Bedienungsanleitung

Version 1.01

Motion Controller mit Sinus-kommutierung für EC-Motoren

MMC 24/5-10

MATTKE AG
Leinenweberstraße 12
D-79108 Freiburg
Germany

Telefon: +49 (0)761- 15 23 4-0
Telefax: +49 (0)761- 15 23 4-56
E-Mail: info@mattke.de
http://www.mattke.de

Inhaltsverzeichnis

Allgemeine Beschreibung **4**
Technische Daten **7**
Anschlüsse **8**
Grundeinstellungen **10**
Drehzahlregelung über Analogeingang **11**
Einfache Möglichkeiten der Sollwertvorgabe über Potentiometer 11
RS-232 Schnittstelle und ASCII-Befehle **13**
Einstellungen der RS-232 Schnittstelle 13
Aufbau der ASCII-Befehle 13
Abspeichern der Konfigurationseinstellungen 14
Ändern der Baudrate 15
Einstellen der Knotenadresse 15
Bestimmen des Motortyps 16
Anwählbare Motortypen und deren Parameter 16
Konfigurationen der Drehzahlregelung **17**
Quelle für den Drehzahlsollwert 17
Einstellen der Reglerparameter 20
Positionsregelung **21**
Standardablauf einer Positionierung 22
Zusammengesetzte Bewegungsprofile 22
Einstellen der Reglerparameter 22
Auswerten von Referenzmarken und Endschalter **23**
Direkte Programmierung über HA-, HL- und HN-Befehle 24
Programmierbare Referenzfahrtsequenz 25
Hard-Blocking Funktion 26
Hall-Indexfahrt 27
Erweiterte Betriebsarten **28**
Schrittmotor-Modus 28
Elektronisches Getriebe (Gearing-Mode) 30
Positionsregelung auf Spannung am Analogeingang 31
Externer Impulsgeber als Istwertgeber 32
Spannungssteller-Modus 34
Fehlerbehandlung und Fehleranschluss **35**
Abweichung zur Solldrehzahl als Fehler ausgeben 35
Verzögerte Aktivierung eines Fehlers 35
Fehleranschluss als Digitalausgang 36
Fehleranschluss als Impulsausgang 36
Speichern und Ausführen von Fahrprogrammen **38**
Steuerung von Ablaufprogrammen 39
Erläuterungen zu Befehlen und Funktionen 41
Technische Information **43**
Sinuskommutierung 43
Stromregler und I^2t-Strombegrenzung 43
Übertemperatursicherung 44
Unterspannungsüberwachung 45
Überspannungsregelung 45
Anhang **46**
Elektromagnetische Verträglichkeit (EMV) 46
RS-232 Multiplexerplatine 47
Der ASCII-Befehlssatz 48
Konfigurations- und Fahrprogrammbeispiele 56
Konfiguration im Auslieferungszustand 63
Indexverzeichnis **65**

Abbildungsverzeichnis

Abbildung 1: Abmessung 5
Abbildung 2: Drehzahlregelung mit Sollwert vom Analogeingang 10
Abbildung 3: einfache Möglichkeit über Potentiometer 11
Abbildung 4: Möglichkeit über Potentiometer mit höherer Präzision 12
Abbildung 5: Speicherungsmethode beim EEPROM 14
Abbildung 6: Übergangscharakteristik bei analoger Drehzahlvorgabe 18
Abbildung 7: Positionsregelung mit Referenzschalter am Analogeingang 21
Abbildung 8: Zusammengesetztes Bewegungsprofil im Vergleich zum Trapezprofil 22
Abbildung 9: Schrittmotorbetrieb mit Drehrichtungseingang 28
Abbildung 10: Gearing-Mode mit Referenzmarke am Fault-Anschluss 30
Abbildung 11: Positionsregelung auf analoge Spannung 31
Abbildung 12: Externer Impulsgeber als Positions- und Drehzahlgeber 33
Abbildung 13: Funktion der I²t-Strombegrenzung 44
Abbildung 14: RS-232 Multiplexerplatine 47

Allgemeine Beschreibung

Der MMC24/5-10 ist für bürstenlose Faulhaber-Motoren mit linearen Hallsensoren konzipiert. Deshalb sind ohne zusätzlichen Geber sehr geringe Drehzahlen und hohe Positionsauflösungen (1/1.000 Umdrehungen) möglich.
Der Motion Controller basiert auf einem leistungsstarken 16-Bit Mikrocontroller der eine hohe Regelgüte, eine genaue Positionierung und sehr niedrige Drehzahlen ermöglicht.

Mit diesem intelligenten Motion Controller können folgende Antriebsaufgaben gelöst werden:

- **Drehzahlregelung** bei hohen Anforderungen an Gleichlauf und geringsten Drehmoment-Schwankungen. Ein PI-Regler sorgt für Einhaltung der Solldrehzahlen.

- **Drehzahlprofile** wie z.B. Rampen-, Dreieck- oder Trapezbewegungen können realisiert werden. Auch kompliziertere Profile sind problemlos umsetzbar.

- **Positionierbetrieb**: Anfahren von definierten Positionen mit hoher Auflösung. Erfassen von **Referenzmarken und Endschalter**.

- **Erweiterte Betriebsarten** (z.B. Schrittmotorbetrieb zum Synchronisieren von Achsen)

- Betrieb als **Drehmomentenregler** durch einstellbare Strombegrenzung.

- Speichern und Ausführen von **Fahrprogrammen** für einen Stand-Alone Positionierbetrieb oder zur Entlastung des HOST-Rechners.

Zur Umsetzung dieser Aufgaben stehen verschiedene Ein- und Ausgänge zur Verfügung:

- **Sollwerteingang** für die Drehzahlvorgabe. Es kann mit Analog- oder PWM Signal gearbeitet werden. Der Eingang kann auch ein Referenzmarkensignal einlesen. Je nach Modus kann hier ein Frequenzsignal oder ein Inkrementalgeber angeschlossen werden.

- **Fehlerausgang** (Open Collector). Auch als Drehrichtungs- oder Referenzmarkeneingang umprogrammierbar.

- Zusätzlicher Digitaleingang.

- **RS-232 Schnittstelle** zur Ankopplung an einen PC.

Die Eingänge sind auch zur Steuerung von Fahrprogrammen geeignet.

Parameter einstellen und Konfiguration festlegen

Sämtliche Parameter können über die RS-232 Schnittstelle eingestellt und im integrierten Speicher (EEPROM) abgelegt werden. Auch die Einstellung der Konfiguration (Betriebsmodi, Ein/Ausgänge,...) erfolgt auf diese Weise. Deshalb kann der Motion Controller je nach Betriebsmodus auch komplett ohne RS-232 Schnittstelle betrieben werden.

Programmierung

Für die Programmierung und Bedienung steht ein umfangreicher ASCII-Befehlssatz zur Verfügung. Dieser kann vom PC z. B. über ein Terminalprogramm, wie in Windows enthalten, oder von einem beliebigen Steuerrechner vorgegeben werden.

Für Windows 95/98/NT ist das „Motion Manager“ - Programm verfügbar, das die Bedienung und Konfiguration wesentlich vereinfacht und zudem eine grafische Online-Analysemöglichkeit der Betriebsdaten bietet.

Einsatzgebiete

Durch die kompakte Bauform ist der Motion Controller in vielfältigen Anwendungen einsetzbar. Die flexiblen Anbindungsmöglichkeiten eröffnen ein breites Einsatzgebiet in allen Bereichen, zum Beispiel in dezentralen Systemen der Automatisierungstechnik, sowie in Handling- und Werkzeugmaschinen.

Optionen

- Für den Betrieb mehrerer Antriebe an einer einzigen RS-232 Schnittstelle ist eine RS-232 Multiplexerplatine erhältlich.
- Auf Anfrage ist eine spezielle Vorkonfiguration der Modi und Parameter möglich.
- Das Motion Manager-Programm ist auf Anfrage oder im Internet erhältlich.
- Einfache Adapter erlauben die Befestigung des MMC24/5-10 auf Hutschienen.

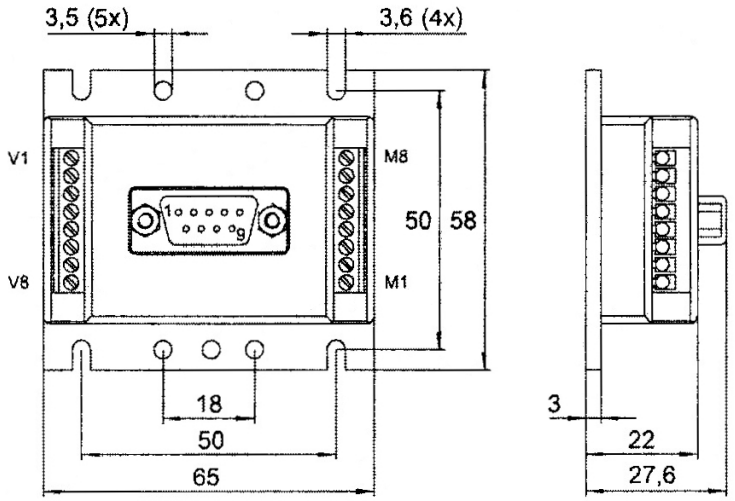

Abbildung 1: Abmessung

Anschlussbelegung

Versorgungsseite:

Nr.	Bezeichnung
V1	RS-232 TXD
V2	RS-232 RXD
V3	Analog GND
V4	Fehlerausgang
V5	Analog Eingang
V6	+24V
V7	GND
V8	Eingang Nr.3

SUB-D Stecker:

Nr.	Bezeichnung
2	RS-232 RXD
3	RS-232 TXD
5	GND

Motorseite:

Nr.	Bezeichnung	Farbe
M1	Phase C	gelb
M2	Hallsensor A	grün
M3	VCC (+5V)	rot
M4	Signal GND	schwarz
M5	Hallsensor B	blau
M6	Hallsensor C	grau
M7	Phase B	orange
M8	Phase A	braun

Technische Daten

Allgemeine Spezifikationen			
Versorgungsspannung	U_B	12 ... 28	V DC
PWM-Schaltfrequenz	f_{PWM}	31,25	kHz
Wirkungsgrad	η	95	%
Max. Dauer-Ausgangsstrom	I_{dauer}	5	A
Max. Spitzen-Ausgangsstrom	I_{max}	10	A
Stromaufnahme der Elektronik	I_{el}	0,06	A
Drehzahlbereich		10 ... 15.000	rpm
Ausgangsspannung für externen Gebrauch	VCC	5	V DC
- Max. Laststrom	I_{CC}	60	mA
Eingang Nr. 1[1)]	Eingangswiderstand	18	kΩ
Drehzahlsollwert analog	Spannungssignal	±10	V
	Steigung der Kennlinie	1.000[2)]	rpm/V
Drehzahlsollwert digital	PWM Signal	low 0... 0,5 / high 4 ... 30	V
	Frequenzbereich	100 … 2.000	Hz
	Tastverhältnis 50%	0	rpm
	Tastverhältnis <50%	linksdrehend	
	Tastverhältnis >50%	rechtsdrehend	
Externer Encoder / Schrittfrequenz	f_{max}	150	kHz
Fehlerausgang (Eingang Nr.2)	Open collector	Max. U_B / 30mA	
	Kein Fehler	Durchgeschaltet nach GND	
	Als Eingang programmiert	low 0 ... 0,5 / high 4 ... U_B	V
Eingang Nr. 3	Signalpegel	low 0... 0,5 / high 4 ... 30	V
Schnittstelle	RS-232	9.600 (1.200, 2.400, 4.800, 19.200)	Baud
Speicher für Fahrprogramme	Serielles EEPROM	7936	Bytes
Betriebstemperaturbereich		0 ... +70	°C
Lagertemperaturbereich		-25 ... +85	°C
Gewicht		110	g

[1]) Kann über die RS-232 Schnittstelle konfiguriert werden (Voreinstellung: Drehzahlsollwert analog).
[2]) Voreingestellter Wert. Dieser kann über die RS-232 Schnittstelle geändert werden.

Bestellhinweise:

Die Motion Controller werden auf einen der nachfolgenden Motoren abgestimmt und sind nur mit diesen verwendbar.

1628T012B K1155	1628T024B K1155
2036U012B K1155	2036U024B K1155
2444S024B K1155	3056K012B K1155
3056K024B K1155	3564K024B K1155

Die Daten dieser Motoren können den Motordatenblättern (Katalog, Internet,...) ohne K1155-Sondernummer entnommen werden.
Die Bezeichnung K1155 bedeutet: Motor mit linearen Hallsensoren, angepasst auf MMC 24/5-10

Anschlüsse

Stromversorgungsanschlüsse (+24V, GND)

Das Netzgerät sollte mindestens 5A liefern können.
Polaritätsanschluss beachten. Bei Falschanschluss wird die interne Sicherung ausgelöst. Diese darf nur werkseitig ersetzt werden!

Analogeingang (Analog Eingang, Analog GND = AGND)

Der Analogeingang ist als Differenzeingang ausgeführt.
Der Analog-GND sollte mit dem Stromversorgungs-GND verbunden werden. Damit wird verhindert, dass der Spannungsabfall in der Versorgungsleitung sich auf den Drehzahlvorgabewert auswirkt.

Der Analogeingang hat je nach Konfiguration unterschiedliche Verwendungszwecke:

- Drehzahlsollwertvorgabe über eine Spannung am Analogeingang (Einstellung bei Auslieferung)
- Drehzahlsollwertvorgabe über PWM-Signal am Analogeingang
- Erfassung einer Referenzmarke (Endschalter) bei Betrieb als Motion Controller
- Eingänge für den externen Impulsgeber im Encodermodus (Analog Eingang gegen GND: Kanal A / Analog GND gegen GND: Kanal B)

RS-232 Anschlüsse

Die RS-232 Hardware Verdrahtung wird über die Anschlüsse TXD, RXD und über den Versorgungs-GND hergestellt. Die eingebaute RS-232 Schnittstelle erlaubt eine direkte Verbindung mit dem PC.

Fehlerausgang

Der Fehlerausgang ist durch folgende Eigenschaften charakterisiert:

- Schalter, der nach GND schaltet (Open Collector)
- Ausgangswiderstand: durchgeschaltet (low Pegel): 47Ω , offen (high Pegel): 10kΩ
- Im Fehlerfall ist der Schalter offen (LED leuchtet nicht, high Pegel)
- Ausgangsstrom maximal 30mA, Spannung im offenen Zustand darf maximal U_B betragen

Der Fehlerausgang wird bei folgenden Situationen aktiviert:

- Strombegrenzung aktiv
- Unterspannung (Versorgungsspannung kleiner 10V)
- Überspannungsregler aktiv (Versorgungsspannung über 32V)
- Übertemperaturabschaltung

Der Fehlerausgangs-Anschluss kann auch für andere Funktionen umkonfiguriert werden:

- Impulsausgang
- Digitalausgang
- Endschaltereingang
- Drehrichtungseingang

Eingang Nr. 3

Der 3. Eingang ist ein Digitaleingang und kann zur Auswertung einer Referenzmarke oder als Steuereingang für Programme verwendet werden.

Grundeinstellungen

Der MMC24/5-10 ist bei der Auslieferung standardmäßig auf den Motortyp 5 (2444S024B K1155) eingestellt.

Für einen sicheren Betrieb mit dem tatsächlich angeschlossenen Motor sind folgende Kapitel zu beachten:

- Bestimmen der Motortypen (Seite 16)
- Einstellen der Reglerparameter (Seite 20)
- Stromregler und I^2t-Strombegrenzung (Seite 43)

Beim Betrieb des MMC24/5-10 in einem erweiterten Betriebsmodus sind die entsprechenden Kapitel zu beachten. Insbesondere sind folgende Punkte wichtig:

- Betriebsmodus: Externer Impulsgeber als Istwertgeber (Seite 32) ... Einstellung auf den angeschlossenen Impulsgeber (Befehl: ENCRES)
- Fehlerausgangs-Anschluss als Eingang: Befehl REFIN bzw. DIRIN und EEPSAV bevor ein Signal an den Fehlerausgangs-Anschluss angeschlossen wird

Bei Nichtbeachtung dieser Grundeinstellungen kann es zur Zerstörung von Komponenten kommen!

Der MMC24/5-10 ist standardmäßig als Drehzahlregler konfiguriert. Die Solldrehzahl wird als Spannungssignal am Analogeingang vorgegeben,

In dieser Betriebsart ist die RS-232 Schnittstelle nicht zwingend erforderlich. Mit ihrer Hilfe können aber Konfigurationseinstellungen vorgenommen werden. Diese werden später noch genauer erläutert.

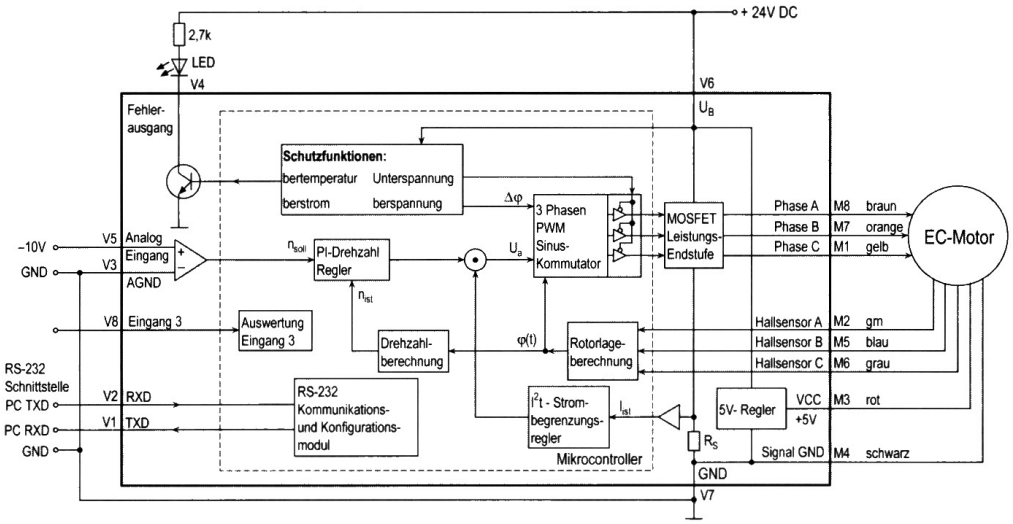

Abbildung 2: Drehzahlregelung mit Sollwert vom Analogeingang

Drehzahlregelung über Analogeingang

Der MMC24/5-10 ist standardmäßig als Drehzahlregler konfiguriert. Die Solldrehzahl wird als Spannungssignal am Analogeingang vorgegeben.

In dieser Betriebsart ist die RS-232 Schnittstelle nicht zwingend erforderlich. Mit ihrer Hilfe können aber Konfigurationseinstellungen vorgenommen werden. Diese werden später noch genauer erläutert.

Einfache Möglichkeiten der Sollwertvorgabe über Potentiometer

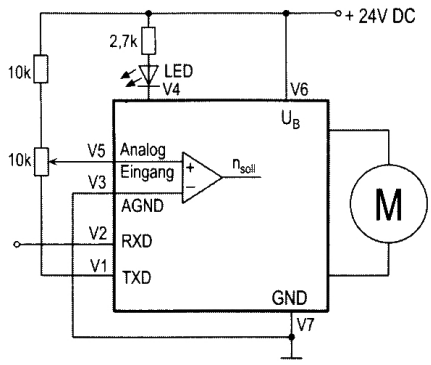

Abbildung 3: einfache Möglichkeit über Potentiometer

Abbildung 3 zeigt die einfachste Möglichkeit, wobei es folgendes zu beachten gilt:

- Solldrehzahl ist von der Versorgungsspannung abhängig
- Geringe Genauigkeit des Sollwertes infolge von Spannungsänderungen am TXD-Anschluss
- Die RS-232 Schnittstelle kann nicht verwendet werden

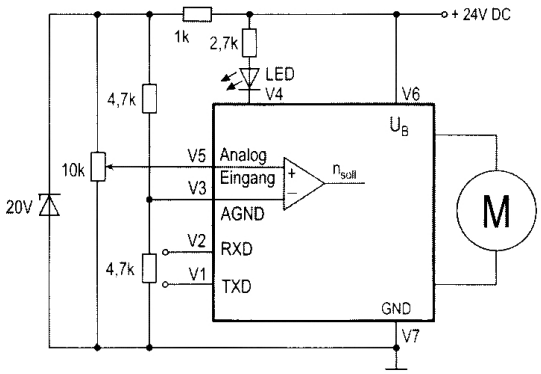

Abbildung 4: Möglichkeit über Potentiometer mit höherer Präzision

Abbildung 4 zeigt eine Schaltung mit höherer Präzision, wobei folgendes gilt:

- Analog GND ist vom Versorgungs-GND getrennt
- Die RS-232 ist voll verwendbar

Bemerkung zur Eingangsschaltung:

Die Eingangsschaltung am Analogeingang ist als Differenzverstärker ausgeführt. Ist der Analogeingang offen, so liegt bereits eine Spannung von ca. 2V an. Das bedeutet, dass in diesem Fall sich der Motor mit ca. 2.000rpm dreht. Der Eingang muss niederohmig mit AGND verbunden werden bzw. auf den Spannungspegel des AGND gesetzt werden, um 0rpm zu erzeugen

RS-232 Schnittstelle und ASCII-Befehle

Die RS-232 Schnittstelle erlaubt eine Ankopplung an einen Personal Computer (PC) oder an eine digitale Steuerung, wie zum Beispiel einer SPS oder eines IPC.

Verwendung der RS-232 Schnittstelle

- Konfiguration des Motors (Betriebmodi, Parameter, ...)
- Online Daten und Werte abfragen
- Online Kommunikation während des Betriebes

Einstellungen der RS-232 Schnittstelle

Das RS-232 Interface ist mit den nachstehenden Einstellungen konfiguriert. Es ist sehr wichtig, dass auch am Personal Computer oder in der Steuerung die gleiche Konfiguration eingestellt wird, da nur so eine korrekte Funktion sichergestellt ist.

Einstellungen:

- 9600 Baud
- 8 Datenbits
- 1 Stoppbit
- No Parity

Arbeitet man mit einem Terminalprogramm am PC, so sollte außerdem „lokales Echo“ und „CR beim Senden“ aktiviert sein.

Aufbau der ASCII-Befehle

Die Kommunikation mit dem PC geschieht üblicherweise über ein einfaches ASCII-Terminalprogramm, wie es im Windows Betriebssystem standardmäßig vorhanden ist. Mit dem verfügbaren „Motion-Manager“-Bedienprogramm ist eine besonders elegante Bedienung des Antriebs möglich. Außerdem können damit Echtzeitdaten, wie die Istdrehzahl, grafisch ausgegeben werden.

Die ASCII-Befehle sind nach folgendem Schema aufgebaut:

1.) Knotenadresse: (optional) ist nur nötig, wenn mehrere Motoren an einer Schnittstelle angeschlossen sind

2.) Befehlswort: eine Zeichenfolge, die nur aus Buchstaben besteht

3.) Zahl: bei manchen Befehlen folgt nach dem Befehlswort eine Zahl

4.) Den Abschluss bildet immer ein CR-Zeichen ➔ Terminalprogramm: Enter-Taste drücken

Beispiele:

V 500 [CR]... Umschalten auf Drehzahlreglermodus und mit Drehzahl 500Upm drehen

GST [CR] ... Get Status ... Der Status des Motors wird zurückgesendet

LA200 [CR]... Absolute Position auf den Wert 200 laden

Leerzeichen werden ignoriert. Es wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Die Antwort der Rückmeldebefehle ist immer eine ASCII-Zeichenfolge. Anschließend folgen ein CR-Zeichen (Return-Zeichen, Dezimalcode 13) und ein LF-Zeichen (Line Feed, Dezimalcode 10).

Beispiel:

Rückmeldung der aktuellen Position (POS-Befehl) Eingabe: POS [CR]
Rückantwort: 50000 [CR] [LF]

➔ Der Antrieb steht aktuell auf der absoluten Position 50000, d.h. 50 Umdrehungen vom Nullpunkt entfernt.

Abspeichern der Konfigurationseinstellungen

Eine Reihe von Parametern und Einstellungen kann in ein sogenanntes EEPROM übertragen werden. Dadurch sind diese Werte auch nach dem Abschalten der Versorgungsspannung noch im System vorhanden. Der Motion Controller startet nach dem Anschluss der Versorgungsspannung immer mit den im EEPROM befindlichen Einstellwerten (Abbildung 5). Dies macht es zum einen möglich, den Motion Controller zu einem bestimmten Antriebstyp (Drehzahlregler, Positionsregler,...) zu konfigurieren, zum anderen können damit bestimmte Parameter, die üblicherweise nur einmal eingestellt werden müssen, fix gespeichert werden.
Die Übernahme der Einstellungen in das EEPROM wird durch den ASCII-Befehl **EEPSAV** ausgelöst.

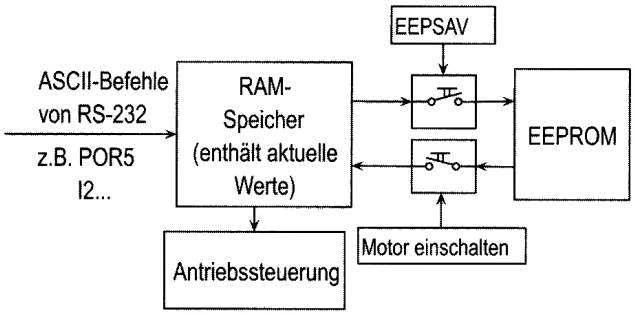

Abbildung 5: Speicherungsmethode beim EEPROM

Achtung:

Während der Ausführung von EEPSAV darf die Stromversorgung nicht unterbrochen werden, da es ansonsten zu Datenverlusten kommen könnte.

Bemerkung zu den Befehlstabellen:

Einstellungen von Befehlen die mit *) gekennzeichnet sind, werden bei **EEPSAV** gespeichert.

Ändern der Baudrate

Die Baudrate kann auf folgende Werte eingestellt werden: 1.200, 2.400, 4.800, 9.600, 19.200 Baud.

Befehl	Funktion	Beschreibung	Beispiel
BAUD *)	Select BAUD Rate	Baudrate für RS-232 Schnittstelle vorgeben	BAUD9600

Default-Einstellung: BAUD9600
Um nach einer Baudratenänderung wieder arbeiten zu können, muss auch am PC diese Änderung vorgenommen werden.

Einstellen der Knotenadresse

Mit Hilfe der sogenannten RS-232 Multiplexerplatine ist es möglich, mehrere Antriebe von einer einzigen RS-232 Schnittstelle aus zu betreiben.

Um die Einheiten zu adressieren, wird vor dem ASCII-Befehl die Knotenadresse gesetzt.

Wird keine Knotenadresse angegeben, so werden alle im Netz angeschlossenen Antriebe angesprochen.

Befehl	Funktion	Beschreibung	Beispiel
NODEADR *)	Define Node Address	Knotenadresse vorgeben(0 bis 255)	NODEADR5
GNODEADR	Get Node Address	Knotenadresse an Schnittstelle senden	GNODEADR ➔ 5

Default-Einstellung: NODEADR0

Vorsicht:

Es darf bei der Vergabe der Knotenadresse nur ein Antrieb angeschlossen sein, da ansonsten mehrere Antriebe die gleiche Adresse bekommen würden.

Vorsicht beim Betrieb mehrerer Antriebe:

Werden Befehle ohne Knotenadresse gesendet, so werden alle Antriebe im Netz angesprochen. Bei Abfragebefehlen führt dies zu einem Fehler, da alle Antriebe gleichzeitig antworten. Dieses Problem muss vom Anwender verhindert werden.

Asynchrone Antworten abschalten:

Bei asynchronen Antwortbefehlen könnte trotz Angabe der Knotenadresse ein Problem auftreten, da die Antwort nicht unmittelbar nach dem Senden des Befehls erfolgt. Beispiel: NP20000 ... Notify Position (wird später noch genauer erläutert).
Die Antwort „p“ erfolgt erst, wenn die Position 20.000 erreicht wurde. Dann könnte aber gerade eine Antwort eines anderen Knoten unterwegs sein und damit beide Informationen zerstören.

Befehl	Funktion	Beschreibung	Beispiel
ANSW *)	Asynchron Answer On / Off	ANSW0 keine automatischen Antworten ANSW1 automatische Antworten freigeben	ANSW1

Bestimmen des Motortyps

Der Motion Controller ist bei der Auslieferung standardmäßig auf den Motortyp 5 (2444S024B K1155) eingestellt.
Über nachfolgenden Befehl muss der tatsächlich angeschlossene Motortyp eingestellt werden.

Befehl	Funktion	Beschreibung	Beispiel
MOTTYP *)	Motor Type	Vorgabe des Motortyps (Siehe Tabelle)	MOTTYP4
GMOTTYP	Get Motor Type	Motortyp an RS-232 senden	GMOTTYP ➔ 8

Anwählbare Motortypen und deren Parameter

Nr.	Motortyp	P-Anteil (POR)	I-Anteil (I)	Spitzenstrom (mA)	Dauerstrom (mA)
1	1628T012B K1155	3	6	3.000	770
2	1628T024B K1155	4	10	3.000	410
3	2036U012B K1155	5	6	3.000	980
4	2036U024B K1155	6	5	3.000	480
5	2444S024B K1155	4	15	5.000	1.370
6	3056K012B K1155	6	10	7.000	1.940
7	3056K024B K1155	12	12	3.000	930
8	3564K024B K1155	8	20	8.000	2.800

Die Parameter können nachträglich wieder einzeln verändert werden. Mit dem Befehl RN werden die Default-Parameter entsprechend des eingestellten Motortyps gesetzt.

Vorsicht: Mit der Motortypeinstellung werden auch interne Parameter verändert. Deshalb ist es wichtig, beim Wechsel des angeschlossenen Motortyps auch am Motion Controller den Typ über den Befehl MOTTYP anzupassen

Konfigurationen der Drehzahlregelung

Bei der Auslieferung ist der Motion Controller so eingestellt, dass er mit einer Drehzahl läuft, die proportional zur Spannung am Analogeingang ist. Durch entsprechende Konfiguration können auch andere Quellen für den Drehzahlsollwert gewählt werden.

Quelle für den Drehzahlsollwert

Befehl	Funktion	Beschreibung	Beispiel
SOR *)	Source For Velocity	Quelle für Drehzahlvorgabe auswählen SOR 0: Solldrehzahl über RS-232 (V-Befehl) SOR 1: Solldrehzahl über Spannungssignal am Analogeingang SOR 2: Solldrehzahl über Pulsweitensignal am Analogeingang	SOR 1

a.) Sollwert über Spannung am Analogeingang

Im Analog-Drehzahlmodus können weitere Einstellungen vorgenommen werden.

Einstellen des Skalierungsfaktors (Maximaldrehzahl):

Befehl	Funktion	Beschreibung	Beispiel
SP *)	Load Maximum Speed	Neue Maximaldrehzahl laden Argument in rpm (Wertebereich: 0 bis 30.000) Einstellung gilt für alle Modi	SP4000
GSP	Get Maximum Speed	Maximaldrehzahl zurücksenden	GSP ➔ 2500

Beispiel: SP5000 ➔ Die Maximaldrehzahl wird auf 5.000rpm eingestellt. Das heißt, bei 10V am Analogeingang beträgt die Solldrehzahl 5.000rpm.

Default-Einstellung: SP10000

Einstellen der Minimaldrehzahl:

Die kleinste Drehzahl, die bei Anliegen der Startspannung vorgegeben wird, kann eingestellt werden.

Befehl	Funktion	Beschreibung	Beispiel
MV *)	Minimum Velocity	Vorgabe der kleinsten Drehzahl in rpm	MV100
GMV	Get Minimum Velocity	Minimale Drehzahl über RS-232 zurücksetzen	GMV ➔0

Default-Einstellung: MV0

Einstellen der Startspannung:

Befehl	Funktion	Beschreibung	Beispiel
MAV *)	Minimum Analog Voltage	Vorgabe der minimalen Startspannung in mV	MAV500
GMAV	Get Minimum Analog Voltage	Minimalen Startspannungswert über RS-232 zurücksenden	GMAV ➔ 25

Beispiel: MAV100 ➔ Erst bei 100mV am Analogeingang läuft der Antrieb los. Von –99mV bis 99mV steht der Rotor.

Default-Einstellung: MAV25

Vorteil: Da 0mV am Analogeingang üblicherweise schwer einstellbar ist, kann auch 0rpm nicht gut hergestellt werden. Das Totband, das durch die minimale Startspannung entsteht, verhindert ein Anlaufen des Motors bei kleinen Störspannungen (Abbildung 6).

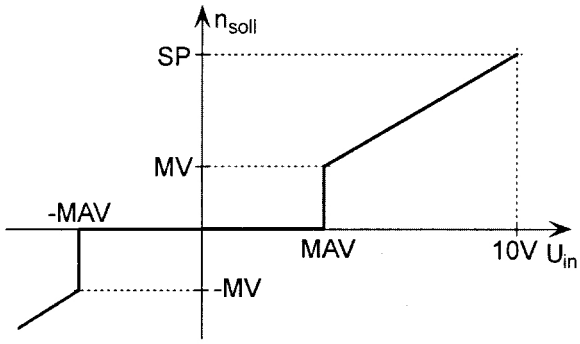

Abbildung 6: Übergangscharakteristik bei analoger Drehzahlvorgabe

Einstellen der Drehrichtung:

Befehl	Funktion	Beschreibung	Beispiel
ADL *)	Analog Direction Left	Positive Spannungen am Analogeingang führen zur Linksdrehung des Rotors	ADL
ADR *)	Analog Direction Right	Positive Spannungen am Analogeingang führen zur Rechtsdrehung des Rotors	ADR

Default-Einstellung: ADR ... Rechtsdrehung bei positiven Spannungen

b.) Sollwert über Pulsweitensignal (PWM) am Analogeingang

Im Auslieferungszustand bedeuten:

- Tastverhältnis > 50% ➔ Rechtsdrehung
- Tastverhältnis = 50% ➔ Stillstand
- Tastverhältnis < 50% ➔ Linksdrehung

Die Befehle SP, MV, MAV, ADL und ADR sind hier ebenfalls anwendbar.

c.) Sollwert über RS-232 Schnittstelle

Befehl	Funktion	Beschreibung	Beispiel
V *)	Select Velocity Mode	Drehzahlmodus aktivieren und mit angegebenen Wert drehen (Drehzahlregelung)	V2000
GV	Get Velocity	Rückmeldung der Solldrehzahl	GV ➔ 500

Unmittelbar nach der Vorgabe des SOR0-Befehls läuft der Motor mit der zu diesem Zeitpunkt eingestellten Drehzahl weiter.

Spezialanwendung:

Motor mit nur einer einzigen Konstantdrehzahl

SOR0 ➔ Auf RS-232 Vorgabe umschalten
V500 ➔ Gewünschte Drehzahl (hier 500rpm) einstellen
EEPSAV ➔ Einstellungen ins EEPROM übernehmen

Mit dieser Konfiguration startet der Motor nach dem Einschalten mit der eingestellten Drehzahl.

Vorgabe einer begrenzten Beschleunigung

Befehl	Funktion	Beschreibung	Beispiel
AC *)	Load Command Acceleration	Neue Beschleunigung laden Argument in U/s^2 (Wertebereich: 0 bis 30.000)	AC100
GAC	Get Acceleration	Beschleunigungswert zurücksenden	GAC ➔ 1000

Default-Einstellung: $30000 U/s^2$

Der Beschleunigungswert bewirkt im Drehzahlreglermodus sanftes Anfahren und Abbremsen.

Drehrichtungseingang

Der Fehlerausgang (Fault-Anschluss) kann zum Drehrichtungseingang umkonfiguriert werden.

Befehl	Funktion	Beschreibung	Beispiel
DIRIN *)	Direction Input	Fehlerausgangs-Anschluss als Drehrichtungseingang umprogrammieren (Auch die Endschalterfunktion wird damit freigegeben)	DIRIN

Pegel und Richtung:
Low: 0 V bis 0,5V ... Linkslauf (entsprechend ADL-Befehl)
High: 4 V bis U_B ... Rechtslauf (entsprechend ADR-Befehl)

Der Pegel am Drehrichtungseingang überstimmt die Einstellungen, die durch ADR und ADL vorgenommen wurden.

Der Drehrichtungseingang wirkt bei jeder Sollwertquelle, d.h. insbesondere auch dann, wenn der Sollwert über die RS-232 vorgegeben wird.

Berücksichtigung der Positionslimits

Mit dem Befehl **APL1** können die Positionslimits (LL-Befehl) berücksichtigt werden. Mit **APL0** werden sie ignoriert (Default-Einstellung).

Einstellen der Reglerparameter

Zur Verbesserung der dynamischen Eigenschaften können die Reglerparameter (Reglerverstärkung, Reglerintegralanteil) erhöht werden. Die Anpassung sollte unbedingt in der praktischen Anwendung erfolgen, da diese erheblichen Einfluss hat.

Befehl	Funktion	Beschreibung	Beispiel
POR *)	Load Proportional Term	Reglerverstärkung laden (Wertebereich 0 - 255)	POR20
I *)	Load Integral Term	Reglerintegralanteil laden (Wertebereich 0 - 255)	I10
GPOR	Get Proportional Term	Reglerverstärkung zurücksenden	GPOR ➔ 8
GI	Get Integral Term	Rückmeldung des Integralterms	GI ➔ 20

Vorgehensweise:

1.) Ausgangskonfiguration setzen:

- Quelle für Drehzahlsollwert: RS-232 ➔ SOR0
- Reglerverstärkung = 8 ➔ POR8
- Integralanteil = 20 ➔ I20
- Drehzahl auf 1/3 der maximalen Anwendungsdrehzahl (Beispiel) ➔ V1000
- Beschleunigung auf höchsten Wert der Anwendung einstellen (Beispiel) ➔ AC10000

2.) Reglerverstärkung erhöhen (Schrittweite 5, später geringer) ➔ POR13

3.) Drehzahlsprung von 1/3 der Maximaldrehzahl auf 2/3 vorgeben (Beispiel) ➔ V2000

4.) Drehzahlsprung von 2/3 auf 1/3 und Verhalten beobachten (Beispiel) ➔ V1000

5.) Schritt 2 bis 4 wiederholen, bis der Regler instabil wird. Danach Reglerverstärkung verringern, bis sichere Stabilität gegeben ist.

6.) Mit Integralanteil entsprechend den Schritten 2 bis 5 verfahren

Positionsregelung

Zum Umschalten vom Drehzahlmodus (Default-Einstellung) auf den Positionierbetrieb ist folgende Befehlssequenz notwendig:

SOR0 ➔ Umschalten auf Drehzahlvorgabe über RS-232
LR0 ➔ Verfahrweg sollte 0 sein
M ➔ Umschalten auf Positionierbetrieb und Start einer Positionierung

Für Positionieraufgaben stehen unter anderem folgende Befehle zur Verfügung:

Befehl	Funktion	Beschreibung	Beispiel
M *)	Initiation Motion	Lageregelung aktivieren und Positionierung starten	M
LA	Load Absolute Position	Neue absolute Sollposition laden Argument: Zahl 1.000 entspricht einer Umdrehung	LA100000
LR	Load Relative Position	Neue relative Sollposition laden	LR5000
SP *)	Load Maximum Speed	Neue Maximaldrehzahl laden Argument in rpm (Wertebereich: 0 bis 30.000)	SP4000
AC *)	Load Command Acceleration	Neue Beschleunigung laden Argument in U/s² (Wertebereich: 0 bis 30.000)	AC100
NP	Notify Position	Rücksenden eines „p" an RS-232, wenn die angegebene Position überfahren wurde (Kein Argument: „p" bei Erreichung der Sollposition)	NP10000 asynch ➔ p
NV	Notify Velocity	Bei Erreichung der vorgegebenen Drehzahl ein „v" an die RS-232 zurücksenden	NV5000 asynch ➔ v

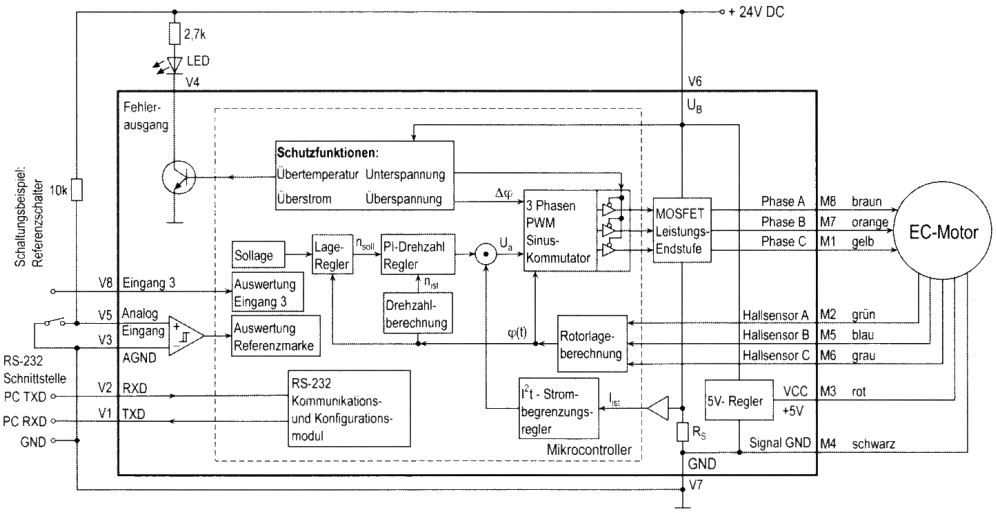

Abbildung 7: Positionsregelung mit Referenzschalter am Analogeingang

Standardablauf einer Positionierung

1.) Laden der Beschleunigung und der Maximaldrehzahl

AC50 ➔ Beschleunigung auf 50 U/s^2 setzen

SP3000 ➔ Maximaldrehzahl auf 3.000 rpm setzen

Diese Werte gelten, bis sie geändert werden oder der Motor abgeschaltet wird.

2.) Sollposition vorgeben

Entweder Variante

a.) **LA40000** ➔ Sollposition auf Absolutposition 40.000 setzen

oder Variante

b.) **LR10000** ➔ Neue Sollposition auf alte Sollposition +10.000 setzen

3.) Positionierung starten

M ➔ Positionierung auf 40.000 bzw. alte Sollposition +10.000 wird gestartet. Der Motor fährt diese Position mit einem Drehzahlprofil entsprechend der eingestellten Beschleunigung und Maximaldrehzahl an (Abbildung 8).

Gemäß Punkt 2.) und Punkt 3.) können verschiedene Positionen hintereinander angefahren werden.

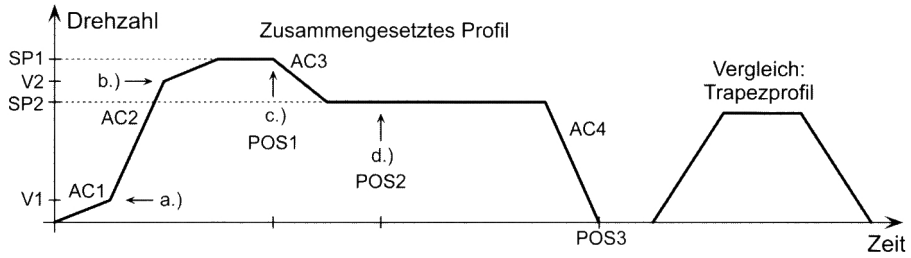

Abbildung 8: Zusammengesetztes Bewegungsprofil im Vergleich zum Trapezprofil

Zusammengesetzte Bewegungsprofile

Durch geschickte Vorgabe neuer Werte (Maximaldrehzahl, Beschleunigung, Endposition), während der Positionierung, können kompliziertere Bewegungsprofile erzeugt werden. Nach einer Werteänderung muss lediglich ein neuer Motionstart (M-Befehl) ausgeführt werden. Die Befehle NP und NV können zur Steuerung des Ablaufs herangezogen werden.

Ablauf (jeweilige Befehlssequenzen nachdem die Notify-Bedingung eingetroffen ist):

Start	a.)	b.)	c.)	d.)
LA[POS3]	AC[AC2]	AC[AC1]	SP[SP2]	AC[AC4]
AC[AC1]	NV[V2]	NP[POS1]	AC[AC3]	NP[POS3]
SP[SP1]	M	M	NP[POS2]	M
NV[V1]			M	
M				

Einstellen der Reglerparameter

Ähnlich wie im Drehzahlreglermodus (siehe vorherige Kapitel) können auch im Positioniermodus die Reglerparameter optimiert werden.

Auswerten von Referenzmarken und Endschalter

Es stehen folgende Eingänge bzw. Signale zur Verfügung:

- Analogeingang
- Fault-Pin als Eingang programmiert
- Eingang Nr. 3
- Nulldurchgang der Hallsensorsignale

Diese Eingänge können über unterschiedliche Methoden ausgewertet werden:

- Direkte Programmierung über HA-, HL- und HN-Befehle
- Programmierbare Referenzfahrtsequenz
- Hard-Blocking Funktion
- Hall-Indexfahrt

Argument in den Endschalterbefehlen:

Die Befehle zur Endschalterprogrammierung (HA, HL, HN, HP, HD und HB) enthalten als Argument eine Zahl, die binär codiert den entsprechenden Endschalter definiert.

Zuordnung:

Endschaltername und Anschluss	Eingangsnummer	Binärstelle	Zahl	Rückmeldezeichen
Analogeingang	1	0	1	h
Fault-Pin	2	1	2	f
Eingang Nr.3	3	2	4	T

Eine „1“ an der korrekten Binärstelle bedeutet üblicherweise eine Aktivierung dieses Eingangs, während eine „0“ den Endschalter deaktiviert.

So bedeuten:

HA0 ... Home Arming an allen Endschaltern deaktivieren

HA1 ... Home Arming am Analogeingang aktivieren, an allen anderen deaktivieren

HA2 ... Home Arming am Fault-Pin aktivieren, an allen anderen deaktivieren

HA3 ... Home Arming am Fault-Pin und am Analogeingang aktivieren

HA4 ... Home Arming am Eingang Nr. 3 aktivieren

HA5 ... Home Arming am Analogeingang und am Eingang Nr. 3 aktivieren

HA6 ... Home Arming am Fault-Pin und am Eingang Nr. 3 aktivieren

HA7 ... Home Arming an allen Endschaltern aktivieren

In den Befehlstabellen wird nur beschrieben, was eine „1“ und was eine „0“ an der entsprechenden Binärstelle bedeutet.

Fault-Pin zum Endschalter umprogrammieren:

Befehl	Funktion	Beschreibung	Beispiel
REFIN *)	Reference Input	Fault-Pin zum Endschaltereingang umprogrammieren	REFIN
ERROUT *)	Error Output	Umschalten auf Fehlerausgangsmodus	ERROUT

Wichtig: Die Endschalterfunktionen für den Fault-Pin werden nur angenommen, wenn REFIN aktiviert ist! (Einstellung mit EEPSAV sichern !)

Festlegen der gültigen Flanke bzw. Polarität:

In Abbildung 6 bzw. Abbildung 9 befinden sich schaltungstechnische Realisierungsbeispiele für den Analogeingang bzw. für den Fault-Pin Endschalter. Je nach dem, wie der Endschalter hardwaremäßig verdrahtet ist, muss die entsprechende Flanke bzw. Polarität festgelegt werden:

Befehl	Funktion	Beschreibung	Beispiel
HP *)	Hard Polarity	Gültige Flanke bzw. Polarität aller Endschalter festlegen: 1: Steigende Flanke bzw. high Pegel gültig 0: Fallende Flanke bzw. low Pegel gültig	HP7

Default-Einstellung: HP7 ... alle Eingänge auf steigende Flanke

Direkte Programmierung über HA-, HL- und HN-Befehle

Über spezielle Befehle kann die Aktion vordefiniert werden, die bei einer Flanke an den Referenzeingängen erfolgen soll.

- Eine programmierte Endschalterfunktion bleibt bestehen, bis die vorgewählte Flanke tatsächlich eintritt
- Über einen erneuten Befehl kann die Programmierung geändert werden, bevor eine Flanke eintritt
- Durch Trennung der Versorgungsspannung wird die Programmierung gelöscht und alle Endschalter sind wieder inaktiv

Befehl	Funktion	Beschreibung	Beispiel
HA	Home Arming	Bei Flanke den Positionswert auf „0“ setzen 1: aktivieren 0: deaktivieren	HA1
HL	Hard Limit	Bei Flanke den Motor stoppen 1: aktivieren 0: deaktivieren	HL3
HN	Hard Notify	Bei Flanken ein Zeichen an RS-232 senden 1: aktivieren 0: deaktivieren	HN2 asynch ➔ f

Zu HL-Befehl:

Drehzahlreglermodus: Der Motor wird beim Eintreten der Flanke mit dem eingestellten Beschleunigungswert abgebremst. Das heißt, er läuft über die Referenzmarke hinaus. Über einen anschließenden Positionierbefehl (Befehl M) kann die Referenzmarke genau angefahren werden. **Vorteil:** keine abrupten Bewegungsänderungen.

Positioniermodus: Bei Eintreten der Flanke positioniert der Motor mit maximaler Beschleunigung auf die Referenzmarke.

Die HA-, HL- und HN-Befehle können gleichzeitig aktiv sein.

Programmierbare Referenzfahrtsequenz

Die programmierbare Referenzfahrtsequenz hat folgende Vorteile:

- Nach der Programmierung kann die Funktion mit einem einzigen Befehl jederzeit aufgerufen werden
- Befindet sich der Antrieb beim Aufruf bereits im Endschalter, so fährt er zuerst aus diesem heraus
- Die Referenzfahrtsequenz kann als erste Aktion nach dem Einschalten gesetzt werden. Dies ermöglicht Positionieranwendungen, die komplett ohne RS-232 auskommen
- Die Referenzfahrt läuft mit einer voreingestellten Drehzahl (HOSP) ab. Die Drehrichtung wird durch das Vorzeichen beim HOSP-Befehl festgelegt

Konfiguration der Referenzfahrt:

1.) Einstellen der Referenzfahrt-Aktion (HA-, HL- und HN-Befehle)

2.) Befehl CAHOSEQ ➔ Capture Homing Sequence ... aktuell eingestellte Referenzfahrt-Aktionen werden in einen Zwischenspeicher geschrieben

3.) Befehl POHOSEQ1 ➔ Power On Homing Sequence ... Homing-Sequence nach dem Einschalten aktivieren (falls gewünscht)

4.) Weitere Konfigurationsbefehle (falls nötig oder gewünscht)

5.) Befehl EEPSAV um die Einstellung in das EEPROM zu übertragen (Werte im Zwischenspeicher werden in das EEPROM übertragen)

Will man die Referenzfahrt-Aktionen ändern, so braucht man einfach nur die obigen Punkte mit den gewünschten Werten wiederholen.

Zum Deaktivieren der Referenzfahrt nach dem Einschalten muss der Befehl POHOSEQ0 vorgegeben werden. Mit EEPSAV wird die Einstellung in das EEPROM übertragen.

Befehl	Funktion	Beschreibung	Beispiel
CAHOSEQ *)	Capture Homing Sequence	Referenzfahrtsequenz in den Zwischenspeicher sichern Die mit den Befehlen HA, HL und HN definierten Aktionen werden zwischengespeichert	CAHOSEQ
POHOSEQ *)	Power On Homing Sequence	Festlegen, ob Referenzfahrtsequenz nach dem Einschalten ausgeführt werden soll POHOSEQ1 ... aktivieren POHOSEQ0 ... deaktivieren	POHOSEQ1
GOHOSEQ	Go Homing Sequence	Referenzfahrtsequenz ausführen Unabhängig vom aktuellen Modus wird eine Referenzfahrt durchgeführt (falls diese programmiert ist)	GOHOSEQ
HOSP *)	Load Homing Speed	Drehzahl für Referenzfahrt laden Wertebereich: -30.000rpm bis 30.000rpm	HOSP100
GHOSP	Get Homing Speed	Drehzahl für Referenzfahrt an RS-232 zurücksenden	GHOSP ➔ 100

Default-Einstellung: HOSP100 ... Rechtsdrehung mit 100 rpm

Hard-Blocking Funktion

Um einen sicheren Schutz gegen das Überfahren der Endschalter zu erreichen, wurde die Möglichkeit geschaffen, die Endschalter in folgender Weise zu programmieren:

- Befindet sich der Antrieb im Endschalter, so wird eine programmierte Drehrichtung gesperrt, d. h. der Antrieb kann sich nur mehr aus dem Endschalter heraus bewegen. Die Drehzahl bleibt auf 0 rpm, auch wenn Solldrehzahlen in die falsche Drehrichtung vorgegeben werden

Befehl	Funktion	Beschreibung	Beispiel
HB *)	Hard Blocking	Hard-Blocking Funktion für entsprechende Endschalter aktivieren oder deaktivieren 1: aktivieren 0. deaktivieren	HB3
HD *)	Hard Direction	Vorgabe der Drehrichtung, die gesperrt wird 1: Rechtslauf sperren 0: Linkslauf sperren	HD2

Der gültige Pegel am Endschalter wird mit dem Befehl „HP“ eingestellt.
Die Hard-Blocking Funktionen können auch gleichzeitig mit den anderen Endschalterfunktionen aktiv sein.

Hall-Indexfahrt

Mit dem Befehl GOHIX kann der Hall-Nullpunkt direkt angefahren werden. Dabei wird auch der Ist-Positionswert auf „0“ gesetzt.

Befehl	Funktion	Beschreibung	Beispiel
GOHIX	Go Hall Index	Auf Hall-Nullpunkt (Hall-Index) fahren und Ist-Positionswert auf „0” setzten	GOHIX

Der Befehl schaltet automatisch auf den CONTMOD.
Die Referenzfahrt wird mit der Homing-Speed (HOSP) durchgeführt.

Erweiterte Betriebsarten

Um die Vielfalt der Anwendungen möglichst gut abzudecken, wurden spezielle Betriebsarten entwickelt. Je nach Anforderung sind auch Modi konfigurierbar, die ohne RS-232 Schnittstelleauskommen. Zur Erhöhung der Auflösung ist ein externer Impulsgeber verwendbar.

Befehl	Funktion	Beschreibung	Beispiel
CONTMOD *)	Continuos Mode	Von einem erweiterten Modus auf Normalbetrieb zurückzuschalten (Hallsensoren als Istwertgeber, Sollwerte für Position über RS-232)	CONTMOD
GMOD	Get Mode	Rücksenden des eingestellten Modus c ... Normalbetrieb s ... Schrittmotor Modus a ... Analoger Positioniermodus e ... Encoder Modus, Istdrehzahl aus Encoder h ... Encoder Modus, Istdrehzahl aus Hallsensoren g ... Elektronisches Getriebe (Gearing-Mode) v ... Spannungssteller-Modus	GMOD ➔ c

Schrittmotor-Modus

Im Schrittmotor-Modus positioniert der Antrieb bei jedem Impuls am Analogeingang umeinen programmierbaren Winkel weiter. Der MMC24/5-10 simuliert somit die Funktionen eines Schrittmotors.

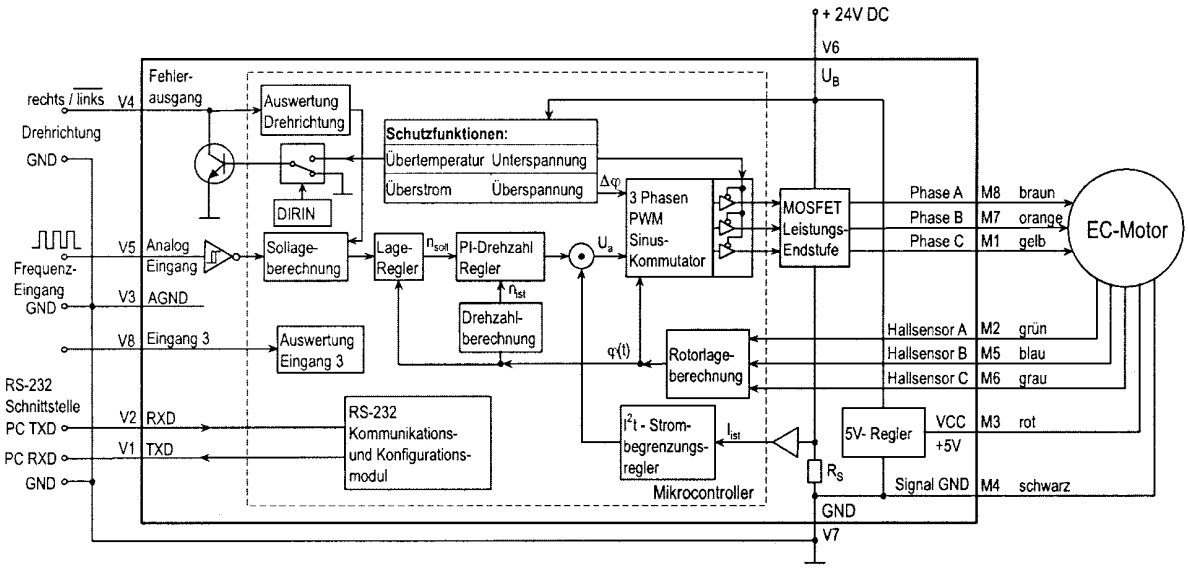

Abbildung 9: Schrittmotorbetrieb mit Drehrichtungseingang

Im Vergleich zu einem echten Schrittmotor kommen einige wesentliche Vorteile hinzu:

- Die Schrittzahl pro Umdrehung ist frei programmierbar und sehr hochauflösend (ca. 3000 Schritte/U)
- Die Schrittweite für einen Schritt ist frei programmierbar
- Kein Rastmoment
- Es ist die volle Dynamik des Bürstenlosmotors nutzbar
- Der Motor ist sehr leise
- Wegen des Istwertgebers gibt es keine Schrittverluste (auch nicht bei höchster Dynamik)
- Im ausgeregelten Zustand (Istposition erreicht) fließt kein Motorstrom
- Hoher Wirkungsgrad

Eingang:

Maximale Eingangsfrequenz: 150kHz
Pegel: low 0 ... 0,5V / high 4V ... 30V

Dieser Modus ermöglicht eine positionstreue Drehzahlregelung, wobei über die Schrittweite und die Schrittzahl beliebige rationale Verhältnisse von der Eingangsfrequenz zur Motordrehzahl eingestellt werden können.

Das Übersetzungsverhältnis kann nach folgender Formel eingestellt werden:

$$UMDREHUNGEN = IMPULSE * \frac{STW}{STN}$$

Umdrehungen ... Umdrehungen, die am Antrieb erzeugt werden

Impulse ... Anzahl der Impulse am Frequenzeingang (= Anzahl der Schritte)

STW ... Step width
(Schrittweitenfaktor ... Anzahl der Schritte pro Impuls am Frequenzeingang)

STN ... Step number
(Schrittanzahl ... Anzahl der Schritte pro Umdrehung)

Wertebereich von STN und STW: 0 bis 65.535

Befehl	Funktion	Beschreibung	Beispiel
STEPMOD *)	Steppermotor Mode	Umschalten auf Schrittmotor Modus	STEPMOD
STW *)	Load Step Width	Schrittweite laden	STW1
STN *)	Load Step Number	Anzahl der Schritte pro Umdrehung laden	STN1000
GSTW	Get Step Width	Rückmeldung der Schrittweite	GSTW ➔ 1
GSTN	Get Step Number	Rückmeldung der Schrittzahl pro Umdrehung	GHOSP ➔ 1000

Default-Einstellungen: STN1000 und STW1

Wenn der Antrieb einmal im Schrittbetrieb konfiguriert ist, benötigt er keine RS-232 Schnittstelle mehr.

Wenn der Drehrichtungseingang nicht programmiert ist, kann die Drehrichtung mit den Befehlen ADR (Rechtsdrehung) und ADL (Linksdrehung) vordefiniert werden.
Für externe Drehrichtungsvorgabe kann der Fault-Pin über den Befehl DIRIN als Drehrichtungseingang programmiert werden.
Im Schrittmotormodus werden auch die **AC**- und **SP**-Werte (siehe Positionsregelung) berücksichtigt. Dadurch kann ein sanftes Anlaufen und Abbremsen erzielt werden.
Mit dem Befehl **APL1** können die Positionslimits (LL-Befehl) berücksichtigt werden. Mit **APL0** werden sie ignoriert (Default-Einstellung).

Elektronisches Getriebe (Gearing-Mode)

Mit dem Gearing-Mode ist es möglich, einen externen Impulsgeber als Sollwertquelle für die Position zu verwenden.

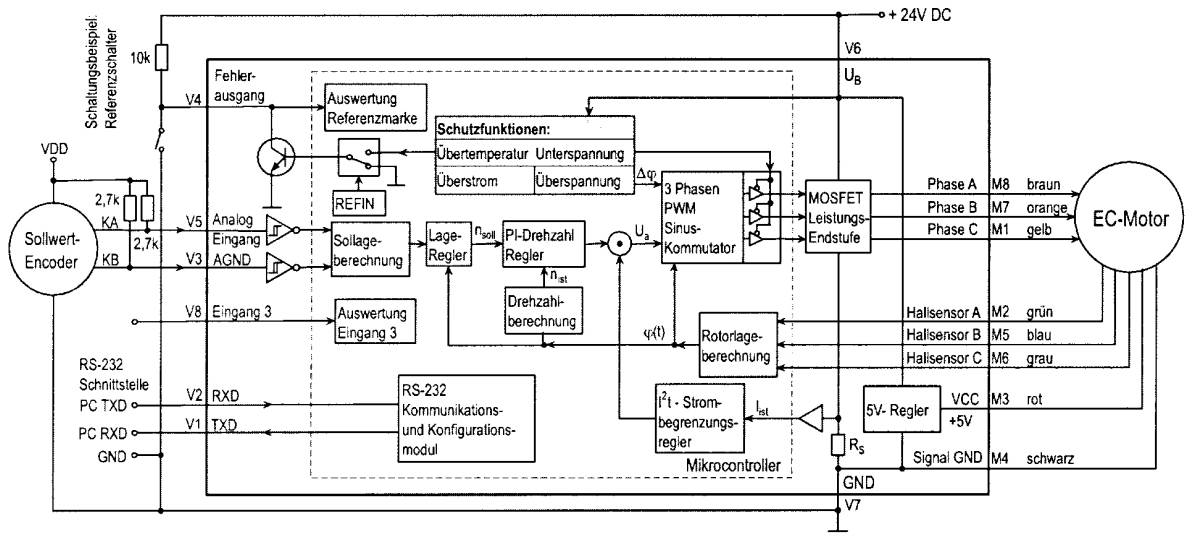

Abbildung 10: Gearing-Mode mit Referenzmarke am Fault-Anschluss

Das Übersetzungsverhältnis kann nach folgender Formel eingestellt werden:

$$UMDREHUNGEN = IMPULSE * \frac{STW}{STN}$$

Umdrehungen ...Umdrehungen, die am Sinusmotor erzeugt werden

Impulse ...tatsächlich gezählte Impulse, die eine Vierflankenauswertung ergibt

STW ...Step width (Schrittweitenfaktor, siehe Schrittmotormodus)

STN ...Step number (Schrittanzahl, siehe Schrittmotormodus)

STN und STW können wie im Schrittmotormodus beschrieben manipuliert werden.

Befehl	Funktion	Beschreibung	Beispiel
GEARMOD *)	Gearing Mode	Umschalten auf Gearing-Modus	GEARMOD

Die Drehrichtung kann mit den Befehlen ADR (Rechtsdrehung) und ADL (Linksdrehung) vordefiniert werden, oder über externes Signal am Fault-Pin (Befehl DIRIN).
Im Gearing-Modus werden auch die **AC**- und **SP**-Werte (siehe Positionsregelung) berücksichtigt.
Dadurch kann ein sanftes Anlaufen und Abbremsen erzielt werden.
Mit dem Befehl **APL1** können die Positionslimits (LL-Befehl) berücksichtigt werden. Mit **APL0** werden sie ignoriert (Default-Einstellung).

Positionsregelung auf Spannung am Analogeingang

Mit der Positionsregelung auf Spannung am Analogeingang kann der Positions-Sollwert über ein Potentiometer oder eine externe Analogspannung vorgegeben werden.

Befehl	Funktion	Beschreibung	Beispiel
APCMOD *)	Analog Position Control Mode	Umschaltung auf Positionsregelung über Analogspannung	APCMOD
LL *)	Load Position Range Limits	Grenzposition laden: Gibt im APCMOD den Positionswert bei 10V an	LL10000

Über den Befehl „LL“ kann die Maximalposition vorgewählt werden. Eine Spannung von 10V entspricht der vorgewählten Maximalposition. Bei –10V positioniert der Antrieb in die entgegengesetzte Richtung.
Unabhängig vom vorgegebenen LL-Wert wird die Maximalposition im APCMOD auf 1.000.000 begrenzt.
Bemerkung: Die Auflösung des Analogeingangs ist auf 10Bit (1.024 Schritte) beschränkt.

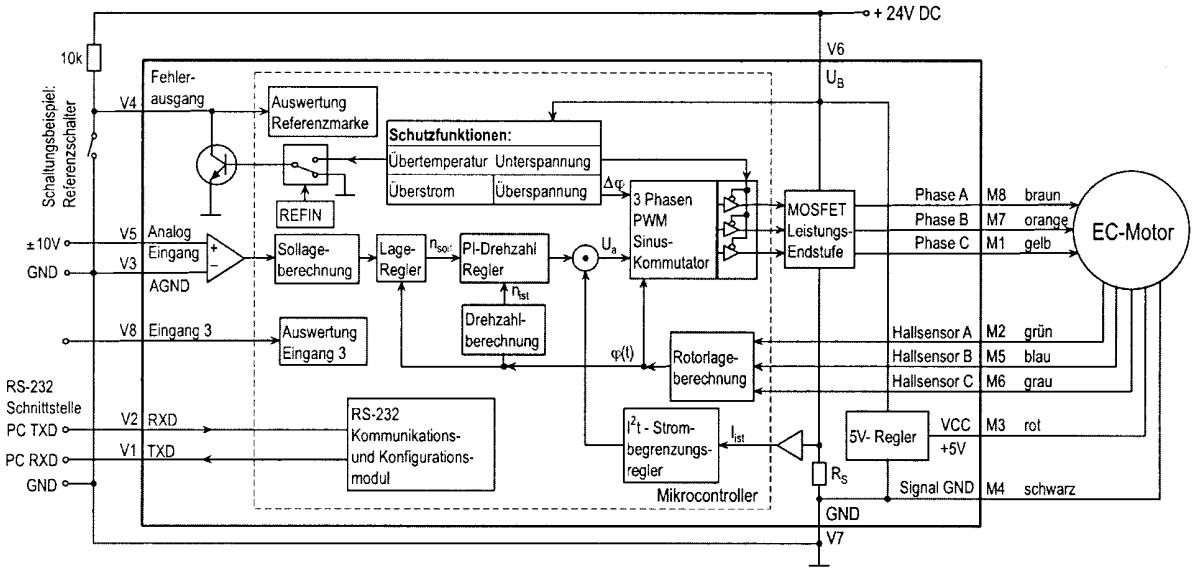

Abbildung 11: Positionsregelung auf analoge Spannung

Über die Befehle ADR und ADL kann die Positionierrichtung vorgegeben werden.

ADR ➔ bei positiven Spannungen auf positive Positionen (Rechtslauf) fahren

ADL ➔ bei positiven Spannungen auf negative Positionen (Linkslauf) fahren

Beim analogen Positioniermodus werden auch die **AC**- und **SP**-Werte (siehe Positionsregelung) berücksichtigt. Dadurch kann ein sanftes Anlaufen und Abbremsen erzielt werden.

Positionierung auf Pulsweitensignal

Sendet man die Befehle **APCMOD** und **SOR2** hintereinander, so wird auf ein Pulsweitensignal (PWM) am Analogeingang positioniert. Das Tastverhältnis bestimmt den Positionssollwert, wobei ein Tastverhältnis von 0,5 (Einschaltzeit = Ausschaltzeit) Solllage = 0 bedeutet. Ein Tastverhältnis größer 0,5 führt zu positiven Solllagewerten, während ein Tastverhältnis kleiner 0,5 zu negativen Solllagewerten führt.

Sonderfunktion:

Wegen der linearen Hallsensoren, ist die absolute Position innerhalb einer Umdrehung erfassbar. Das heißt, auch wenn die Stromversorgung getrennt wird, liefert die Lagebestimmung nach dem Wiedereinschalten den korrekten Positionswert (falls der Rotor nur innerhalb einer Umdrehung verdreht wurde).

Mit den folgenden Befehlen kann erreicht werden, dass der Antrieb im Spannungsbereich 0 V bis 10V genau innerhalb einer Umdrehung positioniert und auch nach dem Abschalten der Versorgung wieder auf die korrekte Position fährt. (Eine Referenzfahrt ist nicht erforderlich !)

ADL ... bei positiven Spannungen auf negative Positionen fahren (nach dem Einschalten steht der Antrieb auf negativen Positionswerten zwischen 0 und –1000)

APCMOD ... auf Analog-Positionierung umschalten

LL1000 ... Maximale Lage auf 1 Umdrehung festlegen

EEPSAV ... Abspeichern der obigen Konfiguration

Externer Impulsgeber als Istwertgeber

Für hochgenaue Anwendungen können die Istwerte von einem externen Impulsgeber abgeleitet werden.

- Die Auflösung der Positionswerte ist in diesem Fall von der Auflösung des Impulsgebers abhängig
- Je nach Anwendung kann die Drehzahl vom Impulsgeber (Abbildung 12) oder von den Hallsensoren abgeleitet werden
- Der externe Impulsgeber kann direkt an der Motorwelle befestigt sein, besonders interessant ist aber ein Impulsgeber der am Abtrieb der Anwendung (z.B. Glasmaßstab) befestigt ist. Dadurch kann die hohe Genauigkeit direkt am Abtrieb eingestellt werden
- Die Kommutierung geschieht weiterhin über die analogen Hallsensoren
- Sämtliche Funktionen der Positionsregelung (zusammengesetzte Profile,...) sind auch hier verfügbar

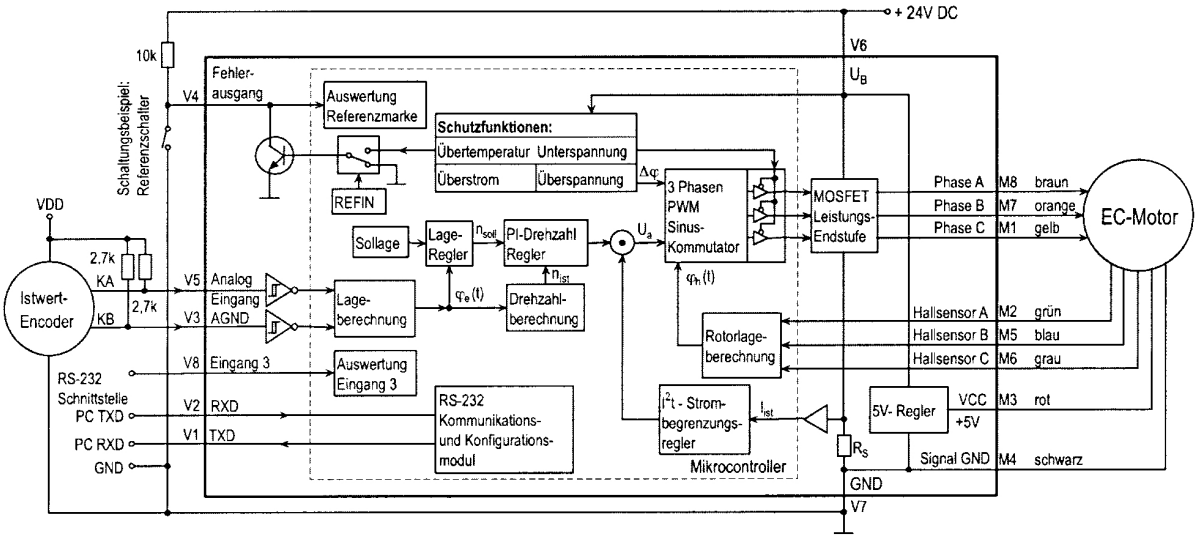

Abbildung 12: Externer Impulsgeber als Positions- und Drehzahlgeber

Die maximale Grenzposition (Wert, der mit dem LL-Befehl vorgegeben wird) umfasst den Wertebereich von 0 bis 2.000.000.000 für die positive bzw. von 0 bis –2.000.000.000 für die negative Grenzposition.

Eingang:
Maximale Eingangsfrequenz: 150kHz
Pegel: low 0 ... 0,5V
high 4V ... 30V

Befehl	Funktion	Beschreibung	Beispiel
ENCMOD *)	Encoder Mode	Umschalten auf Impulsgeber-Modus. Ein externer Impulsgeber dient als Lagegeber (der aktuelle Positionswert wird auf „0“ gesetzt)	ENCMOD
HALLSPEED *)	Hallsensor as Speedsensor	Drehzahl über Hallsensoren	HALLSPEED
ENCSPEED *)	Encoder as Speedsensor	Drehzahl über Encodersignale	ENCSPEED
ENCRES *)	Load Encoder Resolution	Encoderauflösung an den Motor senden. Wertebereich: 0 bis 65.535 (4-fach Imp/U)	ENCRES 2048
GENCRES	Get Encoder Resolution	Eingestellte Impulsgeberauflösung zurücksenden	GENCRES ➔ 2048

Default-Einstellung: ENCRES2048

Wichtig:

Wegen der Vierflankenauswertung muss bei ENCRES immer die vierfache Impulszahl angegeben werden.

Beispiel:
HEDS mit 1024 Impulsen ➔ 4096 Schritte pro Umdrehung ➔ ENCRES4096

Spannungssteller-Modus

Falls der Antrieb als reiner Spannungssteller arbeiten soll, so kann dies mit VOLTMOD konfiguriert werden. Die Motorspannung wird proportional zur Spannung am Analogeingang ausgegeben. Die Strombegrenzung bleibt weiterhin aktiv.

Befehl	Funktion	Beschreibung	Beispiel
VOLTMOD *)	Set Voltage Mode	Spannungssteller-Modus aktivieren	VOLTMOD

Mit diesem Modus ist es möglich einen übergeordneten Regler zu verwenden. Der Controller dient dann als Leistungsverstärker. Man beachte aber, dass die Abtastrate des Stellers bei 1,866 ms liegt und die Auflösung des Analogeingangs 10 Bit beträgt.

Fehlerbehandlung und Fehleranschluss

Die standardmäßigen Fehlerfunktionen und deren Behandlung wird im Kapitel „Technische Informationen“ noch genauer erörtert. Neben diesen Fehlern, die hauptsächlich zum Schutz des Antriebes vor Zerstörung dienen, gibt es noch die Möglichkeit Fehlerbehandlungen zu programmieren, die die Anwendung betreffen.

Abweichung zur Solldrehzahl als Fehler ausgeben

In bestimmten Anwendungen ist es nicht zulässig, dass das vom Antrieb erzeugte Drehzahlprofil vom Sollprofil zu sehr abweicht. Deshalb sollte mit einer Fehlermeldung reagiert werden, wenn dies doch geschieht.

Überschreitet die Abweichung von Istdrehzahl zur Solldrehzahl den vorgegebenen Wert, so wird dies als Fehler ausgegeben. Mit den Befehlen NE, GES und ERI kann der Fehler angezeigt bzw. darauf reagiert werden.

Die Höhe der zulässigen Abweichung kann vorgegeben werden:

Befehl	Funktion	Beschreibung	Beispiel
DEV *)	Load Deviation	Vorgeben der zulässigen Deviationswertes (Betragsmäßige Abweichung der Istdrehzahl zur Solldrehzahl)	DEV500
GDEV	Get Deviation	Rückmelden des Deviationswertes	GDEV ➔ 200

Default-Einstellung: DEV30000 ...die Abweichung kann gar nicht erreicht werden ... dies entspricht einer Deaktivierung des Deviationsfehlers.

Verzögerte Aktivierung eines Fehlers

Mit der Funktion „Delayed Current Error“ kann die Zeit vorgegeben werden, die die Strombegrenzung, die Überspannungsregelung oder der Deviationsfehler aktiv sein muss, bis der Fehler angezeigt wird. Dies ermöglicht es, kurzzeitige Überbelastungen zu ignorieren, wenn zum Beispiel in der Beschleunigungsphase der Strom kurzzeitig in die Begrenzung geht.

Befehl	Funktion	Beschreibung	Beispiel
DCE *)	Delayed Current Error	Verzögerte Fehleraktivierung bei Strombegrenzung, Überspannungsregelung oder Deviationsfehler Argument in 1/100 Sekunden	DCE100
GDCE	Get Delayed Current Error	Gespeicherten Wert für verzögerte Fehleraktivierung an RS232 senden	GDCE ➔ 200

Default-Einstellung: DCE200 ... 2 Sekunden Verzögerung

Beispiel:

DCE400 ➔ Die Verzögerungszeit wird auf 4 Sekunden eingestellt.

Kurzzeitige Strombegrenzungen, wie sie in der Beschleunigungsphase auftreten können, werden ignoriert. Ist der Motor länger in der Strombegrenzung, so wird der Fehler nach 4 Sekunden aktiviert.

Fehleranschluss als Digitalausgang

Neben den Eingangsfunktionen Endschalter (REFIN) und Drehrichtung (DIRIN) die bereits in den vorherigen Kapiteln behandelt wurden, ist eine zusätzliche Ausgangsfunktion möglich.

Der Fehlerausgang kann auch als Digitalausgang konfiguriert werden. Damit ist es zum Beispiel möglich Ventile direkt anzusteuern oder Signale zu senden, wenn ein Ereignis eingetroffen ist.

Befehl	Funktion	Beschreibung	Beispiel
DIGOUT *)	Digital Output	Fehlerausgang auf digitalen Ausgang programmieren Der Ausgang wird auf low Pegel gesetzt	DIGOUT
CO *)	Clear Output	Digitalen Ausgang auf low Pegel setzen	CO
SO *)	Set Output	Digitalen Ausgang auf high Pegel setzen	SO

Der zuletzt gesetzte Pegel am Ausgang wird mit EEPSAV gespeichert.

Fehleranschluss als Impulsausgang

Aus den Hallsensorsignalen werden Impulse abgeleitet und am Fehlerausgang ausgegeben.

Der Impulsausgang ist durch folgende Eigenschaften gekennzeichnet:

- Maximal mögliche Impulszahl: 2.000 Impulse pro Sekunde
- Bei zu hohen Drehzahlen (theoretische Anzahl >2.000) wird die maximale Anzahl ausgegeben
- Die eingestellten Impulse werden genau erreicht. D.h. eine Positionsbestimmung über Impulszählung ist möglich (keine Drift)
- Die ausgegebenen Impulse stimmen zeitlich nicht unbedingt exakt überein (Verzögerungen möglich)
- Da es sich um einen einkanaligen Impulsausgang handelt ist eine Erfassung der Drehrichtung nicht möglich

Befehl	Funktion	Beschreibung	Beispiel
LPN *)	Load Pulse Number	Impulszahl vorgeben (Wertebereich: 1 bis 255)	LPN16
GPN	Get Pulse Number	Rückmeldung der eingestellten Impulszahl	GPN ➔ 16
ERROUT *)	Error Output	Umschalten auf Fehlerausgangsmodus	ERROUT
ENCOUT *)	Encoder Output	Umschalten auf Impulsausgangsmodus	ENCOUT

Default-Einstellungen: 16 Impulse pro Umdrehung (LPN16) Fehlerausgangsmodus (ERROUT)

Beispiel:

LPN64 ➔ 64 Impulse pro Umdrehung

ENCOUT ➔ Impulsausgangsmodus

Bei 1.800rpm ergeben sich: 1.800 / 60 * 64 = 1.920 Impulse pro Sekunde. Bei 3.000rpm ergäbe sich: 3.000 / 60 * 64 = 3.200 Impulse pro Sekunde. Es werden jedoch nur ca. 2.000 Impulse pro Sekunde ausgegeben und es entsteht eine Drift. Eine korrekte Positionszuordnung ist nicht mehr möglich.

Speichern und Ausführen von Fahrprogrammen

Für Stand-Alone Lösungen oder für teilweise autonome Abläufe ist es möglich, Fahrprogramme an den Antrieb zu senden. Diese Programme können mit einem Standard-Text-Editor (Windows-Editor, Word,...) geschrieben werden und dann mit dem Terminalprogramm übertragen werden. Im Controller werden die Programme direkt in das EEPROM gespeichert.

Wichtig: Im Terminalprogramm muss die Handshake-Steuerung „XON/XOFF“ eingestellt sein. Für das Rücksenden der Daten (GPROGSEQ) ist es sinnvoll, am Terminalprogramm "CR beim Empfang" zu aktivieren.

Hinweis: Die Motion Manager Software für Windows 95/98/NT bietet eine komfortable Möglichkeit, um Fahrprogramme zu erstellen und zu übertragen.

Folgende Befehle stehen für Fahrprogramme zur Verfügung:

Befehl	Funktion	Beschreibung	Beispiel
PROGSEQ	Program Sequence	Definiert den Anfang eines Fahrprogramms. Alle nachfolgenden Befehle werden direkt in das EEPROM übertragen (währende der Übertragung Versorgung nicht unterbrechen !!). Ein „End“ markiert das Ende der Übertragung. Alle Befehle nach dem END-Befehl werden wieder sofort ausgeführt	PROGSEQ *LA1000* *NP1000* *M* END
GPROGSEQ	Get Program Sequence	Die gespeicherte Programmsequenz wird an den PC zurückgesendet	GPROGSEQ ➔ <Prog>
ENPROG *)	Enable Program	Die Ausführung des Programms wird freigegeben, d.h. der Ablauf startet. Mit EEPSAV kann dieser Zustand fix gespeichert werden, so dass der Antrieb nach dem Einschalten sofort mit dem Programmablauf startet	ENPROG
DIPROG *)	Disable Program	Ausführung des Programms deaktivieren	DIPROG

Während des Ablaufs eines Programms können weiterhin auch Befehle über die RS-232 gesendet werden.
In Fahrprogrammen sind fast alle ASCII-Befehle verwendbar.

Steuerung von Ablaufprogrammen

Um Programme zu steuern gibt es eine Reihe von zusätzlichen Befehlen, die nur innerhalb von Programmen sinnvoll sind und daher nur dort zur Verfügung stehen.

Bei folgenden Befehlen wird der Ablauf gestoppt bis eine Bedingung erfüllt ist:

NP	... Notify Position	... der Ablauf stoppt am nächsten „M“-Befehl bis die entsprechende Position erreicht ist
HN	... Hard Notify	... der Ablauf stoppt am „GOHOSEQ“-Befehl oder am nächsten „M“- bzw. „V“-Befehl bis der Endschalter überfahren wird
NV	... Notify Velocity	... der Ablauf stoppt am nächsten „M“- oder „V“-Befehl bis die entsprechende Drehzahl erreicht ist
GOHIX	... Go Hall Index	... der Ablauf stoppt am „GOHIX“-Befehl bis die Hall-Null-Position erreicht ist.

Bei mehreren Notify-Bedingungen bewirkt die erste erfüllte Bedingung eine Programmfortsetzung.

Zusatzbefehle (die meisten sind nur innerhalb von Fahrprogrammen verwendbar):

Befehl	Funktion	Beschreibung	Beispiel
DELAY	Delay	Ablauf: eine definierte Zeit anhalten Argument: in 1/100 Sekunden Wertebereich: 0 bis 65.535	DELAY100
TIMEOUT	Timeout	Bei Notify-Befehlen nur die vorgegebene Zeit warten und dann den Ablauf wieder fortfahren. An RS-232 ein „o“, falls Notify-Bedingung nicht erfüllt Der Befehl ist auch von der RS-232 aus verwendbar Argument: in 1/100 Sekunden Werte bereich: 0 bis 65.535	TIMEOUT asynch ➔ o
JMP	Jump	Sprung zur angegebenen Adresse Der Befehl ist auch von der RS-232 aus verwendbar	JMP5
JPH	Jump if Hard-Input activated	Sprung zur angegebenen Adresse, wenn der Analog-Eingang aktiv ist (HP bestimmt die Polarität)	JPH3
JPF	Jump if Fault-Input activated	Sprung zur angegebenen Adresse, wenn der Fault-Pin-Eingang aktiv ist (HP bestimmt die Polarität) Fault-Pin muss als Eingang konfiguriert sein (REFIN)	JPF4
JPT	Jump if Third Input activated	Sprung zur angegebenen Adresse, wenn der 3.Eingang aktiv ist (HP bestimmt die Polarität)	JPT37
SETx	Set Variable x	Variable x (A,B,C) auf den angegebenen Wert setzen, Wertebereich: 0 bis 65.535	SETA10
GETx	Get Variable x	Inhalt der Variable „x” (A,B,C) an RS-232 senden	GETA ➔ 123
DxJNZ	Decrement x, Jump if not Zero	Verringere den Wert der Variable „x” (A,B,C) um 1 und springe, falls der Wert nicht „0“ ist	DBJNZ8
ERI	Error Interrupt	Ab Ausführung dieses Befehls wird ein Fehlerinterrupt aktiviert. Das heißt, wenn irgendwann danach ein Fehler auftritt (Überspannung, Strombegrenzung, ...), dann verzweigt der Ablauf zur angegebenen Adresse. Der Fehlerbehandlungsmodus wird beendet, wenn ein JMP- oder RETI-Befehl ausgeführt wird	ERI5
RETI	Return Error Interrupt	Rücksprung aus der Fehlerbehandlungsroutine, WICHTIG: der unterbrochene Befehl wird nicht mehr fortgeführt, auch wenn er zum Zeitpunkt der Unterbrechung noch nicht beendet war!	RETI
DIERI	Disable Error Interrupt	Der ERI-Befehl wird deaktiviert, d.h. bei einem Fehler wird nicht verzweigt	DIERI
CALL	Call Subroutine	Aufruf eines Unterprogramms	CALL17
RET	Return from Subroutine	Rücksprung aus einem Unterprogramm Es ist zu beachten, dass nur eine Unterprogrammebene möglich ist, d.h. in Unterprogrammen dürfen keine Unterprogramme aufgerufen werden!	RET
A	Define Address	Definition der aktuellen Position als Einsprungadresse für Sprungbefehle, Wertebereich: 0 bis 63	A37

Erläuterungen zu Befehlen und Funktionen

Zu Sprungbefehlen

Mit den Sprungbefehlen ist es möglich, den Programmablauf gezielt zu steuern.
Der **JMP**-Befehl kann auch von der RS-232 aus verwendet werden. Dies ist dann interessant, wenn verschiedene Programme vom Rechner aus aufgerufen werden sollten.

Beispiel:

```
PROGSEQ A1
JMP1        ... Endlosschleife
A2          ... Programmsequenz 2 ... kann nur durch JMP2 von der RS-232 aus
                aufgerufen werden
LA10000
NP
M
JMP1        ... Rücksprung auf die Endlosschleife
A3          ... Programmsequenz 3 ... kann nur durch JMP3 von der RS-232 aus
                aufgerufen werden
LA-10000
NP
M
JMP1        ... Rücksprung auf die Endlosschleife
END
```

Die Programmsequenzen nach A2 bzw. nach A3 können nur durch einen JMP2- bzw. JMP3-Befehl von der RS-232 aus, aufgerufen werden. Ein JMP2 von der RS-232 aus führt hier dazu, dass der Antrieb auf die Lage 10000 fährt und dort stehen bleibt.
Die **DxJNZ**-Befehle dienen zur Bildung von Schleifen mit vordefinierter Anzahl von Durchläufen.
Die Befehle **JPH**, **JPF** und **JPT** ermöglichen Sprünge, die nur ausgeführt werden, wenn der entsprechende Eingang aktiv ist. Dadurch können Programme über externe Schalter aufgerufen werden.
Zur Definition von Einsprungadressen dient der **A-Befehl**. Bei einem Sprung wird der Ablauf an dieser Stelle fortgesetzt.
Der Wertebereich bei Sprungbefehlen geht von 0 bis 63. Entsprechend können mit JMP, JPx, ERI und CALL maximal 64 verschiedene Einsprungpunkte definiert werden.

Zum ERI-Befehl

Bei der Ausführung des ERI-Befehls passiert vorerst nichts. Erst wenn danach eine Fehlersituation eintritt, springt der Ablauf sofort auf die angegebene Adresse.

Dadurch kann im Fehlerfall eine sinnvolle Fortsetzung des Programms erzielt werden.

Durch den RETI-Befehl ist ein Rücksprung auf die Position möglich, an der der Ablauf unterbrochen wurde. Es ist dabei zu beachten, dass der unterbrochene Befehl nicht mehr ausgeführt wird, sondern beim darauffolgenden Befehl fortgesetzt wird.

Innerhalb der Fehlerbehandlungsroutine kann keine erneute Fehlerunterbrechung stattfinden. Der Fehlerbehandlungsstatus wird aufgehoben, sobald der RETI- oder der JMP-Befehl ausgeführt wird. Ab dann werden die Befehle wieder unterbrochen, wenn ein Fehler auftritt. Deshalb sollte in der Fehlerbehandlungsroutine dafür gesorgt werden, dass die Fehlersituation verschwindet. Ansonsten kommt es zum wiederholten Aufruf der Fehlerbehandlung.

Zu Referenzfahrten

Über den HN-Befehl ist es möglich, den Ablauf anzuhalten bis der Endschalter erreicht ist. Um den GOHOSEQ-Befehl innerhalb eines Ablaufs korrekt auszuführen ist es unbedingt erforderlich, bei der Definition der Referenzfahrt-Sequenz den HN-Befehl entsprechend zu setzen.

Dies ist insbesondere auch nötig, wenn man die Einschalt-Referenzfahrt verwenden möchte (POHOSEQ1).

Zu Notify-Befehlen

Über die Notify-Befehle ist es unter anderem möglich, kompliziertere Bewegungsprofile zu erzeugen.

Beispiel:

```
LA100000
SP5000
AC50
NV1000
M AC100
NV2000
M AC50
NP
M
```

Mit dieser Sequenz wird während des Hochlaufs bei 1.000rpm die Beschleunigung erhöht. Bei 2.000rpm wird sie wieder verringert.

Bemerkung: Der NP-Befehl ohne Argument stoppt den Ablauf bis die Sollposition erreicht ist.

Zum CALL-Befehl

Der CALL-Befehl ermöglicht Unterprogramme, die von unterschiedlichen Stellen aus und beliebig oft aufgerufen werden können.

Nur mit dem RET-Befehl kann aus einem Unterprogramm wieder zurückgesprungen werden. Innerhalb eines Unterprogramms sind alle Befehle erlaubt, außer ein erneuter CALL-Befehl.

Allgemeines

Wird ein Ablaufprogramm vollständig abgearbeitet (kein Sprung am Ende eines Programms), so wird ein „n“ an die RS-232 gesendet, falls ANSW1 eingestellt ist.

Um ein Endlosprogramm (für Stand Alone Betrieb sinnvoll) zu erzeugen, ist ein Sprungbefehl am Ende des Programms erforderlich.

Speichergröße: 7936 Bytes

Technische Information

Sinuskommutierung

Der MMC24/5-10 zeichnet sich durch eine sogenannte Sinuskommutierung aus. Dies bedeutet, dass das vorgegebene Drehfeld immer ideal zum Rotor steht. Dadurch gelingt es, Momentenschwankungen auf ein Minimum zu reduzieren, auch dann, wenn die Drehzahlen sehr klein sind. Außerdem läuft der Motor dadurch besonders leise.

Stromregler und I^2t-Strombegrenzung

Der MMC24/5-10 ist mit einem Stromregler ausgerüstet, der es erlaubt, eine Momentenbegrenzung zu realisieren.

Es können zwei Stromwerte programmiert werden:

1.) Spitzenstrom

ASCII-Befehl: LPC8000 ➔ Spitzenstrom auf 8.000mA einstellen

Auf den Spitzenstrom wird begrenzt, solange das thermische Strommodell eine unkritische Temperatur errechnet.

2.) Dauerstrom

ASCII-Befehl: LCC2800 ➔ Dauerstrom auf 2.800mA einstellen

Erreicht das thermische Strommodell eine kritische Temperatur, so wird auf den Dauerstrom umgeschaltet.

Arbeitsweise des Stromreglers

Beim Start des Motors wird dem Stromregler der Spitzenstrom als Sollwert vorgegeben. Mit zunehmender Belastung wird der Strom im Motor immer höher, bis er schließlich den Spitzenstrom erreicht. Ab dann tritt der Stromregler in Kraft und begrenzt auf diesen Stromsollwert.

Parallel dazu läuft ein thermisches Strommodell, das aus dem aktuell fließenden Strom eine Modelltemperatur berechnet. Übersteigt diese Modelltemperatur einen kritischen Wert, so wird auf den Dauerstrom umgeschaltet und der Motorstrom auf diesen geregelt. Erst wenn die Belastung so gering wird, dass die kritische Modelltemperatur unterschritten wird, wird wieder der Spitzenstrom zugelassen.

Das Ziel dieser sogenannten I^2t-Strombegrenzung ist, den Motor bei geeigneter Wahl des Dauerstroms nicht über die thermisch zulässige Temperatur zu erhitzen. Andererseits sollte kurzzeitig eine hohe Belastung möglich sein, um sehr dynamische Bewegungen realisieren zu können (Abbildung 13)

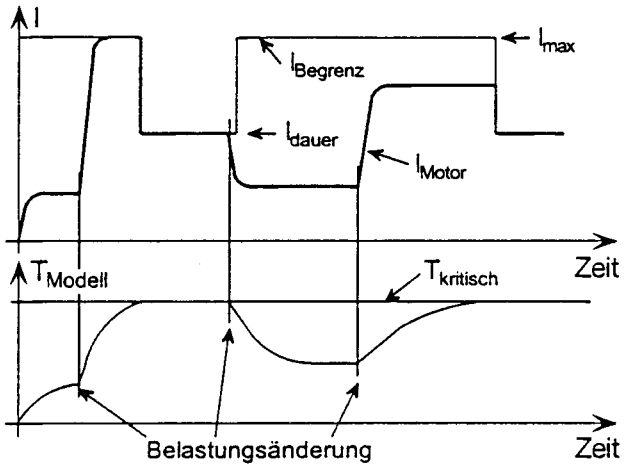

Abbildung 13: Funktion der I²t-Strombegrenzung

Analoge Sollstromvorgabe

- Mit dem Befehl SOR3 kann auf analoge Sollstromvorgabe geschaltet werden. Der Begrenzungsstrom ist proportional zur Spannung am Analogeingang. Die I^2t-Strombegrenzung ist nicht mehr aktiv.
- Der eingestellte Strom wird mit dem Maximalstrom (LPC-Wert) gewichtet. Bei 10V am Analogeingang wird dementsprechend auf den Maximalstrom begrenzt.
- Die Solldrehzahl kann mit dem V-Befehl vorgegeben werden. Mit dem M-Befehl können Positionierungen vorgenommen werden. Erweiterte Betriebsarten sind hier nicht möglich, da diese üblicherweise den Analogeingang verwenden.
- Negative Spannungen am Analogeingang begrenzen den Sollstrom ebenfalls proportional zum Betrag der Spannung. Die Drehrichtung bleibt gleich, wie bei positiven Spannungen.

Übertemperatursicherung

Überschreitet die MOSFET-Temperatur einen vorgegebenen Grenzwert, so wird die Leistungsendstufe des Motion Controllers abgeschaltet. Um die Leistungsendstufe wieder zu aktivieren, müssen folgende Bedingungen erfüllt sein:

- Temperatur unterhalb eines vorgegebenen Grenzwertes
- Solldrehzahl auf 0rpm eingestellt
- Tatsächliche Motordrehzahl kleiner 50rpm

Bemerkung zur Bestimmung der MOSFET-Temperatur:

Es wird die Gehäusetemperatur des Motion Controllers gemessen und über die Strommessung auf die Verlustleistung in den Power-MOSFETs geschlossen. Über ein thermisches Modell wird aus diesen Größen die Temperatur an den MOSFETs berechnet. In den meisten Anwendungsfällen stellt diese Methode einen thermischen Schutz des MMC24/5-10 dar.

Unterspannungsüberwachung

Unterschreitet die Versorgungsspannung 10V, so wird die Leistungsendstufe des Motion Controllers abgeschaltet und der Fehlerausgang aktiviert (LED leuchtet nicht mehr). Die Elektronik bleibt aber noch aktiv. Bei Überschreitung von 11V wird der Controller wieder eingeschaltet.

Überspannungsregelung

Wird der Motor generatorisch angetrieben, so erzeugt er Energie. Üblicherweise sind Netzgeräte nicht in der Lage, diese Energie in das Netz zurückzuspeisen. Aus diesem Grund steigt die Versorgungsspannung am Motion Controller und je nach Drehzahl kann es zur Überschreitung der zulässigen Höchstspannung kommen.

Um eine Zerstörung von Bauteilen zu vermeiden, enthält der MMC24/5-10 einen Regler, der beim Überschreiten einer Grenzspannung (32V) den Polradwinkel des Motors verstellt. Dadurch wird die erzeugte Energie im Motor verbraucht und die Spannung der Elektronik bleibt auf 32V begrenzt.

Diese Methode schützt den Motion Controller bei generatorischem Betrieb des Motors.

Anhang

Elektromagnetische Verträglichkeit (EMV)

Der Motion Controller MMC24/5-10 wurde in Übereinstimmung mit der EMV-Richtlinie 89/336/EWG über die Einhaltung von EMV Schutzanforderungen gemessen und geprüft.

Im Nennbetrieb erfüllt das funktionierende System die Anforderungen folgender Normen:

EMV Emissionen in den Grenzen von VDE 0839 Teil 81-2 (EN50081-2)

EMV Immunität gemäß VDE 0839 Teil 82-2 (EN50082-2) geprüft für:

- Elektrostatische Entladungen ESD mit 4kV (Kontaktentladung) bzw. 8kV (Luftentladung) nach VDE 0847 Teil 4-2 (EN61000-4-2)
- HF-Felder nach VDE 0847 Teil 4-3 (EN61000-4-3)
- Schnelle Transienten nach VDE 0847 Teil 4-4 (EN61000-4-4)
- Leitungsgeführte HF-Ströme nach VDE 0847 Teil 4-6 (EN61000-4-6)

Für die Einhaltung der Anforderungen sind folgende Bedingungen zu erfüllen:

- Betrieb entsprechend den technischen Daten und der Bedienungsanleitung
- Die Versorgungsleitung muss, möglichst nahe an der Steuerung, mit einer Windung durch ein geeignetes Ferritrohr z.B. Kitagawa R1-16-28-7 gesteckt werden
- Abschirmung der Motorspeisekabel (bei Überschreitung von 30cm Leitungslänge)
- Niederohmiger (kleine Impedanz!) Anschluss von Motor und Motion Controller an Masse (Funktionserde)

Unterstützende Maßnahmen für leitungsgebundene Störungen

Die Störungen auf der DC-Anschlussleitung halten die Grenzwerte ein, die nach oben genannter Norm (VDE 0839 Teil 81-2) für die Netzwechselstromanschlüsse vorgeschrieben sind.

RS-232 Multiplexerplatine

Mit Hilfe der sogenannten RS-232 Multiplexerplatine ist es möglich, mehrere Motion Controller von einer einzigen RS-232 Schnittstelle aus zu betreiben. Diese Platine enthält einen Zufuhr-SUB-D Stecker zum Anschluss an den PC und einen Abgangs-SUB-D Stecker zur Verbindung an die nächste Einheit. Mit einer Multiplexerplatine können somit zwei Antriebe angeschlossen werden.

Die erste Abgangsseite kann wieder mit dem Zufuhr-SUB-D Stecker einer weiteren Multiplexerplatine verbunden werden. Mit dieser Verkettung können theoretisch bis zu 255 Antriebe an einer RS-232 Schnittstelle angeschlossen werden. Es können insbesondere auch integrierte Einheiten (3564K024B C) mit externen Controllern (MMC24/5-10) kombiniert werden.

Es ist zu beachten, dass mit dieser Betriebsart eine klare Master-Slave-Struktur eingehalten werden muss. D.h. der PC gibt die Befehle vor und die Antriebe dürfen nur auf Anfrage antworten. Deshalb müssen die asynchronen Antworten mit dem Befehl ANSW0 unterbunden werden.

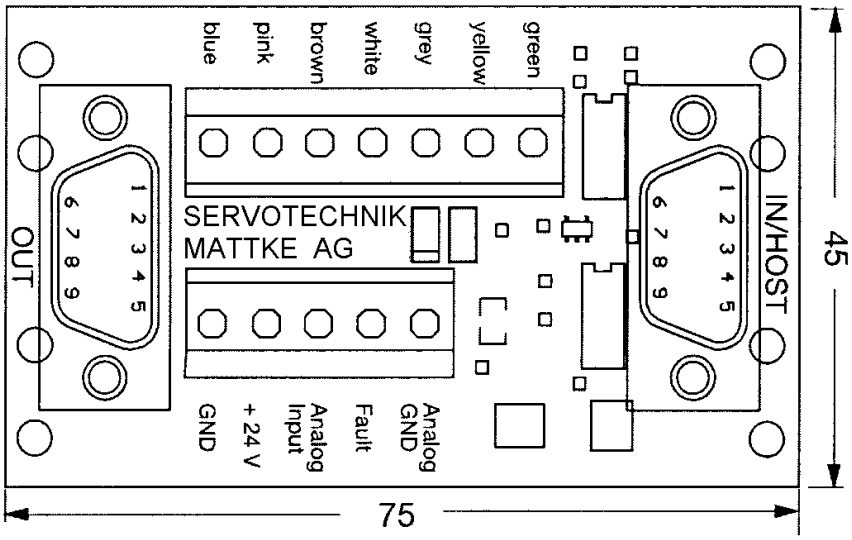

Abbildung 14: RS-232 Multiplexerplatine

Der ASCII-Befehlssatz

Erläuterungen zu den Befehlstabellen

- Einstellungen von Befehlen die mit „*)“ gekennzeichnet sind, werden bei **EEPSAV** gespeichert.
- Bei Rückmeldebefehlen wird mit einem Pfeil „➔“ ein Beispiel der Rückantwort angegeben.
- Bei Befehlen, die erst nach Eintreffen eines Ereignisses antworten, ist die Kennzeichnung „asynch“ davor gestellt.

Befehle zur Grundeinstellung

Befehl	Funktion	Beschreibung	Beispiel
DI *)	Disable Drive	Antrieb deaktivieren (nur Leistungsendstufe)	DI
EN *)	Enable Drive	Antrieb aktivieren (nur Leistungsendstufe)	EN
BAUD *)	Select BAUD Rate	Baudrate für RS-232 Schnittstelle vorgeben	BAUD96000
NODEADR *)	Define Node Address	Knotenadresse vorgeben (0 bis 255)	NODEADR5
GNODEADR	Get Node Address	Knotenadresse an Schnittstelle senden	GNODEADR ➔ 5
MOTTYP *)	Motor Type	Vorgabe des Motortyps (Siehe Tabelle)	MOTTYP4
GMOTTYP	Get Motor Type	Motortyp an RS-232 senden	GMOTTYP ➔ 8
ANSW *)	Asynchron Answer On / Off	ANSW0 ... keine automatischen Antworten ANSW1 ... automatische Antworten freigeben	ANSW1
LL *)	Load Position Range Limits	Grenzpositionen laden (aus diesen Limits wird im Positioniermodus nicht hinausgefahren, positive Werte geben das obere Limit an und negative das untere)	LL2000000 LL-5000000
APL *)	Activate (Deactivate) Position Limits	APL1: Positionslimits auch im Drehzahlregler und im Schrittmotormodus aktivieren APL0: Limits deaktivieren (default)	APL1
GPL	Get Positive Limit	Positive Grenzposition an RS-232 senden	GPL ➔ 60000000
GNL	Get Negative Limit	Negative Grenzposition an RS-232 senden	GNL ➔ -30000000
RN *)	Reset Node	Parameter auf ursprüngliche Werte (ROM-Werte) setzen (Strom, Beschleunigung, Reglerparameter, Maximaldrehzahl, Grenzpositionen, ...)	RN
EEPSAV	Save To EEPROM	Aktuelle Parameter und Konfiguration in EEPROM speichern. Auch beim nächsten Einschalten läuft der Antrieb mit diesen Einstellungen an	EEPSAV
FCONFIG	Factory Configuration	Sämtliche Konfigurationen und Werte werden auf den Auslieferungszustand zurückgesetzt. Der Antrieb wird nach diesem Befehl deaktiviert. Erst durch erneutem Anschluss der Versorgung wird der Antrieb wieder aktiviert (Mit den ROM-Werten)	FCONFIG

Befehle zur Bewegungssteuerung

Befehl	Funktion	Beschreibung	Beispiel
M *)	Initiate Motion	Lageregelung aktivieren und Positionierung starten	M
LA	Load Absolute Position	Neue absolute Position laden Argument: Zahl 1.000 entspricht einer Umdrehung	LA100000
LR	Load Relative Position	Neue relative Position laden	LR5000
V *)	Select Velocity Mode	Drehzahlmodus aktivieren und mit angegebenem Wert drehen (Drehzahlregelung)	V2000
GV	Get Velocity	Rückmeldung der Solldrehzahl	GV ➔ 500
NP	Notify Position	Rücksenden eine „p" an RS-232, wenn die angegebene Position überfahren wurde. (Kein Argument: „p" bei Erreichung der Sollposition)	NP10000 asynch ➔ p
NV	Notify Velocity	Bei Erreichung der vorgegebenen Drehzahl ein „v" an die RS-232 zurücksenden	NV5000 asynch ➔ v
NPOFF	Notify Position Off	Einen noch nicht ausgelösten Notify Position-Befehl wieder deaktivieren	NPOFF
NVOFF	Notify Velocity Off	Einen noch nicht ausgelösten Notify Velocity-Befehl wider deaktivieren	NVOFF

Befehle zur Einstellung der Drehzahlregelung

Befehl	Funktion	Beschreibung	Beispiel
MV *)	Minimum Velocity	Vorgabe der kleinsten Drehzahl in rpm	MV100
GMV	Get Minimum Velocity	Minimale Drehzahl über RS-232 zurücksenden	GMV ➔ 0
MAV *)	Minimum Analog Voltage	Vorgabe der minimalen Startspannung in mV	MAV500
GMAV	Get Minimum Analog Voltage	Minimale Startspannungswert über RS-232 zurücksenden	GMAV ➔ 25
ADL *)	Analog Direction Left	Positive Spannungen am Analogeingang führen zur Linksdrehung des Rotors	ADL
ADR *)	Analog Direction Right	Positive Spannungen am Analogeingang führen zur Rechtsdrehung des Rotors	ADR
SOR *)	Source For Velocity	Quelle für Drehzahlvorgabe auswählen SOR 0: Solldrehzahl über RS-232 (V-Befehl) SOR 1: Solldrehzahl über Spannungssignal am Analogeingang SOR 2: Solldrehzahl über Pulsweitensignal am Analogeingang SOR 3: Sollbegrenzungsstrom über Spannungssignal am Analogeingang (Solldrehzahl wird über RS-232, d.h. V-Befehl vorgegeben)	SOR1

Befehle zur Auswertung von Referenzmarken und Endschaltern

Befehl	Funktion	Beschreibung	Beispiel
HO	Define Home-Position	Kein Argument: Ist-Positionswert auf „0“ setzen Sonst: Ist-Positionswert auf angegeben Zahl setzen	HO
HP	Hard Polarity	Gültige Flanke bzw. Polarität aller Endschalter festlegen: 1**): Steigende Flanken bzw. high Pegel gültig 0**): Fallende Flanken bzw. low Pegel gültig	HP3
HA	Home Arming	Bei Flanke den Positionswert auf „0“ setzen 1**): aktivieren 0**): deaktivieren	HA1
HL	Hard Limit	Bei Flanke den Motor stoppen 1**): aktivieren 0**): deaktivieren	HL3
HN	Hard Notify	Bei Flanke ein Zeichen an RS-232 senden 1**): aktivieren 0**): deaktivieren	HN2 asynch ➔ f
CAHOSE *)	Capture Homing Sequence	Referenzfahrtsequenz in den Zwischenspeicher sichern. Die mit den Befehlen HA, HL und HN definierten Aktionen werden zwischengespeichert	CAHOSEQ
POHOSEQ *)	Power On Homing Sequence	Festlegen, ob Referenzfahrtsequenz nach dem Einschalten ausgeführt werden soll POHOSEQ1 ... aktivieren POHOSEQ0 ... deaktivieren	POHOSEQ1
GOHOSEQ	Get Homing Sequence	Referenzfahrtsequenz ausführen Unabhängig vom aktuellen Modus wird eine Referenzfahrt durchgeführt (falls diese programmiert ist)	GOHOSEQ
HOSP *)	Load Homing Speed	Drehzahl für Referenzfahrt laden Wertebereich: -30.000rpm bis 30.000rpm	HOSP100
GHOSP	Get Homing Speed	Drehzahl für Referenzfahrt an RS-232 zurücksenden	GHOSP ➔ 100
HB	Hard Blocking	Hard-Blocking Funktion für entsprechenden Endschalter aktivieren oder deaktivieren 1**): aktivieren 0**): deaktivieren	HB3
HD	Hard Direction	Vorgabe der Drehrichtung, die gesperrt wird 1**): Rechtslauf sperren 0**): Linkslauf sperren	HD2
GOHIX	Go Hall Index	Auf Hall-Nullpunkt (Hall-Index) fahren und Ist-Positionswert auf „0” setzen	GOHIX
HS	Hard Status	Eine Zahl zwischen 0 und 7 an die RS-232 senden, die angibt welcher/welche Endschalter schon geschalten hat/haben (binärcodiert)	HS ➔ 3
GAHS	Get Actual Homing Status	5 Ziffern zwischen 0 und 7 an die RS-232 senden, die die aktuelle Einstellungen der Referenzschalter anzeigt: 1.Ziffer: HA-Wert 2.Ziffer: HL-Wert 3.Ziffer: HN-Wert 4.Ziffer: HB-Wert 5.Ziffer: HD-Wert	GAHS ➔ 33300
GHSC	Get Homing Sequence Configuration	3 Ziffern zwischen 0 und 7 an die RS-232 senden, die die Einstellung der Homing-Sequence anzeigt: 1.Ziffer: HA-Wert 2.Ziffer: HL-Wert 3.Ziffer: HN-Wert	GHSC ➔ 220

**) „0“ oder „1“ an der dem Endschalter zugeordneten Binärstelle (siehe Beschreibung der Endschalter).

Befehle für aktuelle Parameter

Befehl	Funktion	Beschreibung	Beispiel
SP *)	Load Maximum Speed	Neue Maximaldrehzahl laden Argument in rpm (Wertebereich: 0 bis 30.000) Einstellung gilt für alle Modi	SP4000
GSP	Get Maximum Speed	Maximaldrehzahl zurücksenden	GSP ➔ 10000
AC *)	Load Command Acceleration	Neue Beschleunigung laden	AC100
GAC	Get Acceleration	Beschleunigungswert zurücksenden	GAC ➔ 500
POR *)	Load Proportional Term	Reglerverstärkung laden (Wertebereich 0 – 255)	POR8
GPOR	Get Proportional Term	Reglerverstärkung zurücksenden	GPOR ➔ 8
I *)	Load Integral Term	Reglerintegralanteil laden (Wertebereich 0 – 255)	I20
GI	Get Integral Term	Rückmeldung der Integralterms	GI ➔ 20
LPC *)	Load Peak Current Limit	Spitzenstrom laden Wertebereich: 0 bis 12.000mA	LPC8000
GPC	Get Peak Current	Spitzenstrom an RS-232 senden	GPC ➔ 8000
LCC *)	Load Continuous Current Limit	Dauerstrom laden Wertebereich: 0 bis 12.000mA	LCC2800
GCC	Get continuous Current	Dauerstrom an RS-232 senden	GCC ➔ 2800

Befehle für erweiterte Betriebsarten

Befehl	Funktion	Beschreibung	Beispiel
CONTMOD *)	Continuous Mode	Von einem erweitertem Modus auf Normalbetrieb zurückschalten (Hallsensoren als Istwertgeber, Sollwerte für Lage über RS-232)	CONTMOD
STEPMOD *)	Steppermotor Mode	Umschalten auf Schrittmotor Modus	STEPMOD
STW *)	Load Step Width	Schrittweite laden	STW1
GSTW	Get Step Width	Rückmeldung der Schrittweite	GSTW ➔ 1
STN *)	Load Step Number	Anzahl der Schritte pro Umdrehung laden	STN1000
GSTN	Get Step Number	Rückmeldung der Schrittzahl pro Umdrehung	GSTN ➔ 1000
GEARMOD *)	Gearing Mode	Umschalten auf Gearing-Modus	GEARMOD
APCMOD *)	Analog Position Control Mode	Umschalten auf Positionsregelung über Analogspannung	APCMOD
ENCMOD *)	Encoder Mode	Umschalten auf Impulsgeber-Modus Ein externer Impulsgeber dient als Lagegeber (Der aktuelle Positionswert wird auf „0“ gesetzt)	ENCMOD
HALLSPEED *)	Hallsensor as Speedsensor	Drehzahl über Hallsensoren	HALLSPEED
ENCSPEED *)	Encoder as Speedsensor	Drehzahl über Encodersignale	ENCSPEED
ENCRES *)	Load Encoder Resolution	Encoderauflösung an den Motor senden Wertebereich: 0 bis 65.535 (4-fach Imp/U)	ENCRES512
GENCRES	Get Encoder Resolution	Eingestellte Impulsgeberauflösung zurücksenden	GENCRES ➔ 2048
VOLTMOD *)	Set Voltage Mode	Spannungssteller-Modus aktivieren	VOLTMOD
GMOD	Get Mode	Rücksenden des eingestellten Modus c ... Normalbetrieb s ... Schrittmotor Modus a ... Analoger Positioniermodus e ... Encoder Modus, Istdrehzahl aus Encoder h ... Encoder Modus, Istdrehzahl aus Hallsensoren g ... Elektronisches Getriebe (Gearing Mode) v ... Spannungssteller-Modus	GMOD ➔ c

Befehle zur Konfiguration von Fehlerfunktionen und des Fehleranschlusses

Befehl	Funktion	Beschreibung	Beispiel
DIRIN *)	Direction Input	Fehlerausgangs-Anschluss als Drehrichtungs-eingang umprogrammieren (Auch die End-schalterfunktion wird damit freigegeben)	DIRIN
REFIN *)	Reference Input	Fault-Pin zum Endschaltereingang umpro-grammieren	REFIN
ERROUT *)	Error Output	Umschalten auf Fehlerausgangsmodus	ERROUT
ENCOUT *)	Encoder Output	Umschalten auf Impulsausgangsmodus	ENCOUT
LPN *)	Load Pulse Number	Impulszahl vorgeben (Wertebereich: 1 bis 255)	LPN16
GPN	Get Pulse Number	Rückmeldung der eingestellten Impulszahl	GPN ➔ 16
DIGOUT *)	Digital Output	Fehlerausgang auf digitalen Ausgang pro-grammieren Der Ausgang wird auf low Pegel gesetzt	DIGOUT
CO *)	Clear Output	Digitalen Ausgang auf low Pegel setzen	CO
SO *)	Set Output	Digitalen Ausgang auf high Pegel setzen	SO
DCE *)	Delayed Cur-rent Error	Verzögerte Fehleraktivierung bei Strombe-grenzung, Überspannungsregelung oder Devi-ationsfehler Argument in 1/100 Sekunden	DCE100
GDCE	Get Delayed Current Error	Gespeicherten Wert für verzögerten Fehleraus-gang an RS-232 senden	GDCE ➔ 200
		Vorgeben des zulässigen Deviationswertes (Betragsmäßige Abweichung der Istdrehzahl zur Solldrehzahl)	
GDEV	Get Deviation	Rückmelden des Deviationswertes	GDEV ➔ 200

Befehle für Fahrprogramme

Befehl	Funktion	Beschreibung	Beispiel
PROGSEQ	Program Sequence	Definiert den Anfang eines Fahrprogramms Alle nachfolgenden Befehle werden direkt in das EEPROM übertragen (während der Über-tragung Versorgung nicht unterbrechen !!) Ein „END“ markiert das Ende der Übertragung Alle Befehle nach dem END-Befehl werden wieder sofort ausgeführt	PROGSEQ *LA1000* *NP1000* *M* END
GPROGSEQ	Get Program Se-quence	Die gespeicherte Programmsequenz wird an den PC zurückgesendet	GPROGSEQ ➔ <Prog>
ENPROG *)	Enable Program	Die Ausführung des Programms wird freigege-ben, d.h. der Ablauf startet	ENPROG
DIPROG *)	Disable Program	Ausführung des Programms deaktivieren	DIPROG

Hinweis:
Weitere Befehle für Fahrprogramme können direkt im entsprechenden Kapitel nach-gelesen werden.

Weitere Rückmeldebefehle

Befehl	Funktion	Beschreibung	Beispiel
POS	Get Actual Position	Aktuelle Position zurücksenden	POS ➔ 500000
GN	Get N	Rückmeldung der Istdrehzahl	GN ➔ 4000
GCL	Get Current Limit	Rückmeldung des aktuellen Begrenzungsstroms	GCL ➔ 2800
GCR	Get Real Current	Aktuellen Iststrom zurücksenden	GCR ➔ 2000
TEM	Get Temperature	Aktuelle Gehäusetemperatur zurücksenden (in °C)	TEM ➔ 35
VER	Get Version	Version der aktuellen Software rückmelden	VER ➔ ...3050008..
NE	Notify Error	An RS-232 wird signalisiert, wenn ein Fehler auftritt NE1: Im Fehlerfall ein „r“ zurücksenden NE0: Keine Signalisierung eines Fehlers	NE1 asynch ➔ r
GST	Get Status	Rückmeldung des aktuellen Status (7 Bits) Von links nach rechts: Bit 0: 1 ... Lageregler aktiv 0 ... Drehzahlregler aktiv Bit 1: 1 ... Drehzahlvorgabe analog oder pwm 0 ... Drehzahlvorgabe über RS-232 Bit 2: 1 ... Drehzahlvorgabe über pwm (Bit1 = 1) 0 ... Drehzahlvorgabe analog (Bit1 = 1) Bit 3: 1 ... Antrieb ist aktiv (enabled) 0 ... Antrieb ist inaktiv (disabled) Bit 4: 1 ... Sollposition wurde erreicht 0 ... Sollposition wurde noch nicht erreicht Bit 5: 1 ... Positive Endschalterflanke gültig 0 ... Negative Endschalterflanke gültig Bit 6: 1 ... Endschalter auf high Pegel 0 ... Endschalter auf low Pegel	GST ➔ 0101011
GFS	Get Fault Status	Status des Fehlerausgangs rücksenden (4 Bits), 0 ... kein Fehler, 1 ... Fehler Von links nach rechts: Bit 0: Übertemperatursicherung Bit 1: Strombegrenzungsregler Bit 2: Unterspannungsüberwachung Bit 3: Überspannungsregler	GFS ➔ 0000
GAST	Get Actual Status	4 Bits über aktuellen Status an RS-232 senden Von links nach rechts: Bit 0: 1 ... Endschalter 2 auf high Pegel 0 ... Endschalter 2 auf low Pegel Bit 1: 1 ... Endschalter 3 auf high Pegel 0 ... Endschalter 3 auf low Pegel Bit 2: 1 ... Rechtslauf bei positiven Werten 0 ... Linkslauf bei positiven Werten Bit 3: 1 ... Referenzfahrt nach Einschalten noch aktiv 0 ... Referenzfahrt nach Einschalten beendet	GAST ➔ 1100

Fortsetzung Rückmeldebefehle

Befehl	Funktion	Beschreibung	Beispiel
GSCS	Get Special Configuration Set	8 Konfigurations-Bits an RS-232 senden Von links nach rechts: Bit 0: 1 ... Referenzfahrt nach Einschalten freigegeben 0 ... Referenzfahrt nach Einschalten gesperrt Bit 1: 1 ... Fault-Pin ist Eingang 0 ... Fault-Pin ist Ausgang Bit 2: 1 ... Impulsausgang am Fault-Pin (Bit 1 = 0) 0 ... Fehlersignal am Fault-Pin (Bit 1 = 0) Bit 3: 1 ... Bit 1 = 1: Fault-Pin ist Drehrichtungseingang Bit 1 = 0: Fault-Pin ist Digitalausgang 0 ... Bit 1 = 1: Fault-Pin ist Endschaltereingang 2 Bit 1 = 0: Fault-Pin ist kein Digitalausgang Bit 4: 1 ... Positive Flanke an Endschalter 2 gültig 0 ... Negative Flanke an Endschalter 2 gültig Bit 5: 1 ... Positive Flanke an Endschalter 3 gültig 0 ... Negative Flanke an Endschalter 3 gültig Bit 6: 1 ... Fahrprogramm freigegeben 0 ... Fahrprogramm gesperrt Bit 7: 1 ... Automatische Antworten freigegeben 0 ... Keine automatischen Antworten	GSCS ➔ 00000001
GES	Get Enhanced Status	5 Statusbits an RS-232 senden Von links nach rechts Bit 0: 0 Bit 1: 0 Bit 2: 1... Analoge Sollstromvorgabe aktiv 0... Keine analoge Sollstromvorgabe Bit 3: 1... Positionslimits in allen Modi aktiv 0... Positionslimits nur in der Positionsregelung Bit 4: 1... Deviationsfehler ist gegeben 0... Kein Deviationsfehler vorhanden	GES ➔ 00001

Konfigurations- und Fahrprogrammbeispiele

Bei den folgenden Beispielen wird immer von den Standardeinstellungen, d.h. vom Auslieferungszustand, ausgegangen.

Drehzahlregelung:
Sollwert durch ein PWM-Signal am Analogeingang

Ziele:

- Drehzahlregelung über PWM-Signal am Analogeingang
- Beschleunigung auf den Wert 500 U/s^2 begrenzen
- Skalierung so einstellen, dass eine Maximaldrehzahl von 5000rpm möglich ist
- Die Einstellung der Reglerparameter hat ergeben, dass der Proportionalterm 30 und der Integralterm 15 sein sollte

Umsetzung über ASCII-Befehle:

SOR2
AC500
SP5000
POR30
I15
EEPSAV ➔ Konfigurationseinstellungen im EEPROM abspeichern

Drehzahlregelung:
Sollwert über die RS-232 Schnittstelle

Ziele:

- Drehzahlregelung über RS-232
- Zwei Endschalter (high Pegel aktiv) dürfen nicht überfahren werden
- Spitzenstrom auf 3A und Dauerstrom auf 1A

Umsetzung über ASCII-Befehle:

SOR0
REFIN ➔ Fehler-Anschluss als Endschaltereingang definieren
HP3 ➔ Bei beiden Endschaltern high Pegel aktiv setzen
HB3 ➔ Bei beiden Endschaltern Hard-Blocking Modus
HD1 ➔ Bei Analog-Endschalter Rechtslauf sperren, bei Fehler-Anschluss-Endschalter wird Linkslauf gesperrt
LPC3000
LCC1000
V0 ➔ Umschalten auf Drehzahlreglermodus und mit 0rpm drehen
EEPSAV ➔ Konfigurationseinstellungen im EEPROM abspeichern

Positionsregelung

Ziele:

- Positionsregelung
- Beschleunigung auf den Wert 300 U/s^2 begrenzen
- Fallende Flanke bei Referenzmarke
- Reglerparameter: Proportionalanteil auf 25, Integralanteil auf 8 (Ergebnis der Parameterbestimmung)

Umsetzung über ASCII-Befehle:

AC300

HP0

POR25

I8

SOR0	➔	Auf Vorgabe über RS-232 schalten
LA0	➔	Neue Position laden
M	➔	Positioniermodus aktivieren und positionieren
EEPSAV	➔	Konfigurationseinstellungen im EEPROM abspeichern
Bemerkung:		Die Positionierung sollte erst am Ende durchgeführt werden, damit der Antrieb auch mit den gewünschten Parametern läuft.

Schrittmotorbetrieb ohne RS-232 Schnittstelle

Ziele:

- Schrittmotorbetrieb
- Schrittweite: 3
- Schrittanzahl pro Umdrehung: 557
- Referenzfahrt nach dem Einschalten mit Fehler-Anschluss als Referenzeingang
- Fallende Flanke am Referenzeingang
- Linkslauf bei Referenzfahrt mit 300rpm

Umsetzung über ASCII-Befehle:

STEPMOD

STW3

STN557

REFIN	➔	Fehleranschluss als Referenzeingang festlegen
HA2	➔	Bei Flanke an Endschalter 2 Position „0“ setzen
HL2	➔	Bei Flanke an Endschalter 2 stoppen
CAHOSEQ	➔	Programmierte Referenzfahrt in Zwischenspeicher
POHOSEQ1	➔	Referenzfahrt nach Einschalten durchführen
HP0	➔	Fallende Flanke an allen Referenzeingängen
HOSP-300	➔	Negative Drehzahlwerte definieren den Linkslauf
EEPSAV	➔	Konfigurationseinstellungen im EEPROM abspeichern
Bemerkung:		Die RS-232 Schnittstelle ist nach der Konfiguration nicht mehr nötig

Fahrprogramm:
Positionierroutine über RS-232 aufgerufen

Das Programm ermöglicht den Aufruf verschiedener Routinen von der RS-232 Schnittstelle aus:

- Homing: Zuerst auf einen Endschalter fahren und dann den Hallsensor-Nullpunkt (Hallindex). Dies ermöglicht eine Referenzfahrt mit hoher Wiederkehrgenauigkeit
- Routine 1: Auf Position „0“ fahren und dort stehen bleiben
- Routine 2: Mit geringer Strombegrenzung versuchen eine Position anzufahren. Da in der Anwendung ein Hindernis im Weg sein kann, wird die Sollposition möglicherweise nicht erreicht. Nach 5 Sekunden sollte der Motor auf jeden Fall gestoppt werden. (Die weitere Auswertung geschieht in der übergeordneten Steuerung)
- Routine 3: 1.000 Zyklen von folgendem Ablauf: 10 Umdrehungen vorwärts, 1 Sekunde Pause, 5 Umdrehungen wieder rückwärts und dann 0,5 Sekunden Pause

Konfiguration:

SOR0	➔	Auf RS-232 Drehzahlvorgabe schalten
LR0	➔	Keine Bewegung
M	➔	Auf Positionsregelung schalten (Bewegung „0“)
HA1		
HN1	➔	Ablauf soll stehen bleiben bis Flanke gekommen ist
HL1	➔	Analog-Eingang als Endschalter
CAHOSEQ		Homing Sequenz in den Zwischenspeicher
HOSP200	➔	Homing Drehzahl laden
HP1	➔	Steigende Flanke am Endschalter gültig
ENPROG	➔	Fahrprogramm nach dem Einschalten starten
ANSW0	➔	Keine asynchronen Antworten
EEPSAV	➔	Konfiguration speichern

Programm:

A1		
JMP1	➔	Endlosschleife
A2	➔	Einsprungpunkt für Homing-Sequenz
GOHOSEQ	➔	Homing auf Referenzschalter
GOHIX	➔	Homing auf Hallsensor-Nullpunkt (Hallindex)
JMP1	➔	Rücksprung auf Endlosschleife

A3	➔	Einsprungpunkt für Routine 1
LA0	➔	Sollposition auf „0“ setzen
NP	➔	Notify auf Sollposition ... Ablauf bleibt stehen bis Sollposition erreicht ist
M	➔	Positionierung starten
JMP1	➔	Rücksprung auf Endlosschleife
A4	➔	Einsprungpunkt für Routine 2
LPC500	➔	Strombegrenzungswerte auf 500mA einstellen ... Dauerstrom ≤ Spitzenstrom
LA1000000		
NP		
TIMEOUT500	➔	Nach 5 Sekunden Ablauf fortsetzen, auch wenn Position noch nicht erreicht wurde
M	➔	Positionierung starten
V0	➔	Motor stoppen
LR0		
M	➔	Wieder auf Positioniermodus schalten
JMP1	➔	Rücksprung auf Endlosschleife
A5	➔	Einsprungpunkt für Routine 3
SETA1000	➔	Variable „A“ vordefinieren
A6	➔	Einsprungpunkt für Schleife
LR10000		
NP		
M		
DELAY100		
LR-5000		
NP		
M		
DELAY50		
DAJNZ6	➔	Schleife 1.000 mal wiederholen
JMP1	➔	Rücksprung auf Endlosschleife

Bemerkungen:

- Durch Senden des Befehls JMP2 von der RS-232 aus, wird die Routine „Homing“ aufgerufen. Für die anderen Routinen gilt ähnliches (JMP3 ➔ Routine 1, ...)
- Der Befehl NP (Notify Position) ohne Argument legt fest, dass der Ablauf am Befehl „M“ stehen bleibt, bis die Sollposition erreicht ist

Fahrprogramm:
Ablauf über Digitaleingang gesteuert (keine RS-232)

- Nach dem Einschalten soll zuerst der Endschalter und dann der Hallindex angefahren werden
- Bei positiven Flanken am Digitaleingang (Fehler-Anschluss umprogrammiert) sollte 5 Umdrehungen vorwärts gefahren werden (stehen bleiben, falls Pegel nachher „low“)
- Falls der Pegel nach den 5 Umdrehungen auf „high“ steht, dann soll der Antrieb auf „0“ fahren

Konfiguration:

SOR0	➔	Auf RS-232 Drehzahlvorgabe schalten
LR0	➔	Keine Bewegung
M	➔	Auf Positionsregelung schalten (Bewegung „0“)
REFIN	➔	Fehlerausgang zum Eingang umprogrammieren
HA1		
HN1	➔	Ablauf soll stehen bleiben bis Flanke gekommen ist
HL1	➔	Analog-Eingang als Endschalter
CAHOSEQ	➔	Homing-Sequence in den Zwischenspeicher
POHOSEQ1	➔	Nach dem Einschalten Homing-Sequence ausführen
HOSP-200	➔	Homing Drehzahl laden (rückwärts)
HP1	➔	Steigende Flanke am Endschalter gültig
ENPROG	➔	Fahrprogramm nach dem Einschalten starten
ANSW0	➔	Keine asynchronen Antworten
EEPSAV	➔	Konfiguration speichern

Programm:

GOHIX	➔	Homing auf Hallsensor-Nullpunkt (Hallindex)
A1		
HP3	➔	high Pegel an Eingang 2 (Fault-Pin-Eingang) gültig
A2		
JPF2	➔	Endlosschleife bis low Pegel an Eingang 2
HP1	➔	low Pegel an Eingang 2 (Fault-Pin-Eingang) gültig
A3		
JPF3	➔	Endlosschleife bis high Pegel an Eingang 2 ... Auswertung der positiven Flanke
LR5000		
NP		
M	➔	5 Umdrehungen vorwärts fahren
DELAY50	➔	0,5 Sekunden warten bis Eingang 2 abgefragt wird
JPF1	➔	Bei low Pegel am Eingang 2 wieder auf Anfang springen
LA0		
NP0		
M	➔	Auf Position „0“ fahren, falls high Pegel am Eingang 2
JMP1	➔	Auf Anfang springen

Bemerkungen:

- Bei diesem Programm ist eine RS-232 Schnittstelle nicht mehr erforderlich (Stand Alone Lösung)
- Mit kurzen Impulsen am Eingang (z.B. Taster) wird der gewünschte Ablauf gestartet und mit einem Dauersignal (z.B. Schalter) wird der Rücklauf ausgelöst

Konfiguration im Auslieferungszustand

Die Konfiguration im Auslieferungszustand entspricht folgende Befehlen:

EN	➔	Antrieb ist aktiv
V0	➔	Drehzahlsollwert auf „0“
SOR1	➔	Drehzahlquelle ist Analogeingang
ADR	➔	Bei positiven Analogspannungen Rechtslauf
HP7	➔	Für alle Endschalter gilt steigende Flanke
ERROUT	➔	Fehler-Anschluss auf Fehlerausgang
CONTMOD	➔	Normalmodus (Drehzahlregelung mit Hallsensoren)
HALLSPEED	➔	Hallsensoren als Drehzahlgeber
HB0	➔	Kein Hard-Blocking
HD0	➔	Für alle Endschalter Linkslauf sperren (falls Hard-Blocking aktiv)
HA0		
HL0		
CAHOSEQ	➔	Homing-Sequenz deaktivieren
POHOSEQ0	➔	Keine Homing-Sequenz nach dem Einschalten
HOSP100	➔	Homing-Speed auf Rechtslauf mit 100rpm
ANSW1	➔	Asynchrone Antwort freigeben
DIPROG	➔	Kein Start des Fahrprogramms nach dem Einschalten
BAUD9600	➔	Baudrate auf 9.600 BAUD
NODEADR0	➔	Knotenadresse auf „0“ setzen
MOTTYP5	➔	Motor auf 2444S024B K1155 setzen (entsprechend auch Parameter)
RN	➔	Entspricht folgenden Befehlen: LPC5000 LCC1370 AC30000 I15 POR4 SP10000 MV0 MAV25 LL600000000 LL-600000000 DCE200 DEV30000 LPN16 STW1 STN1000 ENCRES2048

Bemerkungen:

Mit dem Befehl FCONFIG werden (mit Ausnahme von MOTTYP8, LPC8000, LCC2800, I20 und POR8) genau diese Einstellungen vorgenommen.

Notizen

Indexverzeichnis

A

Ablaufprogrammen 39
Analogeingang.....8, 11, 17, 18, 31, 56
Anhang 46
Anschlussbelegung.....6
Anschlüsse.....8
ASCII-Befehle.....13
ASCII-Befehlssatz.....48
Asynchrone Antworten 16
Auslieferungszustand 63

B

Baudrate 15
Befehlstabellen 48
Beschleunigung.....19
Bestellhinweise.....7
Betriebsarten.....28, 52
Bewegungsprofile.....22
Bewegungssteuerung.....49

D

Dauerstrom 43
Digitalausgang.....36
Drehmomentenregler.....4
Drehrichtung.....18
Drehrichtungseingang 19
Drehzahlprofile 4
Drehzahlregelung.....4, 11, 17, 49, 56
Drehzahlreglermodus 25

E

Einsatzgebiete 5
Elektromagnetische Verträglichkeit.....46
Elektronisches Getriebe.....30
Endschalter 23, 24
Endschaltern 50
Erläuterungen zu Befehlen 41
Erweiterte Betriebsarten.....4

F

Fahrprogramm 59
Fahrprogrammbeispiele.....56
Fahrprogramme..... 53
Fahrprogrammen..... 38, 40
Fehleranschluss..... 35, 36
Fehleranschlusses 53
Fehlerausgang 4, 8
Fehlerbehandlung 35
Fehlerfunktionen 53
Funktionen 41

G

Gearing-Mode..... 30
Grundeinstellung..... 48
Grundeinstellungen..... 10

H

Hall-Indexfahrt 27
Hard-Blocking 26

I

I^2t-Strombegrenzung..... 43
Impulsausgang 36
Impulsgeber 10, 32
Istwertgeber 32

K

Knotenadresse..... 15
Konfiguration..... 5, 53, 63
Konfigurationen..... 17
Konfigurations 56
Konfigurationseinstellungen..... 14

M

Minimaldrehzahl..... 17
Motortyps..... 16

N

Notify-Befehlen..... 42

O

Optionen 5

P

Parameter 5, 16, 51
Positionierbetrieb 4
Positioniermodus 25
Positionierroutine 59
Positionierung 22, 32
Positionslimits 20
Positionsregelung 21, 31
Programmierung 5, 24

R

Referenzfahrten 42
Referenzfahrtsequenz 25
Referenzmarken 23, 50
Reglerparameter 20, 22
RS-232 ..... 19, 56, 58, 59
RS-232 Anschlüsse 8
RS-232 Multiplexerplatine 47
RS-232 Schnittstelle 4, 13
Rückmeldebefehle 54, 55

S

Schrittmotorbetrieb 58
Schrittmotor-Modus 28
Sinuskommutierung 43
Skalierungsfaktors 17
Solldrehzahl 35
Sollstromvorgabe 44
Sollwert 18, 56
Sollwerteingang 4
Sollwertvorgabe 11
Spannungssteller-Modus 34
Speichern 38
Spitzenstrom 43
Startspannung 18
Störungen 46
Stromregler 10, 43
Stromreglers 43
Stromversorgungsanschlüsse 8

T

Technische Daten 7
Technische Information 43

Ü

Überspannungsregelung 45
Übertemperatursicherung 44

U

Unterspannungsüberwachung 45

Z

Zusatzbefehle 40