BETRIEBSANLEITUNG
USER MANUAL

PRÄZISION IN PERFEKTION
PRECISION IN PERFEKTION

Universal-Positioniersteuerkarte
Universal Position Control Card

PS 30

9013.0179 / 13.07.2017

OWIS GmbH
Im Gaisgraben 7
79219 Staufen i. Br. (Germany)
Tel. +49 (0) 76 33 / 95 04-0
Fax +49 (0) 76 33 / 95 04-440
info@owis.eu
www.owis.eu
mobile.owis.eu

Inhalt

Content

1. Allgemeines

Die OWIS® Steuerung PS 30 ist eine universelle Positioniersteuerung zum Einbau in einen PC auf Basis einer PCI-Einsteckkarte.

Sie besteht aus einer PCI-Einsteckkarte und einem Endstufenmodul ohne PCI-Steckverbinder, montiert auf einem zweiten Slotblech, im Format 106 x 168 mm (kurzes PCI-Format). Beide Karten sind flexibel über Flachbandkabel miteinander verbunden.

Die PS 30 ist leistungsstark und kann bis zu 3 Achsen mit Schritt- oder DC-Servomotoren betreiben. Es sind beliebige Kombinationen beider Motortypen möglich.

Für die Kommunikation mit unterschiedlicher Peripherie sind zahlreiche Ein- und Ausgänge integriert, zum Beispiel: TTL/Analog und PWM.

Die PS 30 kann Punkt-zu-Punkt-Positionierbetrieb, Trapez- oder S-förmige Geschwindigkeitsprofile, sowie komplexe, mehrachsige Bahnsteuerungen wie Linearinterpolation oder Kreisinterpolation ausführen.

Zum Lieferumfang der Steuerung gehört auch das Softwaretool OWISoft. Damit kann die PS 30 komfortabel konfiguriert und betrieben werden. OWIS® Positioniereinheiten sind in OWISoft hinterlegt und müssen nur dem jeweiligen Antrieb zugeordnet werden. Integration und Betrieb von Fremdmotoren ist ebenfalls möglich.

2. Ausführung und Lieferumfang

Die PS 30 besteht aus einer PCI-Einsteckkarte und einem Endstufenmodul ohne PCI-Steckverbinder, montiert auf einem zweiten Slotblech, im kurzen PCI-Format.

Die PS 30-Endstufe wird über + 12 V vom PC-Netzteil versorgt. Bei hohen Anforderungen an Ausgangsleistung bzw. Fahrgeschwindigkeit kann ein externes 24 V-Netzteil angeschlossen werden.

Die Firmware für die Steuerung kann über PCI- oder RS-232-Schnittstelle aktualisiert werden.

Zum Lieferumfang der Steuerung gehören:

- PS 30-PCI-Einsteckkarte
- PS 30-Endstufenmodul in der gewünschten Motorkonfiguration
- 3 Flachbandkabel 50-polig zum Verbinden von PCI-Einsteckkarte und Endstufenmodul
- 3-fach-Motoradapterkabel
- CD mit OWISoft und Dokumentation in Deutsch und Englisch
- gedruckte Betriebsanleitung in Deutsch und Englisch
- Datenblatt in Deutsch und Englisch

2.1 Standard

Die Steuerung verfügt über:

- PCI-Anschluss
- RS-232-Anschluss
- 2 Eingänge für Referenz- bzw. Endschalter je Achse
- 8 TTL- bzw. Analogeingänge
- 5 TTL-Ausgänge
- 2 PWM-Ausgänge (z.B. zur Ansteuerung von Motorhaltebremsen)
- Anschluss für Freigabe der Motorendstufen

2.2 Zubehör

Folgendes Zubehör ist erhältlich:

- Externes Tischnetzteil AC 100-240 V, DC 24 V, 90 W
- Anschlusskabel mit Stecker für unterschiedliche Positioniersysteme
- Joystick für drei Achsen, analog, mit 3 m Kabel

2.3 Option

Es ist folgende Option verfügbar:

- Stand-Alone-Compiler mit USB-Dongle

3. Sicherheit

Sobald eine Endstufentemperatur von ca. 85°C überschritten wird, schaltet der entsprechende Motorausgang ab, und eine Fehlerstatusmeldung für die betreffende Achse wird im System vermerkt.

Um die Motorendstufen freigeben zu können, muss der Jumper JP14 auf der PCI-Karte gesetzt sein (siehe 7.3). Falls der Jumper nicht gesetzt ist, kann die Freigabe extern über den galvanisch getrennten Freigabe-Eingang erfolgen. Dazu muss der Eingang mit einer Spannung von 5 V versorgt werden. Sind weder der Jumper gesetzt noch der Freigabeeingang mit Spannung versorgt, ist kein Freischalten der Endstufen möglich.

Ferner wird der an einer Motorendstufe angeschlossene Motortyp über einen Codierwiderstand erkannt. So wird verhindert, dass ein versehentlich falsch angeschlossener Motortyp (z.B. ein DC-Motor an einer Schrittmotor-Endstufe) unkontrolliert startet.

4. Normen und Richtlinien

Die Universal-Positioniersteuerkarte PS 30 erfüllt folgende Normen und Richtlinien:

- RoHS-konform
- CE-Richtlinie
- EMV-Richtlinie 2014/30/EU

Störfestigkeit nach Fachgrundnorm EN 61000-6-1 mit:

- Störfestigkeit gegen elektrostatische Entladung (ESD), Basisnorm: EN 61000-4-2
- Störfestigkeit gegen elektromagnetische Felder, Basisnorm: EN 61000-4-3

Störaussendung nach Fachgrundnorm EN 61000-6-3 mit:

- Störstrahlung, Basisnorm: EN 55022 (informationstechnische Einrichtungen/ITE-Geräte)

5. Technische Übersicht

Stromversorgung:	+ 12V vom PC-Netzteil oder max. 24V extern
Leistungsaufnahme:	max. 24W (12V) bzw. max. 90W (24V)
Anzahl der Antriebe:	3 Achsen
Antriebsart:	2-Phasen Schrittmotoren Open Loop (OL), 2-Phasen Schrittmotoren Closed-Loop (CL), DC-Servomotoren
Kommunikation:	PCI-COM-Brücke (max. 115 200 Baud), optional RS-232
Aufbau:	PC-Einsteckkarte
Schutzart:	IP 00
Encoder:	Quadratur-Signale A/B und Index, RS-422- oder TTL-Pegel, mit 4-fach-Auswertung, max. Zählfrequenz 2 MHz (Signal) bzw. 8 MHz (Quadratur)
Funktionen:	Parametrierbare Beschleunigungsrampe (/ Bremsrampe), Dreieckiges- bzw. trapezförmiges Geschwindigkeitsprofil oder S-Kurve
Bewegungsabläufe:	Punkt-zu-Punkt-Positionierbetrieb, Linear- und Kreisinterpolation

6. Aufbau der Steuerung

Die PS 30 besteht aus einer PCI-Einsteckkarte und einem Endstufenmodul ohne PCI-Steckverbinder. Beide Karten haben kurzes PCI-Format (106 x 168 mm) gemäß Spezifikation PCI Local Bus Rev. 3.0 und sind auf separaten Slotblechen montiert.

Die Karten sind flexibel über drei 50-polige Flachbandkabel RM 1,27 mm (Standardlänge: 280 mm) miteinander verbunden. Die Länge der Flachbandkabel ist so ausgelegt, dass das Endstufenmodul bis zu drei Steckplätze von der PCI-Einsteckkarte entfernt montiert werden kann, falls benachbarte Steckplätze belegt sein sollten oder eine Montage des Endstufenmoduls direkt neben der PS 30-PCI nicht möglich ist.

Die Ein- und Ausgänge (TTL/Analog, PWM, Freigabe) sind über den 25-poligen D-Sub-Stecker der PCI-Einsteckkarte zugänglich. Für den Motoranschluss relevante Signale (Endschalter, Encoder usw.) sind auf die 62-polige HD-Buchse des Endstufenmoduls herausgeführt.

Die PS 30-Motorendstufen können wahlweise entweder über den genormten PC-Netzteilstecker (Festplattenversorgung) an der Leiterplattenoberkante des Endstufenmoduls mit + 12V aus dem PC oder extern über die Niederspannungsbuchse unterhalb des 62-poligen Motorsteckers mit bis zu 24V gespeist werden.

6.1 Anschlüsse

Der PCI-Busanschluss und der Ein-/Ausgangsstecker befinden sich auf der PS 30-PCI-Einsteckkarte. Die Buchsen zum Anschluss der Motorendstufen-Versorgung und der Motorstecker sind auf dem Endstufenmodul angebracht.

Eine zusätzliche RS-232-Kommunikationsschnittstelle, über die auch ein Firmware-Update durchgeführt werden kann, ist ebenfalls auf der PCI-Einsteckkarte implementiert.

Anschluss	Funktionen	Buchse
PCI	Kommunikation mit einem PC	PCI-Busanschluss
Motor	Motorversorgung, Endschalter und Encoder	HD 62-polige Buchse
Ein-/Ausgänge	Interaktion mit externer Hardware	D-Sub 25-poliger Stecker
+ 12V intern	Betriebsspannung der Motorendstufen	genormter PC-Netzteilstecker
24V extern	wahlweise: Betriebsspannung der Motorendstufen	DC-Rundstecker 5,5 x 2,1 x 11 mm
RS-232	Kommunikation mit dem PC (optional), Firmware-Update	10-polig IDC

PCI-Bus-Schnittstelle

Die PCI-Schnittstelle der PS 30 ist als sogenannte PCI-COM-Brücke realisiert. Der Windows-Gerätetreiber erkennt die PS 30 als „PCI Serial Port" und weist ihr eine COM-Portnummer zu, die vom Anwender bei Bedarf verändert werden kann. Die PCI-Schnittstelle wird nach erfolgreicher Installation als virtuelle RS-232-Schnittstelle angesprochen.

Die PS 30 kann mit Übertragungsraten von 9 600, 19 200, 38 400, 57 600 oder 115 200 Baud arbeiten. Es ist unbedingt darauf zu achten, dass die Übertragungsrate der PS 30 mit der im Gerätetreiber eingestellten Übertragungsrate übereinstimmt, sonst ist keine Kommunikation möglich. Voreinstellung ist 9600 Baud.

Soll die Kommunikationsgeschwindigkeit der PS 30 verändert werden, ist zunächst die Baudrate der PS 30 per Kommando auf den ewünschten Wert zu setzen. Danach wird die Übertragungsrate des Gerätetreibers auf denselben Wert gesetzt und Windows neu gestartet. Jetzt sollte die PS 30 mit der veränderten Baudrate kommunizieren.

Stromversorgung
Die PCI-Einsteckkarte wird in der Regel über den PCI-Bus mit Leistung versorgt. Falls der Host-PC keine 3,3V-Spannung am PCI-Bus bereitstellen sollte, kann die erforderliche 3,3V-Spannung intern aus + 5V erzeugt werden. Vorwahl ist über einen Jumper möglich (siehe „Bedienelemente und Einstellungen").

Die Betriebsspannung der PCI-COM-Brücke kann ebenfalls per Jumper eingestellt werden. + 3,3V, + 5V und automatische Anpassung sind möglich. Automatische Anpassung sollte bei allen modernen PCs problemlos funktionieren (siehe „Bedienelemente und Einstellungen").

Motorstecker an 3-fach-Motoradapterkabel
Mit dem passenden OWIS® Anschlusskabel werden die OWIS® Positioniereinheiten angeschlossen. Über diesen Anschlussstecker wird der Motor mit Leistung versorgt, die Motor-Haltebremse, falls vorhanden, gesteuert, sowie die Signale des Encoders und der Schalter übertragen.

Die Endstufe hat eine zusätzliche Schutzeinrichtung, die dafür sorgt, dass ein versehentlich falsch angeschlossener Motortyp (z.B. ein DC-Motor an einer Schrittmotor-Endstufe) nicht unkontrolliert startet. Am Motoranschlusskabel ist zwischen Pin 14 und Pin 15 ein Widerstand zur Codierung des Motortyps eingebaut.

Codierung:
- 0 Ohm: DC-Servomotor
- Widerstand unendlich: 2-Phasen-Schrittmotor

Beim Einschalten misst die Steuerung den Widerstandswert und signalisiert einen Fehler, wenn der gemessene Wert nicht zu der jeweiligen Steuerplatine passt. Die Fehlermeldung der Endstufe wird über das Kommando „?ASTAT" ausgelesen (siehe Befehlssatz ab S.24). Der Steckerbelegungsplan ist im Anhang aufgeführt. Die über den 3-fach-Adapter umgesetzten Signale des 62-poligen PS 30-Motorsteckers sind dort aufgelistet. Die Belegung entspricht dem OWIS®-Standard.

End- und Referenzschalter
Pro Achse können maximal zwei Schalter angeschlossen werden. Dies können Mikroschalter, TTL-Hallschalter oder TTL-Lichtschranken mit + 5V-Versorgung sein. An die Eingänge können Öffner oder Schließer, gegen Masse schaltend, angeschlossen werden.

Einer der beiden Schalter ist zusätzlich als Referenzschalter definiert.

Der aktive Pegel und die Zuordnung der Schalter werden per Software konfiguriert.

Encodereingang
Der Encodereingang ermöglicht sowohl den Anschluss von Encodern mit Leitungstreibern (antivalente Signale für CHA, CHB und optional Index I), als auch von Encodern mit TTL-/CMOS-Signalen (siehe „3-fach-Motorstecker an der PCI-Einsteckkarte").

Folgende Eingangssignale sind definiert:

Versorgungsspannung	V_{CC} (+ 5V); GND
Kanal	A, TTL oder CMOS
Kanal	A invertiert
Kanal	B, TTL oder CMOS
Kanal	B invertiert
Kanal	I (Index), TTL oder CMOS
Kanal	I invertiert

Die Umsetzung der antivalenten Signale auf TTL-Signale erfolgt mit RS-422-Leitungsempfängern. Schließt man einen Encoder mit TTL-/ CMOS-Signalen an, so bleibt der Eingang für das invertierte Signal offen und wird intern mit einem hochohmigen Spannungsteiler auf 1,4V gezogen. Die Leiterbahnen der invertierten Signale haben auf der PCI-Einsteckkarte Trennstellen mit beidseitigen Überbrückungs-Pads, um eine Unterbrechung der invertierten Signalleitungen zu ermöglichen, falls dies erforderlich sein sollte. Am nichtinvertierenden Eingang ist ein Pullup-Widerstand nach + 5V vorgesehen.

6.2 Eingänge und Ausgänge

Zur Interaktion mit externen Sensoren und Aktoren sind entsprechende digitale und analoge Ein- und Ausgänge vorgesehen.

An die TTL-kompatiblen Eingänge können z.B. einfache Gabellichtschranken etc. angeschlossen werden.

Mit den TTL-Ausgängen ist es möglich, digitale Hardware in der Anwendung direkt anzusteuern.

Eigenschaften	Pegel	Strom	Sonstiges
TTL-/Analog-Eingänge	0 - 5VDC		können wahlweise als 10 Bit-Analog- oder Digitaleingänge genutzt werden
TTL-Ausgänge	0 - 5V	10mA	
Leistungsausgänge	0 - 24VDC	1,0A	PWM

Die analogen Eingänge können Spannungen zwischen 0V und 5,0V direkt messen und mit 10 Bit-Auflösung wandeln (Referenzspannung: 5,0V). Die Eingänge sind nicht galvanisch getrennt.

Die Abfragebefehle „?ANIN<uv>" und „?INPUTS" beziehen sich auf dieselben Eingänge der PS 30 (siehe Befehlssatz ab S.24). Die Auswertung der Eingänge erfolgt entweder analog oder digital.

Die beiden Leistungsausgänge sind pulsweitenmoduliert und nach Masse schaltend. Sie können induktive Lasten ansteuern, die kurzzeitig einen hohen Anzugsstrom und anschließend nur noch einen geringen Haltestrom brauchen, wie Haltebremsen oder Hubmagnete.

Die Leistungsausgänge können als Haltebremsenansteuerung konfiguriert werden.

7. Steuerungsarchitektur und Funktion

7.1 Aufbau

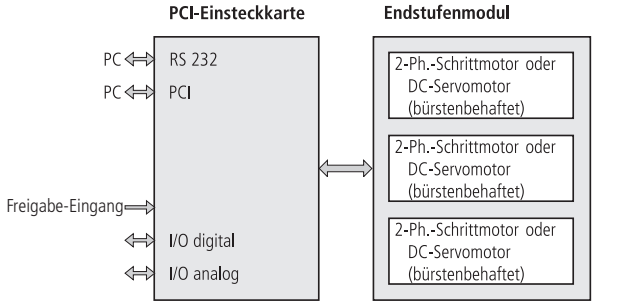

Bild 1: Steuerungsarchitektur

PCI-Einsteckkarte

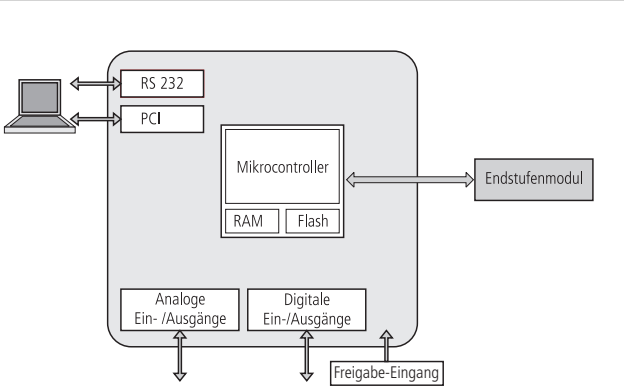

Bild 2: Steuerungsarchitektur PCI-Einsteckkarte

Die PCI-Einsteckkarte (Hauptplatine) ist das Kernstück der PS 30. Sie übernimmt die Steuerung des Hauptablaufs, kommuniziert mit dem PC und mit den Endstufen und verwaltet die digitalen und analogen Ein- und Ausgänge.

Zum Mikrocontroller gehören ein Flash-Speicher mit 512 kB und ein RAM-Speicher mit 128 kB Größe. Der Flash-Speicher wird als Programmspeicher für den Mikrocontroller verwendet, der RAM-Speicher zur Datenspeicherung. Das RAM ist über eine auswechselbare Lithiumbatterie gepuffert.

Die PCI-Einsteckkarte kommuniziert über die PCI-COM-Brücke mit dem PC. Über den PCI-Bus ist ebenfalls ein Update der Firmware möglich. Eine weitere serielle Schnittstelle (RS-232) ist als alternative Kommando- bzw. Update-Schnittstelle zum PC implementiert.

Sie beinhaltet ferner einen Motion-Prozessor, der drei Achsen steuern bzw. regeln kann. Der Motion-Prozessor verarbeitet die Befehle des Mikrocontroller und generiert entsprechend die Ansteuersignale für die Endstufenmodule. Die Schnittstelle zu den Endstufen ist mittels Optokoppler galvanisch getrennt.

Beim Motortyp DC bildet der Motion-Prozessor über die Endstufe, den Motor und den Encoder einen geschlossenen Regelkreis (Closed-Loop-Betrieb). Der Motion-Prozessor generiert dabei Richtungs- und PWM-Signale. Mit einem PWM- und einem Richtungssignal wird beim DC-Motor das Drehmoment in Größe und Richtung vorgegeben. Mit einer Leistungs-H-Brücke auf der Motorplatine wird dann ein entsprechender Strom in die Motorwicklung eingeprägt.

Schrittmotoren werden in der Regel gesteuert betrieben (Open-Loop-Betrieb), d.h. der Motorcontroller steuert den Feldvektor über PWM-Signale, wobei jeweils nur der Winkel des Feldvektors verändert wird, nicht jedoch dessen Länge.

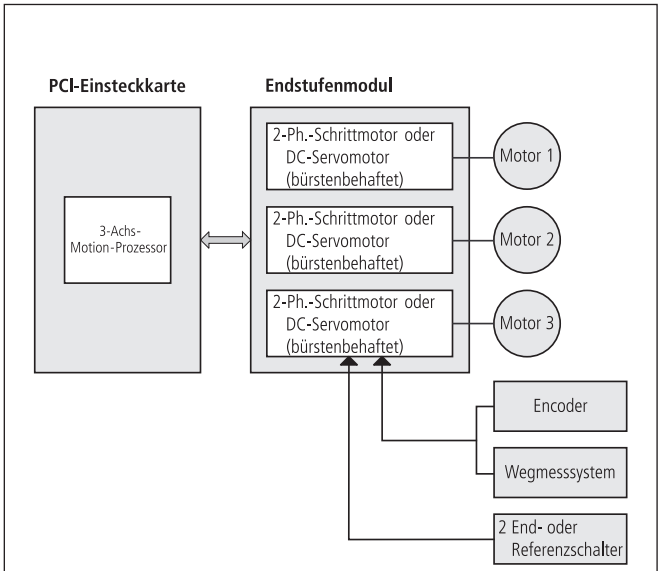

Bild 3: Funktion PCI-Einsteckkarte mit Endstufenmodul

Endstufenmodul

Die PS 30 kann maximal drei Motoren ansteuern. Die Schnittstelle zwischen PCI-Einsteckkarte und Endstufenmodul ist galvanisch getrennt.

Das Endstufenmodul besitzt einen Universal-Anschlussstecker, der alle Motorsignale (Motorwicklungen, Encoder und Endschalter) führt.

Sicherungskonzept

Die verschiedenen Versorgungs- und Hilfsspannungen der PS 30 sind über steckbare SMD-Schmelzsicherungen bzw. selbstrückstellende Sicherungen (Polyswitch) abgesichert, um im Falle eines Hardwaredefektes größeren Schaden zu vermeiden.

PCI-Einsteckkarte

Hilfsspannungsausgänge an 25-poligem D-Sub-Stecker:

- + 12 V bzw. + 24 V PWM: 1 A (flink)
- + 5 V: 300 mA, selbstrückstellend (Polyswitch)

Endstufenmodul

- + 12 V (PC): 2 A (träge)
- + 5 V: 500 mA (träge)
- + 24 V extern: 5 A (träge)
- + 5 V-Hilfsspannungsausgang: 500 mA (flink)

7.2 Betrieb unterschiedlicher Motortypen

Die PS 30 kann sowohl 2-Phasen-Schrittmotoren als auch bürstenbehaftete DC-Servomotoren ansteuern.

Das Endstufenmodul wird kundenspezifisch für beliebige Kombinationen konfiguriert ausgeliefert. In DC-Motor-Konfiguration kann für jede Achse eine Strombegrenzung per Potentiometer eingestellt werden (siehe „Bedienelemente und Einstellungen“).

7.3 Bedienelemente und Einstellungen

Jumpereinstellungen der Hauptplatine

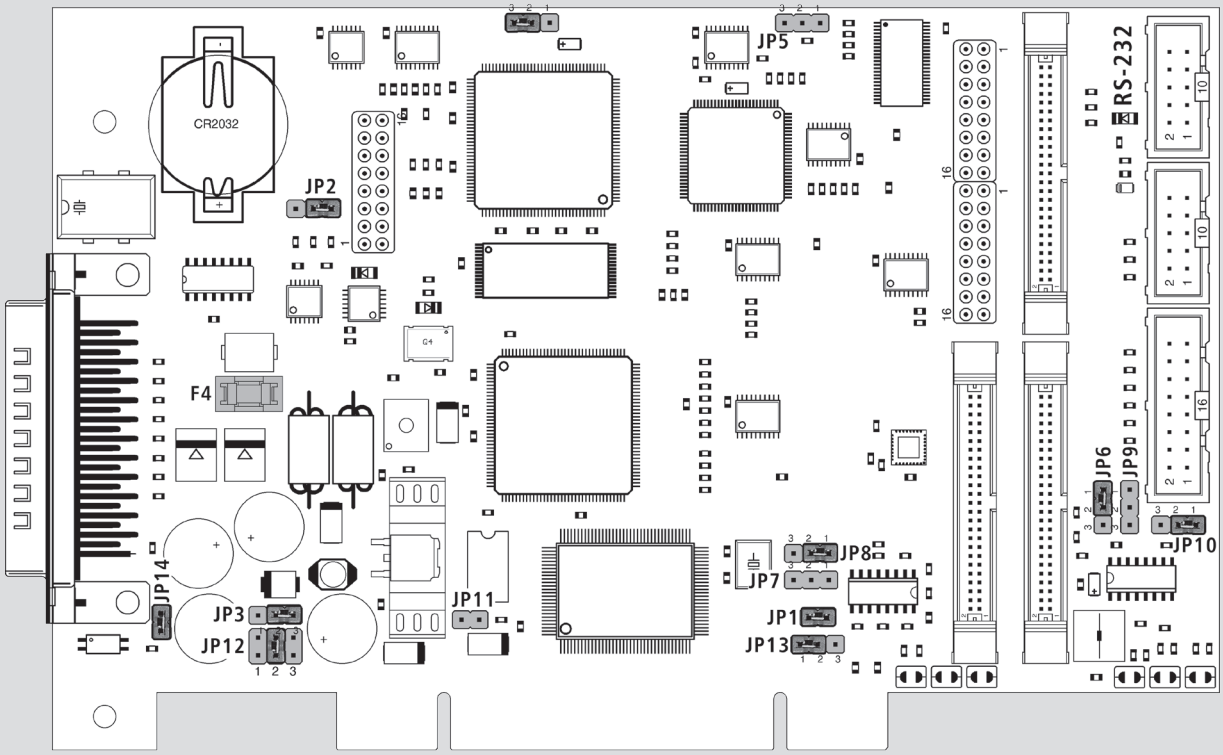

Bild 4: Werkseitige Jumpervoreinstellungen PCI-Einsteckkarte

Jumper	Funktion	Einstellmöglichkeiten	werksseitige Voreinstellung	Anmerkung
JP1	Flash extern/intern	Jumper gesteckt = extern Jumper offen = intern	gesteckt	Werkseinstellung darf nicht verändert werden
JP2	Booten externes/internes Flash	1-2: ext. Flash; 2-3: int. Flash	1-2 (rechts)	Werkseinstellung darf nicht verändert werden
JP3	+3,3V extern/intern	1-2: intern; 2-3: extern	2-3 (rechts)	Werkseinstellung muss ggf. bei sehr altem PC-Mainboard verändert werden
JP5	reserviert		kein Jumper gesteckt	
JP6	Index-Polarität Achse 1	1-2: normal; 2-3: invertiert	1-2 (oben)	
JP7	reserviert		kein Jumper gesteckt	
JP8	Index-Polarität Achse 2	1-2: normal; 2-3: invertiert	1-2 (rechts)	
JP9	reserviert		kein Jumper gesteckt	
JP10	Index-Polarität Achse 3	1-2: normal; 2-3: invertiert	1-2 (rechts)	
JP11	EEPROM freigeben oder sperren	Jumper gesteckt = EEPROM gesperrt Jumper offen = EEPROM freigegeben	offen	Werkseinstellung darf nicht verändert werden
JP12	Betriebsspannung der PCI-COM-Brücke	1: +5V; 2: automatisch; 3: +3,3V	2 (Mitte)	manuelle Vorwahl ggf. bei sehr altem PC-Mainboard erforderlich
JP13	PCI-Bustakt	1-2: 33 MHz; 2-3: 66 MHz	1-2 (links)	Werkseinstellung darf nicht verändert werden
JP14	Endstufenfreigabe	Jumper gesteckt = Freigabe; Jumper nicht gesteckt = Freigabe extern möglich	gesteckt	

Sicherung der Hauptplatine

F4: 1A flink zur Absicherung der +12V- bzw. +24V-Hilfsspannung für die PWM-Ausgänge

Jumper- und Potentiometereinstellungen für PS 30-Endstufenmodul

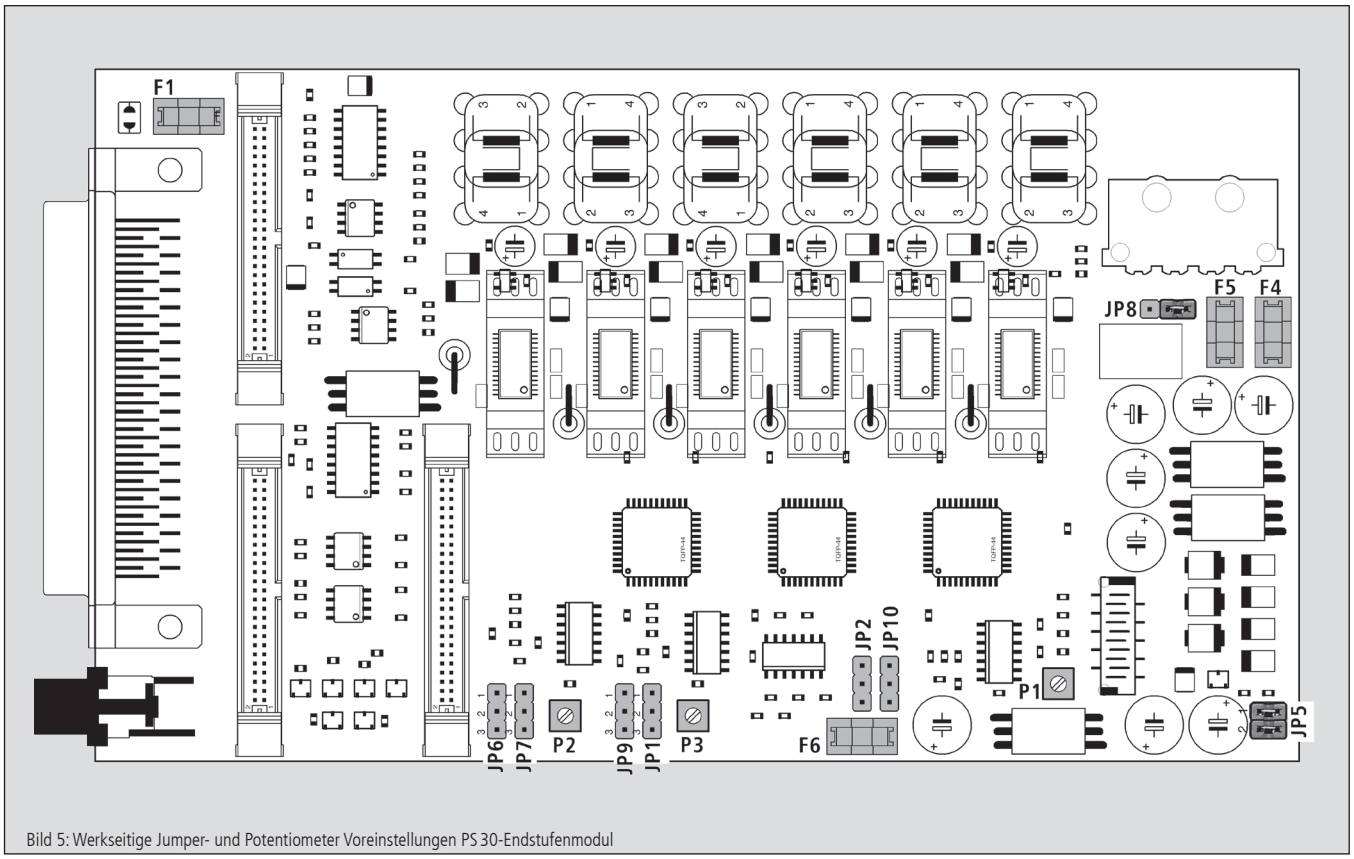

Bild 5: Werkseitige Jumper- und Potentiometer Voreinstellungen PS 30-Endstufenmodul

Jumper/ Potentiometer	Funktion	Einstellmöglichkeiten	werksseitige Voreinstellung	Anmerkung
JP2/JP10	Motortyp Achse 1	1-2 (oben): 2-Phasen-Schrittmotor Open Loop 2-3 (unten): DC-Servomotor	konfigurationsabhängig; beide Jumper müssen immer in der gleichen Stellung sein.	Werkseinstellung darf nicht verändert werden.
JP6/JP7	Motortyp Achse 2			
JP1/JP9	Motortyp Achse 3			
JP8	Hilfsspannung + 5V	1-2: intern 2-3: extern (von PC-Netzteilstecker)	2-3 (links)	Werkseinstellung sollte nicht verändert werden.
JP5	Einspeisung + 12V- bzw. + 24V-Hilfsspannung für PCI-Einsteckkarte	zwei Jumper: 1 (oben) / 2 (unten)	beide Jumper gesteckt	Jumper nur paarweise stecken oder entfernen.
P1	Strombegrenzung DC-Motor, Achse 1	Linksanschlag (Drehung im Gegenuhrzeigersinn): 2,4A (Strombereich 1) 6,6A (Strombereich 2) *) Rechtsanschlag (Drehung im Uhrzeigersinn): Strom Null *) maximal zulässig ist ein Strom von 3,5A	konfigurationsabhängig	Einstellung der Strombegrenzung erfolgt bei Konfiguration werksseitig
P2	Strombegrenzung DC-Motor, Achse 2			
P3	Strombegrenzung DC-Motor, Achse 3			

Sicherungen für PS 30-Endstufenmodul

F4: 3A träge zur Absicherung der + 12V-Versorgung vom Festplatten-Netzteilstecker
F5: 500 mA flink zur Absicherung der internen + 5V-Hilfsspannung
F6: 5A träge zur Absicherung des externen 24V-Versorgungseingangs für die Motorendstufen
F1: 500 mA flink zur Absicherung des 5 V-Hilfsspannungsausgangs

7.4 Strombereichsumschaltung der Motorendstufe

Die PS 30-Endstufe besitzt zwei umschaltbare Strombereiche, um möglichst hohe Auflösung der Stromeinstellung bzw. möglichst feinen Mikroschrittbetrieb zu ermöglichen.

Der gewählte Strombereich wird im statischen RAM abgespeichert. Um einen neuen Strombereich zu aktivieren, ist es erforderlich, die Achse <n> nach der Bereichsumschaltung neu zu initialisieren.

Vorwahl von Strombereich 2 für Achse <n> erfolgt über folgende Kommandofolge:

AMPSHNT<n>=1

INIT<n>

Zurückschalten in Strombereich 1 kann mittels folgender Befehlssequenz vorgenommen werden:

AMPSHNT<n>=0

INIT<n>

Vorwahl des Phasenstromes bei 2-Phasen-Schrittmotoren
Für 2-Phasen-Schrittmotoren können Fahrstrom und Haltestrom separat voreingestellt werden. Die Einstellung für Achse <n> kann wie nachfolgend beschrieben vorgenommen werden. Die Angabe <uv> erfolgt als ganzzahliger Prozentwert des Maximalstromes im vorgewählten Strombereich (1 oder 2).

Fahrstrom: DRICUR<n>=<uv>

Haltestrom: HOLCUR<n>=<uv>

Maxitaler Phasenstrom Strombereich 1 (entsprechend 100%): 1,2A

Maximaler Phasenstrom Strombereich 2 (entsprechend 100%): 3,3A

In Strombereich 2 darf bei der Standardversion der Endstufe maximal ein Phasenstrom von 1,8A, entsprechend 54,5% des Endwerts, eingestellt werden.

Es sollte generell der kleinstmögliche Strombereich gewählt werden, um eine optimale Mikroschrittauflösung zu erhalten.

Hinweis:
Der maximal thermisch zulässige Dauerstrom der Motorendstufen (Abschaltung bei Überschreiten einer Leiterplattentemperatur von ca. 85°C) ist sehr stark von den angeschlossenen Motortypen, der Anzahl der angesteuerten Motoren, deren mechanischer Belastung, den Fahrgeschwindigkeiten, den PC-internen Temperatur-/Lüftungsverhältnissen, der Umgebungstemperatur usw. abhängig.

Es kann deshalb kein exakter Wert für einen garantierten Phasen-Dauerstrom angegeben werden. Bei normaler Raumtemperatur und durchschnittlichen Einbaubedingungen kann davon ausgegangen werden, dass ein Phasenstrom von ca. 1,8A bei drei angeschlossenen Motoren erreichbar ist.

Strombereichseinstellung für DC-Servomotoren
Für DC-Servomotoren ist der geeignete Strombereich unter Berücksichtigung des thermisch zulässigen Dauerstroms des jeweiligen Motortyps vorzuwählen. Eine Strombegrenzung kann auf der jeweiligen Platine konfiguriert werden. (Weitere Hinweise sind im Kapitel „Bedienelemente und Einstellungen" zu finden.)

7.5 Endstufenkonfiguration per Software

Diverse Charakteristika der Motorendstufen, die in den allermeisten Anwendungsfällen nicht gesetzt bzw. verändert zu werden brauchen, können vor der Initialisierung einer Endstufe („INIT..."-Befehl) konfiguriert werden.

Es stehen vier Ausgangsbits zur Verfügung, die mittels „ETTLOUTS<n>=<uv>" gesetzt bzw. mittels „ETTLOUTC<n>=<uv>" gelöscht werden können.
Es werden die Achse <n>, deren Einstellung geändert werden soll, sowie eine 4-stellige Bitmaske <uv> übergeben. Ein gelöschtes Bit (0) bedeutet, dass der Zustand des jeweiligen Ausgangs nicht verändert werden soll, während ein gesetztes Bit (1) vorgibt, dass das betreffende Ausgangsbit gesetzt (ETTLOUTS) bzw. gelöscht (ETTLOUTC) werden soll.

Standardeinstellungen sind in nachfolgenden Tabellen grau markiert.

Bit 1: Timing

Bit Nr. 4-3-2-1	Austastzeit (Blanking Time)
x-x-x-1	1,3 µs
x-x-x-0	0,6 µs

Bits 3 und 4: Entregungsart der Motorwicklungen

Bit Nr. 4-3-2-1	Entregungsart („Decay") bei 2-Phasen-Schrittmotor *)
0-0-x-x	schnell („Fast Decay", 100%)
0-1-x-x	gemischt B („Mixed Decay B", "48%)
1-0-x-x	gemischt A („Mixed Decay A", "15%)
1-1-x-x	langsam („Slow Decay", 0%)

*) Entregung ist aktiv bei fallendem Strom-Ast.

Bit Nr. 4-3-2-1	Entregungsart bei DC-Servomotor
x-0-x-x	schnell, wenn PWM = 0
x-1-x-x	langsam, wenn PWM = 0
0-x-x-x	schnell bei Ansprechen des Strom-Choppers
1-x-x-x	langsam bei Ansprechen des Strom-Choppers

8. Steuerungsfunktionen

8.1 Trapezförmiges Punkt-zu-Punkt-Profil

Die folgende Tabelle umfasst die spezifischen Profilparameter für den trapezförmigen Punkt-zu-Punkt-Modus:

Profilparameter	Format	Wort-länge	Bereich
Position	32.0	32 bit	-2.147.483.648…+2.147.483.647 Counts
Geschwindigkeit	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle
Beschleunigung	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle²
Verzögerung	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle²

Für dieses Profil errechnet der Host eine Beschleunigung, eine Verzögerung, eine Geschwindigkeit und eine Endposition.
Das Profil ist nach der Kurvenform (Bild 6) benannt: Die Achse beschleunigt linear (anhand des programmierten Beschleunigungswertes), bis sie die programmierte Geschwindigkeit erreicht. Die Achse bremst dann linear ab (den Verzögerungsgswert nutzend), bis sie an der vorgegebenen Position stehen bleibt. Falls die programmierte Fahrdistanz so kurz ist, dass die Verzögerung einsetzen muss, bevor die Achse die programmierte Geschwindigkeit erreicht, wird das Profil keinen konstanten Geschwindigkeitsbereich aufweisen, und das Trapez wird zum Dreieck (Bild 7).

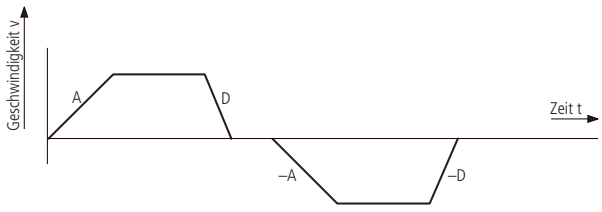

Bild 6: A = Beschleunigung, D = Verzögerung

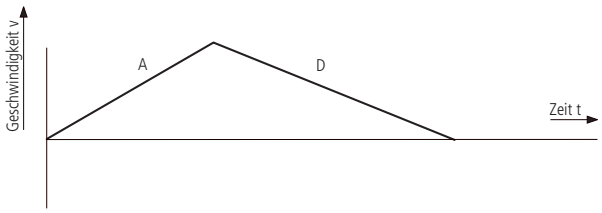

Bild 7: Trapez-Kurven Profil, A = Beschleunigung, D = Verzögerung

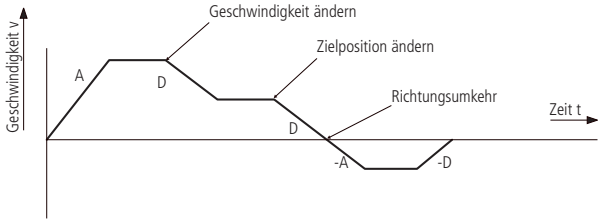

Bild 8: A = Beschleunigung, D = Verzögerung

Die Beschleunigungs- und Verzögerungsrampen können symmetrisch (wenn die Beschleunigung gleich der Verzögerung ist) oder asymmetrisch sein (wenn die Beschleunigung nicht gleich der Verzögerung ist).

Der Beschleunigungsparameter wird immer am Anfang der Bewegungssequenz benutzt. Danach wird der Wert für die Beschleunigung in dieselbe Richtung verwendet, und der Wert für die Verzögerung wird in entgegengesetzter Richtung eingesetzt. Falls keine Bewegungsparameter während der Bewegungssequenz verändert werden, wird der Beschleunigungswert verwendet bis die maximale Geschwindigkeit erreicht wurde. Der Verzögerungswert wird für die Abbremsrampe eingesetzt, bis die Geschwindigkeit auf Null sinkt.

Es ist möglich, einen der Profilparameter zu verändern, während die Achse sich in diesem Profilmodus befindet. Der Profilgenerator wird immer versuchen, die Bewegung innerhalb der durch die Parameter vorgegebenen gesetzten Bedingungen auszuführen. Wird während der Bewegung die Endposition in solch einer Weise verändert, dass die restliche Fahrdistanz das Vorzeichen wechselt, wird die PS 30 mit Rampe bis zum Stopp abbremsen und dann in entgegengesetzte Richtung beschleunigen, um sich zu der neuen angegebenen Position zu bewegen.

8.2 S-Kurven-Punkt-zu-Punkt-Profil

Die folgende Tabelle fasst die Profilparameter für den S-Kurven-Punkt-zu-Punkt-Modus zusammen:

Profilparameter	Format	Wort-länge	Bereich
Position	32.0	32 bit	-2.147.483.648…+2.147.483.647 Counts
Geschwindigkeit	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle
Beschleunigung	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle²
Verzögerung	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle²
Jerk	0.32	32 bit	(1…2.147.483.647)/4.294.967.296 Counts/Cycle³

Das S-Kurven-Punkt-zu-Punkt-Profil fügt im Vergleich zum Trapezprofil einen weiteren Parameter ("Jerk" oder „Ruck") hinzu. Dieser gibt die Änderungsrate der Beschleunigung an.

Wenn in diesem Profilmodus eine Positionierung durchgeführt wird, wird zunächst die Beschleunigung linear mit dem eingestellten Wert Jerk erhöht, bis sie den programmierten Wert erreicht. Der Übergang von konstanter Beschleunigung zu konstanter Geschwindigkeit erfolgt ebenfalls mit einem linearen Anwachsen der Verzögerung. Das Verhalten am Ende der Bewegung ist analog dazu.

Im S-Kurven-Profilmodus muss der gleiche Wert sowohl für die Beschleunigungs- als auch für die Verzögerunsrampe benutzt werden. Asymmetrische Profile sind nicht erlaubt. Dies ist nur im trapezförmigen Profilmodus möglich.

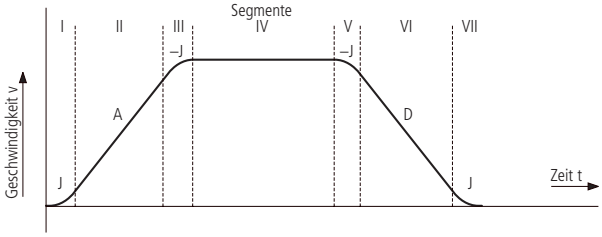

Bild 9: S-Kurven Profil, A = Beschleunigung, D = Verzögerung, J = Beschleunigungsänderung

Bild 9 zeigt ein typisches S-Kurven-Profil. In Segment I erhöht sich der Beschleunigungswert um den per Jerk gesetzten Wert, bis die maximale Beschleunigung erreicht wurde. Im nächsten Segment wird die Achse linear (Jerk = 0) beschleunigt. Das Profil wendet dann im Segment III den negativen Wert des Jerks an, um die Beschleunigung zu reduzieren. Im Segment IV verfährt die Achse jetzt mit maximaler Geschwindigkeit (V). Das Profil wird dann in einer dem Beschleunigungswert ähnlichen Weise abbremsen, indem

in umgekehrter Richtung der negative Jerk verwendet wird, um zuerst die maximale Verzögerung zu erreichen (D), und dann die Achse zu einem Halt an der Endposition zu bringen.

Ein S-Kurven-Profil enthält u.U. nur einen Teil der in Bild 9 gezeigten Segmente. Dies kann z.B. der Fall sein, wenn nicht die maximale Beschleunigung vor dem „Halbweg" in Richtung Endgeschwindigkeit oder Endposition erreicht werden kann. Hier würde das Profil dann nicht die Segmente II und VI enthalten (siehe Bild 10).

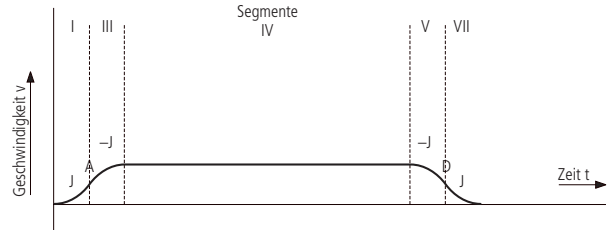

Bild 10: S-Kurve erreicht nicht die maximale Beschleunigung,
A = Beschleunigung, D = Verzögerung, J = Beschleunigungsänderung

Falls eine Position derart angegeben wird, dass die Endgeschwindigkeit nicht erreicht werden kann, wird es kein Segment IV geben (siehe Bild 11).

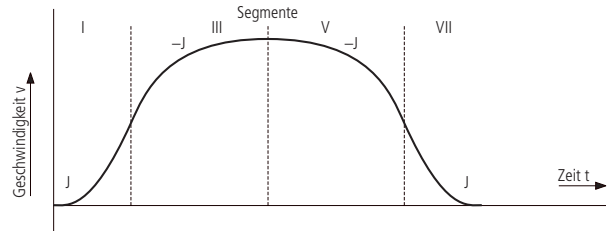

Bild 11: S-Kurve ohne maximales Geschwindigkeitssegment,
A = Beschleunigung, D = Verzögerung, J = Beschleunigungsänderung

Im Gegensatz zum trapezförmigen Profilmodus erlaubt der S-Kurven-Profilmodus keine Änderungen an einem der Profilparameter, während die Achse in Bewegung ist. Ebenfalls darf die Achse nicht in den S-Kurven-Modus geschaltet werden, während die Achse in Bewegung ist. Es ist allerdings erlaubt, vom S-Kurven-Modus zu einem anderen Profilmodus während der Bewegung zu wechseln.

8.3 Geschwindigkeitsmodus

Die folgende Tabelle fasst die Profilparameter für den Geschwindigkeitsmodus zusammen:

Profilparameter	Format	Wort-länge	Bereich
Geschwindigkeit	16.16	32 bit	(-2.147.483.648…+2.147.483.647) /65.536 Counts/Cycle
Beschleunigung	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle²
Verzögerung	16.16	32 bit	(1…2.147.483.647)/65.536 Counts/Cycle²

Im Gegensatz zu den trapezförmigen und S-Kurven-Profilmodi, bei denen die Endposition bestimmt, ob positive oder negative Geschwindigkeit vorgegeben wird, bestimmt das Vorzeichen des im Geschwindigkeitsmodus übergebenen Geschwindigkeitswerts, ob in positiver oder negativer Richtung gefahren werden soll. Deswegen kann der Geschwindigkeitswert der zur PS 30 übermittelt wird, positive Werte (für positive Bewegungsrichtung) oder negative Werte (für entgegengesetzte Bewegungsrichtung) annehmen. Bei diesem Profil wird keine Endposition angegeben.

Die Bahn wird ausgeführt, indem die Achse mit dem angegebenen Wert kontinuierlich beschleunigt, bis die jeweilige Endgeschwindigkeit erreicht wird. Die Achse fängt an abzubremsen, wenn eine neue Geschwindigkeit angegeben wird, die einen kleineren Wert als die aktuelle Geschwindigkeit oder ein anderes Vorzeichen hat als die aktuelle Richtung vorgibt.

Ein einfaches Geschwindigkeitsprofil sieht aus wie ein einfaches trapezförmiges Punkt-zu-Punkt-Profil, wie in Bild 6 dargestellt.

Bild 12 zeigt ein komplizierteres Profil, in dem beides, die Geschwindigkeit als auch die Bewegungsrichtung, zweimal wechseln.

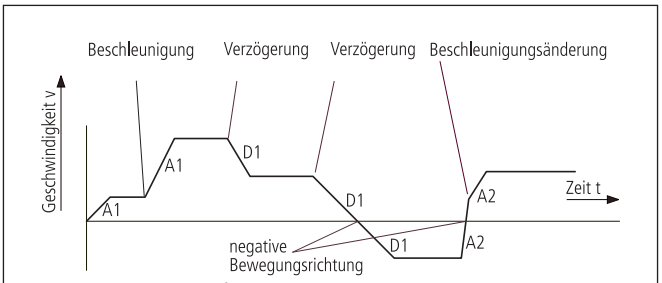

Bild 12: Geschwindigkeitsprofil

Hinweis:
Im Geschwindigkeitsmodus ist die Achsenbewegung nicht an eine Endposition gebunden. Es liegt in der Verantwortung des Anwenders, Geschwindigkeits- und Beschleunigungswerte zu verwenden, die einen sicheren Bewegungsablauf garantieren.

8.4 Referenzierung

Bei der Referenzfahrt wird einer der vier Endschalter angefahren. Die Position kann an dieser Stelle genullt werden. Dazu werden zwei Referenzfahrtgeschwindigkeiten mit Betrag und Vorzeichen und eine Referenzbeschleunigung parametriert. Der Endschalter wird mit großer Geschwindigkeit angefahren und mit kleiner Geschwindigkeit verlassen, dann wird gestoppt.

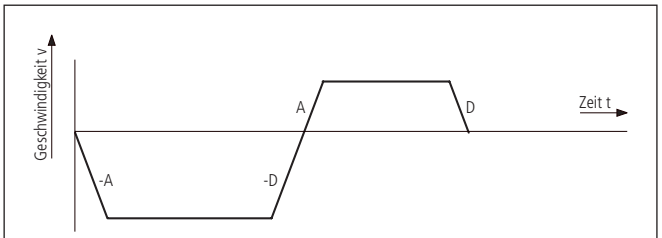

Bild 13: Referenzfahrt, A = Beschleunigung, D = Verzögerung

8.5 Linearinterpolation

Begriffsbestimmung
Linearinterpolation bezeichnet hier die Synchronisation der Bewegung aller beteiligten Achsen derart, dass die Achsen quasi-simultan starten und ihre Ziele praktisch gleichzeitig erreichen. Die Bewegung erfolgt hierbei mittels trapezförmiger Geschwindigkeitsprofile, wobei die Beschleunigungs- und Bremsrampen so angepasst werden, dass alle Achsen ebenfalls synchron beschleunigen bzw. bremsen. Die Bewegung eines aus Linearachsen bestehenden XYZ-Systems, das über Linearinterpolation angesteuert wird, beschreibt somit im kartesischen Koordinatensystem näherungsweise eine Gerade im Raum.

Die Achse mit der niedrigsten Achsnummer, welche den längsten Verfahrweg (umgerechnet in Inkremente) zurückzulegen hat, wird als Führungsachse f bezeichnet. Auf diese Achse werden die restlichen an der Linearinterpolation beteiligten Achsen steuerungsintern per Software synchronisiert.

Funktionsprinzip
Welche der maximal drei Achsen an der Linearinterpolation beteiligt sind, wird über einen Binärcode beim Start der Achsen angegeben. Ein gesetztes Bit bedeutet hierbei, dass die entsprechende Achse aktiv ist.

Für jede Achse muss vor Verwendung der Linearinterpolation ein maximaler Geschwindigkeits- sowie ein maximaler Beschleunigungswert definiert werden, der während des Positioniervorganges nicht überschritten werden darf. Das Geschwindigkeit-Zeit-Profil eines linearinterpolierten Bewegungsablaufes ist symmetrisch.

Unter Berücksichtigung der digitalen Systemzeit (Abtastzeit bzw. Periodendauer des Profilgenerators) für jede Achse werden die Maximalwerte in der Weise umgerechnet, dass die Führungsachse f schnellstmöglich (mit maximal möglicher Geschwindigkeit v_{max} (f) und Beschleunigung a_{max} (f)) ihr Ziel erreicht. Die restlichen Achsen werden auf die Führungsachse synchronisiert, wobei die gegebenen Grenzwerte von der Steuerung einzuhalten sind.

Die Linearinterpolationsachsen seien nachfolgend mit (i) bezeichnet. Das folgende Diagramm zeigt den prinzipiellen Verlauf des Geschwindigkeitsprofils der Führungsachse v_f (t) und einer beliebigen Linearinterpolationsachse v_n (t) an einem Beispiel:

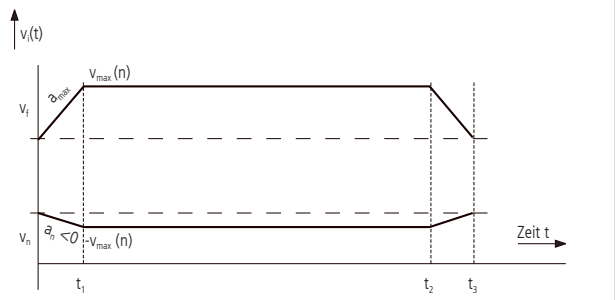

Bild 14: Linearinterpolation

Die Fahrdistanz der Achse (n) ist im Beispiel negativ, die Fahrdistanz der Führungsachse (f) positiv. Zum Zeitpunkt t_1 ist die Beschleunigungsphase beendet. Die Verzögerung wird bei t_2 eingeleitet, und alle drei Achsen stoppen gemeinsam zum Zeitpunkt t_3.

8.6 Funktionsweise der allgemeinen Bahnsteuerung

Definition
Die PS 30 ermöglicht, beliebige Bahnkurven über Ketten von Einzelvektoren zu approximieren, die in Form einer Vektortabelle an die Steuerung übergeben werden. Die allgemeine Bahnsteuerung wird somit über einen Vektormodus realisiert.

In die Vektortabelle werden relative Positionswerte eingetragen, die zu bestimmten, diskreten Zeitpunkten möglichst genau erreicht werden sollen. Bezugspunkt bzw. Startpunkt der Tabellenvektoren ist die jeweilige aktuelle Sollposition der Achsen.

Die approximierten Bahnkurven werden im Geschwindigkeitsmodus mit Trapezprofil gefahren.

Realisierung des Vektormodus
Vektortabelle

Jeder Tabelleneintrag n definiert ein komplettes Fahrsegment und enthält den relativen Fahrtvektor $\Delta\vec{x}$ für maximal drei Achsen (a bis c, entsprechend den Achsnummern 1 bis 3), das für die Fahrt des Vektors vorgegebene Zeitintervall Δt, einen 16-Bit-Funktionscode F, einen 8-Bit-Fehlercode E und einen 8-Bit-Achsfreigabecode T:

n	$\Delta\vec{x}$	Δt	F	E	T
1	$\Delta x_{a1}, \Delta x_{b1}, \Delta x_{c1}$, 0, 0, 0, 0, 0	t_1	t_1	e_1	t_1
...	...	...	...	...	...
N	$\Delta x_{aN}, \Delta x_{bN}, \Delta x_{cN}$, 0, 0, 0, 0, 0	Δt_N	f_N	e_N	t_N

Es können maximal 2000 Vektoren definiert werden ($N_{max} = 2000$).

Die Elemente des Fahrtvektors (Einzeldistanzen) werden als ganzzahlige Werte mit Vorzeichen (Integer 16 Bit) dargestellt. Die maximale Wegdistanz für ein Zeitintervall Δt_n beträgt 32760 Inkremente, d.h. als Wertebereich für einen Positionseintrag ist ein Zahlenwert von - 32760 bis + 32760 zulässig.

Segmentdauer

Das Zeitintervall Δt_n für Fahrsegment <n> wird als ganzzahliges Vielfaches von 1,024 ms angegeben. Der Wertebereich reicht von 20 bis 1638, woraus sich eine definierbare Segmentzeit von minimal 20,48 ms bis maximal 1,677312 s in Schritten von 1,024 ms ergibt:

$\Delta t_{n_{min}} = 20 \cdot 1{,}024\,\text{ms} = 20{,}48\,\text{ms}$

$\Delta t_{n_{max}} = 1638 \cdot 1{,}024\,\text{ms} = 1{,}677312\,\text{s}$

Steuercodes

Alle hier verwendeten Codes (F, E und T) sind prinzipiell Binärcodes, die grundsätzlich als positive ganzzahlige Werte (Integer) repräsentiert und an die Steuerung übergeben werden, unabhängig von dem mittels „TERM=..." vorgewählten Terminalmodus.

Der Funktionscode F wird als 16-Bit-Wert dargestellt. In der aktuellen Firmware-Version wird Bit 15 zur Vorwahl der Betriebsart, d.h. „konstante Geschwindigkeit" (v=const., Bit 15 gelöscht) oder „konstante Beschleunigung" (a=const., Bit 15 gesetzt), verwendet. Die restlichen Bits sind reserviert für eventuelle künftige Erweiterungen. Somit ist für die Standard-Betriebsart „konstante Geschwindigkeit" f = 0 zu setzen und für „konstante Beschleunigung" dementsprechend f = 32768.

Der 8-Bit-Fehlercode E gibt an, ob und gegebenenfalls bei welcher der maximal drei im Vektormodus aktiven Achsen während der Plausibilitätsüberprüfung der Vektortabelle ein Fehler aufgetreten

ist. Hierbei zeigt ein gesetztes Bit 0 einen Fehler bei Achse 1 an, ein gesetztes Bit 1 einen Fehler bei Achse 2 usw. Der 8-Bit-Freigabecode T definiert, welche der Achsen 1 bis 3 im Vektormodus aktiv ist. Die Zuordnung der einzelnen Bits zur Achsnummer entspricht dem Fehlercode E, d.h. ein gesetztes Bit 0 bedeutet, dass Achse 1 aktiv ist usw.

Betriebsarten

In den nachfolgenden Diagrammen werden beide über den Funktionscode F vorwählbaren Betriebsarten anhand des Geschwindigkeit-Zeit-Verlaufes am Beispiel veranschaulicht. Die Zeitintervalle der fünf dargestellten Fahrsegmente werden mit „Δt_1" bis „Δt_5" bezeichnet, die Geschwindigkeitswerte am Ende des jeweiligen Segments mit „v_1" bis „v_5" und die Beschleunigungswerte mit „a_1" bis „a_5".

Geschwindigkeit-Zeit-Diagramm für Betriebsart v=const. (Beispiel):

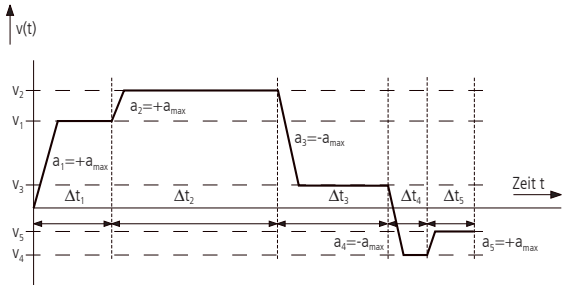

Bild 15: Geschwindigkeit-Zeit-Diagramm für Betriebsart v=const.

Die Fahrgeschwindigkeit wird im Konstantgeschwindigkeits-Modus mit der vorgegebenen Maximalbeschleunigung geändert und bleibt danach konstant. Sie wird während der Abarbeitung der Vektortabelle für jedes Segment zyklisch neu berechnet. Eine eventuell auftretende Lageabweichung am Ende eines Segments fließt im darauffolgenden Segment als Korrekturwert ein, um eine Akkumulation des Positionierfehlers zu vermeiden.

Geschwindigkeit-Zeit-Diagramm für Betriebsart a=const. (Beispiel):

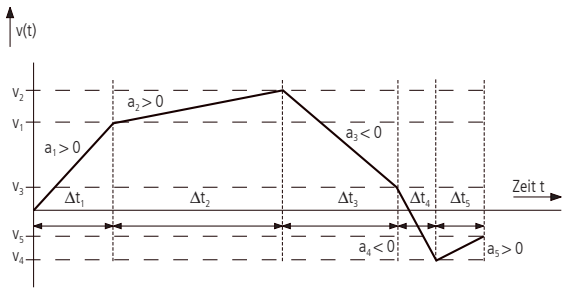

Bild 16: Geschwindigkeit-Zeit-Diagramm für Betriebsart a=const.

Die Fahrgeschwindigkeit ändert sich im Konstantbeschleunigungs-Modus stetig. Der Beschleunigungswert ist innerhalb eines Segments für jede Achse konstant. Endgeschwindigkeit und Beschleunigung innerhalb des Segments werden während der Abarbeitung der Vektortabelle für jedes Segment zyklisch neu berechnet. Eine eventuell auftretende Lageabweichung am Ende eines Segments fließt im darauffolgenden Segment, analog zur Betriebsart v=const., als Korrekturwert ein.

Maximalgeschwindigkeit und -beschleunigung

Die maximal zulässige Geschwindigkeit bzw. Beschleunigung im Vektormodus wird für jede Achse separat mittels der Befehle „IVEL" und „IACC" gesetzt. Diese Grenzwerte gelten gleichermaßen für den Vektormodus als auch für den Betrieb mit Linearinterpolation.

Plausibilitätsprüfung

Über das Kommando „PTABPLAUS" kann eine Vektortabelle auf Plausibilität geprüft werden. Falls die gegebene Zielposition einer Achse nur bei Überschreitung des vorgegebenen Geschwindigkeits- oder Beschleunigungslimits erreicht werden könnte, wird für die betreffende Achse das entsprechende Bit im Fehlercode E gesetzt.

Gesetzte Fehlerbits werden während des Positioniervorgangs ignoriert und dienen nur der Information des Anwenders. Der Tabelleneintrag kann auch dann ausgeführt werden, wenn E ungleich Null ist, jedoch ist dann mit einem sehr großen Positionierfehler zu rechnen.

Achsenfreigabe

Für jede innerhalb eines Fahrsegments aktive Achse muss im Freigabecode T ein Bit gesetzt werden. Achsen mit gelöschtem Bit werden im Fahrtvektor nicht berücksichtigt bzw. mit der programmierten Maximalbeschleunigung auf Geschwindigkeit Null abgebremst, falls die aktuelle Fahrgeschwindigkeit ungleich Null sein sollte.

Syntax

Der Tabelleneintrag <n> wird über den Befehl „POSTAB" generiert und zur Steuerung übertragen. Die Syntax ist wie folgt:

POSTAB <n> = Δx_{an}, Δx_{bn}, Δx_{cn}, 0, 0, 0, 0, 0,

Δt_n,f_n, e_n, t_n

Als Wert für den Fehlercode E sollte immer Null übergeben werden, damit eventuell gesetzte Fehlerbits gelöscht werden.

Die Plausibilitätsprüfung für die Fahrsegmente <n> bis zum Ende der Tabelle wird mittels
PTABPLAUS <n>
vorgenommen.
Hierbei werden für alle aktiven Achsen jedes Segments die Geschwindigkeits- bzw. Beschleunigungswerte berechnet und die Einhaltung der gesetzten Grenzwerte überprüft. Im Fehlerfall wird das der Achse entsprechende Bit im Fehlercode E gesetzt. Der berechnete Geschwindigkeits- und Beschleunigungswert (Vel_i and Acc_i) für Segment <n> der letzten aktiven Achse <i> (d.h. derjenigen aktiven Achse mit der höchsten Achsnummer <i>) wird zu Kontrollzwecken ebenfalls in der Tabelle gespeichert und kann mittels „?POSTAB" ausgelesen werden. Beide Kontrollwerte dienen insbesondere der Fehlersuche bzw. der erweiterten Plausibilitätskontrolle von Fahrsegmenten mit einer einzigen aktiven Achse.

?POSTAB <n>
liefert als Antwort:

Δx_{an}, Δx_{bn}, Δx_{cn}, 0, 0, 0, 0, 0, Δt_n,f_n, e_n, t_n Vel_i, Acc_i

Beispiel:

Das nachfolgende Beispiel soll die grundlegenden Funktionen zur Erstellung der Tabelleneinträge veranschaulichen. Gegeben seien:

Segmentnummer: 0 (erster Tabelleneintrag)

Segmentzeit: ca. 100 ms

aktive Achsen für die Bahnsteuerung: Achsen 1, 2, 3

Geschwindigkeitslimits Achse 1, 2, 3 : 800000, 500000, 300000

Beschleunigungslimits Achse 1, 2, 3 : 2000, 4000, 10000

Fahrdistanzen Achse 1, 2, 3 (relativ, in Inkrementen): 1000, -500, 2000

Betriebsart a=const.

Zu berechnen sind die normierte Segmentzeit Δt_0 und der Freigabecode t_0:

$$\Delta t_0 = \frac{100\,ms}{1{,}024\,ms} \sim 98$$

$$t_0 = 2^0 + 2^1 + 2^2 = 7$$

Folgende Befehle sind zu senden, um die Geschwindigkeits- und Beschleunigungslimits zu setzen sowie den ersten Tabelleneintrag zu definieren:

IVEL1=800000

IVEL2=500000

IVEL3=300000

IACC1=2000

IACC2=4000

IACC3=10000

POSTAB0=1000,-500,2000,0,0,0,0,0,98,32768,0,7

Plausibilitätskontrolle mittels

?PTABPLAUS0

und Auslesen des Tabellenelements über
?POSTAB0
ergibt als Antwort:

1000,-500,2000,0,0,0,0,0,98,32768,4,7,668734,1705,

Der Fehlercode „4" zeigt an, dass der Eintrag für die dritte (und letzte) Achse fehlerhaft ist. Es wurden ein Geschwindigkeitswert von 668734 und eine Beschleunigung von 1705 bei einer gegebenen Fahrstrecke von 2000 Inkrementen für diese Achse berechnet. Der Geschwindigkeitswert liegt über dem zulässigen Grenzwert 300000.

Fahrtende

Nach Abarbeitung des letzten Tabelleneintrags oder bei gelöschtem Freigabe-Bit bremsen die dann nicht mehr aktiven Achsen mit der jeweiligen Verzögerung auf Geschwindigkeit Null ab. Danach wird der Geschwindigkeitsmodus deaktiviert und die Achsen werden von Bahnsteuerungskontrolle auf Positionshaltung umgeschaltet.

Daraus ergibt sich bei Beendung der Bahnkurve ein Nachlaufen um eine gewisse durch die Ausgangsgeschwindigkeit am Ende des letzten Segments und die Maximalbeschleunigung bestimmte Distanz.

Kreisinterpolation

Die approximative Bahnkurvenerzeugung über tabellierte Segmente ermöglicht auch, mit zwei beliebigen Achsen X und Y eine kreisähnliche Figur bzw. einen Teil davon zu generieren. Hierbei wird der gewünschte Kreisbogen durch eine Sequenz von Kreissekanten angenähert.

Über einen speziellen Befehl kann die Vektortabelle ab einem bestimmten Index mit entsprechenden Kreisdaten gefüllt werden, sofern die entsprechenden Basisparameter vorher korrekt gesetzt worden sind.

Über einen Skalierungsfaktor, der die Weginkremente der beiden Achsen in eine bestimmte Beziehung zueinander setzt, ist es möglich, unterschiedliche Achsauflösungen zu kompensieren oder elliptische Konturen zu erzeugen.

Definitionen:

Nummer der Sekanten: $k \in (1,..., m)$; m = Gesamtanzahl Sekanten

Startwinkel (Winkeloffset) des Kreissegments: α

Vom Kreissegment abzudeckender Winkelbereich: $\Delta\alpha$

Radius des Kreissegments: r

Veranschaulichung am Diagramm:

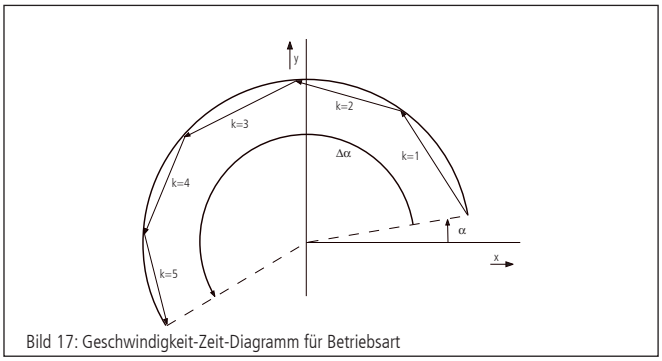

Bild 17: Geschwindigkeit-Zeit-Diagramm für Betriebsart

hier: Teilkreis mit Radius r, Winkeloffset $\alpha = 10°$,
Winkelbereich $\Delta\alpha = +190°$, m = 5 Sekanten

Berechnung

Das zu approximierende Kreissegment wird über den Radius, die Sekantenanzahl, Winkeloffset und Winkelbereich definiert.

Die Drehrichtung wird über das Vorzeichen der Winkelbereichsangabe festgelegt. Hierbei entspricht ein positiver Winkel einer Drehung im Gegenuhrzeigersinn bei entsprechender Anordnung der Achsen (siehe auch Lage des Koordinatensystems in oben genanntem Diagramm).

Der Startwinkel der einzelnen Sekantenvektoren k ergibt sich zu:

$$\alpha_k = \alpha + \Delta\alpha \cdot \frac{k-1}{m}$$

Die x- und y-Koordinaten der Sekantenvektoren sind dann:

$$\Delta x_k = -2r \cdot \sin\left(\frac{\Delta\alpha}{2m}\right) \cdot \sin\left(\alpha_k + \frac{\Delta\alpha}{2m}\right) \quad \text{und}$$

$$\Delta y_k = 2r \cdot \sin\left(\frac{\Delta\alpha}{2m}\right) \cdot \cos\left(\alpha_k + \frac{\Delta\alpha}{2m}\right)$$

Mit $\left|2r \cdot \sin\left(\frac{\Delta\alpha}{2m}\right)\right|$ wird die Länge eines Sekantenvektors bezeichnet.

Skalierungsfaktor

Der Skalierungsfaktor zum Ausgleich unterschiedlicher Auflösungswerte der beiden Kreisinterpolations-Achsen bzw. zur Realisierung von Ellipsen wird über Zähler und Nenner dargestellt, die über zwei separate Kommandos gesetzt werden können. Der Nenner sei mit N, der Zähler mit Z bezeichnet.

Falls N > Z, führt die Y-Achse und die Wegangaben für X werden durch (N / Z) dividiert. Falls Z > N, führt die X-Achse und die Wegangaben für Y werden durch (N / Z) dividiert. Der Standardwert ist Z = N = 1, falls seitens des Anwenders keine Angaben gemacht werden.

Syntax

Ab Tabellenelement <n> werden über den Befehl „PTABCIRCLE" Kreisdaten in Form von <m> Sekantenvektoren generiert und zur Steuerung übertragen. Hierbei bedeutet Angabe von Null für eine Achsnummer, dass die Achse nicht verwendet wird. Die Syntax ist wie folgt:

PTABCIRCLE <n> = <Achsnummer x>, <Achsnummer y>,
Δt_n, f_n, m_n, r_n, α_n, $\Delta\alpha_n$, Z_n, N_n

Beispiel:

PTABCIRCLE0=1,2,326,0,5,1000,10,190,1,1

generiert einen Teilkreis ab Tabellenelement 0 mit Achse 1 als X- und Achse 2 als Y-Achse, Segmentzeit 1/3 Sekunde, Betriebsart v=const., 5 Sekanten, Radius 1000 Inkremente, Startwinkel 10°, Winkelbereich 190° und Skalierung 1/1.

9. Wegerfassung

Encoder
Der Encoder ist ein auch als Drehgeber bezeichnetes Wegerfassungssystem zur Positionsrückmeldung, das für den Motorcontroller im geregelten (Closed-Loop) Betrieb genutzt wird.

Ohne Encoder ist nur der gesteuerte Betrieb (Open Loop) mit Schrittmotoren möglich. Um DC-Motoren betreiben zu können, muss ein Wegerfassungssystem angeschlossen sein. Dies kann ein Encoder sein. Üblicherweise besitzen sie 500, 1250 oder 2500 Linien pro Umdrehung. Über den Encoder erfasst der Motion-Controller die aktuelle Position der Achse und berechnet aus der zeitlichen Veränderung der Positionswerte die aktuelle Geschwindigkeit des Rotors.

Encoder sind fest am Motor angeflanscht und direkt mit dem Rotor verbunden. Die Signale des Encoders sind Kanal A und B (CHA und CHB), 90 Grad versetzt (Quadratur-Signale), und ein Index-Impuls pro Umdrehung. Die PS 30 kann als Encodersignale TTL-Pegel oder antivalente Signale (über Leitungstreiber) verarbeiten.
Die Signale werden nach einer Pegelumwandlung und Filterung direkt an den Motion-Controller weitergegeben.

Linearmesssystem
Ein Messsystem, welches direkt an die Bewegung des Aktors gekoppelt ist, nennt man Linearmesssystem. Das Wegmesssystem kann entweder alternativ zum Encoder der Wegerfassung dienen oder zusätzlich zu einem vorhandenen Encoder zum Nachführen des Positioniersystems auf die Zielposition verwendet werden. Dieses Verfahren nennt sich Nachlaufregelung. Hierbei ist dann Korrektur systematischer Fehler (z.B. Spindelsteigungsfehler) möglich.

Die Zielposition wird bei Verwendung eines Wegmesssystems zur Nachlaufregelung separat (32-Bit Auflösung) angegeben. Der eigentliche Positioniervorgang wird dann vom Motion-Controller über Encoder durchgeführt. Meldet dieser „Position erreicht", dann führt der Hauptprozessor die Position solange nach, bis die vom Messsystem erfaßte exakte Zielposition innerhalb des definierten Zielfensters liegt.

Auswertung des Linearmesssystems
Die PS 30 kann optional mit einer Wegmessplatine bestückt werden. Die Signale des Wegmesssystems entsprechen den vorher genannten Encodersignalen (Quadratur A und B, sowie Index). Auf der Wegmessplatine ist für jede Achse ein 32-Bit-Zähler vorgesehen, der die Signale des Wegmesssystems zählt. Die Signale werden vom Hauptprozessor ausgelesen. Die maximale Zählfrequenz beträgt 5,5 MHz (Signal) bzw. 22 MHz (Quadratur).

Lageregelung
Für den Betrieb von Servomotoren sind zwei Encodereingänge vorgesehen. Der erste Encodereingang dient der Datengewinnung für den Lage-Regelkreis (PID-Lageregelung). Der zweite Encodereingang (optionaler Dual-Loop-Encoder) kann für Positions-Nachlaufregelung benutzt werden.

Funktionsweise der Nachlaufregelung
Um eine Nachlaufregelung für eine bestimmte Positioniereinheit realisieren zu können, ist es erforderlich, die Positioniereinheit mit einem zusätzlichen inkrementalen Linearmesssystem auszustatten, welches die reale Position des Schlittens unter Zuhilfenahme einer eindeutigen Referenzmarke erfasst. Die aus Motor und Antriebsspindel bestehende Antriebseinheit (nachfolgend als "Aktor" bezeichnet) wird über die Steuerung auf die reale Absolutposition nachgeführt (nachgeregelt). Dies kann durch iterative Korrekturbewegungen oder Korrekturfahrt mit konstanter Geschwindigkeit erfolgen. Eine Kombination beider Verfahren ist ebenfalls möglich. Die Auswahl wird über die Betriebsartenvorwahl der Nachlaufregelung vorgenommen. Die Werte für die rechnerische Auflösung von Linearmesssystem und Positioniereinheit sind in der Regel unterschiedlich.

Vor Verwendung der Nachlaufregelung ist eine Referenzierung in Referenzfahrmodus 6 oder 7 durchzuführen. Hierbei wird der insgesamt zur Verfügung stehende Hub in Inkrementen des Linearmesssystems gemessen und der Absolutpositionszähler automatisch bei Überfahren der Referenzmarke des Linearmesssystems auf Null gesetzt.

Die Zielposition einer nachlaufgeregelten Positioniereinheit wird über die nach erfolgreicher Referenzfahrt definierte Absolutposition des Linearmesssystems angegeben, d.h. eine Zielposition wird als Absolut- oder Relativdistanz angegeben, bezogen auf ein ganzzahliges Vielfaches des Weginkrements des Linearmesssystems, den Referenzpunkt und ggf. die aktuelle Position.

Zur steuerungsinternen Berechnung der Wegstrecke des Aktors wird das Verhältnis zwischen Weginkrement des Aktors und Weginkrement des Linearmesssystems über einen Umrechnungsfaktor F= Z / N definiert, der sich aus dem Quotienten beider Auflösungswerte ergibt.

Ein Positioniervorgang mit Nachlaufregelung entspricht folgendem 3-Phasen-Schema, wobei je nach Einstellung nicht alle Phasen auftreten müssen:

- Mittels des gegebenen Umrechnungsfaktors (Z / N) wird aus den gegebenen Positionsdaten die zu verfahrende Relativdistanz des Aktors berechnet.
- Die so berechnete Distanz wird verfahren (Phase 1, Grobpositionierung) und die Abweichung zur Sollposition berechnet.
- Liegt die Istposition außerhalb des definierten Zielfensters, erfolgt, falls gewünscht, eine iterative Annäherung, d.h. es wird zyklisch eine Relativdistanz des Aktors berechnet und an den Motor ausgegeben usw. (Phase 2, Iteration).
- Hierbei gilt als Konvergenzkriterium, dass sich der Betrag der Lageabweichung bei jedem Iterationsschritt verringern muss, bis die Istposition schließlich innerhalb des Zielfensters liegt. Daraus folgt als Divergenzkriterium für die Iteration, dass der Abbruch der Iteration dann erfolgt, wenn der Betrag der Lageabweichung nach Korrekturfahrt (n) größer oder gleich dem Betrag der Lageabweichung nach Korrekturfahrt (n-1) ist.
- Nach erfolgreichem Abschluss (Konvergenz, Istposition liegt innerhalb Zielfenster) oder Abbruch (Divergenz) der Iteration folgt optional eine Korrekturphase im Geschwindigkeitsmodus (Phase 3). Ob Phase 3 aktiv ist oder nicht, ist wählbar, d.h. sie wird über einen Parameter vorgegeben.
- In der anschließenden Korrekturphase wird die Istposition des Linearmesssystems abgefragt. Liegt die Istposition außerhalb des Zielfensters, wird der Geschwindigkeitsmodus mit der vorher definierten Nachlaufgeschwindigkeit als Parameter aufgerufen. Sobald die Istposition innerhalb des Zielfensters liegt, stoppt der Nachführvorgang, d.h. es wird eine Bremsrampe ausgelöst. Fährt der Aktor über das Ziel hinaus, erfolgt eine Drehrichtungsumkehr usw.
- Über einen weiteren Parameter kann vorgegeben werden, ob die Nachführung im Geschwindigkeitsmodus ständig aktiv sein soll oder beim ersten Erreichen des Zielfensters abschaltet.

Berechnung des Umrechnungsfaktors F:

Bei nachlaufgeregeltem Betrieb werden Fahrdistanzen grundsätzlich in Vielfachen der Messsystemauflösung (Weginkrement des Linearmesssystems) angegeben. Die Auflösung des Aktors ist bestimmt durch die Motorauflösung (z.B. Mikroschrittfaktor, Encoderinkrement) und die mechanischen Parameter (z.B. Spindelsteigung).

Aus der gegebenen Fahrdistanz muss die zurückzulegende Relativdistanz des Aktors vor jeder Fahrt berechnet werden.

Nachfolgend soll die Berechnung beispielhaft für einen Lineartisch mit Spindel-Direktantrieb und 2-Phasen-Schrittmotor (ungeregelt) durchgeführt werden.

$$F = \frac{Z}{N} = \frac{r_s}{r_m} = \frac{\text{Auflösung des Aktors}}{\text{Auflösung des Messsystems}}$$

Berechnung von r_s:

$$r_s = \frac{h}{n \cdot m}$$

wobei:

h = Spindelsteigung (Verstellweg pro Motorumdrehung),
n = Motorschrittzahl (Vollschritte pro Motorumdrehung),
m = Mikroschrittfaktor (Mikroschritte pro Vollschritt)

Beispiel:

h = 5 mm,
n = 200,
m = 50

Es ergibt sich:

$$r_s = \frac{5\,\text{mm}}{200 \cdot 50} = 0{,}5\,\mu\text{m}$$

Die Auflösung des Messsystems r_m ist gegeben, z.B.:

$r_m = 0{,}1\,\mu\text{m}$

Somit ist im Beispiel

$$F = \frac{r_s}{r_m} = \frac{0{,}5\,\mu\text{m}}{0{,}1\,\mu\text{m}} = \frac{5}{1} = \; : \frac{Z}{N}$$

und damit:

Z = 5

N = 1.

10. PID-Regelschleifenalgorithmus

Das in der PS 30 benutzte Servofilter arbeitet nach einem PID-Algorithmus. Ein Integrationslimit sichert nach oben gegen einen akkumulierten Fehler ab.

Die PID-Formel lautet wie folgt:

$$\text{Output}_n = K_p E_n + K_d (E_n - E_{(n-1)}) + \sum_{j=0}^{n} E_j \frac{K_i}{256}$$

Hierbei ist:

E_n	Regelabweichung zum diskreten Zeitpunkt n
K_i	Integralanteil des Lagereglers
K_d	Differentialanteil des Lagereglers
K_P	Proportionalanteil des Lagereglers

Alle Filterparameter und die Drehmomentsignalbegrenzung sind programmierbar, so dass der Filter durch den Anwender fein abgestimmt werden kann. Wertebereiche und Formate werden in der folgenden Tabelle aufgelistet:

Terminus	Name	Bereich
I_{lim}	Begrenzung Integralanteil	32 Bit unsigned (0…2.124.483.647)
K_i	Integralanteil des Lagereglers	16 Bit unsigned (0…32.767)
K_d	Differentialanteil des Lagereglers	16 Bit unsigned (0…32.767)
K_p	Proportionalanteil des Lagereglers	16 Bit unsigned (0…32.767)

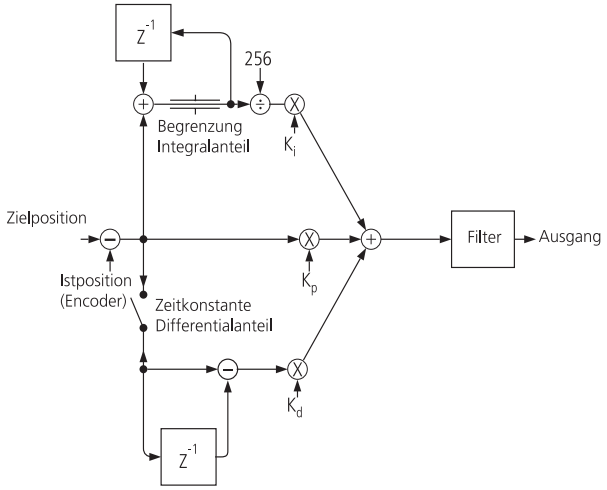

Bild 18: Struktur des digitalen Filters

11. Positioniergeschwindigkeit und - beschleunigung, Berechnung

11.1 2-Phasen-Schrittmotor (Open Loop)

Allgemeines
Jede schrittmotorgetriebene Mechanik besitzt eine insbesondere von Motortyp, Systemreibung und Last abhängige sog. Start-Stop Frequenz. Die Start-Stop-Frequenz bezeichnet die maximale Fahrfrequenz des betreffenden Schrittmotors, mit der dieser noch aus dem Stillstand ohne Beschleunigungsphase starten kann. Es ist üblich, diese und andere Kennfrequenzen von Schrittmotoren in Hertz Vollschritt (HzVS), das heißt Vollschritte pro Sekunde, anzugeben. Die Welle eines Schrittmotors mit Schrittwinkel 1,8°, d.h. R = 200 Vollschritte pro Motorumdrehung, der z.B. mit 400 HzVS läuft, dreht mit einer Geschwindigkeit von zwei Umdrehungen pro Sekunde oder 120 Umdrehungen pro Minute.

Um höhere Geschwindigkeiten als die Start-Stop-Frequenz zu erreichen, muss der Schrittmotor über diese Frequenz hinaus mittels geeigneter Beschleunigungsrampe beschleunigt, bzw. unter diese Frequenz mittels geeigneter Bremsrampe abgebremst werden. Diese Beschleunigung bzw. Bremsung erfolgt mittels trapezförmigem oder S-förmigem Geschwindigkeit-Zeit-Profil. Gegebenenfalls ist eine Dämpfung (Viskosedämpfer, am zweiten Wellenende des Motors montiert) erforderlich, um überhaupt höhere Drehzahlen erreichen zu können.

Fast alle Standard-Schrittmotoren, die bei OWIS® eingesetzt werden, sind in der Lage, einer Frequenz von 400 HzVS im Start-Stop-Betrieb zu folgen.

Die PS 30 besitzt einen digitalen Profilgenerator. Die Geschwindigkeitsprofile werden periodisch berechnet und an den 2-Phasen-Schrittmotor ausgegeben.

Periodendauer
Die Periodendauer des digitalen Profilgenerators ist durch die Hardware festgelegt.

$T_P = 256\,\mu s$

Endgeschwindigkeit
Die Positionierung der Achsen wird im Punkt-zu-Punkt-Verfahren vorgenommen. Hierbei beschleunigt jede Achse wahlweise mit trapezförmigem oder S-förmigem Geschwindigkeits-Profil.

Die Endgeschwindigkeit V nach der Beschleunigungsrampe wird als 32-Bit-Wort angegeben. Ihr Wertebereich reicht von 1 bis 2147483647.

> **! Hinweis:**
> Keinesfalls darf eine höhere Geschwindigkeit vorgegeben werden, als die Mechanik in der Lage ist zu fahren, da sonst die angeschlossene Mechanik beschädigt oder zerstört werden kann.

Bei gegebener Geschwindigkeit V und gegebenem Mikroschrittfaktor Mcstp errechnet sich die Schrittfrequenz f wie folgt:

$$f_{Mcstp} = \frac{1}{T_P} \cdot \frac{V}{65536}$$ (Schrittfrequenz im Mikroschrittmodus)

bzw.

$$f_{VS} = \frac{1}{Mcstp \cdot T_P} \cdot \frac{V}{65536}$$ (auf Vollschrittmodus normierte Schrittfrequenz)

Hieraus ergibt sich die Motordrehzahl n_{RPM} (ohne Berücksichtigung eines evtl. vorhandenen Getriebes) bei einem Schrittmotor mit R Vollschritten pro Motorumdrehung:

$$n_{RPM} = \frac{60}{min} \cdot \frac{1}{Mcstp \cdot R \cdot T_P} \cdot \frac{V}{65536}$$ (Umdrehungen/Minute)

bzw.

$$n_{RPS} = \frac{1}{s} \cdot \frac{1}{Mcstp \cdot R \cdot T_P} \cdot \frac{V}{65536}$$ (Umdrehungen/Sekunde)

Für die Umrechnung von der Motordrehzahl in eine Positioniergeschwindigkeit der Mechanik sind zusätzlich die mechanischen Daten, wie z.B. Spindelsteigung und ggf. die Getriebeübersetzung, zu berücksichtigen.

Beschleunigung bei Trapezprofil
Als Beschleunigung („ACC") ist ein 32-Bit-Wort anzugeben, der Wertebereich reicht von 1 bis 2147483647.

Dauer der Trapezprofil-Beschleunigungsrampe bei gegebener Geschwindigkeit V und Beschleunigung ACC:

$$\Delta t = 1\,s \cdot \frac{V \cdot T_P}{ACC}$$ (Anlauf-/Nachlaufdauer in Sekunden)

Zurückgelegte Distanz während der Trapezprofil-Beschleunigungsrampe:

$$\Delta s = 1\,\text{Mikroschritt} \cdot \frac{V^2}{131072 \cdot ACC}$$ (Nachlaufweg in Mikroschritten)

11.2 DC-Servomotor und 2-Phasen-Schrittmotor (Closed-Loop)

Allgemeines
Die PS 30 hat einen digitalen Lage-/Geschwindigkeits-Regler. Stell- und Regelgröße werden periodisch berechnet. Die Erfassung des Positions-Istwertes geschieht im einfachsten Fall mittels eines Drehgebers, der am 2. Wellenende des Motors angeflanscht ist. Wichtigste Kenngröße des Encoders ist die Encoder-Strichzahl R.
Sie gibt die Anzahl der sog. Linien, d.h. Hell-Dunkel-Perioden je Motorwellenumdrehung, an. Die Signale durchlaufen eine Vierfach-Auswertung, woraus sich generell eine vierfach höhere Auflösung als die Encoder-Strichzahl ergibt.

Abtastzeit
Die Periodendauer des digitalen Reglers wird auch als Abtastzeit bezeichnet und ist durch die Hardware festgelegt. Die minimale Abtastzeit beträgt 204,8 µs. Sie kann bei Bedarf um ganzzahlige Vielfache von 51,2 µs erhöht werden:

$T_s = 204{,}8\,\mu s + n \cdot 51{,}2\,\mu s;\ n \in [0, 1, ..., 386]$

entsprechend einer Abtastzeit von

$T_s = [204{,}8\,\mu s, 256\,\mu s, ..., 19986\,\mu s]$.

Als Abtastzeit können nur ganzzahlige Werte an die PS 30 übergeben werden. Der Wert wird intern auf den nächsten gültigen Wert gerundet.

Standardwert (Voreinstellung): $T_s = 256\,\mu s$.

Endgeschwindigkeit
Die Positionierung der Achsen wird im Punkt-zu-Punkt-Verfahren vorgenommen. Hierbei beschleunigt jede Achse wahlweise mit trapezförmigem oder S-förmigem Geschwindigkeits-Profil.

Die Endgeschwindigkeit V nach der Beschleunigungsrampe wird als 32-Bit-Wort angegeben. Ihr Wertebereich reicht von 1 bis 2147483647.

> **Hinweis:**
> Keinesfalls darf eine höhere Geschwindigkeit vorgegeben werden, als die Mechanik in der Lage ist zu fahren, da sonst die angeschlossene Mechanik beschädigt oder zerstört werden kann.

Bei gegebener Geschwindigkeit V und der Encoder-Linienzahl R errechnet sich die Motordrehzahl (ohne Berücksichtigung eines evtl. vorhandenen Getriebes) wie folgt:

$$n = \frac{60}{\text{min}} \cdot \frac{1}{T_S} \cdot \frac{1}{4R} \cdot \frac{V}{65536} \quad \text{(Umdrehungen pro Minute)}$$

bzw.

$$n = \frac{1}{s} \cdot \frac{1}{T_S} \cdot \frac{1}{4R} \cdot \frac{V}{65536} \quad \text{(Umdrehungen pro Sekunde)}$$

bzw.

$$n = \frac{1\,\text{Inkrement}}{s} \cdot \frac{1}{T_S} \cdot \frac{V}{65536} \quad \text{(Inkremente pro Sekunde)}$$

Die letzte Formel kann auch wie folgt verstanden werden:
Der Controller verfährt V / 65536 Inkremente je Abtastintervall T_S.

Für die Umrechnung von der Motordrehzahl in eine Positioniergeschwindigkeit der Mechanik sind zusätzlich die mechanischen Daten, wie z.B. Spindelsteigung und ggf. die Getriebeübersetzung, zu berücksichtigen.

Beispiel:

Es ist eine Positionierung mit einer Nenndrehzahl n = 1800 U/min auszuführen. Es wird ein Encoder mit R = 500 Linien (entspr. 2000 Impulsen/Umdrehung) am Motor eingesetzt.

Wie ist V zu wählen?

Lösung:

Es ergibt sich allgemein nach Umstellen der Drehzahlgleichung für die Geschwindigkeit:

$$V = \frac{n}{60} \cdot 4 \cdot R \cdot 65536 \cdot T_S$$

Damit wird V = 1006633 für n = 1800 U/min bei Einsatz eines 500-Linien-Encoders. Unter Verwendung einer direktgetriebenen Spindel mit 1 mm Steigung entspricht dies einer Verstellgeschwindigkeit von genau 1,8 m/min. bzw. 30 mm/s.

Beschleunigung bei Trapezprofil

Als Beschleunigung („ACC") ist ein 32-Bit-Wort anzugeben, der Wertebereich reicht von 1 bis 2147483647.

Dauer der Trapezprofil-Beschleunigungsrampe bei gegebener Geschwindigkeit V und Beschleunigung ACC:

$$\Delta t = 1\,s \cdot \frac{V \cdot T_S}{ACC} \quad \text{(Anlauf-/Nachlaufdauer in Sekunden)}$$

Zurückgelegte Distanz während der Trapezprofil-Beschleunigungsrampe:

$$\Delta s = 1\,\text{Inkrement} \cdot \frac{V^2}{131072 \cdot ACC} \quad \text{(Nachlaufweg in Inkrementen)}$$

12. Inbetriebnahme der PS 30

12.1 Einbau und Vorbereitung

Hinweis:
Der Einbau von PCI-Einsteckkarte und Endstufenmodul in den Steuerungs-PC darf nur bei stromlosem PC, d.h. bei gezogenem Netzstecker, vorgenommen werden!

Vor dem Einbau sind geeignete Steckplätze für PCI- und Endstufenkarte auszuwählen, und beide Karten müssen über die mitgelieferten drei 50-poligen Flachbandkabel miteinander verbunden werden. Die Länge der Flachbandkabel ist so dimensioniert, dass das Endstufenmodul bis zu drei Steckplätze benachbart der PCI-Einsteckkarte eingebaut werden kann.

Die Flachbandkabel sollen so montiert werden, dass keine Kreuzungen entstehen. Werden die Kabel von der PCI- zur Endstufenkarte versehentlich über Kreuz angeschlossen, entsteht kein Schaden, jedoch ist dann die Achszuordnung X-Y-Z bzw. 1-2-3 vertauscht und entspricht nicht mehr den Tabellen im vorliegenden Manual bzw. der Adapterkabelkennzeichnung.

Hinweis:
Bitte achten Sie darauf, dass sämtliche Arbeiten an den Karten ausschließlich an einem vor statischer Aufladung gesicherten Arbeitsplatz (ESD-geschützter Bereich) durchgeführt werden, und kontrollieren Sie den korrekten Sitz insbesondere der PCI-Einsteckkarte nach dem Einbau. Der PCI-Stecker darf sich keinesfalls während des Betriebs lösen, sonst können PC und PCI-Einsteckkarte zerstört werden.

12.2 Anschluss der Peripherie und Geräte

Vor dem Einschalten der Steuerung müssen sämtliche Anschlussstecker für Geräte und Peripherie angeschlossen sein, damit sie von der Steuerung erkannt und initialisiert werden.

Es müssen:

- die Positioniereinheit
- die Stromversorgung
- der Computer

angeschlossen werden.

Die Verbindung zum Computer erfolgt über die PCI-Schnittstelle.

Dafür ist eine Treiberinstallation notwendig. Der Treiber befindet sich auf der mitgelieferten CD.

Für die Installation starten Sie bitte „setup. exe".

Hinweis:
Jegliche Geräte und Peripherie müssen vor dem Systemstart angeschlossen sein, da sie sonst von der Steuerung nicht erkannt und initialisiert werden.

12.3 Systemstart

Beim ersten Windows-Start mit eingebauter PS 30 sollte das Betriebssystem die neue Hardware erkennen. Die Treiber können nun installiert werden. Hierzu sind ggf. Administratorrechte erforderlich.

Initialisierung

Nachdem die Stromversorgung eingeschaltet und das Gerät aktiviert wurde, muss jede Achse, die verwendet werden soll, zunächst per INIT-Befehl initialisiert werden.

Achsenparameter, die verändert wurden, werden ebenfalls mit der Initialisierung übernommen.

Software

Zum Lieferumfang der Steuerung gehören das Softwaretool OWISoft, der PCI-Treiber (PCI-COM-Brücke) und die Software-Schnittstelle (SDK/API) für C, C++, C#, LabView (ab Version 8.2) und zusätzliche Programmiersprachen (32/64-Bit). Damit kann die PS 30 komfortabel konfiguriert und betrieben werden.

Unterstützte Betriebssysteme: Windows XP, Windows Vista (32/64-Bit), Windows 7 (32/64-Bit), Windows 8 (32/64-Bit), Windows 8.1 (32/64-Bit), und Windows 10 (32/64-Bit).

Die Software-Schnittstelle enthält Beispielprogramme mit Quellcode und Hilfedateien.

Für die Inbetriebnahme mit OWISoft sind die jeweiligen Parameter der Positionierer für die Achsen hinterlegt, die nur noch angewählt werden müssen.

Hinweis:
Die hinterlegten Parameter sind für unbelastete Positionierer voreingestellt. Für optimalen Lauf müssen die Reglerparameter der konkreten Belastungen angepasst werden.

Lesen Sie hierfür bitte die Bedienungsanleitung OWISoft.

Für die Inbetriebnahme mittels eigener Applikationssoftware lesen Sie bitte das Kapitel "Hinweise zum Aufbau einer eigenen Applikationssoftware". Dort ist im Anschluss auch eine Tabelle mit den Befehlssätzen der PS 30 angefügt.

13. Fehlerüberwachung

13.1 Endschalter

Die PS 30 besitzt zwei Eingänge für Endschalter (MINSTOP, MAXSTOP) sowie Auswertemöglichkeit für einen Referenzschalter je Achse. Als Referenzschalter ist einer der beiden Endschalter definiert.

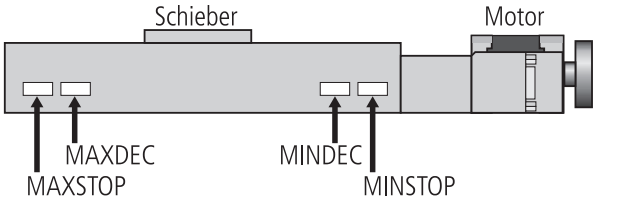

Bild 18: Endschalterkennzeichnung

OWIS® Positioniereinheiten besitzen maximal vier Endschalter. Die Endschalter in negativer Fahrrichtung (Bewegung des Schiebers zum Motor hin) werden mit MINDEC und MINSTOP bezeichnet. Die Endschalter in positiver Fahrrichtung (Bewegung des Schiebers vom Motor weg) werden mit MAXDEC und MAXSTOP bezeichnet. MINDEC und MAXDEC können an der PS 30 nicht ausgewertet werden.

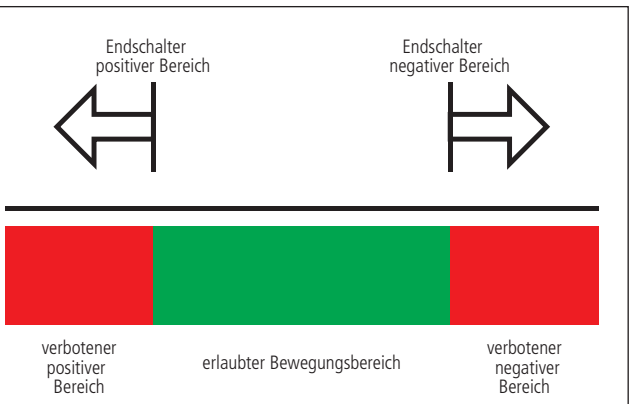

Bild 19: Endschalterfunktionen

Funktion der Endschalter-Überwachung

1. MINSTOP: Auslösen dieses Schalters bei Fahrt in negative Richtung bewirkt nach einer gewissen Reaktionszeit, die einige Millisekunden betragen kann, einen sofortigen, abrupten Motorstop. Der Motor wird hierbei stromlos geschaltet.

   DC-Servomotor: Der Motor wird stromlos geschaltet, jedoch führt die vorhandene kinetische Energie zu einer Restbewegung, bis sie durch Reibung oder mechanische Anschläge verbraucht wurde.

   Schrittmotor (Open Loop): Fals die aktuelle Fahrfrequenz, von der aus gestoppt wurde, höher gewesen ist als die Start-Stop-Frequenz des Systems, führt dies auf Grund der kinetischen Energie im System dazu, dass der Motor noch eine Bewegung ausführt. Dies kann von der Steuerung nicht erfasst werden, so dass der angezeigte Positionswert falsch ist. Eine Referenzfahrt ist nötig, um die Motorschritte wieder mit der angezeigten Position übereinstimmen zu lassen.
2. MAXSTOP: Die Reaktion ist äquivalent zum MINSTOP-Endschalter, jedoch wirkt dieser Schalter nur bei Fahrt in positiver Richtung.

Konfiguration der End- und Referenzschalter

Welche Endschalter an der jeweils angeschlossenen Positioniereinheit vorhanden sind, kann mit dem Befehl „SMK...“ definiert werden. Ein gesetztes Bit (=1) bedeutet, dass der jeweilige Schalter ausgewertet wird.

MINDEC und MAXDEC sind grundsätzlich per Software zu deaktivieren, d.h. die Bits der End- bzw. Referenzschaltermaske („SMK...“/„RMK...“), welche MINDEC und MAXDEC repräsentieren, müssen bei der Konfiguration der Achsen auf Null gesetzt werden. Die Signale werden von der Hardware nicht ausgewertet.

Die Endschalterpolarität wird mit dem Kommando „SPL...“ vorgewählt. Der übergebene Wert definiert, ob Endschalter bzw. Referenzschalter „low“ oder „high“ aktiv sein sollen. Ein gelöschtes Bit bedeutet, dass der jeweilige Schalter „low“ aktiv ist (z.B. Schließerkontakt nach Masse, d.h. offen in nicht betätigtem Zustand). Ein gesetztes Bit (Standardkonfiguration) bedeutet, dass der jeweilige Schalter „high“ aktiv ist (z.B. Öffnerkontakt nach Masse, d.h. geschlossen in nicht betätigtem Zustand).

Die Endschaltereingänge arbeiten standardmäßig mit 5 V-CMOS-Pegel, wobei Open-Collector-NPN- oder Push-Pull-Ausgänge gleichermaßen angeschlossen werden können, da hochohmige Pullup-Widerstände (4,7 kOhm) nach + 5 V bereits geräteintern vorgesehen sind.

Wiederinbetriebnahme nach Achsenfehler

Nachdem ein Achsenfehler durch Betätigung eines Limit-Schalters (MINSTOP oder MAXSTOP) aufgetreten ist, wird die Achse <n> wie folgt wieder in Betrieb genommen:

1. Initialisierung mittels Befehl INIT<n>
2. Freifahren des Limit-Schalters mittels Befehl EFREE<n>

13.2 Endstufen-Fehlerüberwachung

Jede Endstufe meldet mit einer digitalen Leitung ihren Status an den Mikrocontroller zurück. Dieses Signal wird zyklisch kontrolliert. Meldet eine Endstufe einen Fehler, so wird der Antrieb stromlos geschaltet, d.h. die Regelschleife wird geöffnet und das Endstufen-Freigabe-Signal wird inaktiv gesetzt.

13.3 Motion-Controller-Fehlerüberwachung

Die Kommunikation mit den Motion-Controllern wird ebenfalls überwacht. Treten dabei Fehler oder Unplausibilitäten auf, so wird der Antrieb stromlos geschaltet, d.h. die Regelschleife wird geöffnet und das Endstufen-Freigabe-Signal wird inaktiv gesetzt.

13.4 Time-Out-Überwachung

Für jede Achse kann zusätzlich als Parameter eine Timeout-Zeit (in ms, Wertebereich 32 Bit) definiert werden. Die Überwachung kann durch die Einstellung Timeout-Zeit = 0 abgeschaltet werden. Während eine Bewegung (PGO, REF, EFREE, PWMSGO, LIGO) durchgeführt wird, wird zyklisch diese Timeout-Zeit überwacht. Dauert die Bewegung länger als diese Zeit, so wird der Antrieb stromlos geschaltet (?ASTAT → „Z“, siehe Befehlssatz ab S.24), d.h. die Regelschleife wird geöffnet und das Endstufen-Freigabe-Signal wird inaktiv gesetzt. Diese Funktion ist nützlich, wenn z.B. bei der Referenzfahrt der Referenzschalter nicht gefunden wird.

14. Hinweise zum Aufbau einer eigenen Applikationssoftware

Eine PS 30-Applikation besteht allgemein aus einem Initialisierungsteil, welcher die erforderlichen Achsparameter für alle zu verwendenden Achsen <n> setzt und die Achsen einschaltet, einer Schleife, die eine Referenzfahrt für alle Achsen durchführt, und dem eigentlichen Anwenderprogramm, welches die vom Anwender gewünschte Funktionalität beinhaltet.

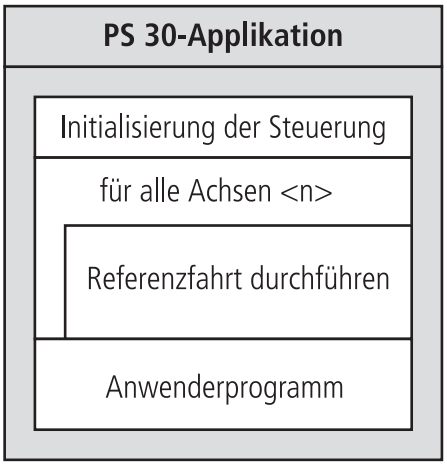

Bild 20: PS 30-Applikations-Architektur

Die Initialisierung der gewünschten Achsen geschieht im einfachsten Fall über das INIT-Kommando, falls die im statischen RAM gespeicherten Parameter übernommen werden sollen. Andernfalls ist es erforderlich, die gewünschten Parameter vor Senden des INIT-Kommandos zu übertragen.

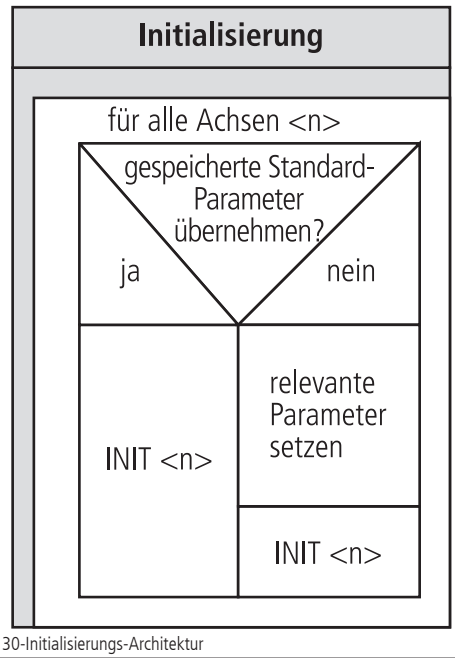

Bild 21: PS 30-Initialisierungs-Architektur

Soll eine Referenzfahrt für eine Achse durchgeführt werden, sind Referenzmaske und Referenzpolarität vorher zu setzen, falls dies nicht bereits erfolgt ist oder entsprechende Werte in den Standardeinstellungen hinterlegt worden sind. Danach wird die Referenzfahrt gestartet.

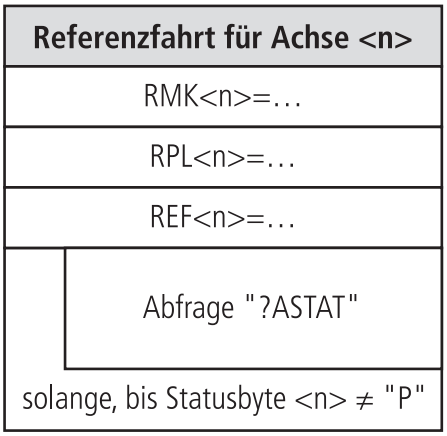

Bild 22: PS 30-Referenzfahrt für Achse

Zwischen zwei einzelnen Befehlen, die zur PS 30 gesendet werden, ist eine Verarbeitungszeit (Interpreterzeit) von ca. 20 bis 40 Millisekunden zu berücksichtigen. Empfangene Gerätemeldungen können z.B. Zeichen für Zeichen im Millisekunden-Takt abgeholt werden, bis die definierte Stringende-Kennung empfangen wird.

Eine Verwendung des mitgelieferten Softwarepakets OWISoft (inklusive SDK und DLL) erleichtert die Inbetriebnahme wesentlich, da häufig verwendete Befehlsfolgen bereits als Funktionen bzw. Prozeduren zusammengefasst sind, und der erforderliche Laufzeitabgleich ebenfalls implementiert ist.

15. Befehlssatz der PS 30

Generelles zum Format der Befehle:

Jeder Befehl wird über die Schnittstelle (PCI oder RS-232) in Form von ASCII-Zeichen übertragen. Die einzelnen Zeichen eines Befehls werden automatisch in Grossbuchstaben umgewandelt. Jeder Befehl wird mit CR oder CR+LF oder LF (einstellbar) abgeschlossen. Weiterhin ist der Antwortmodus einstellbar (TERM). Dazu gibt es drei Einstellungen:

1) Beim Auslesen des Message-Ausgangs-Buffers wird nur eine zweistellige Zahl zurückgegeben (Fehlercode). Diese Einstellung wird vorzugsweise bei Ansteuerung über Software gewählt, da die Gerätemeldungen hier am kürzesten sind, womit der Befehlsdurchsatz optimiert wird.
2) Beim Auslesen des Message-Ausgangs-Buffers wird eine zweistellige Zahl mit Klartext ausgegeben.
3) Wie 2) und zusätzlich wird jeder ausgeführte Befehle, der keinen Wert zurückmeldet, mit „OK" quittiert.

Rückmeldungen werden auch entweder mit CR oder CR + LF oder LF zurückgesendet (einstellbar).

Im ersten Antwortmodus (TERM=0), werden die binären Informationen (z.B. Endschalterkonfiguration, Endschalterstatus, digitale/analoge Eingänge/Ausgänge usw.) als Bits einer Dezimalzahl angegeben. In den anderen Modi (TERM=1, TERM=2) werden diese Werte als binäre Zahl angegeben. Dies gilt sowohl für die Abfrage, als auch für die Einstellung eines Wertes.

Alle Parameter werden resident abgespeichert und mit einer Checksumme versehen. Nach dem Aus- und erneutem Einschalten des Gerätes ist der letzte Stand der Parameter wieder gültig. Sollte die Checksumme nicht mehr stimmen, so werden beim Einschalten automatisch die Werte aus dem FRAM geladen und eine Fehlermeldung in den Fehlerspeicher eingetragen.

Bei Befehlen mit einer Rückantwort (z.B. Abfragen von Parametern) wird die Antwort sofort zum PC zurückgeschickt.

<n> = Achsennummer 1...3 (bzw. höchste Achsennummer)
<uv> = Zahlenwert ohne Vorzeichen
<sv> = Zahlenwert mit Vorzeichen
<v> = vorzeichenbehaftete Wegangabe

Anhang
I Befehlstabelle

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Allgemeine Statusabfragen	?ASTAT	Statusabfrage der Achsen, pro Achse wird ein Zeichen zurückgeschickt, das den aktuellen Zustand der Achse beschreibt: „I" = Achse nicht initialisiert „O" = Achse stromlos in Ruhe „R" = Achse bestromt in Ruhe „T" = Achse positioniert im Trapez-Profil „S" = Achse positioniert im S-Kurven-Profil „V" = Achse arbeitet im Geschwindigkeitsmodus „P" = Achse fährt auf Referenzposition „F" = Achse fährt einen Endschalter frei „J" = Achse arbeitet im Joystick-Betrieb „L" = Achse stromlos, nachdem sie auf Limitschalter (MINSTOP, MAXSTOP) gefahren ist „B" = Achse wird gestoppt, nachdem sie auf einen Bremsschalter (MINDEC, MAXDEC) gefahren ist „A" = Achse stromlos nach Endstufen-Fehler „M" = Achse stromlos nach Motion-Controller-Fehler „Z" = Achse stromlos nach Timeout-Fehler „H" = Phaseninitialisierung aktiv (Schrittmotor-Achse) „U" = Achse nicht freigegeben „E" = Achse stromlos nach Bewegungsfehler „W" = Achse positioniert im Trapez-Profil mit WMS „X" = Achse positioniert im S-Kurven-Profil mit WMS „Y" = Achse arbeitet im Geschwindigkeitsmodus mit WMS „C" = Achse arbeitet im Bahnsteuerungsgeschwindigkeitsmodus „?" = Fehler, unbekannter Achsenstatus	?ASTAT	IOR
	?MSG	Liest den Message-Ausgangs-Buffer aus, der Message-Ausgangs-Buffer wird nur für Fehlermeldungen, die die Kommando-Schnittstelle betreffen, verwendet (falscher Befehl, fehlende Parameter, ungültiger Wert). Folgende Meldungen sind möglich: „00 NO MESSAGE AVAILABLE" (wird ausgegeben, wenn der Meldungspuffer ausgelesen wird, obwohl keine Meldung verfügbar ist) „01 PARAMETER BEFORE EQUAL WRONG" (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpreter den Parameter vor dem Gleichheitszeichen nicht korrekt in einen Zahlenwert umwandeln konnte) „02 AXIS NUMBER WRONG" (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpreter die übergebene Achsennummer nicht auswerten konnte; zulässig z.B. 1 bis 3) „03 PARAMETER AFTER EQUAL WRONG" (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpreter den Parameter nach dem Gleichheitszeichen nicht korrekt in einen Zahlenwert umwandeln konnte) „04 PARAMETER AFTER EQUAL RANGE" (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpreter erkannt hat, dass der Parameter hinter dem Gleichheitszeichen außerhalb des zulässigen Wertebereichs liegt) „05 WRONG COMMAND ERROR" (wird in den Meldungspuffer geschrieben, wenn der gesendete Befehl syntaktisch nicht korrekt war, d.h. vom Befehlsinterpreter nicht erkannt wurde) „06 REPLY IMPOSSIBLE" (wird ausgegeben, wenn die Antwort nicht gesendet werden konnte, z.B., weil der Sendepuffer nocht nicht leer ist) „07 AXIS IS IN WRONG STATE" (wird in den Meldungspuffer geschrieben, wenn ein Fahr- oder Konfigurierungsbefehl gesendet wurde, der nicht ausgeführt werden konnte, da sich die Achse momentan in einem anderen Fahrzustand befindet)	?MSG	00 NO MESSAGE...
	?ERR	Abfrage eines Fehlers aus dem Fehlerspeicher mit einer Speichertiefe von 20. Die Fehlernummer wird immer als 4-stellige Zahl zurückgegeben. Anhand des Fehlercodes kann die Ursache ermittelt werden. Wird 0 zurückgegeben, so sind keine weiteren Fehler mehr gespeichert.	?ERR	1211
	ERRCLEAR	Fehlerspeicher löschen.	ERRCLEAR	
	?EMERGINP	Gibt den aktuellen Zustand des Notaus-Eingangs zurück.	?EMERGINP	1

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Allgemeine Statusabfragen	?ESTAT<n>	Aktuellen logischen Zustand der Endschalter und Endstufenrückmeldung einer Achse auslesen. Bit 0 = MINSTOP Bit 1 = MINDEC Bit 2 = MAXDEC Bit 3 = MAXSTOP Bit 4 = Rückmeldung der Endstufe	?ESTAT	10001
	?AXSIGNALS<n>	Hardware-Achsen-Signale einer Achse abfragen Bit 0 = Encoder CHA, Bit 1 = Encoder CHB, Bit 2 = Encoder Index, Bit 3 = Encoder Home, Bit 4 = MAXSTOP, Bit 5 = MINSTOP, Bit 6 = reserviert, Bit 7 = reserviert, Bit 8 = reserviert, Bit 9 = reserviert, Bit 10 = reserviert, Bit 11-15 = reserviert	?AXSIGNALS1	0000011101101001
	?READOWID<n>=<uv>	Auslesen des Speicherinhaltes des One-Wire-Chips in der Positioniereinheit bis zur 0x00 Endkennung und Übertragen der Daten an den PC. Als Parameter wird die Anfangsadresse 0x00 bis 0x70 im One-Wire-Chip übergeben, ab dieser Adresse werden dann max. 16 Bytes gelesen oder es wird bis zur Enderkennung gelesen.	?READOWID1=0	INFO1|INFO2|....
	?READOWUB<n>	Auslesen des Speicherinhaltes des One-Wire-Chips aus Adresse 0x86 und 0x87 (=UserBytes) in der Positioniereinheit und Übertragen der Daten an den PC.	?READOWUB1	30
Basis-Konfiguration	AXIS<n>=<uv>	Eine Achse freigeben bzw. sperren. Mit diesem Befehl kann eine Achse freigegeben (1) oder gesperrt (0) werden.	AXIS3=1	
	?AXIS<n>	Freigabezustand einer Achse auslesen. Ist die Achse freigegeben, so wird eine 1 angezeigt, ansonsten eine 0.	?AXIS3	1
	MOTYPE<n>=<uv>	0 = DC-Brush 2 = Schrittmotor Open Loop 3 = Schrittmotor Closed-Loop	MOTYPE1=0	
	?MOTYPE<n>	Motortyp für eine Achse auslesen.	?MOTYPE1	0
	AMPSHNT<n>=<uv>	Stombereich für eine Achse einstellen: 0 = Strombereich 1 (niedrig) 1 = Strombereich 2 (hoch).	AMPSHNT1=0	
	?AMPSHNT<n>	Vorgewählten Strombereich für eine Achse auslesen.	?AMPSHNT1	1
	TERM=<uv>	Terminalmodus einstellen: Modus 0 = kurze Antwort, Modus 1 = Antwort mit Klartext, Modus 2 = Antwort mit Klartext und OK nach jedem Befehl ohne Rückmeldung.	TERM=2	
	?TERM	Terminalmodus abfragen.	?TERM	2
	BAUDRATE	Baudrate der seriellen Schnittstelle einstellen, erlaubte Werte sind : 9 600, 19 200, 38 400, 57 600, 115 200. Diese Einstellung wird erst nach dem nächsten Reset aktiv.	BAUDRATE=9600	
	?BAUDRATE	aktuelle Baudrate der seriellen Schnittstelle abfragen.	?BAUDRATE	9600
	COMEND	Befehlsendekennung einstellen: 0 = CR, 1 = CR+LF, 2 = LF.	COMEND=0	
	?COMEND	Befehlsendekennung abfragen.	?COMEND	0
	SAVEGLOB	Globale Parameter im seriellen FRAM abspeichern.	SAVEGLOB	
	LOADGLOB	Globale Parameter aus dem seriellen FRAM abrufen.	LOADGLOB	
	SAVEAXPA<n>	Achsen-Parameter einer Achse im seriellen FRAM abspeichern.	SAVEAXPA1	
	LOADAXPA<n>	Achsen-Parameter einer Achse aus dem seriellen FRAM abrufen.	LOADAXPA1	
	?SERNUM	Serien-Nummer der Steuerung abfragen.	?SERNUM	09080145
	?VERSION	Software-Version HP-Firmware auslesen.	?VERSION	PS30-V5.0-240512
	?MCTRVER	Versionsdaten des Motion-Controller-Chips zurückgeben.		
	?PCHECK	Checksumme über den Programmspeicher berechnen und auslesen.	?PCHECK	12227
	JZONE=<uv>	Inaktive Zone des Joysticks einstellen (0-256).	JZONE=25	
	?JZONE	Inaktive Zone des Joysticks auslesen.	?JZONE	25
	JZEROX=<uv>	Nullpunkt des X-Joysticks setzen.	JZEROX=505	
	?JZEROX	Nullpunkt des X-Joysticks auslesen.	?JZEROX	505
	JZEROY=<uv>	Nullpunkt des Y-Joysticks setzen.	JZEROY=515	

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Basis-Konfiguration	?JZEROY	Nullpunkt des Y-Joysticks auslesen.	?JZEROY	515
	JZEROZ=<uv>	Nullpunkt des Z-Joysticks setzen.	JZEROZ=508	
	?JZEROZ	Nullpunkt des Z-Joysticks auslesen.	?JZEROZ	508
	JBUTTON=<uv>	Auswertung des Joystickbuttons ein-/ ausschalten.	JBUTTON=1	
	?JBUTTON	Auslesen, ob der Joystickbutton ausgewertet wird oder nicht.	?JBUTTON	1
Positionierbetrieb	INIT<n>	Endstufe Freigabe einschalten und Positionsregler aktivieren. Mit diesem Befehl wird die Achse komplett initialisiert und befindet sich anschließend im bestromten Zustand mit aktivem Positionsregler. **Dieser Befehl muss nach dem Einschalten der Steuerung übermittelt werden, damit die Achse anschließend mit den Befehlen REF, PGO, VGO etc. bewegt werden kann.** Vorher müssen folgende Parameter eingestellt worden sein: Motortyp, Endschalter-Maske und -Polarität, Achsparameter/Regelparameter, Strombereich der Motorendstufe.	INIT1	
	PSET<n>=<sv>	Zielposition bzw. Relativweg (ABSOL/RELAT) für eine Achse setzen. Ist absolutes Positionsformat eingeschaltet, so wird der Parameter als absolute Position mit Vorzeichen interpretiert, ist relative Positionsangabe gewählt, so wird der Parameter als Weg mit Vorzeichen interpretiert. Die neue absolute Zielposition berechnet sich dann aus der Summe von letzter absoluter Zielposition und übergebenem Weg.	PSET2= 100000	
	?PSET<n>	Zielposition bzw. Relativweg für eine Achse auslesen.	?PVEL1	10000
	PCHANGE<n>=<sv>	arbeitet wie PSET, aber ändert zusätzlich bei laufender Trapez-Positionierung die Zielposition "on the fly".	PCHANGE2= 50000	
	?CMDPOS	aktuelle Kommando-Position für eine Achse auslesen. Dieser Befehl liefert die aktuelle Zielposition für den Lagerregler zurück.	?CMDPOS1	5000
	VVEL<n>	Sollgeschwindigkeit für Geschwindigkeitsmodus einer Achse setzen. Mit diesem Befehl wird die Startgeschwindigkeit und auch evtl. eine neue Geschwindigkeit, während die Achse im Geschwindigkeitsmodus fährt, übergeben.	VVEL1= -20000	
	?VVEL<n>	Sollgeschwindigkeit für Geschwindigkeitsmodus auslesen	?VVEL1	-20000
	PGO<n>	Positionierung einer Achse starten. Die Achse fährt die neue Zielposition entweder im Trapez- oder S-Kurven-Profil an (siehe „PMOD“).	PGO2	
	VGO<n>	Geschwindigkeitsmodus einer Achse starten.	VGO2	
	STOP<n>	Bewegung einer Achse stoppen. Jegliche aktive Bewegung einer Achse wird abgebrochen. Der Antrieb stoppt mit der programmierten Bremsrampe und bleibt stehen.		
	VSTP<n>	Geschwindigkeitsmodus einer Achse stoppen. Arbeitet eine Achse im Geschwindigkeitsmodus, so wird dieser mit diesem Befehl beendet und die Achse gestoppt.	VSTP2	
	EFREE<n>	Endschalter einer Achse freifahren. Nachdem ein Antrieb in einen Limit-Schalter (MINSTOP, MAXSTOP) oder Bremsschalter (MINDEC, MAXDEC) gefahren ist, kann mit diesem Befehl der Antrieb aus dem Schalter herausgefahren werden. Die Richtung der Bewegung wird dabei selbsttätig entschieden, je nach dem, ob ein positiver oder negativer Endschalter aktiviert ist.		
	MON<n>	Endstufe Freigabe einschalten und Positionsregler aktivieren. Mit diesem Befehl wird die Achse, nachdem der Motor stromlos geschaltet war, wieder eingeschaltet und befindet sich anschließend im bestromten Zustand mit aktivem Positionsregler.	MON1	
	MOFF<n>	Endstufe Freigabe ausschalten und Positionsregler deaktivieren. Mit diesem Befehl wird der Positionsregler deaktiviert und die Freigabe-Leitung für die Endstufe deaktiviert.	MOFF1	
	JOYON	Startet das Verfahren von 1 bis 3 Achsen mit dem Joystick. Anschließend bewegen sich diese Achsen im Geschwindigkeitsmodus. Die Geschwindigkeit und das Vorzeichen werden mit dem Joystick vorgegeben.	JOYON	
	JOYOFF	Stoppt das Verfahren von 1 bis 3 Achsen mit Joystick.	JOYOFF	
	CNT<n>=<sv>	Aktuellen Positionszähler für eine Achse setzen.	CNT1=5000	
	?CNT<n>	Aktuellen Positionszähler für eine Achse auslesen.	?CNT1	5000
	CRES<n>	Aktuellen Positionszähler für eine Achse nullen.	CRES1	
	?POSERR<n>	Auslesen des aktuellen Positionsfehlers einer Achse. Zurückgegeben wird die Differenz zwischen Encoder-Position und Sollposition. Kann auch „on the fly" benutzt werden um den Schleppfehler auszulesen.	?POSERR1	-15

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Positionierbetrieb	?VACT<n>	Aktuelle Geschwindigkeit einer Achse auslesen. Die aktuelle Geschwindigkeit wird im Format 16.16 mit Vorzeichen zurückgegeben, der fraktionale Anteil ist jedoch „0", weil die aktuelle Geschwindigkeit aus der Positionsdifferenz zwischen zwei Sample-Zeiten berechnet wird.	?VACT2	1000
	?ENCPOS<n>	Aktuellen Positionszähler des Encoders für eine Achse auslesen. Dieser Befehl liefert bei Open-Loop-Schrittmotorachsen, die aber mit Encoder betrieben werden, die aktuelle Encoder-Position zurück (umgerechnet in Schritte).	?ENCPOS1	5000
	?MXSTROKE<n>	Gemessenen Tischhub auslesen. Bei der Referenzierung in den Modi 6 und 7 wird der Tischhub ermittelt und kann mit diesem Kommando ausgelesen werden.	?MXSTROKE1	340000
Positionierparameter	RELAT<n>	Positionsangaben für eine Achse auf relativ umschalten (= Angabe des Weges mit Vorzeichen).	RELAT1	
	ABSOL<n>	Positionsangaben für eine Achse auf absolut umschalten (= Angabe der Zielposition mit Vorzeichen).	ABSOL2	
	?MODE<n>	Abfrage des aktuell eingestellten Positionsformates für eine Achse.	?MODE2	ABSOL
	PMOD<n>=<uv>	Positioniermodus Trapez/S-Kurve für eine Achse setzen. (0 = Trapez-Profil, 1 = S-Kurven-Profil).	PMOD1=0 PMOD1=0	
	?PMOD<n>	Positioniermodus Trapez/S-Kurve für eine Achse auslesen.	?PMOD1	1
	PVEL<n>=<uv>	Max. Positioniergeschwindigkeit für eine Achse setzen, wird für das Trapez- und S-Kurven-Profil verwendet.	PVEL1=10000	
	?PVEL<n>	Max. Positioniergeschwindigkeit für eine Achse auslesen.	?PVEL1	10000
	FVEL<n>=<uv>	Endschalterfreifahrtgeschwindigkeit für eine Achse setzen (ohne Vorzeichen).	FVEL1=1000	
	?FVEL<n>	Endschalterfreifahrtgeschwindigkeit für eine Achse auslesen.	?FVEL1	1000
	ACC<n>=<uv>	Beschleunigung (= Anfahrrampe) für eine Achse setzen, wird für alle Modi verwendet (Trapez, S-Kurve, Geschwindigkeitsmodus etc).	ACC1=100	
	?ACC<n>	Beschleunigung für eine Achse auslesen.	?ACC1	100
	DACC<n>=<uv>	Verzögerung (= Bremsrampe) für eine Achse setzen, wird für alle Modi ausser S-Kurve verwendet.	DACC2=68	
	?DACC<n>	Verzögerung für eine Achse auslesen.	?DACC2	68
	JACC<n>=<uv>	Maximalen Jerk („Ruck") für eine Achse setzen, wird nur beim S-Kurven-Profil verwendet.	JACC3=5	
	?JACC<n>	Maximalen Jerk („Ruck") für eine Achse auslesen.	?JACC3	5
	EDACC<n>=<uv>	Notstopp-Verzögerung (Bremsbeschleunigung) für eine Achse einstellen. Diese Verzögerung wird benutzt, wenn ein Bremsschalter angesprochen hat.	EDACC1=1000	
	?EDACC<n>	Notstopp-Verzögerung einer Achse auslesen	?EDACC1	1000
	JVEL<n>=<sv>	Max. Geschwindigkeit der Achse bei „Joystickfahrt" einstellen. Mit diesem Befehl wird die maximale Geschwindigkeit bei voller Auslenkung des Joysticks definiert.	JVEL3=1000	
	?JVEL<n>	Max. Geschwindigkeit der Achse für „Joystickfahrt" auslesen	?JVEL3	1000
	JOYACC<n>=<uv>	Beschleunigung und Verzögerung der Achse bei Fahren mit Joystick einstellen.		
	?JOYACC<n>	Beschleunigung/Verzögerung der Achse bei Fahren mit Joystick auslesen.	?JOYACC3	100
	JPLAX=<n>	Joystickebenenzuordnung X-Ebene Achsen-Nummer setzen. Die übergebene Achsen-Nummer ist danach dem X-Joystick zugeordnet. Übergibt man 0, so ist anschließend keine Achse dem X-Joystick zugeordnet.	JPLAX=2	
	?JPLAX	Joystickebenenzuordnung X-Ebene Achsen-Nummer auslesen.	?PLAX	2
	JPLAY=<n>	Joystickebenenzuordnung Y-Ebene Achsen-Nummer setzen. Die übergebene Achsen-Nummer ist danach dem Y-Joystick zugeordnet. Übergibt man 0, so ist anschließend keine Achse dem Y-Joystick zugeordnet.	JPLAY=3	
	?JPLAY	Joystickebenenzuordnung Y-Ebene Achsen-Nummer auslesen.	?JPLAY	3
	JPLAZ=<n>	Joystickebenenzuordnung Z-Ebene Achsen-Nummer setzen. Die übergebene Achsen-Nummer ist danach dem Z-Joystick zugeordnet. Übergibt man 0, so ist anschließend keine Achse dem Z-Joystick zugeordnet.	JPLAZ=1	

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Positionierparameter	?JPLAZ	Joystickebenenzuordnung Z-Ebene Achsen-Nummer auslesen.	?JPLAZ	1
	LIGO=<uv>	Positionierung mit Linearinterpolation für eine Achsengruppe (binäre Definitionsmaske) starten. Bit-Reihenfolge : <Achse 3, Achse 2, Achse 1>.	LIGO=111	
	IVEL<n>=<uv>	Maximalgeschwindigkeit <uv> für Linearinterpolationsachse <n> einstellen.	IVEL1=50000	
	?IVEL<n>	Maximalgeschwindigkeit für Linearinterpolationsachse <n> auslesen.	?IVEL1	50000
	IACC<n>=<uv>	Maximalbeschleunigung <uv> für Linearinterpolationsachse <n> einstellen.	IACC3=2000	
	?IACC<n>	Maximalbeschleunigung für Linearinterpolationsachse <n> auslesen.	?IACC3	2000
	POSTAB<uv>=<v>, <v>,...	Eine Tabellenzeile in die Bahntabelle herunterladen; vor dem „="-Zeichen steht die Tabellen-Zeilennummer (0 bis ...), hinter dem „="-Zeichen folgen durch Komma getrennt die Werte für die Tabellen-Spalten. Parameter-Liste: 1. Weg mit Vorzeichen Achse 1 (-32760 bis 32760) 2. Weg mit Vorzeichen Achse 2 (-32760 bis 32760) 3. Weg mit Vorzeichen Achse 3 (-32760 bis 32760) 4. reserviert 5. reserviert 6. reserviert 7. reserviert 8. reserviert 9. Segmentzeit in ms 10. Funktionscode Bit 15 im Funktionscode gesetzt: Verfahren mit a=const. Bit 15 im Funktionscode gelöscht: Verfahren mit v=const. 11. Fehlerbyte 12. Freigabe-Byte (immer als Zahlenwert)	POSTAB0=1000, 2000, 0, 0, 0, 0, 0, 0, 100, 0, 0, 3	
	?POSTAB<uv>	Eine Tabellenzeile aus der Bahntabelle hochladen. Als Parameter wird die Tabellen-Zeilennummer (0 bis ...) übergeben. Die Tabellenwerte werden durch Komma getrennt aufgelistet. Parameter-Liste: 1. Weg mit Vorzeichen Achse 1 (-32760 bis 32760) 2. Weg mit Vorzeichen Achse 2 (-32760 bis 32760) 3. Weg mit Vorzeichen Achse 3 (-32760 bis 32760) 4. reserviert 5. reserviert 6. reserviert 7. reserviert 8. reserviert 9. Segmentzeit in ms (16 Bit) 10. Funktionscode (16 Bit) 11. Fehlerbyte (8 Bit) 12. Freigabe-Byte (8 Bit) 13. Bei der Plausibilitätskontrolle berechnete Geschwindigkeit (32 Bit) 14. Bei der Plausibilitätskontrolle berechnete Beschleunigung (32 Bit)	?POSTAB0	1000, 2000, 0, 0, 0, 0, 0, 0, 100, 0, 0, 3, 2100, 50,
	PTABPLAUS<uv>	Plausibilitätskontrolle der Bahntabelle durchführen. Die Grenzwerte für Geschwindigkeit und Beschleunigung werden geprüft und die Fehlerbytes entsprechend gesetzt. Die Geschwindigkeiten und Beschleunigungen in den einzelnen Tabellenzeilen werden berechnet und deren Werte für die aktive Achse mit der höchsten Achsnummer eingetragen.	PTABPLAUS0	
	PTABGO<uv>	Bahnsteuerung ab einem bestimmten Tabelleneintrag starten.	PTABGO0	
	PTABSTP	Eine laufende Bahnsteuerung abbrechen; die beteiligten Achsen verhalten sich wie am Ende der Bahntabelle.	PTABSTP	
	PTABCLR	Tabelle für Bahnsteuerung löschen.		
	PTABCPY <uv>=<uv>,<uv>	Einen Bereich der Tabelle für Bahnsteuerung kopieren. Der Wert vor dem „="-Zeichen gibt den Zielindex in der Positionstabelle an, der Wert hinter dem „="-Zeichen gibt den Quellindex an und der Wert hinter dem Komma die Anzahl Zeilen die kopiert werden sollen.	PTABCPY 50=10, 20	
	PTABCLR<uv>=<uv>	Einen Bereich der Tabelle für Bahnsteuerung löschen. Der Wert vor dem „="-Zeichen gibt den Zeilenindex an, ab dem gelöscht werden soll; der Wert hinter dem „="-Zeichen gibt die Anzahl Zeilen an, die gelöscht werden sollen.	PTABCLR50=20	

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Positionierparameter	PTABCIRCLE <uv>=<uv>, ...	Kreisinterpolation berechnen und in die Positionstabelle ab der angegebenen Startzeile eintragen. Die Parameter werden als Liste durch Komma getrennt übergeben.Die Achsen-Freigabebits werden mit den evtl. bereits bestehenden Einträgen der aktuellen Tabelle bitweise ODER-verknüpft. 1. Achsennummer für X (0 für keine X-Achse) 2. Achsennummer für Y (0 für keine Y-Achse) 3. Segmentzeit in ms (16 Bit) 4. Funktionscode (OR-Maske) für die Segmente (16 Bit) 5. Anzahl der Kreissegmente (16 Bit) 6. Kreisradius mit Vorzeichen (32 Bit) 7. Anfangswinkel in Grad mit Vorzeichen (16 Bit) 8. Winkelbereich in Grad mit Vorzeichen (16 Bit) 9. Optional Skalierung Zähler mit Vorzeichen (16 Bit) 10. Optional Skalierung Nenner mit Vorzeichen (16 Bit)	PTABCIRLCE0= 1,2,10000,0,12, 50000,0,360,1,1	
Achsparameter	MCSTP<n>=<uv>	Mikroschrittauflösung bei Schrittmotorachsen einstellen.	MCSTP1=50	
	?MCSTP<n>	Mikroschrittauflösung bei Schrittmotorachsen auslesen.	?MCSTP1	50
	DRICUR<n>=<uv>	Fahrstrom bei Schrittmotorachsen als ganzzahliger Prozentwert des Maximal-stromes im vorgewählten Strombereich (1 oder 2) in Prozent einstellen.	DRICUR1=50	
	?DRICUR<n>	Fahrstrom bei Schrittmotorachsen in Prozent auslesen.	?DRICUR1	50
	HOLCUR<n>=<uv>	Haltestrom bei Schrittmotorachsen in Prozent einstellen.	HOLCUR1=30	
	?HOLCUR<n>	Haltestrom bei Schrittmotorachsen in Prozent auslesen.	?HOLCUR1	30
	ATOT<n>=<uv>	Achsen-Timeout-Zeit einstellen in Millisekunden, 0 schaltet die Timeout-Überwachung ab.	ATOT1=20000	
	?ATOT<n>	Achsen-Timeout-Zeit abfragen.	?ATOT1	20000
	FKP<n>=<uv>	Regelparameter KP für eine Achse einstellen.	FKP1=25	
	?FKP<n>	Regelparameter KP für eine Achse abfragen.	?FKP1	25
	FKD<n>=<uv>	Regelparameter KD für eine Achse einstellen.	FKD1=5	
	?FKD<n>	Regelparameter KD für eine Achse abfragen.	?FKD1	5
	FKI<n>=<uv>	Regelparameter KI für eine Achse einstellen.	FKI1=10	
	?FKI<n>	Regelparameter KI für eine Achse abfragen.	?FKI1	10
	FIL<n>=<uv>	Regelparameter Integrationslimit für eine Achse einstellen.	FIL1=100000	
	?FIL<n>	Regelparameter Integrationslimit für eine Achse abfragen.	?FIL1	100000
	FST<n>=<uv>	Sample-Zeit für eine Achse einstellen (in Mikrosekunden).	FST1=500	
	?FST<n>	Sample-Zeit für eine Achse abfragen (in Mikrosekunden).	?FST1	500
	FDT<n>=<uv>	Verzögerungseit des D-Anteils für eine Achse einstellen (in Sample-Zeit-Zyklen).	FDT1=5	
	?FDT<n>	Verzögerungseit des D-Anteils für eine Achse abfragen (in Sample-Zeit-Zyklen).	?FDT1	5
	MXPOSERR<n>=<uv>	Maximalen Positionsfehler für eine Servo-Achse setzen. Wird dieser Wert überschritten, so schaltet die Achse ab. Diese Abschaltung gilt nur für die Motortypen DC-Brush, Schrittmotor Closed-Loop.	MXPOSERR1=50	
	?MXPOSERR<n>	Maximalen Positionsfehler einer Achse abfragen.	?MXPOSERR1	50
	MAXOUT<n>=<uv>	Maximalen Ausgabewert der Servoregelschleife in Prozent einstellen. Mit diesem Befehl kann der maximale Wert für eine Achse, der an den Servo-Verstärker ausgegeben wird, eingestellt werden. **Max. zulässiger Wert: 99 %**	MAXOUT1=95	
	?MAXOUT<n>	Maximalen Ausgabewert in Prozent auslesen.	?MAXOUT1	95
	INPOSMOD<n>=<uv>	Bewegungsfertigmeldemodus einstellen, 0 = Zielposition erreicht, 1 = für eine gewisse Zeit im Fenster um die Zielposition.	INPOSMOD1=0	
	?INPOSMOD<n>	Bewegungsfertigmeldemodus abfragen.	?INPOSMOD1	0
	INPOSTIM<n> = <uv>	Bewegungsfertigmeldezeit einstellen in Sample-Zeit-Zyklen.	INPOSTIM1=1000	
	?INPOSTIM<n>	Bewegungsfertigmeldezeit abfragen.	?INPOSTIM1	1000
	INPOSWND<n>=<uv>	Bewegungsfertigmeldefenster einstellen in Encoder-Counts.	INPOSWND1=50	
	?INPOSWND<n>	Bewegungsfertigmeldefenster abfragen.	?INPOSWND1	50

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Achsparameter	AMPPWMF<n>=<uv>	PWM-Frequenz für Endstufe einstellen, 20000 oder 80000 ist möglich.	AMPPWMF1= 20000	
	?AMPPWMF<n>	PWM-Frequenz für Endstufe abfragen.	?AMPPWMF1	?AMPPWMF1
	ENCLINES<n>=<uv>	Linien-Anzahl des Encoders für eine Achse einstellen.	ENCLINES1=500	
	?ENCLINES<n>	Linien-Anzahl des Encoders für eine Achse abfragen.	?ENCLINES1	500
	MOTPOLES<n>=<uv>	Polanzahl des Motors für eine Achse einstellen.	MOTPOLES1=50	
	?MOTPOLES<n>	Polanzahl des Motors für eine Achse auslesen.	?MOTPOLES1	50
	ELCYCNT<n>=<uv>	Encoder-Counts für einen elektrischen Kommutierungszyklus einstellen.	ELCYCNT1=128	
	?ELCYCNT<n>	Encoder-Counts für einen elektrischen Kommutierungszyklus abfragen.	?ELCYCNT1	128
	PHINTIM<n>=<uv>	Phasen-Initialisierungszeit in Sample-Zeit-Zyklen einstellen.	PHINTIM1=10	
	?PHINTIM<n>	Phasen-Initialisierungszeit in Sample-Zeit-Zyklen abfragen.	?PHINTIM1	10
	PHINAMP<n>=<uv>	Phasen-Initialisierungsamplitude in % einstellen.	PHINAMP1=50	
	?PHINAMP<n>	Phasen-Initialisierungsamplitude in % abfragen.	?PHINAMP1	50
Endschalterkonfiguration und Referenzfahrt	REF<n>=<uv>	Referenzfahrt mit Angabe des Referenzfahrtmodus für eine Achse starten: Modus 0 = nächsten Index-Impuls suchen und stehenbleiben Modus 1 = Referenzschalter anfahren und stehenbleiben Modus 2 = Referenzschalter anfahren, nächsten Index-Impuls suchen und stehen bleiben Modus 3 = Modus 0, zusätzlich akt. Positon auf 0 setzen Modus 4 = Modus 1, zusätzlich akt. Positon auf 0 setzen Modus 5 = Modus 2, zusätzlich akt. Positon auf 0 setzen Modus 6 = Maximalen Referenzschalter anfahren, minimalen Referenzschalter anfahren, aktuelle Position auf 0 setzen Modus 7 = Minimalen Referenzschalter anfahren, maximalen Referenzschalter anfahren, aktuelle Positon auf 0 setzen	REF2=4	
	RVELS<n>=<sv>	Referenzfahrtgeschwindigkeit „langsam" für eine Achse setzen. Mit dieser Geschwindigkeit wird der Index gesucht bzw. aus dem Referenzschalter herausgefahren (vorzeichenbehaftet).	RVELS2=2000	
	?RVELS<n>	Referenzfahrtgeschwindigkeit „langsam" für eine Achse auslesen.	?RVELS2	2000
	RVELF<n>=<sv>	Referenzfahrtgeschwindigkeit „schnell" für eine Achse setzen. Mit dieser Geschwindigkeit fährt der Antrieb auf den Referenzschalter (vorzeichenbehaftet).	RVELF2= -20000	
	?RVELF<n>	Referenzfahrtgeschwindigkeit „schnell" für eine Achse auslesen.	?RVELF2	-20000
	RDACC<n>=<uv>	Referenzfahrt-Verzögerung für eine Achse einstellen. Diese Verzögerung wird benutzt, wenn der Referenzpunkt angefahren wird.	RDACC1=2000	
	?RDACC<n>	Referenzfahrt-Verzögerung einer Achse auslesen.	?RDACC1	2000
	SMK<n>=<uv>	Endschaltermaske für eine Achse setzen. Mit diesem Befehl werden die Endschalter und die Bremsschalter aktiv bzw. inaktiv gesetzt. Wird auf einen Endschalter gefahren, so wird die Bewegung abrupt gestoppt und der Motor danach stromlos geschaltet. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	SMK3=1001	
	?SMK<n>	Endschaltermaske für eine Achse auslesen.	?SMK3	1001
	SPL<n>=<uv>	Endschalterpolarität für eine Achse setzen. Mit diesem Befehl wird der aktive Pegel für die Endschalter und Bremsschalter festgelegt. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	SPL3=1111	
	?SPL<n>	Endschalterpolarität für eine Achse auslesen.	?SPL3	1111
	RMK<n>=<uv>	Referenzschaltermaske für eine Achse setzen. Mit dem Befehl wird definiert, welcher der 4 Endschalter einer Achse als Referenzschalter interpretiert werden soll. Es muss eine Maske mit genau einer Eins übergeben werden. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	RMK3=0001	
	?RMK<n>	Referenzschaltermaske für eine Achse auslesen.	?RMK3	0001
	RPL<n>=<uv>	Referenzschalterpolarität für eine Achse setzen. Dieser Befehl definiert den aktiven Pegel des Referenzschalters. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	RPL3=1111	
	?RPL<n>	Referenzschalterpolarität für eine Achse auslesen.	?RPL3	1111
	?HYST<n>	Referenzschalterhysterese einer Achse auslesen. Nach erfolgter Referenzfahrt kann mit dem Kommando die Hysterese des Schalters ausgelesen werden.	?HYST1	28
	?REFST<n>	Abfrage der Gültigkeit der Referenzfahrt. Nach erfolgter Referenzfahrt wird der Status auf 1 =„gültig" gesetzt. Schaltet man einen Antrieb ohne Encoder (z.B. Schrittmotor Open Loop) stromlos, so wird die Gültigkeit auf 0 zurückgesetzt.	?REFST	1

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Endschalterkonfiguration und Referenzfahrt	LMK<n>=<uv>	Limit-Positionsüberwachungsmaske für die Achse setzen. Mit diesem Befehl wird die Limit-Positionsüberwachung für die untere und/oder die obere Grenzposition aktiv bzw. inaktiv geschaltet. Die Limit-Positionsüberwachung verhält sich beim Überschreiten der Grenzen wie der entsprechende DEC-Schalter. Bit-Reihenfolge: <MAXDEC, MINDEC>	LMK1=01	
	?LMK<n>	Limit-Positionsüberwachungsmaske für die Achse auslesen.	?LMK1	01
	?LSTAT<n>	Aktuellen, logischen Zustand der Limit-Positionsüberwachung der Achse auslesen. Bit 0 = MINDEC untere Grenze überschritten Bit 1 = MAXDEC obere Grenze überschritten	?LSTAT1	01
	SLMIN<n>=<uv>	Negative Limit-Position für die Achse einstellen.	SLMIN1=100	
	?SLMIN<n>	Negative Limit-Position für die Achse auslesen.	?SLMIN1	100
	SLMAX<n>=<uv>	Positive Limit-Position für die Achse einstellen.	SLMAX1= 100000	
	?SLMAX<n>	Positive Limit-Position für die Achse auslesen.	?SLMAX1	100000
Ein-/Ausgänge	ETTLOUTS<n>=<bin>	TTL-Ausgänge zur Endstufe einer Achse setzen. Übergeben wird die Achsennummer und eine binäre Setzmaske.	ETTLOUTS1=10	
	ETTLOUTC<n>=<bin>	TTL-Ausgänge zur Endstufe einer Achse rücksetzen. Übergeben wird die Achsennummer und eine binäre Löschmaske.	ETTLOUTC1=01	
	?INPUTS	Aktuellen Zustand der Eingänge auslesen (16-Bit-Binärzahl).	?INPUTS	10100100101101
	?INPTTL	aktuellen Zustand der TTL-Eingänge auslesen (8-Bit Binärzahl).	?INPTTL	00110011
	OUTPUT<uv>=<uv>	Aktuellen Zustand eines Ausgangs ändern.	OUTPUT1=0	
	?OUTPUTS	Aktuellen Zustand aller Ausgänge auslesen.	?OUTPUTS	10100100101101
	OUTTTL<uv>=<uv>	Aktuellen Zustand eines TTL-Ausgangs ändern.	OUTTTL1=0	
	?OUTTTL	Aktuellen Zustand aller TTL-Ausgänge auslesen.	?OUTTTL	00101001
	?ANIN<uv>	Analog-Eingang abfragen, angegeben wird die Kanal-Nummer von 1 bis 8, zurückgegeben wird der gewandelte 10-Bit-Wert.	?ANIN3	234
	OPWM<uv>=<uv>	PWM-Ausgang setzen, angegeben wird die Kanal-Nummer von 1 bis 2 und der Aussteuerungswert von 0 bis 100%.	OPWM1=55	
	?OPWM<uv>	PWM-Ausgang abfragen, angegeben wird die Kanal-Nummer von 1 bis 2 und zurückgegeben wird der zuletzt eingestellte Aussteuerungswert von 0 bis 100%.	?OPWM1	55
	AXOUTPUT<n>=<uv>	AxisOut-Pin einer Achsen auf High/Low setzen.		
Nachlaufregelung	?APWMS<n>	Aktuelle Position des Wegmesssystems einer Achse auslesen.	APWMS4	3000
	WMSRES<n>	Aktuelle Position des Wegmesssystems einer Achse auf 0 setzen (wird nicht benötigt bzw. darf nach erfolgter Referenzierung nicht mehr benutzt werden, da sonst die Position verloren geht).	WMSRES4	
	?MXWMSSTRK<n>	Den nach Referenzfahrt mit Modus 6 oder 7 ermittelten maximalen Gesamthub in Inkrementen des Wegmesssystems abfragen.	?MXWMSSTRK2	100000
	WMSFAKZ<n>=<uv>	Faktor für die Positionierung mit Nachlaufregelung Zähler setzen.	WMSFAKZ1=1	
	?WMSFAKZ<n>	Faktor für die Positionierung mit Nachlaufregelung Zähler abfragen.	?WMSFAKZ1	1
	WMSFAKN<n>=<uv>	Faktor für die Positionierung mit Nachlaufregelung Nenner setzen.	WMSFAKN1=5	
	?WMSFAKN<n>	Faktor für die Positionierung mit Nachlaufregelung Nenner abfragen.	?WMSFAKN1	5
	PWMSSET<n>=<sv>	Zielposition bzw. Relativweg (Vorwahl erfolgt, analog zur normalen Positionierung ohne Nachlaufregelung, über die Kommandos ABSOL bzw. RELAT) für eine Achse setzen; ist die absolute Positionsangabe eingeschaltet, so wird der Parameter als absolute Position mit Vorzeichen interpretiert, ist relative Positionsangabe gewählt, so wird der Parameter als Weg mit Vorzeichen interpretiert. Die neue absolute Zielposition berechnet sich dann aus der letzten abs. Zielposition plus Weg.	PWMSSET2= 100000	
	?PWMSSET<n>	Zielposition bzw. Relativweg für eine Achse auslesen.	?PWMSSET2	100000
	PWMSGO<n>	Positionierung mit WMS bei einer Achse starten, die Achse fährt die neue Zielposition entweder im Trapez- oder S-Kurven-Profil an (siehe PMOD).	PWMSGO2	
	PWMSWIN<n>=<uv>	Halbe Zielfensterbreite für die Positionierung mit WMS einstellen (gesamte Breite des Zielfensters = ±PWMSWIN).	PWMSWIN1=10	
	?PWMSWIN<n>	Halbe Zielfensterbreite für die Positionierung mit WMS abfragen.	?PWMSWIN1	10
	?PWMSMODE<n>	Positioniermodus für die Positionierung mit WMS abfragen.	?PWMSMODE1	1

Befehls-gruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Nachlaufregelung	PWMSMODE<n>=<uv>	Positioniermodus für die Positionierung mit WMS einstellen. Modus 0: Nur Grobpositionierung mit Phase 1, Iteration durch mehrmaligen Aufruf. Modus 1: Grobpositionierung Phase 1 und Iteration mit Phase 2 Modus 2: Grobpositionierung Phase 1 und Iteration mit Phase 2 und Korrekturfahrt mit Phase 3, Phase 3 bleibt aktiv und muss mit PWMSSTP vor nächster Positionierung beendet werden. Modus 3: Grobpositionierung Phase 1 und Korrekturfahrt mit Phase 3, Phase 3 bleibt aktiv und muss mit PWMSSTP vor nächster Positionierung beendet werden. Modus 4: Grobpositionierung Phase 1 und Iteration mit Phase 2 und Korrekturfahrt mit Phase 3, Phase 3 wird bei Erreichen des Zielfensters beendet. Modus 5: Grobpositionierung Phase 1 und Korrekturfahrt mit Phase 3, Phase 3 wird bei Erreichen des Zielfensters beendet.	PWMSMODE1=1	
	WMSVEL<n>=<uv>	Nachfahrgeschwindigkeit beim Positionieren mit WMS einstellen (ohne Vorzeichen).	WMSVEL1=100	
	?WMSVEL<n>	Nachfahrgeschwindigkeit beim Positionieren mit WMS auslesen.	?WMSVEL1	100
	PWMSSTP<n>	Positionierung mit WMS bei einer Achse stoppen, befindet sich die Achse beim Positionieren mit WMS in Phase 3, so muss vor dem Verfahren der Achse mit einem neuen Befehl diese Betriebsart mit diesem Befehl beendet werden.	PWMSSTP1	
	?PWMSSTATE<n>	Zustand beim Positionieren einer Achse mit WMS auslesen Bit 0: Achse positioniert mit WMS Bit 1: Achse positioniert mit WMS und ist in Phase 1 Bit 2: Achse positioniert mit WMS und ist in Phase 2 Bit 3: Achse positioniert mit WMS und ist in Phase 3 Bit 4: Achse ist im vorgegebenen Zielfenster	?PWMSSTATE1	
	?PWMSERR<n>	Auslesen des aktuellen Positionsfehlers einer Achse beim Positionieren mit WMS.	?PWMSERR1	
	WMSINV<n>=<uv>	Zählrichtung des WMS invertieren (1=ja / 0=nein).	WMSINV1=0	
	?WMSINV<n>	Auslesen, ob Zählrichtung WMS invertieren ja/nein.	?WMSINV1	0
Haltebremsenansteuerung	HBCH<n>=<uv>	PWM-Ausgang für Haltebremse einer Achse zuordnen: <Achsennummer> = <PWM-Kanal> PWM-Kanal = 0 für Haltebremsenfunktion aus.	HBCH1=1	
	?HBCH<n>	Zuordnung Haltebremse PWM-Kanal einer Achse abfragen.	?HBCH1	1
	HBFV<n>=<uv>	Ersten PWM-Wert (zum Anziehen) bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Prozentwert>.	HBFV1=50	
	?HBFV<n>	Ersten PWM-Wert (zum Anziehen) bei der Ansteuerung der Haltebremse abfragen.	?HBFV1	50
	HBSV<n>=<uv>	Zweiten PWM-Wert (zum Halten) bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Prozentwert>.	HBSV1=20	
	?HBSV<n>	Zweiten PWM-Wert (zum Halten) bei der Ansteuerung der Haltebremse abfragen.	?HBSV1	20
	HBTI<n>=<uv>	Zeit für ersten PWM-Wert bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Zeit für ersten PWM-Wert in ms>.	HBTI1=300	
	?HBTI<n>	Zeit für ersten PWM-Wert bei der Ansteuerung der Haltebremse abfragen.	?HBTI1	300
Reset	RESETAC	Reset Endstufenmodul auslösen.	RESETAC	
	RESETMB	Reset Hauptprozessor auslösen.	RESETMB	
Stand -Alone-Programmierung	SAMEM	Merker-Wert setzen.	SAMEM38=50	
	?SAMEM	Merker-Wert abfragen.	?SAMEM38	50
	SAEXEC	Stand-Alone-Programm-Ausführung starten (1) /stoppen (0).	SAEXEC0	
	SASTEP	Eine Stand-Alone-Programm-Zeile ausführen, der Zeilen-Index wird übergeben und zurückgegeben wird der Zeilen-Index der nächsten Zeile.	SASTEP1	2
	SALOAD	Eine Stand-Alone-Programm-Zeile laden, übergeben wird der Zeilen-Index und der Inhalt der Programm-Zeile (16 Byte) als Hex-Dump im ASCII-Format.	SALOAD11= 04000015F900...	
	SACHKS	Checksumme über das Stand-Alone-Programm aktualisieren, nachdem ein neues Stand-Alone-Programm geladen wurde.		

II Relevanz der Parameter für verschiedene Motortypen

Parameter	DC-Brush	2-Phasen-Schrittmotor Open Loop	2-Phasen-Schrittmotor Closed-Loop
MOTYPE	+	+	+
AXIS	+	+	+
FKP	+	–	+
FKD	+	–	+
FDT	+	–	+
FKI	+	–	+
FIL	+	–	+
FST	+	–	+
MAXOUT	+	+[1)]	+
MXPOSERR	+	–	+
SMK	+	+	+
SPL	+	+	+
RMK	+	+	+
RPL	+	+	+
RVELF	+	+	+
RVELS	+	+	+
ACC	+	+	+
DACC	+	+	+
JACC	+	+	+
PVEL	+	+	+
EDACC	+	+	+
FVEL	+	+	+
ABSOL	+	+	+
RELAT	+	+	+
PMOD	+	+	+
AMPPWMF	+	+	+
MCSTP	–	+	–
DRICUR	–	+	+
HOLCUR	–	+	+
AMPSHNT	+	+	+

Parameter	DC-Brush	2-Phasen-Schrittmotor Open Loop	2-Phasen-Schrittmotor Closed-Loop
MOTPOLES	–	–	+
ENCLINES	–	–	+
ELCYCNT	–	–	+
PHINTIM	–	+	+
PHINAMP	–	–	+
ATOT	+	+	+
INPOSTIM	+	–	+
INPOSWND	+	–	+
INPOSMOD	+	–	+
HBCH	(+)	(+)	(+)
HBFV	(+)	(+)	(+)
HBTI	(+)	(+)	(+)
HBSV	(+)	(+)	(+)
JPLAX	(+)	(+)	(+)
JPLAY	(+)	(+)	(+)
JPLAZ	(+)	(+)	(+)
JOYACC	(+)	(+)	(+)
JVEL	(+)	(+)	(+)
JZONE	(+)	(+)	(+)
JZEROX	(+)	(+)	(+)
JZEROY	(+)	(+)	(+)
JBUTTON	(+)	(+)	(+)

\+ benötigt
– nicht benötigt
(+) optional

1) Verwendung ist möglich, jedoch ist darauf zu achten, dass der hier gesetzte Wert größer oder gleich dem maximalen PWM-Wert für DRICUR bzw. HOLCUR ist. Der Ausgang wird auf jeden Fall auf den per MAXOUT definierten Wert begrenzt. Wird ein zu kleiner Wert gewählt, funktioniert der Mikroschrittbetrieb nicht mehr ordnungsgemäß.

III Belegungstabellen

Ein-/Ausgänge

Pinbelegung des 25-poligen D-Sub (male) an der PCI-Einsteckkarte

Funktion	Pin
Input 1 (TTL/Analog)	6
Input 2 (TTL/Analog)	5
Input 3 (TTL/Analog)	4
Input 4 (TTL/Analog)	3
Input 5 (TTL/Analog)	10
Input 6 (TTL/Analog)	9
Input 7 (TTL/Analog)	8
Input 8 (TTL/Analog)	7
TTL-Output 1	16
TTL-Output 2	15
TTL-Output 3	14
TTL-Output 4	13
TTL-Output 5	17
+5V, max. 300 mA Gesamtstrom	1, 2
PWM-Output 1, max. 1A *)	20
PWM-Output 2, max. 1A *)	21
+12V/+24V, max. 1A Gesamtstrom **)	18, 19
GND	11, 12, 24, 25
Freigabe-Eingang + (5V) ***)	22
Freigabe-Eingang - (GND)	23

*) nach Masse (GND) schaltend
**) +12V oder +24V je nach Leistungsversorgung (intern über PC-Netzteil oder extern)
***) Freigabe der Motorendstufe über Optokoppler (U_B = 5V) erforderlich; z.B. Brücke Pin 2 –> Pin 22 und Pin 23 –> Pin 24

RS-232

Pinbelegung des 10-poligen IDC-Steckverbinders

RS-232	Pin
n.c.	1
DSR	2
RXD	3
RTS	4
TXD	5
CTS	6
DTR	7
n.c.	8
GND	9
GND	10

3-fach-Motorstecker an der PCI-Einsteckkarte

Pinbelegung des 62-poligen HD (female)

Pin	DC-Motor	Schrittmotor
1	Motor1 +	Motor1 Phase 1+
2	Motor1 +	Motor1 Phase 1+
3	Motor1 -	Motor1 Phase 1 -
4	Motor1 -	Motor1 Phase 1 -
5	Motor1 +	Motor1 Phase 2+
6	Motor1 +	Motor1 Phase 2+
7	Motor1 -	Motor1 Phase 2 -
8	Motor1 -	Motor1 Phase 2 -
9	Motorcodierung	
10	Motor1 MAXSTOP	
11	Motor1 MINSTOP	
12	Encoder1 A	
13	Encoder1 $\overline{A}$	
14	Encoder1 B	
15	Encoder1 $\overline{B}$	
16	Encoder1 Index	
17	Encoder1 $\overline{Index}$	
18	+5V	
19	GND	
20	OWISid	
21	+24V-Einspeisung für PWM-Ausgänge (optional)	
22	Motor3 +	Motor3 Phase 1+
23	Motor3 +	Motor3 Phase 1+
24	Motor3 -	Motor3 Phase 1 -
25	Motor3 -	Motor3 Phase 1 -
26	Motor3 +	Motor3 Phase 2+
27	Motor3 +	Motor3 Phase 2+
28	Motor3 -	Motor3 Phase 2 -
29	Motor3 -	Motor3 Phase 2 -
30	Motorcodierung	
31	Motor3 MAXSTOP	
32	Motor3 MINSTOP	
33	Encoder3 A	
34	Encoder3 $\overline{A}$	
35	Encoder3 B	
36	Encoder3 $\overline{B}$	
37	Encoder3 Index	
38	Encoder3 $\overline{Index}$	
39	+5V	
40	GND	
41	OWISid	
42	GND-Einspeisung für PWM-Ausgänge (optional)	
43	Motor2 +	Motor2 Phase 1+
44	Motor2 +	Motor2 Phase 1+
45	Motor2 -	Motor2 Phase 1 -
46	Motor2 -	Motor2 Phase 1 -
47	Motor2 +	Motor2 Phase 2+
48	Motor2 +	Motor2 Phase 2+
49	Motor2 -	Motor2 Phase 2 -
50	Motor2 -	Motor2 Phase 2 -
51	Motorcodierung	
52	Motor2 MAXSTOP	
53	Motor2 MINSTOP	
54	Encoder2 A	
55	Encoder2 $\overline{A}$	
56	Encoder2 B	
57	Encoder2 $\overline{B}$	
58	Encoder2 Index	
59	Encoder2 $\overline{Index}$	
60	+5V	
61	GND	
62	OWISid	

3-fach-Motoradapterkabel

Pinbelegung des im Lieferumfang enthaltenen 3-fach-Adapterkabels (3x D-Sub 37-polig, female)

Pin-Nr. HD 62-polig male	Pin-Nr. D-Sub 37-polig female Achse X (1)	Pin-Nr. D-Sub 37-polig female Achse Y (2)	Pin-Nr. D-Sub 37-polig female Achse Z (3)
1	19		
2			
3	18		
4			
5	17		
6			
7	16		
8			
9	14		
10	2		
11	5		
12	11		
13	10		
14	9		
15	8		
16	7		
17	6		
18	12		
19	13+15		
20	29		

22			19
23			
24			18
25			
26			17
27			
28			16
29			
30			14
31			2
32			5
33			11
34			10
35			9
36			8
37			7
38			6
39			12
40			13+15
41			29

Pin-Nr. HD 62-polig male	Pin-Nr. D-Sub 37-polig female Achse X (1)	Pin-Nr. D-Sub 37-polig female Achse Y (2)	Pin-Nr. D-Sub 37-polig female Achse Z (3)
43		19	
44			
45		18	
46			
47		17	
48			
49		16	
50			
51		14	
52		2	
53		5	
54		11	
55		10	
56		9	
57		8	
58		7	
59		6	
60		12	
61		13+15	
62		29	

Motorstecker an 3-fach-Motoradapterkabel

Nachfolgend sind die über 3-fach-Adapter umgesetzten Signale des 62-poligen PS 30-Motorsteckers aufgelistet.
Die Kabelbelegung entspricht dem OWIS®-Standard.

Pinbelegung des 37-poligen D-Sub (female)

	Pin	DC-Motor	Schrittmotor
Leistung	19	Motor +	Phase 1 +
	18	Motor -	Phase 1 -
	17	Motor +	Phase 2 +
	16	Motor -	Phase 2 -

Signale	15	Motorcodierung
	14	Motorcodierung
	13	GND
	12	+ 5 V
	11	Quad A
	10	Quad $\overline{A}$
	9	Quad B
	8	Quad $\overline{B}$
	7	Index
	6	$\overline{Index}$

Schalter + Signale	5	MINSTOP
	4	(reserviert)
	3	(reserviert)
	2	MAXSTOP
	1	(reserviert)
	37	(reserviert)
	36	(reserviert)
	35	(reserviert)
	34	(reserviert)
	33	(reserviert)
	32	(reserviert)
	31	(reserviert)
	30	(reserviert)
	29	OWISid
	28	(reserviert)
	27	(reserviert)
	26	(reserviert)
	25	(reserviert)
	24	(reserviert)
	23	(reserviert)
	22	(reserviert)
	21	(reserviert)
	20	(reserviert)

Anschlusskabel

1. Signalkabel mit Gesamtschirm Twisted Pair 8 x 2 x 0,15 mm²
   + Sternvierer innen, geschirmt, 4 x 0,25 mm²

Paar Nr.	Farbe Ader 1	Farbe Ader 2	Querschnitt
1	rot	blau	0,15 mm²
2	weiß	braun	0,15 mm²
3	grün	gelb	0,15 mm²
4	grau	rosa	0,15 mm²
5	schwarz	violett	0,15 mm²
6	grau/rosa	rot/blau	0,15 mm²
7	orange	orange/schwarz	0,15 mm²
8	transparent	transparent/rot	0,15 mm²
9 a	grün/weiß	grün/braun	0,25 mm²
9 b	gelb/weiß	gelb/braun	0,25 mm²

2. Motorkabel mit Gesamtschirm

Ader Nr.	Farbe	Querschnitt
1	rot	0,6 mm²
2	blau	0,6 mm²
3	weiß	0,6 mm²
4	schwarz	0,6 mm²
5	braun	0,6 mm²
6	rosa	0,5 mm²
7	grau	0,5 mm²

EU/UE Konformitätserklärung/Declaration of conformity

Wir
We

OWIS GmbH
Im Gaisgraben 7
79219 Staufen / Germany
+49(0)7633/9504-0
+49(0)7633/9504-440
www.owis.eu
info@owis.eu

erklären in alleiniger Verantwortung, dass das Produkt
declare under our sole responsibility that the product

PS 30

auf das sich diese Erklärung bezieht, mit den folgenden Normen oder normativen Dokumenten übereinstimmt.
to which this declaration relates is in conformity with the following standards or other normative documents.

EN 61000-6-1:2007 mit/with EN 61000-4-2:2009, EN 61000-4-3:2011
EN 61000-6-3:2011 mit/with EN 55022:2011

Gemäss den Bestimmungen der Richtlinie:
Following the provisions of directive:

2014/30/EU

Ort und Datum der Ausstellung
Place and date of issue

Staufen, 27.09.2017

Name und Unterschrift
Name and signature

D. J. Schuhen
Leitung Vertrieb

i.A. Dr. Peter Hilgers
Leitung Entwicklung

Aktuelle Ausgabe: 27.09.17 DB / DSCH, 2.01.112 FO Konformitätserklärung