

Universal-Positioniersteuerung

PS 90+

9013.0013 / 31.07.2020





## Inhalt

1. Allgemeines .....	5
2. Ausführung und Lieferumfang .....	5
2.1 Standard .....	5
2.2 Zubehör .....	5
2.3 Optionen .....	5
3. Sicherheit .....	6
Ströme und Spannungen .....	6
Erwärmung des Kühlkörpers bis max. 70° C .....	6
4. Normen und Richtlinien .....	6
5. Technische Übersicht .....	7
Eingang .....	7
Anschlüsse .....	7
6. Aufbau der Steuerung .....	8
Ein-/Aus-Taster .....	8
Reset-Taster .....	8
6.1 Anschlüsse .....	9
USB- und RS-232-Schnittstelle .....	9
Ethernet-Schnittstelle .....	9
Anybus®-Schnittstelle .....	9
NOT-AUS-Funktion .....	9
Stromversorgung .....	9
Universal-Motoranschluss .....	9
End- und Referenzschalter .....	10
Encodereingang .....	10
6.2 Eingänge und Ausgänge .....	10
7. Steuerungsarchitektur und Funktion .....	11
7.1 Aufbau .....	11
Hauptplatine .....	11
Antriebsplatine .....	11
Motorplatine .....	11
Sicherungskonzept .....	11
7.2 Betrieb unterschiedlicher Motortypen .....	11
Schrittmotoren .....	11
DC-Motoren .....	11
BLDC-Motoren .....	11
7.3 Konfiguration der Motorendstufe .....	11
2-Phasen-Schrittmotor (Open Loop) .....	11
DC-Servomotor .....	11
7.4 Strombereichumschaltung der Motorendstufe .....	12
Vorwahl des Phasenstromes für 2-Phasen-Schrittmotoren .....	12
Strombereichseinstellung für DC-Servomotoren .....	12
8. Steuerungsfunktionen .....	12
8.1 Trapezförmiges Punkt-zu-Punkt-Profil .....	12
8.2 S-Kurven-Punkt-zu-Punkt-Profil .....	13
8.3 Geschwindigkeitsmodus .....	13
8.4 Referenzierung .....	14
8.5 Linearinterpolation .....	14
Begriffsbestimmung .....	14
Funktionsprinzip .....	14
8.6 Synchroner Start .....	14
8.7 Funktionsweise der allgemeinen Bahnsteuerung .....	14
Definition .....	14
Realisierung des Vektormodus .....	15
Kreisinterpolation .....	16
8.8 Automatisches Reagieren auf externe Auslöser und Setzen von Ausgängen .....	17
8.8.1 Automatisches Reagieren auf Eingänge .....	17
Definition des Eingangs .....	17
Mögliche Aktionen .....	17
Zuweisung einer Aktion zu einem Auslöser .....	18
Verhalten .....	18

## Content

1. General information .....	48
2. Setup and Scope of Delivery .....	48
2.1 Standard .....	48
2.2 Accessories .....	48
2.3 Options .....	48
3. Safety .....	49
Currents and voltages .....	49
Heat Sink Temperature up to 65° C max. ....	49
4. Standards and Directives .....	49
5. Technical Overview .....	50
Input .....	50
Connections .....	50
6. Setup of the Control Unit .....	51
On/Off Switch .....	51
Reset Button .....	51
6.1 Connections .....	52
USB and RS-232 Interfases .....	52
Ethernet interface .....	52
Anybus® Interface .....	52
Emergency-Stop Function .....	52
Power Supply .....	52
Universal Motor Connector .....	52
Limit and Reference Switches .....	53
Encoder Input .....	53
6.2 Inputs and Outputs .....	53
7. Control Unit Architecture and Function .....	54
7.1 Assembly .....	54
Main Board .....	54
Drive Controller Board .....	54
Motor Driver Board .....	54
Safety Fuse Concept .....	54
7.2 Operation of Different Motor Types .....	55
Step Motors .....	55
DC Motors .....	55
BLDC Motors .....	55
7.3 Settings of the Motor Output Stage .....	55
2-Phase Step Motor (Open Loop) .....	55
DC Servo and BLDC Motor .....	55
7.4 Selection of the Current Range for the Motor Power Stage ..	55
Phase Current Setting for 2-Phase Step Motors .....	55
Current Range Setting for DC Servo and BLDC Motors .....	55
8. Control Functions .....	55
8.1 Trapezoidal Point-to-Point Profile .....	55
8.2 S-Curve Point-to-Point Profile .....	56
8.3 Velocity Mode .....	57
8.4 Reference run .....	57
8.5 Operating Mode of Linear Interpolation .....	57
Definition .....	57
Functional Principle .....	57
8.6 Synchronous Start .....	58
8.7 Operation mode of the General Continuous Path Control ..	58
Definition .....	58
Realisation of Vector Mode .....	58
Circular interpolation .....	60
8.8 Automatic Reaction To External Triggers And Setting of Outputs .....	60
8.8.1 Automatic Reaction To External Triggers .....	60
Definition Of An Input .....	60
Possible Actions .....	60
Assigning An Action To A Trigger .....	61
Behaviour .....	61

8.8.2 Automatisches Setzen von Ausgängen .....	18	8.8.2 Automatic Setting Of Outputs .....	61
Definition der Ereignisse .....	18	Definition Of Events.....	61
Zuordnung zu Ausgängen .....	18	Assigning Of Outputs .....	61
Verhalten .....	18	Behaviour .....	62
8.8.3 Konfiguration.....	19	8.8.3 Configuration .....	62
9. Wegerfassung.....	19	9. Travel Measuring.....	62
Encoder.....	19	Encoder .....	62
Linearmesssystem.....	19	Linear Measuring System.....	62
Funktionsweise der Nachlaufregelung.....	19	Function of the follow up control.....	62
10. PID-Regelschleifenalgorithmus .....	21	10. PID Servo Loop Algorithm .....	64
11. Positioniergeschwindigkeit und -beschleunigung, Berechnung ....	21	11. Positioning Velocity and Acceleration, Calculation .....	65
11.1 2-Phasen-Schrittmotor (Open Loop) .....	21	11.1 2-Phase Step Motor (Open Loop).....	65
Allgemeines .....	21	General Information .....	65
Periodendauer .....	21	Cycle Time.....	65
Endgeschwindigkeit.....	21	Final Velocity .....	65
Beschleunigung bei Trapezprofil.....	22	Acceleration for Trapezoidal Velocity Profiling .....	65
11.2 DC-Servomotor und 2-Phasen-Schrittmotor (Closed-Loop) ....	22	11.2 DC Servo Motor and 2-Phase Step Motor (Closed-Loop)..	65
Allgemeines .....	22	General Information .....	65
Abtastzeit.....	22	Servo Loop Cycle Time.....	65
Endgeschwindigkeit.....	22	Final Velocity .....	65
Beschleunigung bei Trapezprofil.....	22	Acceleration for Trapezoidal Velocity Profiling .....	66
12. Nano-Hybrid-Ansteuerung.....	23	12. Nano-Hybrid Control .....	67
Allgemeines .....	23	General Information .....	67
Technische Übersicht und Aufbau der Steuerung.....	23	Technical Overview and Setup of the Control Unit.....	67
Sicherheit .....	23	Safety.....	67
Steuerungsarchitektur und Funktion.....	23	Control Architecture and Function .....	67
Anschluss .....	23	Connection.....	67
Positionierung im Nano-Hybrid-Betrieb .....	23	Positioning in nano-hybrid mode .....	67
Allgemeine Beschreibung der Nachlaufregelung für Piezo-		General Description of Follow-up Control for	
Antriebe .....	24	piezo-drives.....	68
13. Inbetriebnahme der PS 90+ .....	25	13. Initial Operation of the PS 90+ .....	69
13.1 Vorbereitung der Steuerung .....	25	13.1 Installation and Preparing .....	69
Aufstellung.....	25	Installation .....	69
NOT-AUS-Funktion .....	25	Emergency-Stop Function .....	69
13.2 Anschluss der Peripherie und Geräte.....	25	13.2 Connection of Peripherals and Devices .....	69
13.3 Systemstart .....	25	13.3 Getting Started .....	69
Initialisierung.....	25	Initialization .....	69
Software .....	25	Software .....	69
14. Fehlerüberwachung.....	26	14. Malfunction Monitoring.....	70
14.1 Endschalter .....	26	14.1 Limit Switches .....	70
Funktion der Endschalter-Überwachung.....	26	Working Principle of the Limit Switch Monitoring .....	70
Konfiguration der End- und Referenzschalter.....	26	Configuration of Limit and Reference Switches.....	70
Wiederinbetriebnahme nach Achsenfehler .....	26	Reconnection after Axis Error .....	70
14.2 Endstufen-Fehlerüberwachung .....	26	14.2 Output-Stage Error Monitoring .....	70
14.3 Motion-Controller-Fehlerüberwachung .....	26	14.3 Motion-Controller Error Monitoring .....	70
14.4 Time-Out-Überwachung.....	26	14.4 Time-Out Monitoring .....	70
15. Joystick .....	27	15. Joystick .....	71
16. Hinweise zum Aufbau einer eigenen Applikationssoftware....	27	16. Instructions Concerning the Setup of an Own Application Software..	71
17. Befehlssatz der PS 90+ .....	28	17. Command Set for the PS 90+ .....	72
Anhang.....	29	Attachment.....	73
I Befehlstabelle .....	29	I Command Table .....	73
II Relevanz der Parameter für verschiedene Motortypen.....	42	II Parameter Relevance for the different Motor Types .....	85
III Belegungstabellen.....	43	III Connecting Table .....	86
TTL-Ein- /Ausgänge .....	43	TTL In- / Outputs .....	86
Analog-Ein- /Ausgänge .....	43	Analog In- / Outputs.....	86
SPS-Ein- /Ausgänge.....	43	SPS In- / Outputs .....	86
RS-232 .....	43	RS-232 .....	86
Universal-Motorstecker .....	44	Universal Motor Connector.....	87
Anschlusskabel .....	45	Connecting Cable.....	88
Kabelvorschlag für RS-232-Schnittstelle .....	45	Recommendation for a RS-232 Interface Cable .....	88
Konformitätserklärung .....	89	UE Declaration of Conformity .....	89

## 1. Allgemeines

Die OWIS® Steuerung PS 90+ ist eine universelle Positioniersteuerung, die für anspruchsvolle Steuerungsaufgaben eingesetzt wird.

Sie ist modular aufgebaut und wird flexibel auf den jeweiligen Anwendungsbereich konfiguriert.

Die PS 90+ ist sehr leistungsstark und kann bis zu neun Achsen mit Schrittmotoren, DC-, oder BLDC- Servomotoren oder bis zu sechs Nano-Hybrid-Achsen betreiben.

Die in einem stabilen Metallgehäuse untergebrachte Steuerung kann eigenständig (Stand-Alone) oder mit einem Rechner betrieben werden.

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

Ist eine erhöhte Präzision gefordert, kann an jeder Achse ein zusätzliches Inkremental- oder Wegmesssystem angeschlossen werden. Die Steuerung bietet außerdem die Möglichkeit, Schrittmotoren mit einem zusätzlichen Encoder im Closed-Loop-Modus zu betreiben.

Bei Applikationen, für die höchste Präzision gefordert ist, können mit der PS 90+ bis zu sechs Nano-Hybrid-Achsen betrieben werden. Die Hybrid-Technologie verbindet die Vorteile der spindelbetriebene Positionierung mit der Präzision von Piezo-Aktoren.

Die PS 90+ kann Punkt-zu-Punkt-Positionierbetrieb, Trapez- oder S-förmige Geschwindigkeitsprofile, sowie komplexe, mehrachsige Bahnsteuerungen, wie Linearinterpolation oder Kreisinterpolation, ausführen.

Die PS 90+ kann selbständig auf Zustandswechsel der digitalen Eingänge reagieren und zum Beispiel Positioniervorgänge starten oder stoppen. Sie ist außerdem in der Lage, in Abhängigkeit von bestimmten Ereignissen automatisch Ausgänge zu setzen.

Zum Lieferumfang der Steuerung gehört auch die Software OWISoft. Damit kann die PS 90+ 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 90+ besteht aus einem Grundgerät für unterschiedliche Motorspannungen und wird entsprechend den Kundenanforderungen mit Achsmodulen, zusätzlichen Funktionen und Anschlüssen bestückt. Ein Nachrüsten mit Achsmodulen, Funktionen und Anschlüssen ist ebenso möglich. Das Gerät wird bei OWIS® komplett aufgebaut, getestet und anschlussfertig geliefert. Die gültige Firmware für die Steuerung ist eingespielt. Sie kann gegebenenfalls über die USB- oder RS-232-Schnittstelle aktualisiert werden.

Zum Lieferumfang der Steuerung gehören:

- PS 90+ in der gewünschten Motorkonfiguration
- Netzkabel mit 2,5 m Länge
- USB-Kabel mit 2 m Länge
- CD mit OWISoft
- Betriebsanleitung als PDF Version in Deutsch und Englisch
- Kurzanleitung gedruckt und als PDF Version in Deutsch und Englisch
- Datenblatt gedruckte Version in Deutsch und Englisch

### 2.1 Standard

Die Steuerung verfügt über:

- USB-Anschluss
- RS-232-Anschluss
- Anschluss für externen NOT-AUS-Taster
- 4 Eingänge für Referenz- bzw. Endschalter je Achse
- 8 TTL- und Analogeingänge
- 8 TTL- und Analogausgänge
- 8 SPS-Ein- und Ausgänge
- Motoranschluss D-Sub 37-polig mit Anschluss für Motorhaltebremse (Option), End-/Referenzschalter und weitere Signale (siehe Pinbelegung, S.38) + je nach Version bis zu 3 Motorhaltebremsen Anschlüsse

### 2.2 Zubehör

Folgendes Zubehör ist erhältlich:

- Anschlusskabel mit Stecker für unterschiedliche Positioniersysteme
- Joystick für drei Achsen, analog, mit 3 m Kabel
- NOT-AUS-Taster mit 3 m Kabel
- bis zu 4 Ausgänge für Motor-Haltebremsen

### 2.3 Optionen

Es sind folgende Optionen verfügbar:

- Anybus®-Schnittstelle (Modbus/TCP)

### 3. Sicherheit

- Vor der Inbetriebnahme des Gerätes die Gerätebeschreibung lesen und diese für den späteren Gebrauch aufbewahren.
- Die Warnhinweise, Sicherheits- und Installationshinweise müssen beachtet werden.
- Technische Daten und Anschlussbelegungen beachten.
- Das Gerät darf nur für den bestimmungsgemäßen Gebrauch verwendet werden.
- Das Gerät ist für Innenraumanwendungen konzipiert und darf nicht im Freien verwendet werden.
- Das Gerät muss vor zu hoher Luftfeuchtigkeit (80%), Erschütterungen sowie explosiven Gasen geschützt werden.
- Das Gerät darf nur von dafür qualifiziertem Fachpersonal in Betrieb genommen und verwendet werden.
- Die geltenden Installations-, Sicherheits- und Unfallverhütungsvorschriften sind einzuhalten.
- Das Gerät darf nur in seinem geschlossenen Metallgehäuse betrieben werden.
- Anschluss-, Montagearbeiten und Sicherungswechsel dürfen nur im spannungsfreien Zustand des Gerätes ausgeführt werden.
- Sicherungswechsel darf nur von einer autorisierten Fachkraft vorgenommen werden.
- Es darf nur die vorgeschriebene Sicherung verwendet werden.
- Nicht verwendete Slots müssen mit der vorgesehenen metallischen Slot-Blende abgedeckt sein.
- Vor dem Öffnen des Gerätes muss das Gerät spannungsfrei geschaltet und vom Stromnetz getrennt werden. Das Gerät ausschalten und das Kaltgerätekabel vom Gerät und der Netzversorgung entfernen!
- Das Gerät erzeugt Wärme (Netzteil/Endstufen). Die Lüftungsschlitze nicht abdecken, genügend Abstand zu anderen Gegenständen einhalten (mind. 15cm).
- Es dürfen nur dafür vorgesehene Komponenten und Betriebsmittel, so wie Kabel und Leitungen, welche den geltenden Bestimmungen/Normen entsprechen, angeschlossen werden.
- Es dürfen keine Leitungen mit Netzspannungs- oder gefährlichen Potentialen am Gerät angeschlossen werden (ausgenommen Netzanschluss).
- Schäden die durch Nichtbeachtung dieser Hinweise entstehen, sind von Ansprüchen jeglicher Art ausgeschlossen

Die Steuerung hat je nach Ausführung ein Gewicht von etwa 15 kg. An der Frontseite unten befindet sich eine Griffmulde und an der Rückseite oben ein Handgriff. Damit kann die PS 90 sicher transportiert werden.

Das Steuergerät ist für Betriebstemperaturen von + 10 bis + 40°C und Lagertemperaturen von - 10 bis + 50°C konzipiert.

Die PS 90 hat eine NOT-AUS-Schaltung, deren Funktion an die EN 418 angelehnt ist. Sie unterbricht die Leistungsversorgung der Motorendstufen auf der Sekundärseite (Klein Spannungsbereich 24 V bzw. 48 V). 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

losläuft.

Die jeweiligen Achsmodule der Steuerungen dürfen nur mit den für sie konfigurierten Motortypen betrieben werden. Andere oder weiterführende Nutzungen entsprechen nicht dem vorgesehenen Verwendungszweck.

#### Ströme und Spannungen

Das Schaltnetzteil der PS 90+ besitzt einen Weitbereichseingang für eine Primärspannung von 100VAC bis 240VAC mit 50/60 Hz. Der Netzeingang ist über eine Feinsicherung 15AT (480W) abgesichert.

Ausgangseitig sind keine besonderen Sicherheitsvorkehrungen erforderlich, da die PS 90+ ausschließlich mit Kleinspannung (PELV) bis 48VDC arbeitet. Falls die PS 90+ zur Ansteuerung von Nano-Hybrid-Achsen konfiguriert ist, wird der Piezozweig mit Spannungen im Bereich von -71V bis +71V betrieben. Besondere Sicherheitshinweise finden Sie im Kapitel „Nano-Hybrid-Ansteuerung“.

#### Erwärmung des Kühlkörpers bis max. 70° C

Während des Betriebs der Steuerung wird die Abwärme der eingebauten Motorplatinen (Endstufen) über den seitlich angebrachten Kühlkörper an die Außenluft abgegeben.

Je nach Anzahl und Größe (Stromaufnahme) der angeschlossenen Motoren, sowie der Betriebsart (Kurzzeit-, Aussetz-, Dauerbetrieb) erwärmt sich der Kühlkörper und kann eine Temperatur von maximal 70° C erreichen. Wärmestau in der Steuerung oder am Kühlkörper ist zu vermeiden.

Es muss ein Mindestabstand von 15 cm zu geschlossenen Flächen und Wänden eingehalten werden.

Bei Nichtbeachtung der Sicherheitshinweise der Betriebsanleitung sind Sachschäden sowie Personenschäden möglich. Daher müssen diese jedem Nutzer zugänglich gemacht und eingehalten werden.

Die Universal-Positioniersteuerung PS 90+ ist nach den anerkannten sicherheitstechnischen Regeln gebaut und erfüllt die im folgenden Kapitel aufgeführten Normen und Richtlinien.

### 4. Normen und Richtlinien

#### Richtlinien:

##### 2014/30/EU (EMV-Richtlinie)

Harmonisierte Normen

EN 55011:2016 + A1:2017

EN 61000-6-2:2005

EN 61000-3-2:2014

EN 61000-3-3:2013

##### 2014/35/EU (Niederspannungs-Richtlinie)

Harmonisierte Norm

EN 61010-1:2010

##### 2011/65/EU (RoHS-Richtlinie)

Harmonisierte Norm

EN 50581:2012

## 5. Technische Übersicht

Hersteller	OWIS GmbH
Name	Universal-Positioniersteuerung PS 90+

### Eingang

Netzanschluss	Kaltgerätebuchse (C20), (Leitungslängen max. 3m, Schutzleiteranschluss)
Spannung:	100...240V AC 50/60Hz ±10%
Strom	max. 15 A
Sicherung	interne Sicherung 15 AT

**! Hinweis:**  
 ■ Die Stromaufnahme ist abhängig von der Ausstattung/Ausbau-  
 stufe und von den angeschlossenen Komponenten/Peripherie.

Betriebstemperatur	+10...40°C, 80% relative Feuchte, nicht kondensierend
Lagertemperatur	+10...50°C, 80% relative Feuchte, nicht kondensierend
Schutzart	IP20
Verwendungsort	Innenräume
Betriebshöhe	bis zu max. 2000m
Verschmutzungsgrad	2 (trocken nicht leitend)
Überspannungskategorie II	
Schutzklasse	I (Schutzleiter)
Gehäuse	Metallgehäuse
Maße	ca. 20x50x50cm BxHxT
Gewicht	ca. max. 15 kg, (abhängig von der Ausführung/Ausstattung)

### Anschlüsse

PC/RS232	D-SUB Buchse 9pol. female; Kabellänge max. 3 m, geschirmt
PC/USB	USB 2.0, USB-Buchse Typ-B, female; Kabellänge max. 3 m, geschirmt
Anybus	Abhängig vom Modul
STOP	Rundstecker, female, max. 3 m. geschirmt
TTL I/O	D-SUB 37pol. male; Kabellänge max. 3 m, geschirmt
SPS I/O	D-SUB 37pol. female; Kabellänge max. 3 m, geschirmt
Analog I/O	D-SUB 37pol. male; Kabellänge max. 3 m, geschirmt
2x LAN	2x Ethernet, RJ45-Buchse LAN-Kabel, CAT5; geschirmt, Kabellänge max. 10m
M1 bis M9	Motoren/Antriebe, D-SUB 37 pol. female; Kabellänge Standard 3m geschirmt, längere Kabel auf Anfrage

### Bedienelemente:

on/off

I/O

### ! ACHTUNG:

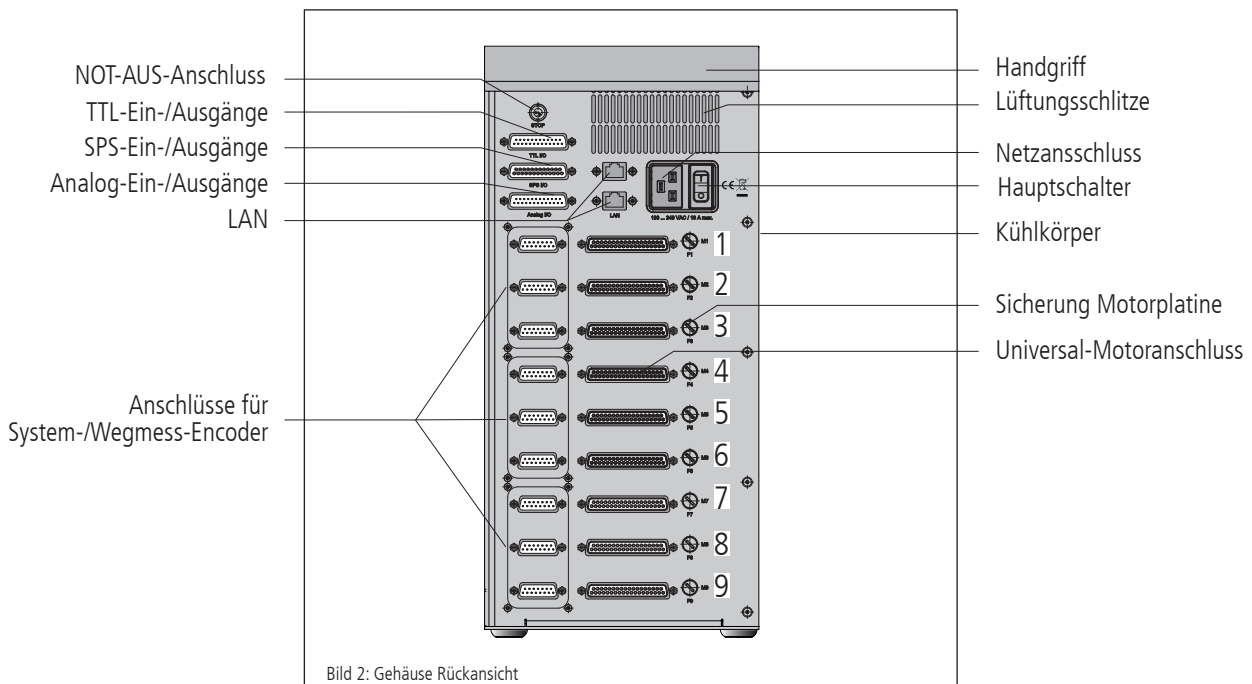
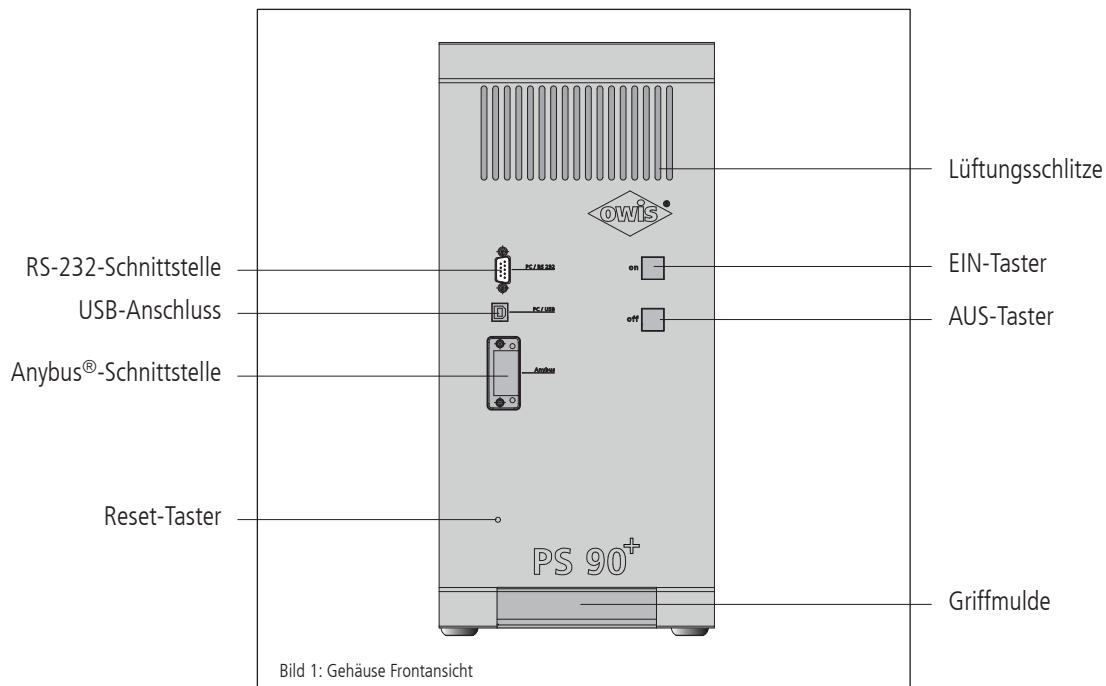
- Für den Netzanschluss darf nur das zugehörige Kaltgerätekabel verwendet werden!

### ! VORSICHT:

- Zur sicheren Trennung von der Netzspannung, das Gerät ausschalten und das Kaltgerätekabel vom Gerät und vom Netz entfernen!

WARNUNG	Sicherheits- und Installationsvorschriften beachten! Anschlussbelegungen beachten, bei Fehlanschluss Funktionsausfall oder Zerstörung möglich!
Stromversorgung	Schaltnetzteil mit Strombegrenzung
Anzahl der Antriebe	bis zu 9 Achsen (SM, DC und BLDC), bis zu 6 Nano-Hybrid-Achsen (SM)
Antriebsart	2-Phasen Schrittmotoren Open Loop (OL), 2-Phasen Schrittmotoren Closed-Loop (CL), DC-Servomotoren, BLDC-Servomotoren, Nano-Hybrid-Achsen
Kommunikation	USB 2.0, RS-232, optional Anybus <sup>®</sup> -Schnittstelle (Modbus/TCP)
Aufbau	Tischgerät in Metallgehäuse Schutzart: IP 20
Encoder	Quadratur-Signale A/B und Index, RS-422- oder TTL-Pegel, mit 4-fach- Auswertung, max. Zählfrequenz 5 MHz (Signal) bzw. 20 MHz (Quadratur)
Funktionen	Parametrierbare Beschleunigungsrampe ( / Bremsrampe), Dreieckiges- bzw. trapezförmiges Geschwindigkeitsprofil oder S-Kurve
Bewegungsabläufe	Punkt-zu-Punkt Positionierbetrieb und Linear- und Kreisinterpolation

## 6. Aufbau der Steuerung



Die PS 90+ ist in einem hochwertigen und stabilen Metallgehäuse untergebracht.

Zur internen Kühlung sind auf der Gehäusevorder- und Rückseite im oberen Bereich Lüftungsschlitze angebracht. Die Abwärme der eingebauten Motorplatinen (Endstufen) wird über den seitlich angebrachten Kühlkörper an die Außenluft abgegeben.

### Ein-/Aus-Taster

Hauptschalter sowie Ein-/Aus-Taster der Steuerung sind beleuchtet. Die Beleuchtung des Hauptschalters zeigt die Aktivierung der Steuerung an. Die Taster werden nach dem Einschalten der Endstufen-Leistungsversorgung durch Betätigung des Ein-Tasters zur Betriebskontrolle mit maximaler Intensität beleuchtet. Betätigung des Aus-Tasters schaltet die Motorendstufen aus und reduziert die Hintergrundbeleuchtung der Taster auf ein Minimum.

### Reset-Taster

Falls die PS 90+ nicht mehr reagiert oder unerwartete Fehler auftreten, kann der versenkte Reset-Taster z.B. mittels einer Kugelschreiber Spitze betätigt werden. Alternativ ist es möglich, das Gerät aus- und wieder einzuschalten.



## 6.1 Anschlüsse

Die Anschlüsse der PS 90+ befinden sich auf der Gehäusevorder- und auf der Gehäuserückseite. Dies sind Schnittstellen zur Kommunikation, Ein- und Ausgänge für Peripherie, sowie Anschlüsse für die Positionierer (siehe Bild 1 und 2).

Anschluss	Funktionen	Buchse
USB-Slave	Kommunikation mit einem PC	USB-Buchse Typ B
RS-232	Kommunikation mit einem PC	D-Sub 9-poliger Stecker
Ethernet-Schnittstelle	Verbindung mit einem PC über Ethernet / Integrierte Switch-Funktion	RJ 45
RS-485	Fernbedienung der Steuerung mit dem Handterminal	D-Sub 9-polige Buchse
Joystick	Manuelles Verfahren von maximal 3 Achsen	Analogeingänge 1, 2, 3
TTL-Ein-/Ausgänge	Interaktion mit externen Sensoren und Aktoren	D-Sub 25-poliger Stecker
Analog-Ein-/Ausgänge	Interaktion mit externen Sensoren und Aktoren, Joystick	D-Sub 25-poliger Stecker
SPS-Ein-/Ausgänge	Interaktion mit externer (SPS)-Steuerung	D-Sub 25-polige Buchse
Universal-Anschlussstecker	Motorversorgung mit Motor-Haltebremse und Encoder-/Endschalteranschluss	D-Sub 37-polige Buchse
Wegmesssystem/ Encoder	Anschluss von Wegmesssystemen	CONNEI 12-polige Buchse
Netzanschluss	Spannungsversorgung	Kaltgerätebuchse

### Option:

Anybus-Modul Modbus/TCP	Kommunikation mit einem PC über Ethernet	RJ 45
-------------------------	--	-------

### USB- und RS-232-Schnittstelle

Die PS 90+ hat eine USB 2.0-Slave-Schnittstelle, der Anschluss befindet sich auf der Geräterückseite. Die Schnittstelle ist USB 1.1 und 2.0 kompatibel. Die USB-Schnittstelle der PS 90+ ist als sogenannte

COM-Brücke realisiert. Der Windows-Gerätetreiber erkennt die PS 90+

als „USB-Serial-Port“ und weist ihr eine COM-Portnummer zu, die vom Anwender bei Bedarf verändert werden kann. Die USB-Schnittstelle wird nach erfolgreicher Installation als virtuelle RS-232-Schnittstelle angesprochen.

Alternativ zur USB-Schnittstelle kann die Steuerung über die RS-232 mit einem PC kommunizieren.

Die PS 90+ 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 90+ mit der im Gerätetreiber eingestellten Übertragungsrate übereinstimmt, sonst ist keine Kommunikation möglich. Voreinstellung ist 9 600 Baud. (Kann aus dem Abnahmeprotokoll entnommen werden.)

### Ethernet-Schnittstelle

Die PS 90+ besitzt zwei Ethernet-Schnittstellen mit integrierter Switchfunktion. Über einen der beiden Anschlüsse kann die Steuerung mit dem lokalen Netzwerk verbunden werden. Der zweite Anschluss steht zum Verbinden weiterer Geräte zur Verfügung. Somit kann bei nur

einem vorhandenen Netzwerkanschluss die Steuerung mit dem Netzwerk und ein PC über die Steuerung ebenfalls mit dem Netzwerk verbunden werden. Zum Verbinden muss der Port 8777 gewählt werden.

### Anybus®-Schnittstelle

Die PS 90+ kann optional mit einem Anybus®-Modul „Modbus/TCP“ geliefert werden. Mit diesem Modul ist es möglich, über Ethernet Kommandos zur PS 90+ zu schicken und entsprechende Rückmeldungen zu erhalten.

### NOT-AUS-Funktion

An der Geräterückseite ist ein Anschluss für einen externen NOT-AUS-Taster vorgesehen, an welchem standardmäßig ein Kurzschlussstecker eingesteckt ist. Soll ein NOT-AUS-Taster angeschlossen werden, ist der Kurzschlussstecker zu entfernen.

#### ! Hinweis:

- Wird der Kurzschlussstecker entfernt und kein NOT-AUS-Taster angeschlossen, ist die Funktion der Motorplatinen (Motorendstufen) blockiert

Die NOT-AUS-Schaltung der PS 90+ ist angelehnt an die EN 418 und unterbricht die Leistungsversorgung der Motorendstufen auf der Sekundärseite (Kleinspannungsbereich 24V oder 48V). Die Funktion wird durch ein selbthaltendes Relais mit zwangsgeführten Kontakten (2 Öffnerkontakte in Reihe) umgesetzt. Bei Abschaltung der Endstufen wird zusätzlich zur Endstufenversorgung die Endstufenfreigabe weggeschaltet (doppelte Sicherheit).

### Stromversorgung

Die Stromversorgung der PS 90+ ist für eine Eingangsspannung von 100VAC bis 240VAC mit 50/60 Hz ausgelegt (Weitbereichseingang). Ein Schaltnetzteil generiert 24VDC und versorgt die Ein- und

Ausgänge auf der Hauptplatine. Die Logikspannungen + 5V, + 2,5V und + 3,3V für Haupt- und Antriebsplatine werden aus dieser 24VDC-Versorgung erzeugt. Ein zweites Schaltnetzteil generiert die Zwischenkreisspannung für die Motorplatinen (wahlweise 24 oder 48VDC). Diese Spannung speist die Leistungsstufen der Motorplatinen.

Die Versorgungsspannungen für Logik und Leistung sind galvanisch getrennt.

### Universal-Motoranschluss

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

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
- 470 Ohm: BLDC

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“ und „?MPUNISTAT<n>“ ausgelesen (siehe Befehlssatz ab S. 26).

Der Steckerbelegungsplan ist im Anhang aufgeführt. Die Belegung entspricht dem OWIS®-Standard.

### End- und Referenzschalter

Pro Achse können maximal 4 Schalter angeschlossen werden. Dies können 24V-Induktivschalter, Mikroschalter, TTL-Hall-Effekt-Endschalter oder TTL-Lichtschranken sein. An die Eingänge können beliebige Pegel,  $\pm 5V - \pm 24V$ , Öffner oder Schließer, gegen  $+U_b$  oder Masse schaltend, angeschlossen werden.

Einer der vier 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.

Folgende Eingangssignale sind definiert:

Versorgung	$V_{CC}$ (+ 5V); GND
Kanal	A (TTL oder CMOS)
Kanal	A invertiert
Kanal	B (TTL oder CMOS)
Kanal	B invertiert
Kanal	I (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 hocho

## 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 einfache Gabellichtschranken etc. angeschlossen werden.

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

Die SPS-kompatiblen Eingänge ermöglichen die Verwendung der im Anlagenbau üblichen 24VDC-Induktiv-Sensoren in Zweidraht- und Dreidraht-Technik. Die Arbeitswiderstände der SPS-Eingänge können per Software gemeinsam als Pull-Up oder Pull-Down geschaltet werden.

Die SPS-Ausgänge steuern Magnetventile oder sonstige induktive und ohmsche Lasten direkt an (gegen + 24V schaltend). Die Ausgänge sind kurzschlussfest.

Eigenschaften	Pegel	Strom	Sonstiges
TTL-Eingänge	0-5V		—
SPS-Eingänge	0-24VDC		2-Draht/3-Draht
Analogeingänge	0-4,096VDC		Auflösung 10 Bit
TTL-Ausgänge	0-5V	10 mA	—
SPS-Ausgänge	0-24VDC	300 mA	kurzschlußfest
Analogausgänge	0-4,096VDC	10 mA	Auflösung 10 Bit
Leistungsausgänge	0-24VDC	1,0A	PWM

Die analogen Eingänge können Spannungen zwischen 0V und 4,096V direkt messen und mit 10-Bit-Auflösung wandeln (Referenzspannung: 4,096 V). Die Ein- und Ausgänge sind nicht galvanisch getrennt.

Die Abfragebefehle „?ANIN<uv>“ und „?INPUTS“ beziehen sich auf dieselben Eingänge der PS 90 (siehe Befehlssatz ab S. 26).

Die Auswertung der Eingänge erfolgt entweder analog oder digital.

Die vier Leistungsausgänge sind pulswertenmoduliert 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.

Die NOT-AUS-Schaltung der PS 90 ist angelehnt an die EN 418 und unterbricht die Leistungsversorgung der Motorendstufen auf der Sekundärseite (Klein Spannungsbereich 24V oder 48V). Die Funktion wird durch ein selbsthaltendes Relais mit zwangsgeführten Kontakten (2 Öffnerkontakte in Reihe) umgesetzt. Bei Abschaltung der Endstufen wird zusätzlich zur Endstufenversorgung die Endstufenfreigabe weggeschaltet (doppelte Sicherheit).

## 7. Steuerungsarchitektur und Funktion

Die Steuerung besteht im Wesentlichen aus folgenden Komponenten:

1. ein eingebautes Netzteil
2. eine Hauptplatine
3. max. 9 Antriebsplatinen
4. max. 9 Motorplatinen (Endstufen)

### 7.1 Aufbau

#### Hauptplatine

Die Hauptplatine ist das Kernstück der PS 90+. Sie übernimmt die Steuerung des Hauptablaufs, kommuniziert mit dem PC und mit den Antriebsplatinen und verwaltet die digitalen und analogen Ein- und Ausgänge.

Die Hauptplatine hat einen USB-Anschluss für die Kommunikation mit einem PC. Eine weitere RS-232-Schnittstelle ist als alternative Kommando-Schnittstelle zum PC implementiert.

Mit dem optionalen Anybus®-Modul „Modbus/TCP“ ist die Kommunikation mit einem PC über Ethernet möglich.

#### Antriebsplatine

Jede Antriebsplatine beinhaltet ferner einen Motion-Prozessor, der eine Achse steuern bzw. regeln kann. Der Motion-Prozessor verarbeitet die Befehle des Mikrocontrollers und generiert entsprechend die Ansteuersignale für die Endstufenmodule. Die Schnittstelle zu den Endstufen ist mittels Optokoppler galvanisch getrennt.

#### Motorplatine

Die PS 90+ kann mit maximal neun Motorplatinen bestückt werden.

Auf der Motorplatine befindet sich die Endstufe, die die Wicklung(en) des Motors mit Strom versorgt und damit das Drehmoment steuert.

Die Motorplatine ist mit dem Universal-Anschlussstecker verbunden, an dem der Motor mit seinen Wicklungen, der Encoder, evtl. die Hall-Effekt-Kommutierungssensoren und alle Schalter, die zu dieser Antriebsachse gehören, angeschlossen sind.

#### Sicherungskonzept

Für jede Motorplatine ist eine eigene Schmelzsicherung (5 x 20 mm) vorhanden, die entsprechend dem maximal auftretenden Strom ausgelegt ist. Sie soll helfen, im Falle eines Hardwaredefektes größeren Schaden zu vermeiden. Die Sicherung ist von der Geräte-rückseite zugänglich und kann von außen getauscht werden.

Standardmäßige Absicherung: 6,3 AT. Zusätzlich ist jede Motorplatine mit einer elektronischen Sicherung versehen. Wird der maximal zulässige Phasenstrom überschritten, so wird die Platine abgeschaltet. Zusätzlich wird die Freigabe entfernt.

### 7.2 Betrieb unterschiedlicher Motortypen

#### Schrittmotoren

Die PS 90+ ist für den Betrieb von 2-Phasen-Schrittmotoren ausgelegt, die sowohl gesteuert (Open Loop), als auch geregelt (Closed-Loop) betrieben werden können.

#### DC-Motoren

Die PS 90+ kann ebenfalls DC-Motoren (bürstenbehaftete Servomotoren) ansteuern.

Die Endstufe ist als H-Brücke mit Strombegrenzung ausgeführt, die mit einem PWM-Signal und einem Richtungssignal angesteuert wird. Es ist eine automatische Strombegrenzung eingebaut, die beim Überschreiten des maximalen Motorstroms anspricht.

#### BLDC-Motoren

Ein Betrieb von BLDC-Motoren (bürstenlose Servomotoren) mit drei Motorphasen ist auch möglich.

Die Endstufe steuert drei Motorwicklungen mit drei vom Motion-Controller generierten 50/50-PWM-Signalen an. In allen 3 Brückenweigen wird der Summenstrom gemessen. Treten zu hohe Motorströme auf, so wird der Strom mittels Stromchopper begrenzt.

### 7.3 Konfiguration der Motorendstufe

Die Endstufen sind ab Werk auf einen Motortyp fest vorkonfiguriert. Diese Einstellung kann vom Anwender nicht geändert werden. Im Folgenden werden die Konfigurationsmöglichkeiten für die jeweiligen Typen beschrieben.

#### 2-Phasen-Schrittmotor (Open Loop)

Für diesen Motortyp ist keine Strombegrenzung vorgesehen. Die Einstellung des Motorstromes ist im folgenden Abschnitt 7.4 beschrieben.

Die Stromregelung erfolgt über einen PID-Regler. Dieser darf nicht mit dem PID-Regler zur Positionslagebestimmung verwechselt werden, auch wenn die Begriffe identisch sind.

Über vier Parameter (P-, I-Schnell-, I-Langsam und D-Wert) wird die Reglercharakteristik bestimmt. Ungünstige, zu hoch eingestellte Werte können dazu führen, dass der Motor pfeift. Zu niedrige Werte reduzieren die maximal erreichbare Geschwindigkeit. Für jeden Motortyp müssen die optimalen Parameter individuell eingestellt werden. Die häufig bei Schrittmotoren auftretenden Geräusche im Betrieb können durch eine günstige PID-Einstellung stark reduziert werden. Insbesondere bei niedrigen Maximalgeschwindigkeiten ist so ein besonders geräuscharmer Betrieb möglich.

Die PS 90 wird in Kombination mit OWIS®-Positioniereinheiten bereits mit passenden Reglereinstellungen ausgeliefert. In OWISoft sind außerdem Parametersätze hinterlegt, die entweder für geräuschreduzierten, langsamen oder sehr dynamischen Betrieb optimiert sind.

#### DC-Servomotor

Beim Betrieb von DC-Servomotoren oder BLDC-Motoren ist es üblich, eine Strombegrenzung einzustellen. Dies erfolgt über den Befehl DRICUR (siehe Befehlsreferenz). Die Strombegrenzung wird nach dem Einschalten der Steuerung mit der ersten Initialisierung übernommen. Um die Begrenzung zu ändern ist ein Neustart der Steuerung notwendig. DRICUR<n>=100 entspricht dabei 100% von 12 A. Die Werte müssen entsprechend kleiner gewählt werden. Eine zu niedrig eingestellte Begrenzung reduziert die verfügbare Dynamik deutlich, da DC-Motoren und BLDC-Motoren beim Beschleunigen und Verzögern kurzzeitig größere Ströme benötigen. Dies stellt in der Regel keine Gefahr für den Motor dar.

## 7.4 Strombereichumschaltung der Motorendstufe

Die PS 90+-Endstufe besitzt zwei umschaltbare Strombereiche, um möglichst hohe Auflösung der Stromeinstellung bzw. möglichst feinen Mikroschrittbetrieb zu ermöglichen.

Der gewählte Strombereich wird abgespeichert. Um den neuen Strombereich zu aktivieren, ist es erforderlich, die Achse <n> nach der Bereichumschaltung neu zu initialisieren.

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

```
AMPSHNT<n>=1
```

```
INIT<n>
```

Zurückschalten in Strombereich 1 (niedrig) kann mittels folgender Befehlssequenz vorgenommen werden:

```
AMPSHNT<n>=0
```

```
INIT<n>
```

### Vorwahl des Phasenstromes für 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>
```

Maximaler Phasenstrom Strombereich 1  
(entsprechend 100%): 2,4A

Maximaler Phasenstrom Strombereich 2  
(entsprechend 100%): 5,45A

#### **Hinweis:**

- Alle Strombereich 2 darf maximal ein Phasenstrom von 3,6A, entsprechend 66% des Endwerts, eingestellt werden.

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

### Strombereichseinstellung für DC-Servomotoren

Für DC-Servomotoren ist der geeignete Strombereich unter Berücksichtigung des thermisch zulässigen Dauerstroms des jeweiligen Motortyps vorzuwählen. Eine Strombegrenzung kann durch das Setzen der jeweiligen Parameter eingestellt werden. (Weitere Hinweise sind im Kapitel „Einstellelemente der Motorendstufe“ zu finden.)

## 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	Wortlänge	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/Cycle <sup>2</sup>
Verzögerung	16.16	32 bit	(1...2.147.483.647)/65.536 Counts/Cycle <sup>2</sup>

Für dieses Profil errechnet der Host eine Beschleunigung, eine Ver-

zögerung, eine Geschwindigkeit und eine Endposition.

Das Profil ist nach der Kurvenform (Bild 3, 5) benannt: Die Achse beschleunigt linear (anhand des programmierten Beschleunigungswertes), bis sie die programmierte Geschwindigkeit erreicht. Die Achse bremst dann linear ab (den Verzögerungswert 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 4).

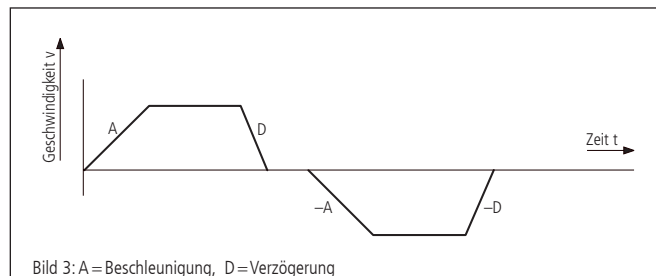


Bild 3: A = Beschleunigung, D = Verzögerung

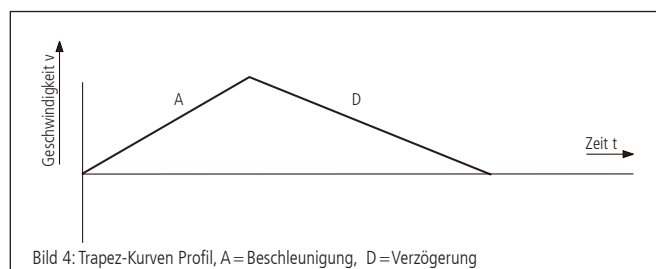


Bild 4: Trapez-Kurven Profil, A = Beschleunigung, D = Verzögerung

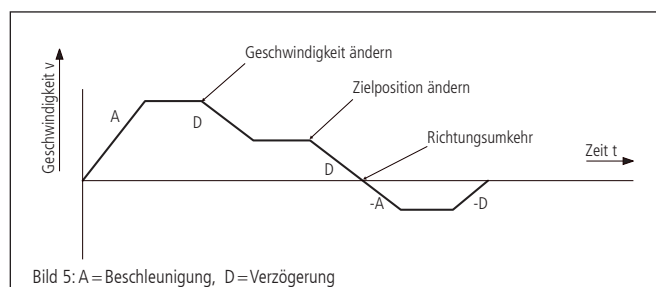


Bild 5: A = Beschleunigung, D = Verzögerung

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 90+ 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	Wortlänge	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/Cycle <sup>2</sup>
Verzögerung	16.16	32 bit	(1...2.147.483.647)/65.536 Counts/Cycle <sup>2</sup>
Jerk	0.32	32 bit	(1...2.147.483.647)/4.294.967.296 Counts/Cycle <sup>3</sup>

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 Verzögerungsrampe benutzt werden. Asymmetrische Profile sind nicht erlaubt. Dies ist nur im trapezförmigen Profilmodus möglich.

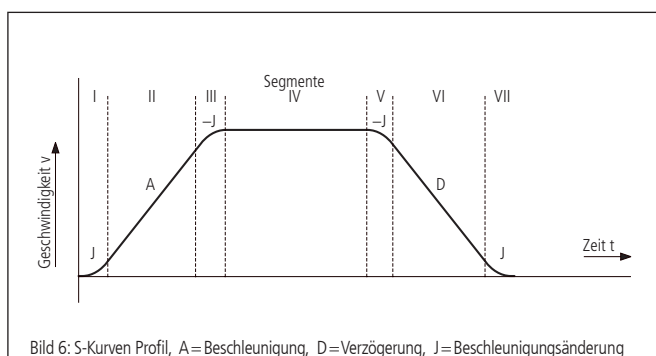
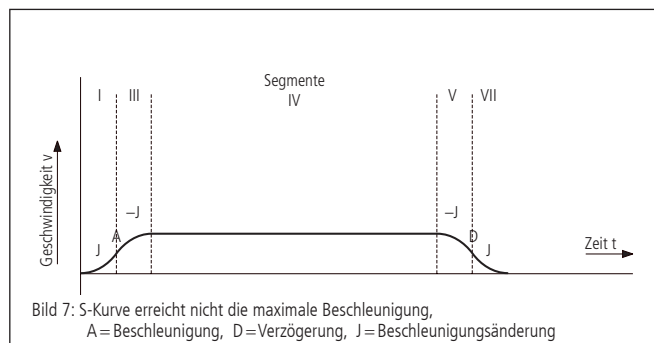
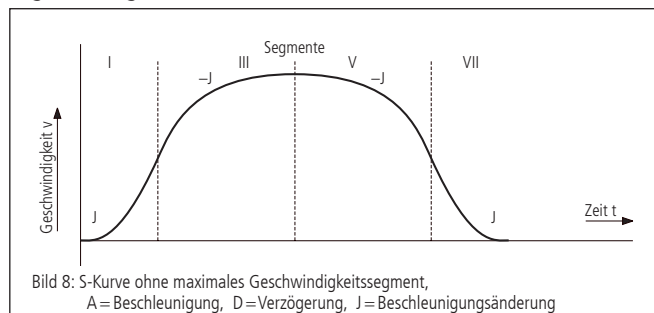


Bild 7 zeigen ein typisches S-Kurven-Profil. In Segment I erhöht sich der Beschleunigungswert um den per Jerk gesetzten Wert, bis die maximale Beschleunigung erreicht wurde. Im nächsten Segment wird die Achse linear (Jerk = 0) beschleunigt. Das Profil wendet dann im Segment III den negativen Wert des Jerks an, um die Beschleunigung zu reduzieren. Im Segment IV verfährt die Achse jetzt mit maximaler (programmierter) 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 (A), 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 7 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 7).



Falls eine Position derart angegeben wird, dass die Endgeschwindigkeit nicht erreicht werden kann, wird es kein Segment IV geben (siehe Bild 8).



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	Wortlänge	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/Cycle <sup>2</sup>
Verzögerung	16.16	32 bit	(1...2.147.483.647)/65.536 Counts/Cycle <sup>2</sup>

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 90+ 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 hat 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 4 dargestellt. Bild 10 zeigt ein komplizierteres Profil, in dem beides, die Geschwindigkeit als auch die Bewegungsrichtung, zweimal wechseln.

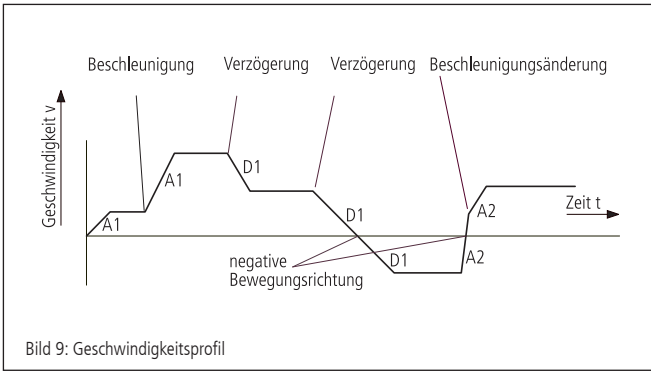


Bild 9: Geschwindigkeitsprofil

**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 parametrisiert. Der Endschalter wird mit großer Geschwindigkeit angefahren und mit kleiner Geschwindigkeit verlassen, dann wird gestoppt.

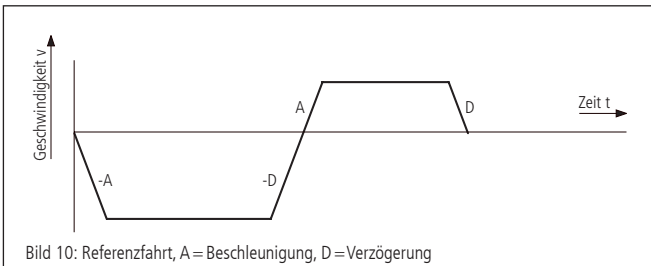


Bild 10: Referenzfahrt, A=Beschleunigung, D=Verzögerung

### 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 Verfahrensweg (umgerechnet in Inkremente) zurückzulegen hat, wird als Führungssachse f bezeichnet. Auf diese Achse werden die restlichen an der Linearinterpolation beteiligten Achsen steuerungsintern per Software synchronisiert.

#### Funktionsprinzip

Welche der maximal 9 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 Führungssachse f schnellstmöglich (mit maximal möglicher Geschwindigkeit  $v_{max}(f)$  und Beschleunigung  $a_{max}(f)$ ) ihr Ziel erreicht. Die restlichen Achsen werden auf die Führungssachse 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 Führungssachse  $v_f(t)$  und einer beliebigen Linearinterpolationsachse  $v_n(t)$  an einem Beispiel:

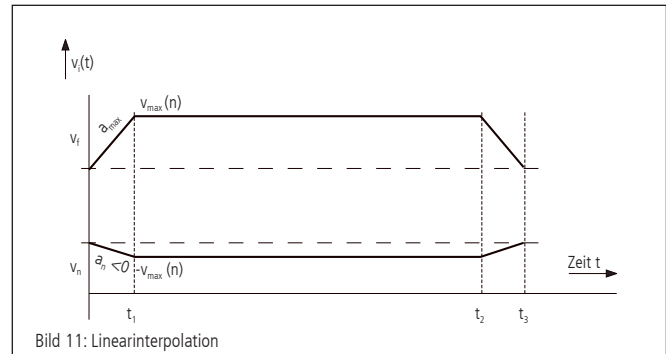


Bild 11: Linearinterpolation

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

### 8.6 Synchroner Start

Ähnlich der Linearinterpolation ist es möglich, den Positioniervorgang oder den Geschwindigkeitsmodus mehrerer Achsen synchron starten zu lassen.

Mit Hilfe der dazu notwendigen Befehle (siehe Befehlsreferenz) werden alle Berechnungen zunächst intern durchgeführt. Anschließend erfolgt ein nahezu gleichzeitiger Start aller ausgewählter Achsen. Im Unterschied zur Linearinterpolation führt jede Achse ihre Bewegung so aus, wie sie es auch beim einzelnen Start geschehen würde.

### 8.7 Funktionsweise der allgemeinen Bahnsteuerung

#### Definition

Die PS 90+ 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  $\vec{\Delta X}$  für maximal neun Achsen ( $a$  bis  $i$ , entsprechend den Achsnummern 1 bis 9), das für die Fahrt des Vektors vorgegebene Zeitintervall  $\Delta t$ , einen 16-Bit-Funktionscode  $F$ , einen 8-Bit-Fehlercode  $E$  und einen 8-Bit-Achsfreigabecode  $T$ :

$n$	$\vec{\Delta X}$	$\Delta t$	$F$	$E$	$T$
1	$\Delta X_{a1}, \dots, \Delta X_{i1}$	$t_1$	$t_1$	$e_1$	$t_1$
...	...	...	...	...	...
$N$	$\Delta X_{aN}, \dots, \Delta X_{iN}$	$\Delta t_N$	$f_N$	$e_N$	$t_N$

Es können maximal 4000 Vektoren definiert werden ( $N_{\max} = 4000$ ).

Die Elemente des Fahrtvektors (Einzeldistanzen) werden als ganzzahlige Werte mit Vorzeichen (Integer 16-Bit) dargestellt. Die maximale Wegdistanz für ein Zeitintervall  $\Delta t_n$  beträgt 2147483648 Inkremente, d.h. als Wertebereich für einen Positionseintrag ist ein Zahlenwert von -2147483648 bis +2147483648 zulässig.

### Segmentdauer

Das Zeitintervall  $\Delta t_n$  für Fahrsegment  $\langle n \rangle$  wird als ganzzahliges Vielfaches von 1,024 ms angegeben. Der Wertebereich reicht von 20 bis 65535, woraus sich eine definierbare Segmentzeit von minimal 20,48 ms bis maximal 67,10784 s in Schritten von 1,024 ms ergibt:

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

$$\Delta t_{n_{\max}} = 65535 \cdot 1,024 \text{ ms} = 67,10784 \text{ s}$$

### 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. Bit 15 wird zur Vorwahl der Betriebsart, d.h. „konstante Geschwindigkeit“ ( $v = \text{const.}$ , Bit 15 gelöscht) oder „konstante Beschleunigung“ ( $a = \text{const.}$ , Bit 15 gesetzt), verwendet.

Die restlichen Bits werden benutzt, um bis zu drei Ausgänge pro Zeile jeweils entweder zu setzen oder zu löschen. Über Bits 0-3 wird binär der Ausgang gewählt. Bit 4 entscheidet dann, ob der Ausgang gesetzt oder gelöscht wird. Dieses Schema wiederholt sich für Bits 5 bis 9 und Bits 10 bis 14.

Werte für die Wahl des Ausgangs zwischen 1 und 8 entsprechen den TTL-Ausgangsnummern 1 bis 8. Für die SPS-Ausgänge 1 bis 7 muss entsprechend 9 bis 15 gewählt werden. Wird der Ausgangswert auf Null gesetzt, so wird keine Aktion ausgeführt.

Somit ist für die Standard-Betriebsart „konstante Geschwindigkeit“  $f = 0$  zu setzen, wenn keine Ausgänge angesteuert werden sollen.

Der 8-Bit-Fehlercode  $E$  gibt an, ob und gegebenenfalls bei welcher der maximal 8 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 8 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 „ $\Delta t_1$ “ bis „ $\Delta t_5$ “ 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  $v = \text{const.}$  (Beispiel):

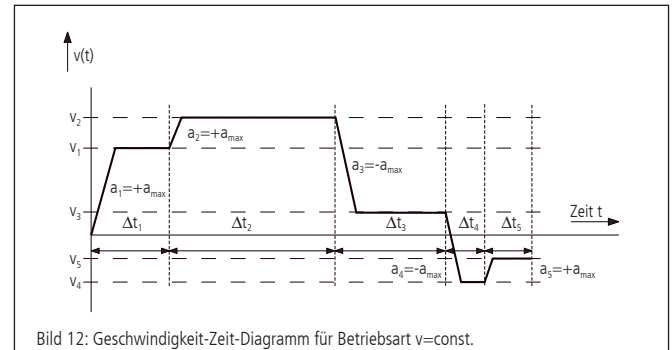


Bild 12: Geschwindigkeit-Zeit-Diagramm für Betriebsart  $v = \text{const.}$

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  $a = \text{const.}$  (Beispiel):

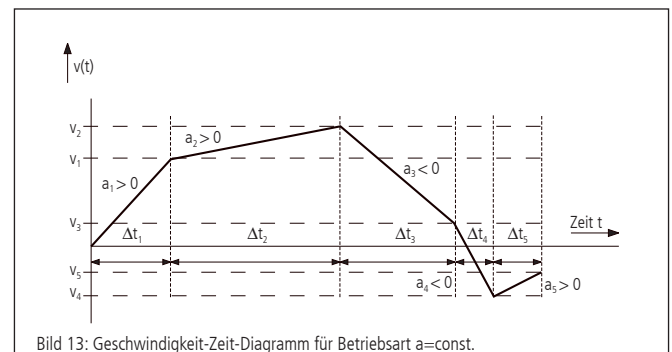


Bild 13: Geschwindigkeit-Zeit-Diagramm für Betriebsart  $a = \text{const.}$

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  $v = \text{const.}$ , 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> =  $\Delta x_{an}, \dots, \Delta x_{in}, \Delta t_n, f_n, e_n, t_n$

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:

$\Delta x_{an}, \dots, \Delta x_{in}, \Delta t_n, f_n, e_n, t_n, Vel_i, Acc_i$

#### 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  $\Delta t_0$  und der Freigabecode  $t_0$ :

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

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

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,-500,2000,0,0,0,0,0,0,98,32768,0,7

Plausibilitätskontrolle mittels

?PTABPLAUS0

und Auslesen des Tabellenelements über

?POSTAB0

ergibt als Antwort:

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

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 Maximalbeschleunigung auf Geschwindigkeit Null ab. Danach wird der Geschwindigkeitsmodus deaktiviert und die Achsen werden von Bahnsteuerungskontrolle auf Positionshaltung umgeschaltet.

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

#### Auswahl von Segmenten

Neben der Möglichkeit, über Freigabebits die gesamte Tabelle zu segmentieren, kann dies auch direkt über den Startbefehl PTABGO erfolgen. PTABGO<n> startet die Tabelle ab Zeile n. PTABGO<n><m> führt die Zeilen n bis m aus.

#### 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 Achsaufösungen zu kompensieren oder elliptische Konturen zu erzeugen.

#### Definitionen:

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

Startwinkel (Winkeloffset) des Kreissegments:  $\alpha$

Vom Kreissegment abzudeckender Winkelbereich:  $\Delta\alpha$

Radius des Kreissegments:  $r$

Veranschaulichung am Diagramm:



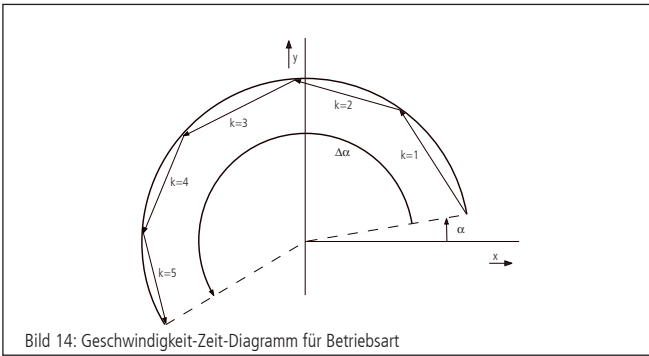


Bild 14: Geschwindigkeit-Zeit-Diagramm für Betriebsart

hier: Teilkreis mit Radius  $r$ , Winkeloffset  $\alpha = 10^\circ$ , Winkelbereich  $\Delta\alpha = +190^\circ$ ,  $m = 5$  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 Auflösungsgrade 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  $\langle n \rangle$  werden über den Befehl „PTABCIRCLE“ Kreisdaten in Form von  $\langle m \rangle$  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  $\langle n \rangle = \langle \text{Achsnummer } x \rangle, \langle \text{Achsnummer } y \rangle,$

$\Delta t_n, f_n, m_n, r_n, \alpha_n, \Delta\alpha_n, Z_n, N_n$

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  $v = \text{const.}$ , 5 Sekanten, Radius 1000 Inkremente, Startwinkel  $10^\circ$ , Winkelbereich  $190^\circ$  und Skalierung 1 mit Trapezprofil gefahren.

## 8.8 Automatisches Reagieren auf externe Auslöser und Setzen von Ausgängen

Die PS 90+ besitzt 16 digitale Eingänge und Ausgänge. Die TTL-Eingänge und Ausgänge werden im Folgenden mit 1 bis 8, die SPS-Eingänge und Ausgänge mit 9 bis 16 durchnummeriert.

### 8.8.1 Automatisches Reagieren auf Eingänge

Ein automatisches Reagieren auf externe Auslöser (Trigger-Funktion) bedeutet, dass vordefinierte Aktionen mit einem Zustandswechsel eines Eingangs verknüpft werden können. Sobald dieser Zustandswechsel eintritt, wird die Aktion selbständig durch die Steuerung ausgeführt.

#### Definition des Eingangs

Jeder Eingang kann den logischen Zustand 0 oder 1 annehmen. Jeder Zustandswechsel wird als Flanke bezeichnet. Ein Wechsel von 0 auf 1 ist eine steigende, ein Wechsel von 1 auf 0 ist eine fallende Flanke. Somit gibt es insgesamt 32 Auslöser (16 Eingänge mit je zwei möglichen Flanken).

#### Mögliche Aktionen

Es gibt insgesamt acht Aktionen, die ausgelöst werden können. Jeder Aktion ist eine Aktions-ID zugewiesen. Außerdem kann jede Aktion mit bis zu zwei Parametern konfiguriert werden:

Aktion	Aktions-ID	Parameter	Verhalten
Geschwindigkeitsmodus starten	1	Achsen-Nr. (1...9)	VGO
Geschwindigkeitsmodus stoppen	2	Achsen-Nr. (1...9)	VSTP
Positioniervorgang starten	3	Achsennummer (1...9)	PGO
Mehrere Achsen im Geschwindigkeitsmodus starten	4	Bitmaske Bit 0 = Achse 1 Bit 8 = Achse 9	MVGO
Geschwindigkeitsmodus mehrerer Achsen stoppen	5	Bitmaske Bit 0 = Achse 1 Bit 8 = Achse 9	MSTOP
Achse stoppen	6	Achsennummer (1...9)	STOP
Achse anschalten	7	Achsennummer (1...9)	MON
Achse ausschalten	8	Achsennummer (1...9)	MOFF

Die Aktion beinhaltet bei allen Fahrbefehlen lediglich das Starten der betroffenen Achsen. Alle zu diesem Zeitpunkt gesetzten Parameter, insbesondere auch eine relative Distanz oder absolute Position besitzen Gültigkeit.

Falls ein Befehl wegen eines unzulässigen Zustands der Achse nicht ausgeführt werden kann, wird die Ausführung ignoriert. Bei dem nächsten Auftreten des Auslösers wird die Ausführung erneut versucht.

## Zuweisung einer Aktion zu einem Auslöser

Es können Aktionen einem Auslöser zugewiesen und in einer Aktionstabelle gespeichert werden. Ein Tabelleneintrag beinhaltet die folgenden Informationen:

Eingang	Flanke	Aktions-ID	Parameter
1-16	0 (fallend) oder 1 (steigend)	1 bis 8	2 x 32-bit Signed decimal oder signed hex String mit Präfix 0x

### Verhalten

Wenn ein Auslöser auftritt, wird die PS 90+ die zugeordneten Aktionen in kürzest möglicher Zeit ausführen. Die Aktionstabelle wird im Hintergrund sequentiell abgearbeitet.

Die Tabelle besitzt eine fixe Größe, daher kann der Anwender alle Einträge bearbeiten, löschen, abfragen und die Ausführungsreihenfolge festlegen.

Die Ausführung beginnt immer in Zeile 1. Wenn eine Bedingung erfüllt ist, wird die Aktion sofort ausgeführt und die nächste Zeile bearbeitet. Es wird nicht auf das Ende einer Aktion einer Zeile gewartet.

Durch die Programmierung der Tabelle muss durch den Anwender sichergestellt sein, dass kein widersprüchliches oder ungewolltes Verhalten auftritt. Es findet keine Prüfung statt.

Unter diesen Voraussetzungen ist es möglich, den gleichen Auslöser für mehrere Aktionen zu benutzen, das heißt dass ein Auslöser mehrfach in der Tabelle verwendet werden darf. Genauso ist es möglich, gleiche Aktionen mehrfach zu verwenden.

Die Tabelle besteht aus maximal 64 Einträgen.

Zur Erläuterung sollen die folgenden Beispiele dienen:

Starte und stoppe Achse 1 mit Eingang 14. Starte und stoppe Achse 2 mit Eingang 0. Starte Achse 3, wenn Achse 2 über Eingang 0 gestoppt wird.

<b>Hinweis:</b>	Es ist nicht definiert, ob erst Achse 2 stoppt oder Achse 3 startet. Beide Aktionen werden so schnell wie möglich gestartet.
-----------------	--

Zeile	Eingang	Flanke	Aktions-ID	Parameter
1	14	1	1	1
2	14	0	2	1
3	0	1	1	2
4	0	0	2	3
5	0	0	1	3

Stoppe Achse 1 über Eingang 14 und starte Achse 1 über Eingang 3 und 12.

Zeile	Eingang	Flanke	Aktions-ID	Parameter
1	14	0	1	1
2	2	1	2	1
3	12	1	1	1

## 8.8.2 Automatisches Setzen von Ausgängen

Ein automatisches Setzen von Ausgängen (Event-Funktion) bedeutet, dass Ausgänge auf einen vordefinierten Zustand gesetzt werden, wenn bestimmte interne Ereignisse eintreten. Interne Ereignisse beziehen sich auf Achsenzustände.

## Definition der Ereignisse

Es gibt sechs interne Ereignisse, die benutzt werden können. Jedem sind eine Ereignis-ID und gegebenenfalls mehrere Parameter zugeordnet.

Internes Ereignis	Ereignis-ID	Parameter	Anmerkung
Geschwindigkeit einer Achse ist größer als ein bestimmter Wert	1	Achsennummer (1...9) Geschwindigkeitswert: signed decimal	Aktuelle Geschwindigkeit > Wert
Geschwindigkeit einer Achse ist kleiner als ein bestimmter Wert	2	Achsennummer (1...9) Geschwindigkeitswert: signed decimal	Aktuelle Geschwindigkeit < Wert
Position einer Achse ist größer als ein bestimmter Wert	3	Achsennummer (1...9) Positionswert: signed decimal	Aktuelle Position > Wert
Position einer Achse ist kleiner als ein bestimmter Wert	4	Achsennummer (1...9) Positionswert: signed	Aktuelle Position < Wert
Achse wechselt in einen Bewegungszustand	5	Achsennummer (1...9)	Basierend auf Achsenzustand
Achse wechselt in einen ruhenden Zustand	6	Achsennummer (1...9)	Basierend auf Achsenzustand

Für die beiden letzten Einträge gilt:

Die Achsenzustände (siehe ?ASTAT) T, S, V, P, F, W, X, Y, C und N gelten als Bewegungszustand. Ein Wechsel von einem anderen in einen dieser Zustände wird als Wechsel in einen Bewegungszustand interpretiert. Ein Wechsel aus einem dieser Zustände in einen nicht aufgeführten wird als Wechsel in einen ruhenden Zustand interpretiert.

### Zuordnung zu Ausgängen

Die Zuordnung zu Ausgängen erfolgt über eine Ereignistabelle. Ein Tabelleneintrag beinhaltet die folgenden Daten:

Ereignis-ID	2 Parameter	Ausgangsmaske	Ausgangswert
1-8	2 * 32 bit	16 bit	16 bit

Ein Ereignis beeinflusst nur die Ausgänge, die in der Ausgangsmaske mit 1 gewählt sind. Die Ausgänge der so ausgewählten Ausgänge werden auf die zugehörigen Ausgangswerte gesetzt.

Die Tabelle kann bis zu 64 Einträge enthalten.

### Verhalten

Die Ausgänge sind nicht synchronisiert. Es gibt auch kein Clock-Signal. Dies heißt, dass bei einer gleichzeitigen Änderung mehrerer Ausgänge Übergangszustände entstehen können.

Die Liste wird zyklisch abgearbeitet. Falls ein Ereignis eintritt, werden die entsprechenden Ausgänge gesetzt. Es kann keine Aussage über die exakte Ausführungsreihenfolge getroffen werden.

Eine Ereignis-ID kann mehrfach verwendet werden. Das Verhalten soll an zwei Beispielen veranschaulicht werden:

Achse 2 startet und beschleunigt bis zu einer Geschwindigkeit von 20000 und wird dann nach einiger Zeit wieder gestoppt. Sobald die Geschwindigkeit 10000 überschritten ist, soll Ausgang 2 auf 0 gesetzt werden. Falls die Geschwindigkeit von einem höheren Wert als 9000 diesen unterschreitet, soll der Ausgang 2 auf 1 gesetzt werden. Der Ausgang ist zu Beginn der Sequenz nicht definiert.

Ereignis-ID	2 Parameter	Ausgangsmaske	Ausgangswert
1	Achse 2 10000	0x0002	0x0000
2	Achse 2 9000	0x0002	0x0002

Wenn Achse 1 in einen Bewegungszustand wechselt, soll Ausgang 15 auf 0 gesetzt werden. Wenn Achse 3 in einen Ruhezustand wechselt, soll Ausgang 15 auf 1 gesetzt werden. Wenn Achse 5 in einen Bewegungszustand wechselt, soll Ausgang 3 auf 1 gesetzt werden.

Ereignis-ID	2 Parameter	Ausgangsmaske	Ausgangswert
5	Achse 1	0x8000	0x0000
6	Achse 3	0x8000	0x0001
5	Achse 5	0x0004	0x0004

In diesem Beispiel wirken sowohl die erste als auch die dritte Zeile auf Ausgang 15. Falls beide Ereignisse gleichzeitig auftreten ist nicht definiert, welches zuerst bearbeitet wird. Somit ist auch der Wert des Ausgangs nicht eindeutig. Diese Konflikte können nur durch den Benutzer erkannt und verhindert werden.

### 8.8.3 Konfiguration

Jede Tabelle beinhaltet bis zu 64 Einträge.

Folgende Funktionen stehen über die Kommandoschnittstelle zur Verfügung:

- Die komplette Funktionalität „Reagieren auf externe Auslöser und Setzen von Ausgängen“ kann über einen Befehl aktiviert oder deaktiviert werden. Da diese Funktionalität permanent Rechenleistung beansprucht, ist es ratsam, Sie nur bei tatsächlicher Benutzung zu aktivieren.
- Der Zustand aktiviert oder deaktiviert kann abgefragt werden.
- Parameter können in die beiden Tabellen geladen werden.
- Einzelne Zeilen der Tabellen können ausgelesen werden.
- Einzelne Zeilen der Tabelle können gelöscht werden.
- Befehle mit Zeilennummern größer als der zulässige Bereich werden ignoriert.
- Falls beim Eintragen von Zeilen in die Aktionstabelle ein Fehler auftritt, weil zum Beispiel ein unzulässiger Parameter gesetzt wird, so wird die Aktions-ID auf 0 gesetzt.
- Leere Zeilen in der Aktionstabelle sind durch die ID 0 gekennzeichnet.
- Falls beim Eintragen von Zeilen in die Ereignistabelle ein Fehler auftritt, weil zum Beispiel ein unzulässiger Parameter gesetzt wird, so wird die Ereignis-ID auf 0 gesetzt.
- Leere Zeilen in der Ereignistabelle sind durch die ID 0 gekennzeichnet.

## 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 BLDC- oder 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 (sog. Quadratur-Signale), und ggf. ein Index-Impuls pro Umdrehung. Die PS 90+ 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 (ebenfalls mit einer Auflösung von 32 Bit) 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.

Die Signale des Wegmesssystems entsprechen den vorher genannten Encodersignalen (Quadratur A und B, sowie Index). Die maximale Zählfrequenz beträgt 5 MHz (Signal) bzw. 20 MHz (Quadratur).

### 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 Absolutposition des Schlittens unter Zuhilfenahme einer eindeutigen Referenzmarke erfasst. Die aus Motor und Antriebs-spindel 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 Relativedistanz 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 Auflösungsweite ergibt.

Ein Positioniervorgang mit Nachlaufregelung entspricht folgendem 3-Phasen-Schema:

- Mittels des gegebenen Umrechnungsfaktors ( $Z/N$ ) wird aus den gegebenen Positionsdaten die zu verfahrenende Relativedistanz 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 Relativedistanz 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 Relativedistanz 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:

$h = 5 \text{ mm}$ ,  
 $n = 200$ ,  
 $m = 50$

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

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 90+ benutzte Servofilter arbeitet nach einem PID-Algorithmus. Ein Integrationslimit sichert nach oben gegen einen akkumulierten Fehler ab.

Die PID-Formel lautet wie folgt:

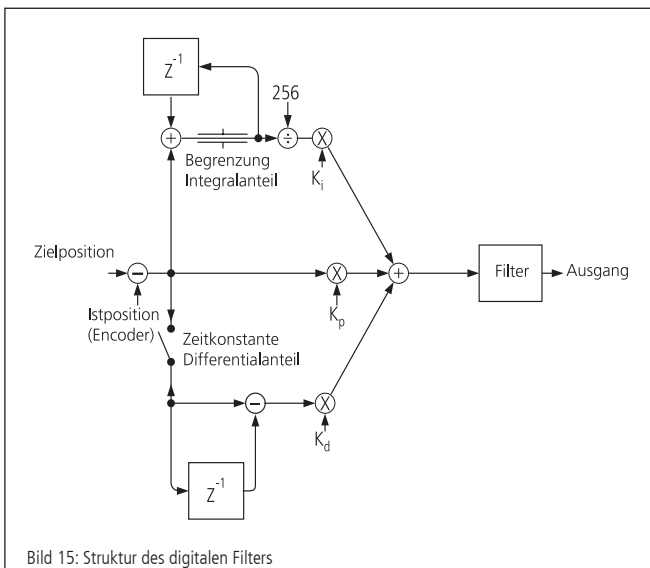
$$\text{Output}_n = K_p E_n + K_d (E_n - E_{(n-1)}) + \sum_{j=0}^n E_j \cdot \frac{K_i}{256}$$

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)



## 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 welcher dieser noch aus dem Stillstand ohne Beschleunigungsphase loslaufen kann. Es ist üblich, diese und andere Kennfrequenzen von Schrittmotoren in Hertz Vollschritt („HzVS“), d.h. Vollschritte pro Sekunde, anzugeben. Die Welle eines Schrittmotors mit Schrittwinkel  $1,8^\circ$ , 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 90+ 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.

$$T_p = 256 \mu\text{s}$$

#### 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:

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

bzw.

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

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

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

bzw.

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

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:

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

Zurückgelegte Distanz während der Trapezprofil-Beschleunigungsrampe:

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

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

### Allgemeines

Die PS 90+ hat einen digitalen Lage-/Geschwindigkeits-Regler. Stell- und Regelgröße werden periodisch berechnet. Die Erfassung des Positions-Istwertes geschieht im einfachsten Fall mittels eines Drehgebers (auch „Encoder“ genannt), 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 4-fach 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 51,2  $\mu\text{s}$ . Sie kann bei Bedarf um ganzzahlige Vielfache von 51,2  $\mu\text{s}$  erhöht werden:

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

entsprechend einer Abtastzeit von

$$T_s = [51,2 \mu\text{s}, 102,4 \mu\text{s}, 153,6 \mu\text{s}, 204,8 \mu\text{s}, 256 \mu\text{s}, \dots, 19986 \mu\text{s}]$$

Als Abtastzeit können nur ganzzahlige Werte an die PS 90+ übergeben werden. Der Wert wird intern auf den nächsten gültigen Wert gerundet.

Standardwert (Voreinstellung):  $T_s = 256 \mu\text{s}$

### 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.

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

bzw.

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

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  $n = 1800 \text{ U/min}$  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  $n = 1800 \text{ U/min}$  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.

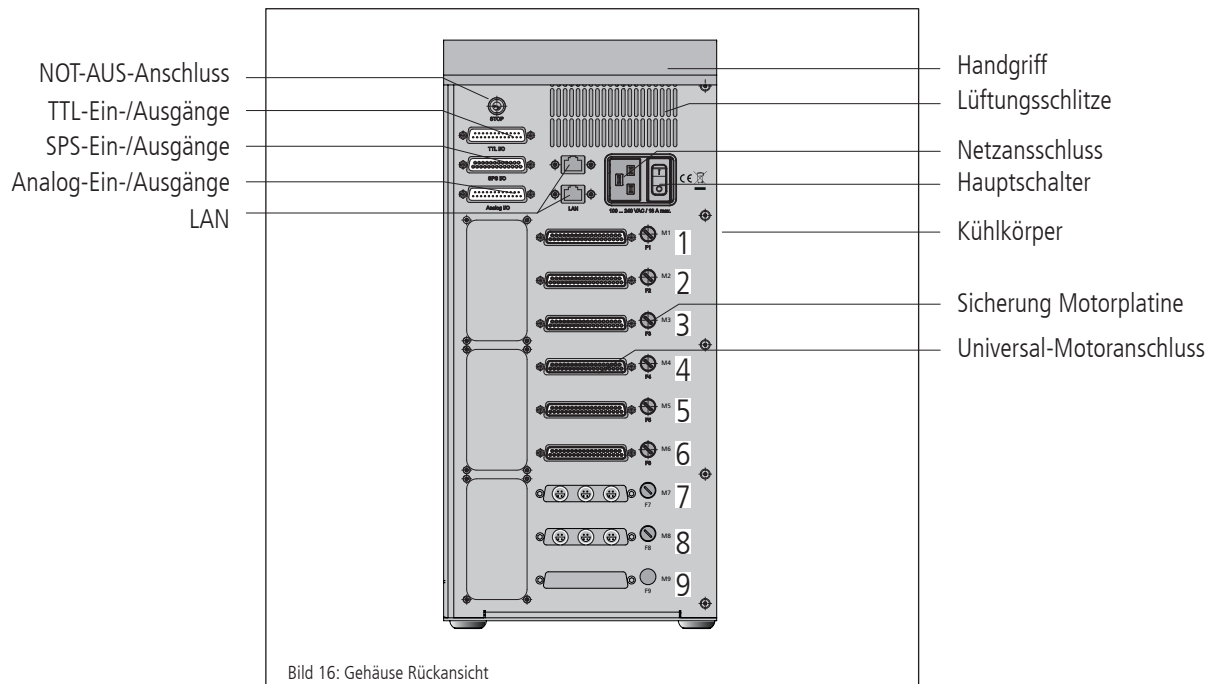
Dauer der Trapezprofil-Beschleunigungsrampe bei gegebener Geschwindigkeit V und Beschleunigung ACC:

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

Zurückgelegte Distanz während der Trapezprofil-Beschleunigungsrampe:

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

## 12. Nano-Hybrid-Ansteuerung



### Allgemeines

Um OWIS® Positioniereinheiten mit Nano-Hybrid-Technik ansteuern zu können, ist eine entsprechend ausgestattete PS 90 notwendig. In diesem Kapitel werden die Unterschiede und Besonderheiten dieser Ausstattungsvariante sowie die Betriebsmodi erklärt. Alle allgemeinen Eigenschaften, insbesondere die Bedien- und Sicherheitshinweise gelten uneingeschränkt. Vor der Benutzung der PS 90 zur Ansteuerung von OWIS® Nano-Hybrid-Positioniereinheiten ist die Kenntnis der vorherigen Kapitel notwendig.

### Technische Übersicht und Aufbau der Steuerung

Nano-Hybrid Positioniereinheiten von OWIS® besitzen einen hybriden Antrieb. Die Grobpositionierung erfolgt über einen hochauflösenden Schrittmotor. Die Feinpositionierung erfolgt durch einen Piezoaktor.

Eine PS 90, die zur Ansteuerung dieser Nano-Hybrid-Positioniereinheiten ausgerüstet ist, kann nur noch für sechs ausgerüstet sein. Statt der Achsen 7-9 ist die Ansteuerelektronik für die Piezoaktoren vorhanden.

Die Positioniereinheiten werden über den Universal-Motorstecker angeschlossen. Zusätzlich wird der Piezozweig über ein weiteres Kabel verbunden. Die Anschlüsse dafür befinden sich für die ersten drei Nano-Hybrid-Positioniereinheiten auf Position 7. Falls vier bis sechs solcher Einheiten angesteuert werden sollen, ist auch Position 8 belegt. Position 9 ist immer unbelegt.

### Sicherheit

Die Ansteuerung der Piezozweige erfolgt mit einer Spannung im Bereich zwischen -71 V und +71 V und kann schwerste Verletzungen verursachen. Die Benutzung darf nur durch Personal erfolgen, welches im Umgang mit solchen Spannungen unterwiesen ist. Die allgemeinen Unfallverhütungsvorschriften müssen beachtet werden.

### Steuerungsarchitektur und Funktion

Eine Nano-Hybrid-PS 90 besteht im Wesentlichen aus folgenden Komponenten:

1. Ein eingebautes Netzteil
2. Eine Hauptplatine
3. Maximal zwei Antriebsplatinen
4. Maximal sechs Motorplatinen (Endstufen)
5. Maximal zwei Wegmessplatinen
6. Eine D/A-Wandlerkarte für die Piezoaktoren
7. Maximal zwei Steuermodule für die Piezoaktoren

Anstelle der dritten Antriebsplatine wird eine D/A-Wandlerkarte für die Piezoaktoren eingesetzt. Diese kann drei oder sechs Achsen ansteuern. Besteht nur der erste Antriebsverband aus Nano-Hybrid-Positioniereinheiten, so wird zusätzlich ein Steuermodul für Piezoaktoren benötigt. Sind beide Antriebsverbände in Nano-Hybrid-Technik, so werden davon zwei benötigt. Das Steuermodul für den Piezozweig besitzt eine Schutzeinrichtung, die prüft, ob auch eine entsprechende OWIS®-Nano-Hybrid-Einheit angeschlossen ist. Die grundsätzliche Ansteuerung der Schrittmotoren bleibt unverändert.

### Anschluss

Der Anschluss erfolgt über zwei Kabel. Zum einen wird ein Motor-kabel an den 37-poligen D-Sub-Stecker der Steuerung angeschlossen, welches am Tisch an der 18-poligen Lemo-Buchse befestigt wird. Zum anderen wird das Piezo-Kabel sowohl am Tisch als auch an der Steuerung am 4-poligen Lemo-Anschluss befestigt.

### Positionierung im Nano-Hybrid-Betrieb

Zur Positionierung mit einer Nano-Hybrid-Positioniereinheit stehen drei Möglichkeiten zur Verfügung. Normale Positionierungen können wie mit jeder Einheit mit Schrittmotor erfolgen. Da jede Nano-Hybrid-Positioniereinheit mit einem integrierten Messsystem ausgestattet ist, kann auch die Nachlaufregelung (Modi 0 bis 5) als Betriebsart gewählt werden.

Um die Möglichkeiten der hochgenauen Positionierung mit Hilfe des Piezo-Antriebs nutzen zu können, sind spezielle Modi (6 bis 9) der Nachlaufregelung vorhanden, die im Folgenden erklärt werden:

### Allgemeine Beschreibung der Nachlaufregelung für Piezo-Antriebe

Bei der Nachlaufregelung erfolgt die Positionierung in mehreren, aufeinander folgenden Schritten. Der erste Schritt ist dabei immer eine Grobpositionierung über den Motor. Anschließend erfolgt eine Korrektur des Positionierfehlers durch den Schrittmotor und dann durch den Piezo-Antrieb. Folgende Werte sind für die Positionierung mit Hilfe des Piezo-Antriebs relevant:

#### - PWMSSET

Dieser Wert gibt die Sollposition für den zu startenden Positionierungsvorgang an. Im absoluten Positioniermodus wird der Wert als Absolutposition betrachtet. Im relativen Positioniermodus entscheidet das Vorzeichen die Fahrtrichtung.

#### - PWMSPWIN

Dieser Wert beschreibt ein zulässiges Zielfenster, welches vor und hinter der Zielposition PWMSSET liegt. Er wird in Inkrementen des Messsystems angegeben. Die Positionierung wird erfolgreich beendet, sobald die Istposition zwischen  $PWMSSET - PWMSPWIN$  und  $PWMSSET + PWMSPWIN$  liegt. PWMSPWIN muss immer positiv sein und sollte im Bereich zwischen 2 und 10 Inkremente liegen. Dieser Wert kann die Geschwindigkeit, mit der eine Positionierung erfolgt, beeinflussen. Wird er zu groß gewählt, so ist die Positionier- und Wiederholgenauigkeit zu grob. Wird er zu klein gewählt, so kann die Positionierung möglicherweise nicht zu einer stabilen Endposition führen.

#### - WMSOFFS

Der Schrittmotor muss den Schieber mit einem Versatz zur Sollposition positionieren, damit die Sollposition durch den Piezo-Antrieb angefahren werden kann. Dieser Versatz wird durch WMSOFFS in Messsysteminkrementen angegeben. WMSOFFS muss immer negativ sein und sollte im Bereich zwischen -20 und -100 liegen. Wird der Wert unpassend gewählt, so muss möglicherweise eine zusätzliche Korrektur durch den Motor erfolgen und der Positionierungsvorgang dauert länger.

#### - PWMSWIN

Dieser Wert beschreibt ein zulässiges Zielfenster, welches vor und hinter dem Ziel der Positionierung mit dem Schrittmotor liegt. Er wird in Inkrementen des Messsystems angegeben. Die Positionierung mit dem Schrittmotor wird erfolgreich beendet, wenn zu diesem Zeitpunkt die Istposition in einem Bereich zwischen  $(PWMSSET + WMSOFFS) - PWMSWIN$  und  $(PWMSSET + WMSOFFS) + PWMSWIN$  liegt. PWMSWIN muss immer positiv sein und sollte im Bereich zwischen 10 und 50 Inkrementen liegen. Dieser Wert kann die Dauer der Positionierung beeinflussen. Ist er ungünstig gewählt, so muss gegebenenfalls eine zusätzliche Korrektur durch den Schrittmotor oder den Piezo-Antrieb erfolgen.

#### - WMSVEL

Dieser Wert legt die Geschwindigkeit fest, mit der in den Modi 7 und 9 die Korrekturfahrt in Phase 2 durchgeführt wird. WMSVEL muss immer positiv sein.

Es wird dabei je nach Modus das folgende Schema abgearbeitet:

- Mittels des gegebenen Umrechnungsfaktors (Z/N) wird aus den gegebenen Positionsdaten die zu verfahrenende Relativedistanz des Aktors berechnet.
- Die so berechnete Distanz wird verfahren (Phase 1) und die Abweichung zur Sollposition berechnet.
- Liegt die Istposition außerhalb des Fensters  $(PWMSSET + WMSOFFS) \pm PWMSWIN$ , so wird eine Korrekturphase (Phase 2) gestartet. Liegt die Istposition innerhalb des Fensters, so wird die Positionierung mit dem Piezo-Antrieb gestartet (Phase 3).
- Falls Phase 2 notwendig ist, so hängt die Art und Weise der Korrektur vom gewählten Modus ab. In den Modi 6 und 7 wird eine erneute Grobpositionierung (Phase 1) durchgeführt. Das heißt, dass von der aktuellen Position ausgehend eine neue Zielposition berechnet wird, die anschließend vom Schrittmotor mit den eingestellten Werten durchgeführt wird. Phase 2 wird beendet, falls am Ende die Istposition innerhalb des Fensters liegt oder falls sich der Fehler im Vergleich zum vorherigen Schritt vergrößert hat. In den Modi 8 und 9 wird die Korrektur durch eine Bewegung im Geschwindigkeitsmodus durchgeführt. Die Geschwindigkeitsfahrt wird beendet, sobald das Fenster erreicht ist. Sollte das Fenster überfahren werden, so wird eine erneute Fahrt im Geschwindigkeitsmodus in die andere Richtung gestartet.
- Nachdem die Positionierung mit dem Schrittmotor abgeschlossen ist, wird die Feinpositionierung mit dem Piezo-Antrieb gestartet. Die Positionierung endet, sobald die Istposition im Bereich  $PWMSSET \pm PWMSPWIN$  liegt. Sollte der Piezo-Antrieb das Zielfenster nicht erreichen, wird in Phase 2 gesprungen und eine erneute Korrektur mit dem Schrittmotor gestartet.
- In den Modi 7 und 9 bleibt die Phase 3 auch nach erfolgreicher Positionierung aktiv. Das heißt, dass Änderungen der Istposition zum Beispiel aufgrund von externen Kräften auf den Schieber fortlaufend durch den Piezo-Antrieb korrigiert werden. Sollte der Piezo-Antrieb alleine nicht in der Lage sein, das Zielfenster zu erreichen, wird automatisch Phase 2 gestartet.



## 13. Inbetriebnahme der PS 90+

### 13.1 Vorbereitung der Steuerung

#### Aufstellung

Die Steuerung ist für den Einsatz in Forschung, Entwicklung sowie für industrielle Anwendungen konzipiert. Sie darf nur in trockener, staubarmer Umgebung betrieben werden. Grundsätzlich wird sie freistehend betrieben.

Zur internen Kühlung sind auf der Gehäusevorder- und Rückseite im oberen Bereich Lüftungsschlitze angebracht. Die Abwärme der Motorplatinen (Endstufen) wird über den seitlich angebrachten Kühlkörper an die Außenluft abgegeben. Die Steuerung darf nicht in ein Gehäuse oder einen Schrank ohne ausreichende Luftzirkulation eingebaut werden.

#### Hinweis:

- Wärmestau in der Steuerung oder am Kühlkörper ist zu vermeiden. Es soll ein Mindestabstand von 15 cm zu geschlossenen Flächen und Wänden eingehalten werden.

#### NOT-AUS-Funktion

An der Geräterückseite ist ein Anschluss für einen externen NOT-AUS-Taster vorgesehen, an welchem standardmäßig ein Kurzschlussstecker eingesteckt ist. Soll ein NOT-AUS-Taster angeschlossen werden, ist der Kurzschlussstecker zu entfernen.

#### Hinweis:

- Wird der Kurzschlussstecker entfernt und kein NOT-AUS-Taster angeschlossen, ist die Funktion der Motorplatinen (Motorendstufen) blockiert.

### 13.2 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.

Zunächst sind die Positioniereinheiten an die entsprechenden Achsen anzuschließen (siehe Abnahmeprotokoll). Es müssen:

- die Positioniereinheit
- die Stromversorgung
- der Computer

angeschlossen werden.

Die Verbindung zum Computer erfolgt über die USB, RS-232- oder Ethernet- Schnittstelle.

Mit dem optionalen Anybus<sup>®</sup>-Modul „Modbus/TCP“ ist die Kommunikation mit einem PC über Ethernet möglich.

Für USB-Schnittstelle 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 initialisiert und somit erkannt werden.

### 13.3 Systemstart

Durch das Betätigen des Hauptschalters wird die Steuerung aktiviert. Der Mikrocontroller startet alle vorhandenen Programme und Parameter und initialisiert sich und seine Peripherie. Der Initialisierungsvorgang dauert ca. 10 Sekunden. Danach ist die Steuerung bereit, Kommandos

vom PC zu empfangen und zu bearbeiten. Beim ersten Windows-Start mit angeschlossener PS 90+ 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

Für die Inbetriebnahme gehören zum Lieferumfang der Steuerung das Softwaretool OWISoft, der USB-Treiber und die Software-Schnittstelle (SDK/API) für C, C++, C#, LabView (ab V 8.2) und zusätzliche Programmiersprachen (32/64-Bit). Damit kann die PS 90+ 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 dem 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.

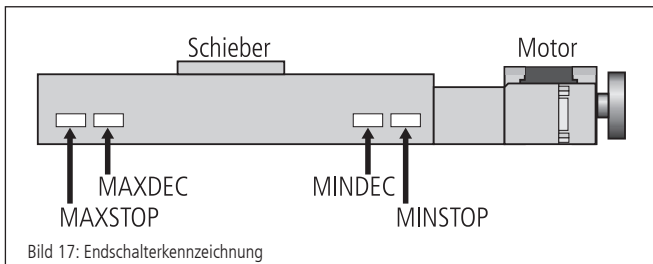
Für die Inbetriebnahme mittels eigener Applikationssoftware lesen Sie bitte das Kapitel „Hinweise zum Aufbau einer eigenen Applikationssoftware“. Dort ist im Anschluss auch eine Tabelle mit den Befehlssätzen der PS 90+ angefügt.

# 14. Fehlerüberwachung

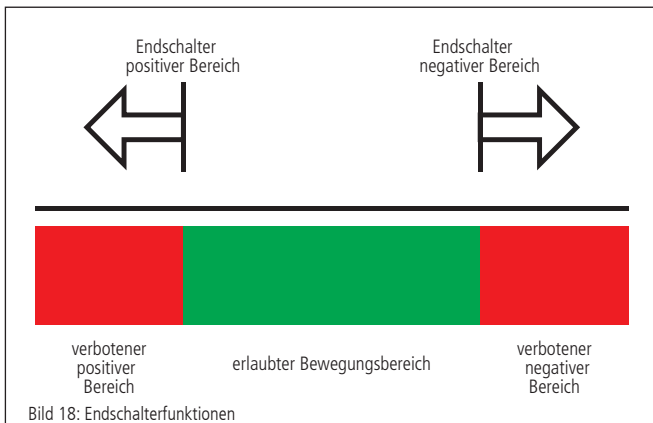
## 14.1 Endschalter

Die PS 90+ besitzt vier Endschaltereingänge, davon jeweils zwei Eingänge für Limit-Schalter (MINSTOP, MAXSTOP) und zwei für Bremschalter (MINDEC, MAXDEC), sowie Auswertemöglichkeit für einen Referenzschalter je Achse.

Einer der vier Schalter ist als Referenzschalter definiert.



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.



### 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): Falls 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. MINDEC:** Dieser Endschalter löst bei Betätigung während negativer Fahrt eine Bremsrampe mit programmierbarer Verzögerung aus. Der Motor wird nach ausgeführter Bremsrampe nicht abgeschaltet, sondern bleibt weiterhin aktiv. Falls der Nachlaufweg der Bremsrampe zu groß gewesen sein sollte, und die Positioniereinheit anschließend den MINSTOP-Endschalter erreicht, siehe 1.
- 3. MAXDEC:** Die Reaktion ist äquivalent zum MINDEC-Endschalter, jedoch wirkt dieser Endschalter nur bei Fahrt in positiver Richtung.

- 4. MAXSTOP:** Die Reaktion ist äquivalent zum MINSTOP-Endschalter, jedoch wirkt dieser Endschalter 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.

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. Schließkontakt 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 5V-CMOS-Pegel, wobei Open-Collector-NPN- oder Push-Pull-Ausgänge gleichermaßen angeschlossen werden können, da hochohmige Pullup-Widerstände (4,7 kOhm) nach +5V bereits geräteintern vorgesehen sind. Die Endschaltereingänge sind tolerant gegen eine Fremdspannung von bis zu +24V.

### 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>

## 14.2 Endstufen-Fehlerüberwachung

Jedes Endstufen-Feld 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.

## 14.3 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.

## 14.4 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.25), 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.

## 15. Joystick

Zusätzlich zum Handterminal besteht die Möglichkeit, einen Joystick an die Steuerung anzuschließen, der als Zubehör erhältlich ist. Mit ihm können maximal drei Achsen manuell verfahren werden. Der XYZ-Joystick wird an den Analogeingänge (1, 2, 3) der PS 90+ angeschlossen.



Bild 19: Joystick

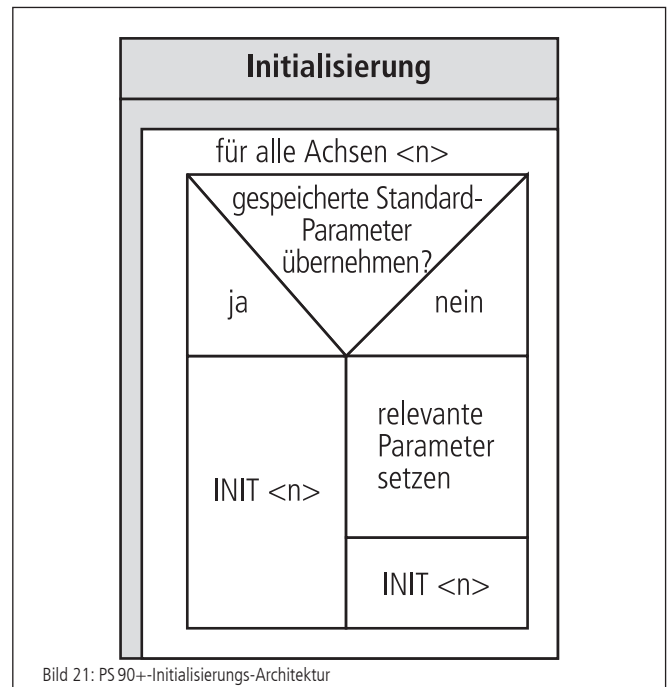


Bild 21: PS 90+-Initialisierungs-Architektur

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.

## 16. Hinweise zum Aufbau einer eigenen Applikationssoftware

Eine PS 90+-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.

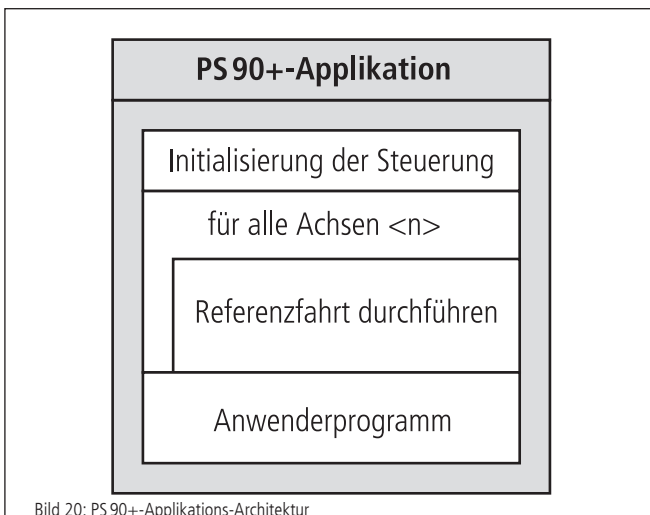


Bild 20: PS 90+-Applikations-Architektur

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.

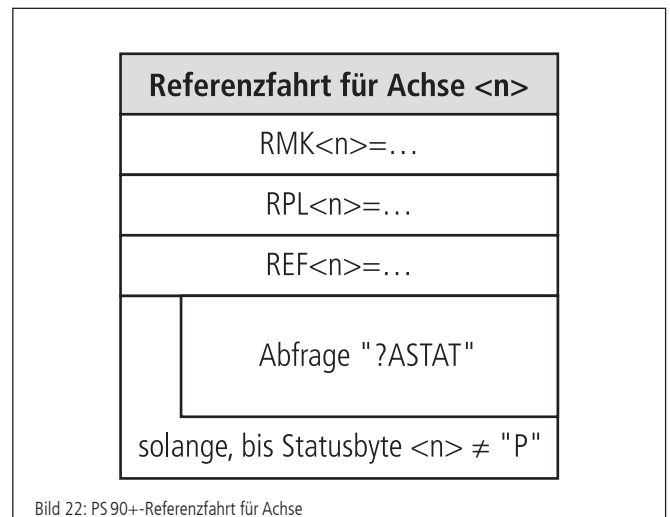


Bild 22: PS 90+-Referenzfahrt für Achse

Zwischen zwei einzelnen Befehlen, die zur PS 90+ 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 Laufzeit-ableich ebenfalls implementiert ist.

## 17. Befehlssatz der PS 90+

Generelles zum Format der Befehle:

Jeder Befehl wird über die Schnittstelle (RS-232, USB oder Ethernet) 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 Befehl, 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...9 (bzw. höchste Achsennummer)
- <uv> = Zahlenwert ohne Vorzeichen
- <sv> = Zahlenwert mit Vorzeichen
- <v> = vorzeichenbehaftete Wegangabe

# Anhang

## I Befehlstabelle

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Allgemeine Statusabfragen	?ASTAT	<p>Statusabfrage der Achsen, pro Achse wird ein Zeichen zurückgeschickt, das den aktuellen Zustand der Achse beschreibt:</p> <p>„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            „N“ = Achse arbeitet im Piezo-Nachführmodus mit WMS            „?“ = Fehler, unbekannter Achsenstatus</p>	?ASTAT	IIOURTTJV
	?MSG	<p>Liest den Message-Ausgangs-Buffer aus, der Message-Ausgangs-Buffer wird nur für Fehlermeldungen, die die Kommando-Schnittstelle betreffen (falscher Befehl, fehlende Parameter, ungültiger Wert) verwendet. Folgende Meldungen sind möglich:</p> <p>„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 9)            „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 noch nicht leer ist)            „07 AXIS IS IN WRONG STATE“ (wird in den Meldungspuffer geschrieben, wenn ein Fahr- oder Konfigurationsbefehl gesendet wurde, der nicht ausgeführt werden konnte, da sich die Achse momentan in einem anderen Fahrzustand befindet)            „08 AXIS NOT RELEASED“ (wird ausgegeben, wenn eine nicht freigegebene Achse initialisiert wird)            „09 ERROR IN POSITION TABLE“ (wird ausgegeben, wenn es ein Problem in der Tabelle für Bahnsteuerung gibt)            „10 MPUNI CAN ERROR“ (wird in den Meldungspuffer geschrieben, wenn es einen Fehler in der internen Kommunikation gibt)</p>	?MSG	00 NO MESSAGE...

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Allgemeine Statusabfragen	?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	
	?ESTAT<n>	Auslesen des aktuellen logischen Zustandes der Endschalter und Endstufenrückmeldung einer Achse. Bit 0 = MINSTOP, Bit 1 = MINDEC, Bit 2 = MAXDEC, Bit 3 = MAXSTOP, Bit 4 = Rückmeldung der Endstufe.	?ESTAT	10101
	?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 = AxisIn-Pin, Bit 7 = Hall A, Bit 8 = Hall B, Bit 9 = Hall C, Bit 10 = AxisOut-Pin, Bit 11-15 = reserviert.	?AXSIGNALS1	0000011101101001
	?MPUNISTAT<n>	Statusinformationen über eine Achse abfragen: 1. Achsennummer 2. Fehlercode 3. Typ 4. AD-Digits 5. Version 6. Signale		
	?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	90
Basis-Konfiguration	AXIS<n>=<uv>	Eine Achse freigeben bzw. sperren. Mit diesem Befehl kann eine Achse freigegeben (1) oder gesperrt (0) werden.	AXIS5=1	
	?AXIS<n>	Freigabestatus einer Achse auslesen. Ist die Achse freigegeben, so wird eine 1 angezeigt, ansonsten eine 0.	?AXIS5	1
	MOTYPE<n>=<uv>	0 = DC-Brush 2 = Schrittmotor Open Loop 3 = Schrittmotor Closed-Loop 4 = BLDC	MOTYPE1=0	
	?MOTYPE<n>	Motortyp für eine Achse auslesen.	?MOTYPE1	0
	AMPSHNT<n>=<uv>	Strombereich 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: 9600, 19200, 38400, 57600, 115200. 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	PS90-V8.0-xxxxxxx	

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Basis-Konfiguration	?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	
	?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
	IPADDR=<uv>	Feste IPv4-Adresse setzen als Dezimalzahl Aktiv nach globalem Speichern und Neustart	IPADDR= 168428041 (für 10.10.2.9) IPADDR= 3232235628 (für 192.168.0.108)	
	IPADDR1=<uv>	Feste IPv4-Adresse setzen. Notation mit Punkten Aktiv nach globalem Speichern und Neustart	IPADDR1= 10.10.2.9 IPADDR1= 192.168.0.108	
	?IPADDR	IPv4-Adresse abfragen als Dezimalzahl. Gibt die zuletzt gespeicherte Adresse zurück.	?IPADDR	168428041
	?IPADDR1	IPv4-Adresse abfragen. Notation mit Punkten. Gibt die zuletzt gespeicherte Adresse zurück.	?IPADDR1	10.10.2.9
	IPGWADDR=<uv>	Feste IPv4-Gateway-Adresse setzen als Dezimalzahl Aktiv nach globalem Speichern und Neustart	IPGWADDR = 168248287 (für 10.10.2.255)	
	IPGWADDR1=<uv>	Feste IPv4-Gateway-Adresse setzen. Notation mit Punkten Aktiv nach globalem Speichern und Neustart	IPGWADDR1= 10.10.2.255	
	?IPGWADDR	IPv4-Gateway-Adresse abfragen als Dezimalzahl Gibt die zuletzt gespeicherte Adresse zurück.	?IPGWADDR	168248287
	?IPGWADDR1	IPv4-Gateway-Adresse abfragen. Notation mit Punkten Gibt die zuletzt gespeicherte Adresse zurück.	?IPGWADDR1	10.10.2.255
	IPNETMASK=<uv>	Feste IPv4-Netzmaske-Adresse setzen als Dezimalzahl Aktiv nach globalem Speichern und Neustart.	IPGWADDR = 168428032 (für 10.10.2.0)	
	IPNETMASK1=<uv>	Feste IPv4-Netzmaske-Adresse setzen. Notation mit Punkten Aktiv nach globalem Speichern und Neustart.	IPGWADDR1= 10.10.2.0	
	?IPNETMASK	IPv4-Netzmaske abfragen als Dezimalzahl Gibt die zuletzt gespeicherte Adresse zurück.	?IPGWADDR	168428032
	?IPNETMASK1	IPv4-Netzmaske abfragen. Notation mit Punkten Gibt die zuletzt gespeicherte Adresse zurück.	?IPGWADDR1	10.10.2.0
DHCP=<uv>	DHCP aktivieren (=1) oder deaktivieren (=0) Aktiv nach globalem Speichern und Neustart.	DHCP=0 DHCP=1		
?DHCP	Status des DHCP abfragen. Gibt den zuletzt gesetzten Status zurück.	?DHCP	0	
?IPADDRACTIVE	Gibt die aktuell verwendete IPv4-Adresse als Dezimalzahl zurück.	?IPADDRACTIVE	168428041	
?IPADDRACTIVE1	Gibt die aktuell verwendete IPv4-Adresse zurück. Notation mit Punkten.	?IPADDRACTIVE1	10.10.2.9	
?MACADDR	Gibt die MAC-Adresse der Ethernet-Schnittstelle zurück.	?MACADDR	ab:bc:cd:01.02.03	

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
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. <b>Dieser Befehl muss nach dem Einschalten der Steuerung übermittelt werden, damit die Achse anschließend mit den Befehlen REF, PGO, VGO etc. bewegt werden kann.</b> Vorher müssen folgende Parameter eingestellt worden sein: Motortyp, Limitschalter-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>=<sv>	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	
	MPGO=<uv>	Positionieren mit mehreren Achsen starten. Übergeben wird ein Zahlenwert, der angibt, welche Achsen gestartet werden sollen. Bit 0 = Achse 1... Bit 8 = Achse 9	MPGO=00000011	
	MVGO=<uv>	Geschwindigkeitsmodus mit mehreren Achsen starten. Übergeben wird ein Zahlenwert, der angibt, welche Achsen gestartet werden sollen. Bit 0 = Achse 1... Bit 8 = Achse 9	MVGO=00000011	
	STOP<n>	Bewegung einer Achse stoppen. Jegliche aktive Bewegung einer Achse wird abgebrochen. Der Antrieb stoppt mit der programmierten Bremsrampe und bleibt stehen.		
	MSTOP=<uv>	Mehrere Achsen stoppen. Übergeben wird ein Zahlenwert, der angibt, welche Achsen gestoppt werden sollen. Bit 0 = Achse 1... Bit 8 = Achse 9	MSTOP = 00000011	
	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	



Befehlsgruppe	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	
	?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.	JACC9=5	
	?JACC<n>	Maximalen Jerk („Ruck“) für eine Achse auslesen.	?JACC9	5
	EDACC<n>=<uv>	NOT-AUS-Verzögerung (Bremsbeschleunigung) für eine Achse einstellen. Diese Verzögerung wird benutzt, wenn ein Bremsschalter angesprochen hat.	EDACC1=1000	
	?EDACC<n>	NOT-AUS-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 bei Fahren mit Joystick 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
	JAUTOMOFF<n>=<uv>	Nur bei DC-Betrieb: Motor im Joystickmodus automatisch stromlos schalten, wenn die Zielposition erreicht ist.	JAUTOMOFF1=0	
	?JAUTOMOFF<n>	Abfrage, ob im Joystickmodus automatisch stromlos geschaltet wird.	?JAUTOMOFF1	0
	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	

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Positionierparameter	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=3	
	?JPLAZ	Joystickebenenzuordnung Z-Ebene Achsen-Nummer auslesen.	?JPLAZ	
	LIGO=<uv>	Positionierung mit Linearinterpolation für eine Achsengruppe (binäre Definitionsmaske) starten. Bit-Reihenfolge :<Achse 9, Achse 8, Achse 7, Achse 6,..., Achse 2, Achse 1>.	LIGO=000000111	
	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-Zeilenummer (0 bis ...), hinter dem „=“ -Zeichen folgen durch Komma getrennt die Werte für die Tabellen-Spalten. Parameter-Liste: 1. Weg mit Vorzeichen Achse 1 (-2147483648 bis 2147483648) 2. Weg mit Vorzeichen Achse 2 (-2147483648 bis 2147483648) 3. Weg mit Vorzeichen Achse 3 (-2147483648 bis 2147483648) 4. Weg mit Vorzeichen Achse 4 (-2147483648 bis 2147483648) 5. Weg mit Vorzeichen Achse 5 (-2147483648 bis 2147483648) 6. Weg mit Vorzeichen Achse 6 (-2147483648 bis 2147483648) 7. Weg mit Vorzeichen Achse 7 (-2147483648 bis 2147483648) 8. Weg mit Vorzeichen Achse 8 (-2147483648 bis 2147483648) 9. Weg mit Vorzeichen Achse 9 (-2147483648 bis 2147483648) 10. Funktionscode Bit 3 bis 0: Ausgangsnummer 1 Bit 4: Ausgangspegel 1 Bit 8 bis 5: Ausgangsnummer 2 Bit 9: Ausgangspegel 2 Bit 13 bis 10: Ausgangsnummer 3 Bit 14: Ausgangspegel 3 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, 0, 100, 0, 0, 192	
	?POSTAB<uv>	Eine Tabellenzeile aus der Bahntabelle auslesen. Als Parameter wird die Tabellen-Zeilenummer (0 bis ...) übergeben.	?POSTAB0	1000, 2000, 0, 0, 0, 0, 0, 100, 0, 0, 192, 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>, <uv>	Mit einem Parameter: Bahnsteuerung ab einem bestimmten Tabelleneintrag starten. Mit zwei Parametern: Bahnsteuerung ab einem bestimmten Tabelleneintrag starten und vor einem anderen Tabelleneintrag stoppen.	PTABGO0, 15	
	PTABSTP	Eine laufende Bahnsteuerung abbrechen; die beteiligten Achsen verhalten sich wie am Ende der Bahntabelle.	PTABSTP	
	PTABCLR	Tabelle für Bahnsteuerung löschen.		
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)			

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Positionierparameter	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	
	PTABDEL<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.	PTABDEL50=20	
	PTABSAVE	Tabelle in Flash speichern.	PTABSAVE	
	PTABSAVE<uv,uv>	Teile der Tabelle im Flash speichern. Der erste Wert gibt die Startzeile an, der zweite Wert die Anzahl der Zeilen. Alle anderen Zeilen im Flash werden nicht geändert.	PTABSAVE3,10	
	PTABLOAD	Komplette Tabelle aus dem Flash in den Arbeitsspeicher laden.	PTABLOAD	
	PTABLOAD<uv,uv>	Teile der Tabelle aus dem Flash in den Arbeitsspeicher laden. Der erste Wert gibt die Startzeile an, der zweite Wert die Anzahl der Zeilen.		
	?PTABFLASHBUSY	Prüfen, ob eine Speicheraktion in den Flash aktiv ist (=1). Falls ja, sollte die Steuerung nicht ausgeschaltet oder neugestartet werden.	?PTABFLASHBUSY	0
PositionAutomatisches Reagieren und Setzen von Ausgängen	TRIG<uv>=<a>, <b>,<c>,<d>,<e>]]	Schreibe einen Eintrag in die Aktionstabelle <uv>=Zeilennummer <a>=Aktions-ID <b>=Eingang <c>=Flanke (0=fallend, 1=steigend) <d>=Parameter 1, signed 32 bit decimal oder Hex-String, z.B. Achsennummer oder Achsenmaske <e>=ggf. Parameter 2 Keine Leerzeichen! Vorzeichen sind möglich. Hex-Strings müssen mit 0x oder 0X beginnen. Parameter d und e hängen von der gewählten Aktion ab.	TRIG2=10,0,3,5 Zeile 2: Fallende Flanke auf Eingang 10 startet Positioniervorgang von Achse 5	
	?TRIG<uv>	Zeile <uv> aus der Aktionstabelle ausgeben. Ausgaben immer als Dezimalzahl.	?TRIG2	10,0,3,5
	EVENT<uv>=<a>, <b>,<c>,<d>,<e>]]	Schreibe einen Eintrag in die Ereignistabelle <uv>=Zeilennummer <a>=Ereignis-ID <b>=Ausgangsmaske 16 bit decimal oder Hex-String <c>=Ausgangswertemaske 16 bit decimal oder Hex-String <d>=Parameter 1, signed 32 bit decimal oder Hex-String, z.B. Achsennummer oder Achsenmaske <e>=ggf. Parameter 2 Keine Leerzeichen! Vorzeichen sind möglich. Hex-Strings müssen mit 0x oder 0X beginnen. Parameter d und e hängen von der gewählten Aktion ab.	EVENT6=2,0x24, 0x4,8,10000 EVENT6=2,36, 4,8, 10000 Zeile 6: Wenn die Geschwindigkeit von Achse 8 unter 10000 fällt, Ausgang 3 auf 1 setzen und Ausgang 6 auf 0 setzen	
	?EVENT<uv>	Zeile <uv> aus der Ereignistabelle ausgeben. Ausgaben immer als Dezimalzahl.	?EVENT6	2,36,4,8,10000
	TRIGDEL[<a>,<b>]]	Ohne Parameter: Gesamte Aktionstabelle löschen Ein Parameter: Zeile a löschen Zwei Parameter: Zeilen a bis a+b löschen.	TRIGDEL3,10	
	EVENTDEL[<a>,<b>]]	Ohne Parameter: Gesamte Ereignistabelle löschen Ein Parameter: Zeile a löschen Zwei Parameter: Zeilen a bis a+b löschen.	EVENTDEL3,10	
	TRIGENABLE=<uv>	Ereignis- und Aktionsfunktion aktivieren (<uv>=1) oder deaktivieren (<uv>=0).	TRIGENABLE=1	
	?TRIGENABLE	Zustand der Funktion abfragen.	?TRIGENABLE	1
	TRIGTABSVE	Aktionstabelle in Flash speichern.	TRIGTABSVE	
	EVENTTABSVE	Ereignistabelle in Flash speichern.	EVENTTABSVE	
	?TRIGEVENTBUSY	Abfragen, ob aktuell Speichervorgänge in den Flash laufen 0: keine Speicherung 1: Speicherung Aktionstabelle 2: Speicherung Ereignistabelle 3: Speicherung beide Tabellen Während einer Speicherung die Steuerung nicht ausschalten oder neu starten.	?TRIGEVENTBUSY	0
	TRIGTABLOAD	Aktionstabelle aus Flash in Arbeitsspeicher laden.	TRIGTABLOAD	
	EVENTTABLOAD	Ereignistabelle aus Flash in Arbeitsspeicher laden.	EVENTTABLOAD	

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Achspanparameter	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 Maximalstromes im vorgewählten Strombereich (1 oder 2) in Prozent einstellen. Strombegrenzung bei DC-Achsen einstellen. 100 % entsprechen 12 A. Diese Einstellung wird nur nach einem Neustart der Steuerung mit der ersten Initialisierung übernommen.	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>	Verzögerungszeit des D-Anteils für eine Achse einstellen (in Sample-Zeit-Zyklen).	FDT1=5	
	?FDT<n>	Verzögerungszeit des D-Anteils für eine Achse abfragen (in Sample-Zeit-Zyklen).	?FDT1	5
	MXPOSERR<n>=<uv>	Maximalen Positionierfehler 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 und BLDC.	MXPOSERR1=50	
	?MXPOSERR<n>	Maximalen Positionierfehler 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. <b>Max. zulässiger Wert: 99 %</b>	MAXOUT1=95	
	?MAXOUT<n>	Maximalen Ausgabewert in Prozent auslesen.	?MAXOUT1	95
	MPUNIPID<n>=<uv1>, <uv2>, <uv3>, <uv4>	PID-Stromregler für Schrittmotorbetrieb einstellen. uv1 = KP uv2 = KI fast uv3 = KI slow uv4 = KD	MPUNIPID1=900,0,40,1500	
	?MPUNIPID<n>	PID-Stromreglerwerte abfragen	?MPUNIPID1	900 0 40 1500
	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
AMPPWMF<n>=<uv>	PWM-Frequenz für Endstufe einstellen, 20000 oder 80000 ist möglich.	AMPPWMF1=20000		
?AMPPWMF<n>	PWM-Frequenz für Endstufe abfragen.	?AMPPWMF1	20000	
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=25		
?MOTPOLES<n>	Polanzahl des Motors für eine Achse auslesen.	?MOTPOLES1	25	
BLDCCT<n>=<uv>	Kommutierungsmodus bei BLDC einstellen: 0 = Blockkommutierung mit Hallensoren 1 = Sinuskommutierung mit Encoder	BLDCCT1=0		
?BLDCCT<n>	Kommutierungsmodus bei BLDC abfragen.	?BLDCCT1	0	

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Achsparparameter	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 stehenbleiben 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		
	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	
Endschalterkonfiguration und Referenzfahrt	?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=1000	
	?RDACC<n>	Referenzfahrt-Verzögerung einer Achse auslesen.	?RDACC1	1000
	SMK<n>=<uv>	Endschaltermaske für eine Achse setzen. Mit diesem Befehl werden die Endschalter und die Bremschalter 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=0110	
	?SMK<n>	Endschaltermaske für eine Achse auslesen.	?SMK3	0110
	SPL<n>=<uv>	Endschalterpolarität für eine Achse setzen. Mit diesem Befehl wird der aktive Pegel für die Endschalter und Bremschalter 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 „1“ übergeben werden. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	RMK3=0001	
	?RMK<n>	Referenzschaltermaske für eine Achse auslesen.	?RMK3	1
	RPL<n>=<uv>	Referenzschalterpolarität für eine Achse setzen. Dieser Befehl definiert den aktiven Pegel des Referenzschalters. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	RPL3=1110	
	?RPL<n>	Referenzschalterpolarität für eine Achse auslesen.	?RPL3	1110
	?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

Befehlsgruppe	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 Löschrmaske.	ETTLOUTC1=01	
	?INPUTS	Aktuellen Zustand der Eingänge auslesen (16-Bit Binärzahl).	?INPUTS	0010100100101101
	?INPTTL	aktuellen Zustand der TTL-Eingänge auslesen (8-Bit Binärzahl).	?INPTTL	00110011
	?INSPS	aktuellen Zustand der SPS-Eingänge auslesen (8-Bit Binärzahl).	?INSPS	00110011
	OUTPUT<uv>=<uv>	Aktuellen Zustand eines Ausgangs ändern.	OUTPUT1=0	
	?OUTPUTS	Aktuellen Zustand aller Ausgänge auslesen.	?OUTPUTS	0010100100101101
	OUTTTL<uv>=<uv>	Aktuellen Zustand eines TTL-Ausgangs ändern.	OUTTTL1=0	
	?OUTTTL	Aktuellen Zustand aller TTL-Ausgänge auslesen.	?OUTTTL	00101001
	OUTSPS<uv>=<uv>	Aktuellen Zustand eines SPS-Ausgangs ändern.	OUTSPS1=0	
	?OUTSPS	Aktuellen Zustand aller SPS-Ausgänge auslesen.	?OUTSPS	00101001
	INMODE=<uv>	Eingangsspegel TTL/SPS umschalten (0 = TTL, 1 = SPS).	INMODE0	
	?INMODE	Aktuellen eingestellten Eingangsspegel TTL/SPS abfragen.	?INMODE	0
	?ANIN<uv>	Analog-Eingang abfragen, angegeben wird die Kanal-Nummer von 1 bis 8, zurückgegeben wird der gewandelte 10-Bit Wert.	?ANIN3	234
	DAOUT<uv>=<uv>	Analog-Ausgang setzen, angegeben wird die Kanal-Nummer von 1 bis 8 und der Ausgabewert für den DA-Wandler.	DAOUT2=250	
	?DAOUT<uv>	Analog-Ausgang abfragen, angegeben wird die Kanal-Nummer von 1 bis 8, zurückgegeben wird der zuletzt eingestellte Digital-Wert.	?DAOUT2	250
	OPWM<uv>=<uv>	PWM-Ausgang setzen, angegeben wird die Kanal-Nummer von 1 bis 4 und der Aussteuerungswert von 0 bis 100%.	OPWM1=55	
	?OPWM<uv>	PWM-Ausgang abfragen, angegeben wird die Kanal-Nummer von 1 bis 4 und zurückgegeben wird der zuletzt eingestellte Aussteuerungswert von 0 bis 100%.	?OPWM1	55
	AXOUTPUT<n>=<uv>	AxisOut-Pin einer Achsen auf High/Low setzen.		
	?IOCONFIG	aktuell eingestellte IO-Konfiguration auslesen.	?IOCONFIG	15
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	
	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 absoluten 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	

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Nachlaufregelung	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>=<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. Modus 6: Grobpositionierung (Phase 1), Iteration (Phase 2), Phase 3 mit Hybrid-Nachführung, Phase 3 wird im Zielfenster beendet. Modus 7: Grobpositionierung (Phase 1), Iteration (Phase 2), Phase 3 mit Hybrid-Nachführung, Phase 3 bleibt aktiv. Modus 8: Grobpositionierung (Phase 1), Korrekturfahrt (Phase 2), Phase 3 mit Hybrid-Nachführung, Phase 3 wird im Zielfenster beendet. Modus 9: Grobpositionierung (Phase 1), Korrekturfahrt (Phase 2), Phase 3 mit Hybrid-Nachführung, Phase 3 bleibt aktiv.	PWMSMODE1=6	
	?PWMSMODE<n>	Positioniermodus für die Positionierung mit WMS abfragen.	?PWMSMODE1	6
Nachlaufregelung	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	16
	?PWMSERR<n>	Auslesen des aktuellen Positionsfehlers einer Achse beim Positionieren mit WMS.	?PWMSERR1	0
	WMSINV<n>=<uv>	Zählrichtung des WMS invertieren (1=ja / 0=nein).	WMSINV1=0	
	?WMSINV<n>	Auslesen, ob Zählrichtung WMS invertieren (ja/nein).	?WMSINV1	0
	WMSOFFS<n>=<sv>	Positionsoffset mit Vorzeichen für die Grobposition bei Piezo-Positionierung mit WMS einstellen.	WMSOFFS1=-80	
	?WMSOFFS<n>	Positionsoffset mit Vorzeichen für die Grobposition bei Piezo-Positionierung mit WMS abfragen.	?WMSOFFS1	80
	PWMSPWIN<n>=<uv>	Zielfensterbreite für die Feinpositionierung mit WMS und Piezo einstellen (Phase 3).	PWMSPWIN1=0	
	?PWMSPWIN<n>	Zielfensterbreite für die Feinpositionierung mit WMS und Piezo abfragen (Phase 3).	?PWMSPWIN1	0
?PVOLTG<n>	Anfragen des aktuellen Piezo-Ausgangswertes bei WMS-Positionierung mit Piezo.	?PVOLTG1	487	

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Nachlaufregelung	?DACINPUTS	Hybrid-Fehlerstatus abfragen (als Bitmuster). Bit 0: Fehler, Hybrid-Achse 1 Bit 1: Fehler, Hybrid-Achse 2 Bit 2: Fehler, Hybrid-Achse 3 Bit 3: Fehler, Hybrid-Betriebsspannung 1 Bit 4: Fehler, Hybrid-Achse 4 Bit 5: Fehler, Hybrid-Achse 5 Bit 6: Fehler, Hybrid-Achse 6 Bit 7: Fehler, Hybrid-Betriebsspannung 2	?DACINPUTS	11110001
	PWMSPTIM<n>=<uv>	Zykluszeit der Hybrid-Positionierung einer Achse mit Nachlaufregelung einstellen.	PWMSPTIM1=1	
	?PWMSPTIM<n>	Zykluszeit der Hybrid-Positionierung einer Achse mit Nachlaufregelung abfragen.	?PWMSPTIM1	1
	PWMSPMXO<n>=<uv>	Maximalen Hybrid-Ausgabewert einer Achse setzen.	PWMSPMXO1=4095	
	?PWMSPMXO<n>	Maximalen Hybrid-Ausgabewert einer Achse abfragen.	?PWMSPMXO1	4095
Haltebremsenansteuerung	HBCH<n>=<uv>	PWM-Ausgang für Haltebremse einer Achse zuordnen: <Achsennummer> = <PWM-Kanal> PWM-Kanal = 0 für Haltebremsenfunktion aus.	HBCH8=3	
	?HBCH<n>	Zuordnung Haltebremse PWM-Kanal einer Achse abfragen.	?HBCH8	3
	HBFV<n>=<uv>	Ersten PWM-Wert (zum Anziehen) bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Prozentwert>.	HBFV8=50	
	?HBFV<n>	Ersten PWM-Wert (zum Anziehen) bei der Ansteuerung der Haltebremse abfragen	?HBFV8	50
	HBSV<n>=<uv>	Zweiten PWM-Wert (zum Halten) bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Prozentwert>.	HBSV8=20	
	?HBSV<n>	Zweiten PWM-Wert (zum Halten) bei der Ansteuerung der Haltebremse abfragen.	?HBSV8	20
	HBTI<n>=<uv>	Zeit für ersten PWM-Wert bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Zeit für ersten PWM-Wert in ms>	HBTI8=300	
?HBTI<n>	Zeit für ersten PWM-Wert bei der Ansteuerung der Haltebremse abfragen	?HBTI8	300	
Reset	RESETAC	Reset Antriebsplatinen auslösen.	RESETAC	
	RESETMB	Reset Hauptplatine auslösen.	RESETMB	
Stand-Alone-Programmierung	SAMEM	Merker-Wert setzen.	SAMEM38=50	
	?SAMEM	Marker-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.		
Anybus®-Modul	ABNETADR=<uv>	Feste Anybus-IPv4-Adresse setzen als Dezimalzahl Aktiv nach globalem Speichern und Neustart.	ABNETADR=168428041 (für 10.10.2.9) ABNETADR=3232235628 (für 192.168.0.108)	
	ABNETADR1=<uv>	Feste Anybus-IPv4-Adresse setzen. Notation mit Punkten Aktiv nach globalem Speichern und Neustart.	ABNETADR1=10.10.2.9 ABNETADR1=192.168.0.108	



Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Anybus®-Modul	?ABNETADR	Anybus-IPv4-Adresse abfragen als Dezimalzahl. Gibt die zuletzt gespeicherte Adresse zurück.	?ABNETADR	168428041
	?ABNETADR1	Anybus-IPv4-Adresse abfragen. Notation mit Punkten. Gibt die zuletzt gespeicherte Adresse zurück.	?ABNETADR1	10.10.2.9
	ABNETSUB=<uv>	Anybus-IPv4-Netzmaske-Adresse setzen als Dezimalzahl Aktiv nach globalem Speichern und Neustart.	ABNETSUB= 168428032 (für 10.10.2.0)	
	ABNETSUB1=<uv>	Anybus IPv4-Netzmaske-Adresse setzen. Notation mit Punkten Aktiv nach globalem Speichern und Neustart.	ABNET- SUB1=10.10.2.0	
	?ABNETSUB	Anybus-IPv4-Netzmaske abfragen als Dezimalzahl Gibt die zuletzt gespeicherte Adresse zurück.	?ABNETSUB	168428032
	?ABNETSUB1	Anybus-IPv4-Netzmaske abfragen. Notation mit Punkten Gibt die zuletzt gespeicherte Adresse zurück.	?ABNETSUB1	10.10.2.0
	ABNETGW=<uv>	Anybus-IPv4-Gateway-Adresse setzen als Dezimalzahl Aktiv nach globalem Speichern und Neustart.	ABNETGW= 168248287 (für 10.10.2.255)	
	ABNETGW1=<uv>	Anybus-IPv4-Gateway-Adresse setzen. Notation mit Punkten Aktiv nach globalem Speichern und Neustart	ABNETGW1= 10.10.2.255	
	?ABNETGW	Anybus-IPv4-Gateway-Adresse abfragen als Dezimalzahl Gibt die zuletzt gespeicherte Adresse zurück.	?ABNETGW	168248287
	?ABNETGW1	Anybus-IPv4-Gateway-Adresse abfragen. Notation mit Punkten Gibt die zuletzt gespeicherte Adresse zurück.	?ABNETGW1	10.10.2.255
	ABNETDHCP=<uv>	Anybus-DHCP aktivieren (=1) oder deaktivieren (=0) Aktiv nach globalem Speichern und Neustart.	DHCP=0 DHCP=1	
	?ABNETDHCP	Status des DHCP abfragen. Gibt den zuletzt gesetzten Status zurück.	?DHCP	0
	ABNETCOM=<uv>	Anybus-Netzwerk-Kommunikationsparameter einstellen, z.B. die Bitrate.	ABNETCOM=10	
	?ABNETCOM	Anybus-Netzwerk-Kommunikationsparameter abfragen.	?ABNETCOM	10

## II Relevanz der Parameter für verschiedene Motortypen

Parameter	DC-Brush	2-Phasen-Schrittmotor Open Loop	2-Phasen-Schrittmotor Closed-Loop	BLDC
MOTYPE	+	+	+	+
AXIS	+	+	+	+
FKP	+	-	+	+
FKD	+	-	+	+
FDT	+	-	+	+
FKI	+	-	+	+
FIL	+	-	+	+
FST	+	-	+	+
MAXOUT	+	+ <sup>1)</sup>	+	+
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	BLDC
MOTPOLES	-	-	+	+
ENCLINES	-	-	+	+
ELCYCNT	-	-	+	+
BLDCCT	-	-	+	+
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

#### TTL-Ein-/Ausgänge

Pinbelegung des 25-poligen D-Sub (male)

TTL-I/O	Pin
Input 1	16
Input 2	17
Input 3	18
Input 4	19
Input 5	20
Input 6	21
Input 7	22
Input 8	23
Output 1	3
Output 2	4
Output 3	5
Output 4	6
Output 5	7
Output 6	8
Output 7	9
Output 8	10
+ 5V, max. 300 mA Gesamtstrom	1, 2, 14, 15
GND	11, 12, 24, 25
n. c.	13

#### SPS-Ein-/Ausgänge

Pinbelegung des 25-poligen D-Sub (female)

SPS-I/O	Pin
Input 1	16
Input 2	17
Input 3	18
Input 4	19
Input 5	20
Input 6	21
Input 7	22
Input 8	23
Output 1	3
Output 2	4
Output 3	5
Output 4	6
Output 5	7
Output 6	8
Output 7	9
Output 8	10
+24V, max. 1000 mA Gesamtstrom	1, 2, 14, 15
GND	11, 12, 24, 25
n. c.	13

#### Analog-Ein-/Ausgänge

Pinbelegung des 25-poligen D-Sub (male)

Analog-I/O	Pin
Input 1	6
Input 2	5
Input 3	4
Input 4	3
Input 5	10
Input 6	9
Input 7	8
Input 8	7
Output 1	23
Output 2	22
Output 3	21
Output 4	20
Output 5	19
Output 6	18
Output 7	17
Output 8	16
+ 5V, max. 300 mA Gesamtstrom	1, 2, 14, 15
GND	11, 12, 24, 25
U <sub>ref</sub> Output 4,096V	13

#### RS-232

Pinbelegung des 9-poligen D-Sub (female)

RS-232	Pin
CD	1
RX	2
TX	3
DTR	4
GND	5
DSR	6
RTS	7
CTS	8
RI	9

## Universal-Motorstecker

Mit dem passenden OWIS® Anschlusskabel werden die OWIS® Positioniereinheiten angeschlossen. Über diesen Anschlussstecker wird der Motor mit Leistung versorgt, die Signale des Encoders, evtl. der Hall-Effekt-Sensoren und der Schalter übertragen, sowie die Motor-Haltebremse, falls vorhanden, gesteuert.

Pinbelegung des 37-poligen D-Sub (female):

	Pin	DC-Motor	Schrittmotor OL	BLDC
Leistung	19	Motor +	Phase 1 +	U
	18	Motor -	Phase 1 -	V
	17	Motor +	Phase 2 +	W
	16	Motor -	Phase 2 -	-

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

Schalter + Signale	5	MINSTOP		
	4	MINDEC		
	3	MAXDEC		
	2	MAXSTOP		
	1	GND		
	37	Motorhaltebremse +24V		
	36	Motorhaltebremse -		
	35	(reserviert)		
	34	(reserviert)		
	33	(reserviert)		
	32	(reserviert)		
	31	GND		
	30	+ 5V		
	29	(reserviert)		
	28			Hallsensor A+
	27			Hallsensor A-
	26			Hallsensor B+
	25			Hallsensor B-
	24			Hallsensor C+
	23			Hallsensor C-
	22	+ 5V		
	21	GND		
20	+24V			

## Anschlusskabel

1. Signalkabel mit Gesamtschirm Twisted Pair 8x2x0,15 mm<sup>2</sup> und Sternvierer innen, geschirmt, 4x0,25 mm<sup>2</sup>

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

2. Motorkabel mit Gesamtschirm

Ader Nr.	Farbe	Querschnitt
1	rot	0,6 mm <sup>2</sup>
2	blau	0,6 mm <sup>2</sup>
3	weiß	0,6 mm <sup>2</sup>
4	schwarz	0,6 mm <sup>2</sup>
5	braun	0,6 mm <sup>2</sup>
6	rosa	0,5 mm <sup>2</sup>
7	grau	0,5 mm <sup>2</sup>

## Kabelvorschlag für RS-232-Schnittstelle

Zur Herstellung einer Kommunikationsverbindung mit einem PC wird ein Standardkabel mit 1:1 Verdrahtung verwendet.





## EU/UE Konformitätserklärung/Declaration of conformity

Wir  
We

OWIS GmbH  
Im Gaisgraben 7  
79219 Staufen / Germany  
+49(0)7633/9504-0  
+49(0)7633/9504-44  
www.owis.eu  
info@owis.eu

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

PS 90+

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 55011:2016 + A1:2017; EN 61000-6-2:2005; EN 61000-3-2:2014; EN 61000-3-3:2013  
EN 61010-1:2010; EN 50581:2012

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

2014/30/EU; 2014/35/EU; 2011/65/EU

Ort und Datum der Ausstellung  
Place and date of issue

Name und Unterschrift  
Name and signature

Staufen im Breisgau, 31.07.2020

  
D. J. Schuhen

  
P. Hilgers