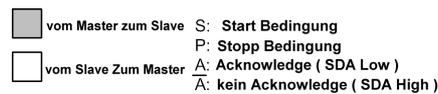


Bild 26: Daten vom Slave über die serielle Schnittstelle lesen

7.1.6 Timing Anforderungen der seriellen Schnittstelle

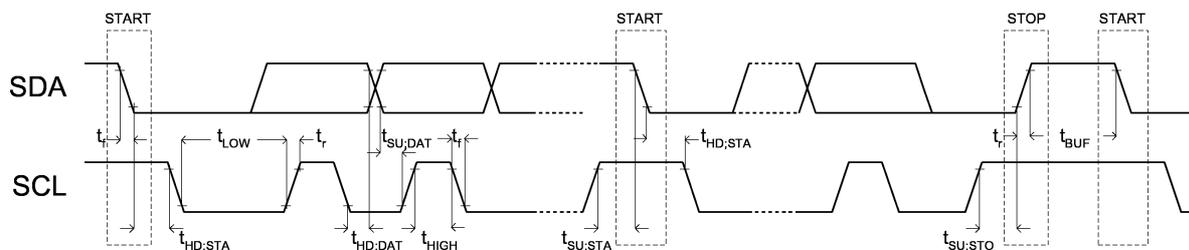


Bild 27: Definition des Timings

Parameter	Symbol	SCL Clk Frequenz <= 100KHz		SCL Clk Frequenz <= 350KHz		Unit
		Min.	Max.	Min.	Max.	
Low Level Eingangsspannung Fixed input levels	V_{IL}	-0.5 ⁽¹⁾	1.5	-0.5 ⁽¹⁾	0.3V _{DD}	V
High Level Eingangsspannung Fixed input levels	V_{IH}	3.0	⁽²⁾	0.7V _{DD}	⁽²⁾	V
Länge der vom Eingangsfilter unterdrückten Impulsspitzen	t_{SP}	n/a	n/a	50	50	Ns
Eingangskapazität der I/O Pins	C_i	-	10	-	10	pF

Tabelle 16: Serielle Schnittstelle – Charakter der SDA und SCI I/O Stufe

Notiz:

(1): Wenn Eingangsspannung $\leq -0,3$ V, dann muß ein 20 ... 100 Ohm Widerstand parallel dazu

(2): Maximum $V_{IH} = V_{DDmax} + 0,5V$

n/a: nicht anwendbar

Parameter	Symbol	SCL Clk frequency <= 100KHz		SCL Clk frequency <= 350KHz		Unit
		Min.	Max.	Min.	Max.	
SCL clock frequency	f _{SCL}	0	100	0	350	KHz
Hold time (repeated) START condition. After this period, the first clock pulse is generated.	t _{HD,STA}	4.0	-	0.6	-	µs
LOW period of the SCL clock	t _{LOW}	4.7	-	1.3	-	µs
HIGH period of the SCL clock	t _{HIGH}	4.0	-	0.6	-	µs
Set-up time for a repeated START condition	t _{SU,STA}	4.7	-	0.6	-	µs
Data set-up time	t _{SU,DAT}	250	-	100	-	ns
Rise time of both SDA and SCL signals	t _r	-	1000	20+0.1C _b ⁽¹⁾	300	ns
Fall time of both SDA and SCL signals	t _f	-	300	20+0.1C _b ⁽¹⁾	300	ns
Set-up time for STOP condition	t _{SU,STO}	4.0	-	0.6	-	µs
Bus free time between a STOP and START condition	t _{BUF}	4.7	-	1.3	-	µs
Capacitive load for each bus line	C _b	0	400	-	400	pF
Noise margin at the LOW level for each connected device (including hysteresis)	V _{nL}	0.1V _{DD}	-	0.1V _{DD}	-	V
Noise margin at the HIGH level for each connected device (including hysteresis)	V _{nH}	0.2V _{DD}	-	0.2V _{DD}	-	V

Tabelle 17: Serielle Schnittstelle – Charakter der SDA und SCL Busleitung

7.1.7 Beschreibung von Anwendungsbefehlen

Die Kommunikation zwischen dem TMC222 und einem I2C Schnittstellenmaster findet über eine große Anzahl von Befehlen statt.

Lesebefehle werden gebraucht bei:

- Ausgabe der aktuellen Statusinformationen wie z.B. Error Flags
- Ausgabe der aktuellen Position des Schrittmotors
- Vergleich auf richtige Programmierung und Konfiguration des TMC222

Schreibbefehle werden gebraucht bei:

- Programmierung des OTP Speichers
- Konfiguration des TMC222 mit Bewegungsparametern (z.B. max/min Geschwindigkeit, Beschleunigung, Schrittmodus, u.s.w.)
- Lieferung der Zielposition für den Schrittmotor

7.1.8 Befehlsübersicht

Befehls Mnemonic	Function	Befehls-Byte	
		Binär	Hexadecimal
GetFullStatus1	ausgeben des kompletten Status des Chips	"1000 0001"	0x81
GetFullStatus2	ausgeben der aktuellen, der Ziel und der sicheren Pos.	"1111 1100"	0xFC
GetOTPParam	OTP Parameter ausgeben	"1000 0010"	0x82
GotoSecurePosition	fährt Motor zur sicheren Position	"1000 0100"	0x84
HardStop	unverzögerlicher Stop	"1000 0101"	0x85
ResetPosition	setzt die aktuelle Position zu Null	"1000 0110"	0x86
ResetToDefault	überschreibt den RAM mit den Inhalt des OTP	"1000 0111"	0x87
RunInit	Referenzsuche	"1000 1000"	0x88
SetMotorParam	setzt die Motorparameter	"1000 1001"	0x89
SetOTP	überschreibt den OTP Speicher	"1001 0000"	0x90
SetPosition	zum program. der Zielposition und der sicheren Pos.	"1000 1011"	0x8B
SoftStop	Motor stoppt mit Verzögerung	"1000 1111"	0x8F

Tabelle 18: Serielle Schnittstelle – Befehlsübersicht

7.1.9 Befehlsbeschreibung

7.1.9.1 GetFullStatus1 (\$81hex)

Dieser Befehl wird vom Master benutzt, um einen vollständigen Status des Schaltkreises und des Schrittmotors zu bekommen. Die Parameter die über den seriellen Bus zum Master gesendet werden, sind:

- Den Betrag des Max Spulenstroms und des Haltespulenstroms (Irun und Ihold)
- Maximal- und Minimalgeschwindigkeit für den Schrittmotor (Vmax und Vmin)
- Richtung der Bewegung Urzeigersinn/gegen den Uhrzeigersinn (Shaft)
- Schrittmodus (StepMode) (Tabelle 13: StepMode auf Seite 21)
- Beschleunigung (Verzögerung) für den Schrittmotor (Acc)
- Beschleunigungsform (AccShape)
- Statusinformation (sehen Sie bald)
 - Bewegungs Status <motion [2:0]>
 - Überstrom Flags für Phase1 <OVC1> und Phase2 <OVC2>
 - Digitaler Strom Reset <VddReset>
 - Ladunspumpenstatus <CPFail>
 - Status über externen Schalter <ESW>
 - Schrittverlust <StepLoss>
 - Elektrischer Defekt <EIDef>
 - Unterspannung <UV2>
 - Temperatur Info <Tinfo>
 - Temperatur Warnung <TW>
 - Temperaturabschaltung <TSD>

GetFullStatus1 command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	GetFullStatus1	1	0	0	0	0	0	0	1

Tabelle19: GetFullStatus1 Befehl

GetFullStatus1 command (Response)									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	1
1	Address	1	1	1	1	OTP3	OTP2	OTP1	OTP0
2	Irun & lhold	Irun (3:0)				lhold (3:0)			
3	Vmax & Vmin	Vmax (3:0)				Vmin (3:0)			
4	Status 1	AccShape	StepMode(1:0)		Shaft		ACC(3:0)		
5	Status 2	VddReset	StepLoss	EIDef	UV2	TSD	TW	Tinfo(1:0)	
6	Status 3	Motion(2:0)			ESW	OVC1	OVC2	1	CPFail
7	N/A	1	1	1	1	1	1	1	1
8	N/A	1	1	1	1	1	1	1	1

Tabelle20: GetFullStatus1 Befehl (Antwort)

7.1.9.2 GetFullStatus2 (\$FChex)

Dieser Befehl wird dem Schaltkreis vom Master geliefert, um die aktuelle Position des Schrittmotors zu bekommen. Die Position wird vom TMC im 16-Bit Format geliefert, mit den 3 LSBs auf '0' wenn er sich im Halbschrittmodus (StepMode = "00") befindet. Weiterhin werden auch die programmierte Zielposition und sichere Position geliefert.

Schreibweisen:

- Aktuelle Position des Schrittmotors, obere 8Bits <ActPos (15:8)>
- Aktuelle Position des Schrittmotors, untere 8Bits <ActPos (7:0)>
- Zielposition des Schrittmotors, obere 8Bits <TagPos (15:8)>
- Zielposition des Schrittmotors, untere 8Bits <TagPos (7:0)>
- Sichere Position des Schrittmotors, untere 8Bits <SecPos (7:0)>
- Sichere Position des Schrittmotors, obere 3Bits <SecPos (10:8)>

GetFullStatus2 command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	GetFullStatus2	1	1	1	1	1	1	0	0

Tabelle21: GetFullStatus2 Befehl

GetFullStatus2 command (Response)									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	1
1	Address	1	1	1	1	OTP3	OTP2	OTP1	OTP0
2	Actual Position 1	ActPos(15:8)							
3	Actual Position 2	ActPos(7:0)							
4	Target Position 1	TagPos(15:0)							
5	Target Position 2	TagPos(7:0)							
6	Secure Position	SecPos(7:0)							
7	Secure Position	1	1	1	1	1	SecPos(10:8)		
8	N/A	1	1	1	1	1	1	1	1

Tabelle22: GetFullStatus2 Befehl (Antwort)

7.1.9.3 GetOTPPParam (\$82hex)

Dieser Befehl wird zum TMC geschrieben, um den Inhalt eines OTP Speichers zu lesen. Zwecks weiterer Informationen beziehen Sie sich auf Tabelle 11: OTP Speicherstruktur auf Seite 21.

GetOTPPParam command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	GetOTPPParam	1	0	0	0	0	0	1	0

Tabelle23: GetOTPPParam Befehl

GetOTPPParam command (Response)									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	1
1	OTP byte 0	OTP@0x00							
2	OTP byte 1	OTP@0x01							
3	OTP byte 2	OTP@0x02							
4	OTP byte 3	OTP@0x03							
5	OTP byte 4	OTP@0x04							
6	OTP byte 5	OTP@0x05							
7	OTP byte 6	OTP@0x06							
	OTP byte 7	OTP@0x07							

Tabelle24: GetOTPPParam Befehl (Antwort)

7.1.9.4 GotoSecurePosition (\$84hex)

Dieser Befehl wird vom Master zu einen oder allen Schrittmotoren gesendet, um ihn zu der sicheren Position SecPos [10:0] zu fahren.

GotoSecurePosition command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	GotoSecurePosition	1	0	0	0	0	1	0	0

Tabelle25: GotoSecurePosition Befehl

7.1.9.5 HardStop (\$85hex)

Dieser Befehl wird intern ausgelöst, wenn ein elektrisches Problem in einer oder beiden Spulen wahrgenommen wird und dadurch die H Brücken ausschaltet. Wenn dieses Problem während sich der Motor bewegt auftritt, wird das <StepLoss> Flag beim nächsten GetStatus Befehl gesetzt, um den Master vor möglichen Schrittverlusten zu warnen. Der HardStop Befehl kann auch zu eigenen Sicherheitsgründen vom Master erteilt werden.

HardStop command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	HardStop command	1	0	0	0	0	1	0	1

Tabelle26: HardStop Befehl

7.1.9.6 ResetPosition (\$86hex)

Dieser Befehl wird zum TMC vom Master gesendet, um das ActPos und das TagPos Register zu löschen, um eine Initialisierung der Schrittmotorposition zu erlauben.

ResetPosition command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	ResetPosition	1	0	0	0	0	1	1	0

Tabelle27: ResetPosition Befehl

7.1.9.7 ResetToDefault (\$87hex)

Dieser Befehl wird zum TMC vom Master gesendet, um den gesamten Slave in den Ausgangszustand zu bringen. ResetToDefault wird zum Beispiel zum überschreiben des RAM's mit den Grundzustandsparametern der Register benutzt. Dies ist ein anderer Weg für den Master einen Slave zu initialisieren, ob im Falle eines Notfalls oder einfach nur um den RAMinhalt zu aktualisieren.

Notiz: ActPos wird nicht von einem ResetToDefault Befehl modifiziert, und sein Wert wird in das TagPos Register kopiert, um den Versuch, den Motor zur Position '0' zu fahren, zu vermeiden .

ResetToDefault command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	ResetToDefault	1	0	0	0	0	1	1	1

Tabelle28: ResetToDefault Befehl

7.1.9.8 RunInit (\$88hex)

Dieser Befehl wird zum TMC vom Master gesendet, um durch suchen des Nullpunktes (oder Referenzposition) die Ausgangsposition des Motors zu initialisieren. Beziehen Sie sich auf 6.1.11 Referenz Suche/Positions Initialisierung auf Seite 15.

Es beginnt mit einer Sequenz folgender Befehle:

- SetMotorParam (Vmax, Vmin)
- SetPosition (Pos1)
- SetMotorParam (Vmin, Vmax)
- SetPosition (Pos2)
- ResetPosition

Sobald der Befehl RunInit gestartet ist, kann es nicht von irgendeinem anderen Befehl unterbrochen werden. Außer es tritt eine interne Sicherheitsabschaltung, (6.1.10 Motorshutdown) auf oder der Befehl HardStop wird erhalten. Weiterhin muß der Master überprüfen, ob die aktuelle Position des Schrittmotors nicht der Zielposition der ersten Bewegung entspricht. Dies ist sehr wichtig, sonst blockiert sich der TMC selbst. Befindet sich der TMC erst einmal in diesem Zustand, kann dieser nur noch mit dem HardStop Befehl gefolgt von einem GetFullStatus Befehl verlassen werden.

RunInit command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	RunInit command	1	0	0	0	1	0	0	0
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	Vmax Vmin	Vmax(3:0)				Vmin(3:0)			
5	Position2 byte 1	TagPos1(15:8)							
6	Position2 byte 2	TagPos1(7:0)							
7	Position1 byte 1	TagPos2(15:8)							
8	Position1 byte 2	TagPos2(7:0)							

Tabelle29: RunInit Befehl

7.1.9.9 SetMotorParameter (\$89hex)

Dieser Befehl wird an den TMC vom Master gesendet, um die Werte für die (unten aufgeführten) Schrittmotorparameter in den RAM zu schreiben.

- Spitzenstromwert durch die Spulen (Irun)
- Haltestrom durch die Spulen (Ihold)
- Maximalgeschwindigkeit des Schrittmotors (Vmax)
- Minimalgeschwindigkeit des Schrittmotors (Vmin)
- Beschleunigungsform (AccShape) 0 = mit Beschl./ 1 = ohne Beschl.
- Schrittmodus (StepMode)
- Bewegungsrichtung (Shaft)
- Beschleunigung (Verzögerung) für den Schrittmotor (Acc)
- Sichere Position des Schrittmotors (SecPos)

SetMotorParameter command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	SetMotorParam command	1	0	0	0	1	0	0	1
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	Irun & I hold	Irun(3:0)				Ihold(3:0)			
5	Vmax & Vmin	Vmax(3:0)				Vmin(3:0)			
6	Status	SecPos(10:8)			Shaft		Acc(3:0)		
7	SecurePos	SecPos(7:0)							
8	StepMode				AccShape	StepMode[1:0]			

Tabelle30: SetMotorParam Befehl

7.1.9.10 SetOTP (\$90hex)

Dieser Befehl wird zu dem TMC vom dem Master gesendet um den OTP Speicher zu beschreiben.
Achtung: Einmal auf „1“ programmierte Bits sind nicht wieder löschar!

SetOTP command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	ZapOTP command	1	0	0	1	0	0	0	0
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	OTP Address	1	1	1	1	1	OTPA(2:0)		
5	Pbit	Pbit(7:0)							

Tabelle31: SetOTP Befehl

7.1.9.11 SetPosition (\$8Bhex)

Dieser Befehl wird zu dem TMC vom dem Master gesendet, um den Motor zu einer angegebenen Position zu fahren, in Abhängigkeit der Lage zur Nullposition des Motors und in welchem Schrittmodus er sich befindet. SetPosition wird nicht ausgeführt, wenn eine der folgenden Flags auf eins gesetzt sind:

- Übertemperatur <TSD>
- Unterspannung <UV2>
- Schrittverlust <StepLoss>
- Elektrischer Fehler <EIDef>

SetPosition command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	SetPosition command	1	0	0	0	1	0	1	1
2	N/A	1	1	1	1	1	1	1	1
3	N/A	1	1	1	1	1	1	1	1
4	Position byte1	TagPos(15:8)							
5	Position byte2	TagPos(7:0)							

Tabelle32: SetPosition Befehl

7.1.9.12 SoftStop (\$8Fhex)

Wenn der Befehl SoftStop während einer Bewegung des Schrittmotors auftritt, provoziert er eine unmittelbare Verzögerung bis zu VMIN, gefolgt von einem Stop ohne Rücksicht auf die erreichte Position.

Dieser Befehl tritt in den folgenden Fällen auf:

- Die Chiptemperatur übersteigt die Thermal Shutdown Schwelle
- Der Master fordert einen SoftStop an

SoftStop command									
Byte	Content	Structure							
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	Slave Address	1	1	OTP3	OTP2	OTP1	OTP0	HW2	0
1	SoftStop command	1	0	0	0	1	1	1	1

Tabelle33: SoftStop Befehl

7.1.10 Positionierungsbeispiel

Das TMC222 muß eine Positionierung ausführen, wobei die aktuelle Position des Schrittmotors unbekannt ist. Die gewünschte Zielposition ist 3000 µSchritte von Position 0 entfernt. Siehe Bild28: Positionierungsbeispiel: Anfangssituation auf Seite 35

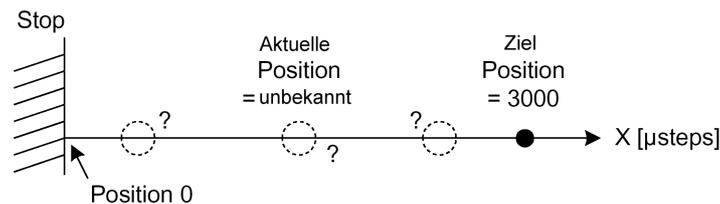


Bild28: Positionierungsbeispiel: Anfangssituation

Die folgende Sequenz von Befehlen muß an den Slave gesendet werden, um das oben beschriebene Szenario auszuführen (angenommen nach dem Einschalten):

GetFullStatus1

Der Befehl wird verwendet, um den aktuellen Status des TMC222s zu lesen. Elektrische oder Umgebungsprobleme wurden gemeldet und der TMC verlässt den Systemruhestatus und ist für Aktionen bereit.

GetFullStatus2

Der Schaltkreis begibt sich in einen Blockierzustand, wenn die aktuelle Position der ersten Zielposition des Befehls RunInit entspricht. Dieser Befehl wird verwendet, um die aktuelle Position zu lesen. Der Master muß achtgeben daß beide Positionen verschiedene Werte enthalten. Für Blockierungsbedingungen sehen Sie 6.1.11 Referenzsuche/Positionsinitialisierung auf Seite 15.

SetMotorParam

Um den Schrittmotor mit einer gewünschten Bewegung zu treiben, müssen die Parameter für Drehmoment, Geschwindigkeit, usw. durch den Befehl SetMotorParam ausgegeben werden. Siehe 7.1.9.9 SetMotorParam auf Seite 33.

RunInit

Ist die aktuelle Position unbekannt, sollte eine Positionsinitialisierung ausgeführt werden. Die erste Bewegung muß den Schrittmotor in den sicheren Stop fahren. Die zweite Bewegung ist eine sehr kurze Bewegung, die den Motor aus der Stopposition heraus bringt. Die aktuelle Position wird dann automatisch auf Null gesetzt, nachdem die zweite Bewegung beendet ist. Siehe 7.1.9.8 RunInit auf Seite 33.

Nach der Referenzsuche sieht die aktuelle Situation wie in Bild 29 aus: Positionierungsbeispiel: Situation nach Referenzsuche.
Aktuelle Position des Schrittmotors entspricht "0", die Zielposition ist 3000 μ Schritte von der aktuellen Position entfernt.



Bild 29: Positionierungsbeispiel: Situation nach Referenzsuche

Jetzt kann der Positions-Befehl `SetPosition` eingegeben werden, um den Schrittmotor zur gewünschten Stelle zu fahren.

`SetPosition`

Dieser Befehl bewirkt, daß sich der Schrittmotor zur gewünschten Zielstelle begibt. Siehe 17.1.9.11 `SetPosition` auf Seite 34. Nachdem die Bewegung beendet worden ist, sieht die Situation wie in Abbildung 30 aus: Positionierungsbeispiel: Bewegung beendet.



Bild 30: Positionierungsbeispiel: Bewegung beendet

Danach können der aktuelle Status und die aktuelle Position durch die Verwendung der zwei **GetFullStatus** Befehle geprüft werden. Der Master kann überprüfen, ob ein elektrisches oder ein Temperaturproblem aufgetreten ist. Weiterhin wird die aktuelle Position gelesen.

GetFullStatus1: Sehen Sie 7.1.9.1 `GetFullStatus1` auf Seite 29.

GetFullStatus2: Sehen Sie 7.1.9.2 `GetFullStatus2` auf Seite 30.

8. Gehäusemaße

8.1 SOIC-20

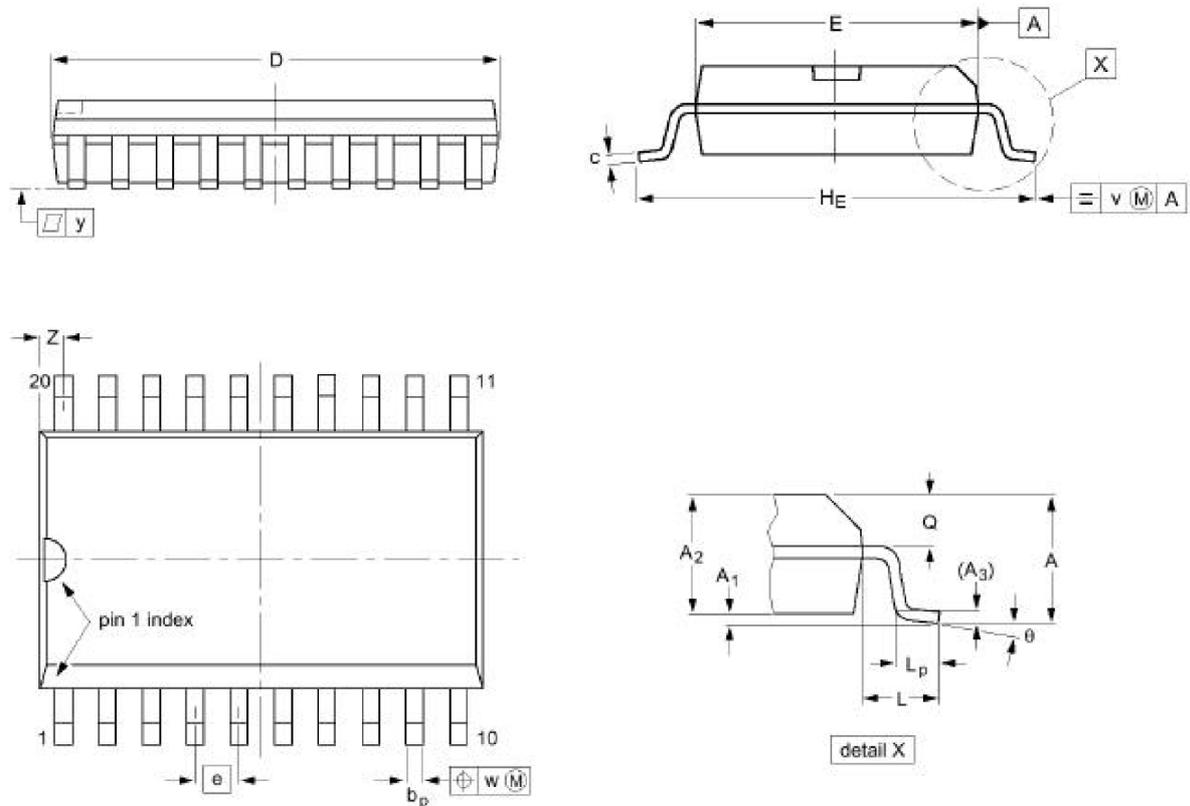


Bild31: Gehäusemaße SOIC-20

Einheit	A max	A ₁	A ₂	A ₃	b _p	c	D ⁽¹⁾	E ⁽¹⁾	e	H _E	L	L _p	Q	v	w	y	Z ⁽¹⁾	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	13.0 12.6	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.51 0.49	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	0°

Tabelle34: SOIC-20 Mechanische Daten

Elektrische Eigenschaften

8.2 Maximale Leistungseinschätzung

Parameter		Min	Max	Einheit
Vbat	Versorgungsspannung	-0.3	+35	V
Vlin	Bus Eingangs Spannung	-80	+80	V
Tamb	Umgebungstemperatur unter Abweichung(*)	-50	+150	°C
Tst	Lagertemperatur	-55	+160	°C
Vesd (**)	Elektrostatische Entladungsspannung am LIN Pin	-4	+4	kV
	Elektrostatische Entladungsspannung an den anderen Pins	-2	+2	kV

Tabelle35: Maximale Leistungseinschätzung

(*) Die Funktion des IC's ist nicht mehr garantiert

(**) Menschliches Körpermodell (100pF über 1,5kOhm)

8.3 Betriebsbereich

Parameter		Min	Max	Einheit
Vbat	Versorgungs Spannung	+8	+29	V
Top	Betriebstemperaturbereich	Vbat <= 18V	+125	°C
		Vbat <= 29V	+85	°C

Tabelle36: Betriebsbereich

8.4 DC Parameter

Motor Driver								
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit	
IMSm _{max} Peak	OA1 OA2 OB1 OB2	Max current through motor coil in normal operation			800		mA	
IMSm _{RMS}		Max RMS current through coil in normal operation			570		mA	
RD _{son}		On resistance for each pin (including bond wire)	To be confirmed by characterization				1	Ω
IMSL		Leakage current	HZ Mode, 0V < V(pin) < V _{bb}		-50		+50	μA

Tabelle37: DC Parameter Motortreiber

Thermal Warning and shutdown							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
T _{tw}		Thermal Warning		138	145	152	°C
T _{t_{sd}} (*)		Thermal Shutdown			T _{tw} + 10		°C
T _{low}		Low Temperature Warning				T _{tw} - 155	

Tabelle38: DC Parameter Thermische Warnung und Abschaltung

Supply and Voltage regulator							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
V _{bb}	VBB	Nominal operating supply range		6.5		18	V
V _{bb} OTP		Supply Voltage for OTP zapping		8.5		9.5	V
UV1		Low voltage high threshold		8.8	9.4	9.8	V
UV2		Stop voltage low threshold		8.1	8.5	8.9	V
I _{bat}		Total current consumption	Unloaded Outputs			10	
V _{dd}	VDD	Internal regulated output (**)	8V < V _{bb} < 18V Load = 1μF (+100nF cer.)	4.75	5	5.25	V
I _{dd} Stop		Digital current consumption	V _{bb} < UV2		2		MA
V _{dd} Reset		Digital supply reset level (***)				4.4	V
I _{dd} Lim		Current limitation	Pin shorted to ground			40	MA

Tabelle39: DC Parameter Spannungsstabilisierung

(*) um von Messungen bestätigt zu werden

(**) der RAM Inhalt wird durch diese Spannung nicht geändert

(***) externer Widerstand gesehen von Pin SW1 oder HW2, enthält 1kOhm parallel Widerstand

Switch Input and hardwired address input HW[2]							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
Rt_OFF	SW1 HW2	Switch OFF resistance ⁽⁵⁾	Switch to GND or Vbat	10			k Ω
Rt_ON		Switch ON resistance ⁽⁵⁾				2	k Ω
Vbb_sw		Vbb range for guaranteed operation of SW1 and HW2		6		18	V
Vmax_sw		Maximum Voltage	T < 1s			40	V
Ilim_sw		Current limitation	Short to GND or Vbat		30		mA

Tabelle40: DC Parameter Eingang Schalter und Hardwired Adresse

Hardwired address inputs and test pin							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
Vlow	HW0	Input level high		0.7			Vdd
Vhigh	HW1	Input level low				0.3	Vdd
HWhyst	TST	Hysteresis		0.075			Vdd

Tabelle41: DC Parameter und Hardwired Adresseingang und Test Pin

Charge Pump							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
Vcp	VCP	Output Voltage External Buffer Capacitor	Vbb > 15V	Vbb+10	Vbb+12.5	Vbb+15	V
Cbuffer			Vbb > 8V	Vbb+5.8 220		470	V nF
Cpump	CPP CPN	External pump Capacitor		220		470	nF

Tabelle42: DC Parameter Ladungspumpe

8.5 AC Parameter

Power-Up							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
Tpu		Power-Up time				10	ms

Tabelle43: AC Parameter Einschalten

Switch Input and hardwired address input HW[2]							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
Tsw	SW1	Scan Pulse Period		921	1024	1127	µs
Tsw_on	HW2	Scan Pulse Duration			1/16		Tsw

Tabelle44: AC Parameter Eingang Schalter und Hardwired Adresseingang

Motor Driver							
Symbol	Pin(s)	Parameter	Test condition	Min	Type	Max	Unit
Fpwm	Oxx	PWM frequency		18	20	22	KHz
Tbrise		Turn-On transient time	Between 10% and 90%		350		ns
Tbfall		Turn-Off transient time			250		ns

Tabelle46: AC Parameter Motortreiber

Überarbeitungsprotokoll

Version	Date	Comments
up to 0.90p	July 9 th , 2003	before v. 0.90 changes on unpublished versions only
0.91p	September 18 th , 2003	pins renamed according to TRINAMIC conventions; corrections concerning cross references, drawings in PDF