

## Positioniersteuerung

**PS 10-32**

9013.0185 / 13.07.2017





# Inhalt

1. Allgemeines .....	5
2. Ausführung und Lieferumfang .....	5
2.1 Standard .....	5
2.2 Zubehör .....	5
3. Sicherheit .....	6
Ströme und Spannungen .....	6
4. Normen und Richtlinien .....	6
5. Technische Übersicht .....	6
6. Aufbau der Steuerung .....	7
Status-LEDs .....	7
Anzeige der Status-LEDs .....	7
6.1 Anschlüsse .....	8
USB-Schnittstelle .....	8
Universal-Motoranschluss .....	8
End- und Referenzschalter .....	8
Encodereingang .....	8
Stromversorgung .....	8
Sicherungskonzept .....	8
BUS-Schnittstelle .....	8
Anschluss der Stromversorgung .....	9
6.2 Eingänge und Ausgänge .....	9
6.3 Strombereichsumschaltung der Motorenstufe .....	9
Vorwahl des Phasenstromes bei 2-Phasen-Schrittmotoren .....	9
Einstellung der Strombegrenzung für DC-Servomotoren .....	9
6.4 Einstellungen der Steuerplatine .....	10
Jumpereinstellungen .....	10
7. Steuerungsfunktionen .....	11
7.1 Trapezförmiges Punkt-zu-Punkt-Profil .....	11
7.2 Geschwindigkeitsmodus .....	11
7.3 Referenzierung .....	11
8. Wegerfassung .....	12
Lageregelung .....	12
Encoder .....	12
Linearmesssystem .....	12
9. PID-Regelschleifenalgorithmus .....	12
10. Positioniergeschwindigkeit und -beschleunigung, Berechnung .....	13
10.1 2-Phasen-Schrittmotor (Open Loop) .....	13
Allgemeines .....	13
Periodendauer .....	13
Endgeschwindigkeit .....	13
Beschleunigung bei Trapezprofil .....	13
10.2 DC-Servomotor .....	13
Allgemeines .....	13
Abtastzeit .....	13
Endgeschwindigkeit .....	13
Beschleunigung bei Trapezprofil .....	14
11. Inbetriebnahme der PS 10-32 .....	14
11.1 Vorbereitung der Steuerung .....	14
Aufstellung .....	14
11.2 Anschluss der Peripherie und Geräte .....	14
11.3 Systemstart .....	14
Initialisierung .....	14
Software .....	14
CANopen-Vernetzung .....	15
12. Fehlerüberwachung .....	15
12.1 Endschalter .....	15
Funktion der Endschalter-Überwachung .....	15
Konfiguration der End- und Referenzschalter .....	16
Wiederinbetriebnahme nach Achsenfehler .....	16
12.2 Endstufen-Fehlerüberwachung .....	16
12.3 Time-Out-Überwachung .....	16

# Content

1. General Information .....	29
2. Setup and Scope of Delivery .....	29
2.1 Standard .....	29
2.2 Accessories .....	29
3. Safety .....	30
Currents and Voltages .....	30
4. Standards and Directives .....	30
5. Technical Overview .....	30
6. Setup of the Control Unit .....	31
Status LEDs .....	31
Display of the status LEDs .....	31
6.1 Connections .....	32
USB Interface .....	32
Universal Motor Connector .....	32
Limit and Reference Switches .....	32
Encoder Input .....	32
Power Supply .....	32
Safety Fuse Concept .....	32
BUS Interface .....	32
Connection for Power Supply .....	33
6.2 In- and Outputs .....	33
6.3 Selection of the Current Range for the Motor Power Stage .....	33
Phase Current Setting for 2-Phase Step Motors .....	33
Current Limiting Setting for DC Servo Motors .....	33
6.4 Control Board Settings .....	34
Jumper Settings .....	34
7. Control Functions .....	35
7.1 Trapezoidal Point-to-Point Profile .....	35
7.2 Velocity Mode .....	35
7.3 Reference run .....	35
8. Travel Measuring .....	36
Position Feedback Control .....	36
Encoder .....	36
Linear Measuring System .....	36
9. PID Servo Loop Algorithm .....	36
10. Positioning, Velocity and Acceleration, Calculation .....	37
10.1 2-Phase Stepper Motor (Open Loop) .....	37
General Information .....	37
Cycle Time .....	37
Final Velocity .....	37
Acceleration for Trapezoidal Velocity Profiling .....	37
10.2 DC Servo Motor .....	37
General Information .....	37
Servo Loop Cycle Time .....	37
Final Velocity .....	37
Acceleration for Trapezoidal Velocity Profiling .....	38
11. Initial Operation of the PS 10-32 .....	38
11.1 Preparation of the Control Unit .....	38
Installation .....	38
11.2 Connection of Peripherals and Devices .....	38
11.3 System Start-up .....	39
Initialization .....	39
Software .....	39
CANopen-Networking .....	39
12. Malfunction Monitoring .....	39
12.1 Limit Switches .....	39
Working Principle of the Limit Switch Monitoring .....	40
Configuration of Limit and Reference Switches .....	40
Reconnection after Axis Error .....	40
12.2 Output Stage Error Monitoring .....	40
12.3 Time-Out Monitoring .....	40

13. Hinweise zum Aufbau einer eigenen Applikationssoftware .....	17
14. Befehlssatz der PS 10-32 .....	18
14.1 CANopen-Vernetzung .....	18
Anhang .....	19
I Befehlstabelle .....	19
II Relevanz der Parameter für verschiedene Motortypen .....	25
III Belegungstabelle .....	25
Ein-/Ausgänge .....	25
BUS-System .....	25
Universal-Motorstecker .....	26
EU Konformitätserklärung .....	51

13. Instructions Concerning the Setup of User Application Software .....	41
14. Command Set for the PS 10-32 .....	42
14.1 CANopen-Networking .....	42
Attachment .....	43
I Command Table .....	43
II Relevance of the Parameters for different Motor Types .....	49
III Connecting Table .....	49
In-/Outputs .....	49
BUS System .....	49
Universal Motor Connector .....	50
UE Declaration of Conformity .....	51

## 1. Allgemeines

Die OWIS® Steuerung PS 10-32 ist eine 1-Achs-Positioniersteuerung, die für einfache Steuerungsaufgaben eingesetzt werden kann.

Sie kann entweder 2-Phasen-Schrittmotoren (Open Loop) bis 1,8A oder DC-Motoren mit Encoder (Closed-Loop) bis 3,5A steuern.

Für die Kommunikation mit einem PC ist eine USB-Schnittstelle integriert. Weitere Peripherie kann über zahlreiche Ein- und Ausgänge wie z.B. TTL-, Analog-Eingänge so wie SPS-Ausgänge angebunden werden.

Die PS 10-32 kann Punkt-zu-Punkt-Positionierbetrieb und trapezförmige Geschwindigkeitsprofile ausführen.

Bei dieser Steuerung können bis zu 32 Einheiten vernetzt werden. Beliebige Kombinationen beider Motortypen sind dabei möglich. Die Vernetzung erfolgt über ein rudimentäres, vereinfachtes CANopen-Protokoll.

Zwei der insgesamt vier digitalen SPS Ausgänge sind in der PS 10-32 als PWM konfiguriert, um beispielsweise eine Bremse steuern zu können.

Zum Lieferumfang der Steuerung gehört auch das Softwaretool OWISoft. Damit kann die PS 10-32 komfortabel konfiguriert und betrieben werden. OWIS® Positioniereinheiten sind in OWISoft hinterlegt und müssen nur den jeweiligen Antrieb zugeordnet werden. Integration und Betrieb von Fremdmotoren ist ebenfalls möglich.

## 2. Ausführung und Lieferumfang

Die PS 10-32 besteht aus einem Grundgerät mit vorkonfigurierter Endstufe. Das Gerät wird bei OWIS® getestet und anschlussfertig geliefert.

Zum Lieferumfang der Steuerung gehören:

- USB-Kabel mit 2 m Länge
- CD mit OWISoft und Dokumentation in Deutsch und Englisch
- gedruckte Version der Betriebsanleitung in Deutsch und Englisch

### 2.1 Standard

Die Steuerung verfügt über:

- USB-Anschluss
- 4 Eingänge für Referenz- bzw. Endschalter
- 4 TTL-Eingänge
- 4 Analogeingänge
- 1 TTL-Ausgang
- 4 SPS-Ausgänge
- Anschluss für Freigabe der Motorenendstufe (I/O-Anschluss)
- Motoranschluss D-Sub 37-polig mit Anschluss für Motorhaltebremse (Option), End-/Referenzschalter und weitere Signale (siehe Pinbelegung, S.26)
- Rundstecker zum Durchschleifen der Stromversorgung von PS 10-32 zu PS 10-32
- 2 der 4 SPS-Ausgänge als PWM konfiguriert (z.B. zur Ansteuerung der Motorhaltebremse)
- BUS-Schnittstelle (an CANopen-Bus angelehnt)

### 2.2 Zubehör

Folgendes Zubehör ist erhältlich:

- Externes Tischnetzteil zur Stromversorgung AC 100-240V, DC 24 V, 90 W
- Anschlusskabel mit Stecker für unterschiedliche Positioniersysteme
- Verbindungskabel für Bus-Schnittstelle 2 m lang
- Abschlussstecker für Bus-Schnittstelle (Abschlusswiderstand)
- Verbindungskabel zum Durchschleifen der Stromversorgung 2 m lang

### 3. Sicherheit

Die Verwendung der PS 10-32 sollte ausschließlich von autorisiertem Fachpersonal und unter Berücksichtigung der Vorschriften der Unfallverhütung sowie den Vorschriften der Elektroindustrie durchgeführt werden. Beachten Sie hierzu auch die entsprechenden (im Datenblatt aufgeführten) Sicherheitshinweise.

Nicht qualifizierte Personen dürfen die Steuerung nicht in Betrieb nehmen.

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

Es muss vor hoher Luftfeuchtigkeit, Erschütterungen sowie explosiven Gasen geschützt werden.

Vor dem Öffnen des Gerätes muss es spannungsfrei geschaltet und von der Stromversorgung getrennt sein.

Anschluss- und Montagearbeiten dürfen nur im stromlosen Zustand des Gerätes ausgeführt werden. Montage und Einsatz von Betriebsmitteln muss gemäß den Normen der Konformitätserklärung erfolgen.

Um die Motorenstufe freigeben zu können, muss der Jumper JP7 auf der Hauptplatine gesetzt sein (siehe 6.4). Falls der Jumper nicht gesetzt ist, kann die Freigabe extern über den galvanisch getrennten Freigabe-Eingang erfolgen. Dazu muss der Eingang mit einer Spannung von 5 V versorgt werden. Sind weder der Jumper gesetzt noch die Freigabeeingang mit Spannung versorgt, ist kein Freischalten der Endstufen möglich.

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

Die jeweilige Konfiguration der Steuerung darf nur mit den für sie konfigurierten Motortypen betrieben werden. Andere oder weiterführende Nutzungen entsprechen nicht dem vorgesehenen Verwendungszweck.

### Ströme und Spannungen

Der Stromeingang ist über eine Feinsicherung 5AT abgesichert. Ausgangsseitig sind keine besonderen Sicherheitsvorkehrungen erforderlich, da die PS 10-32 ausschließlich mit Kleinspannung (PELV) bis 24VDC arbeitet.

Die Positioniersteuerung PS 10-32 ist nach den anerkannten sicherheitstechnischen Regeln gebaut und erfüllt die im folgenden Kapitel aufgeführten Normen und Richtlinien.

### 4. Normen und Richtlinien

Die Positioniersteuerung PS 10-32 erfüllt folgende Normen und Richtlinien:

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

Störfestigkeit nach Fachgrundnorm EN 61000-6-1 mit:

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

- Störfestigkeit gegen elektromagnetische Felder, Basisnorm: EN 61000-4-3

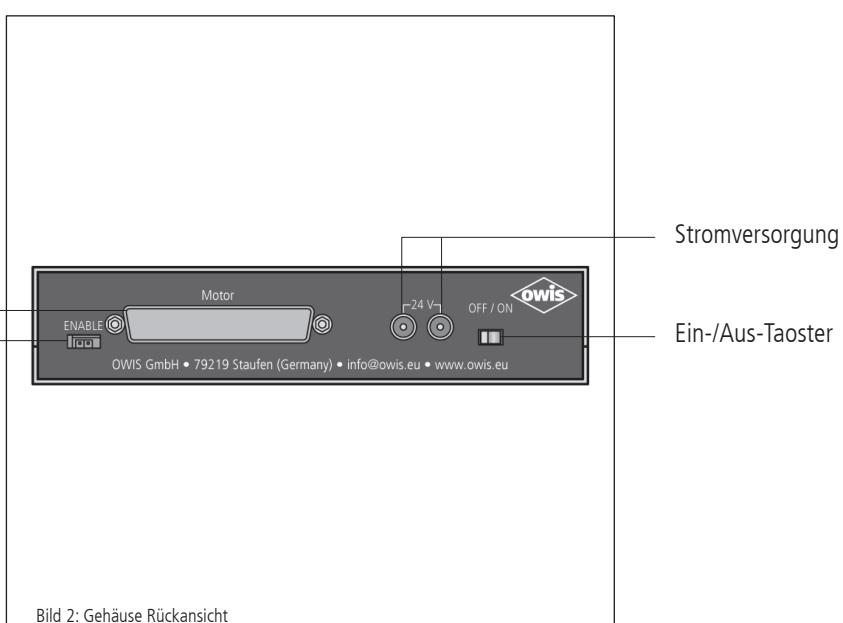
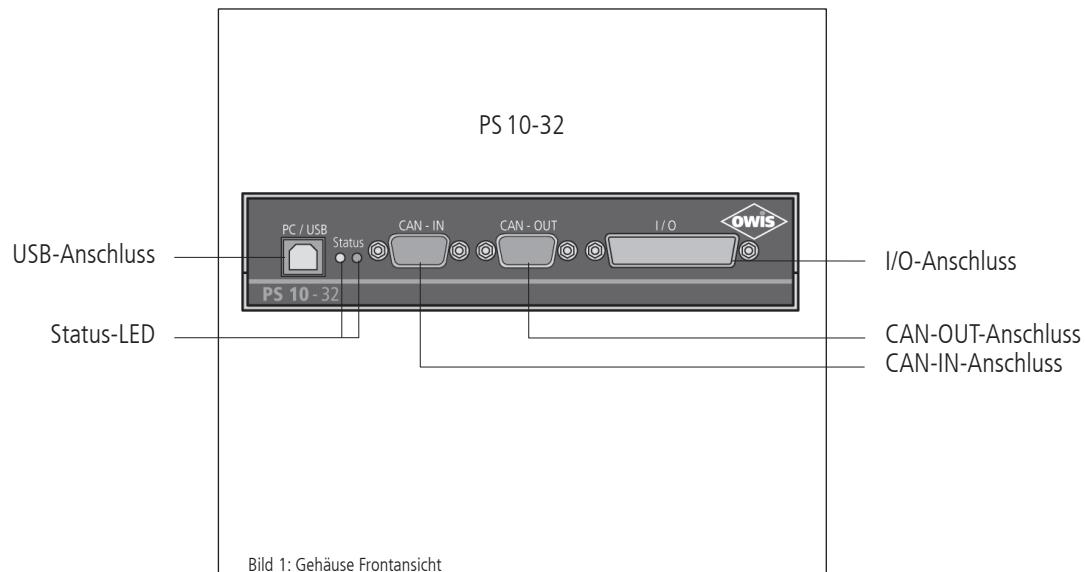
Störaussendung nach Fachgrundnorm EN 61000-6-3 mit:

- Störstrahlung, Basisnorm: EN 55014-1 (Hausgeräte)

### 5. Technische Übersicht

Stromversorgung:	Netzteil 24V, 90W extern (PS 10-32: Verbindungskabel zum Durchschleifen der Stromversorgung)
Anzahl der Antriebe:	1 Achse
Antriebsart:	(PS 10-32: Vernetzung bis zu 32 Einheiten) 2-Phasen-Schrittmotoren, Open Loop (OL) oder DC-Servomotoren, Closed-Loop (CL)
Kommunikation:	USB 2.0
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 1,875 MHz (Signal) bzw. 7,5 MHz (Quadratur)
Funktionen:	Parametrierbare Beschleunigungsrampe (/ Bremsrampe), Dreieckiges- bzw. trapezförmiges Geschwindigkeitsprofil
Bewegungsabläufe:	Punkt-zu-Punkt-Positionierbetrieb

## 6. Aufbau der Steuerung



Die PS 10-32 ist in einem stabilen Metallgehäuse untergebracht.

Die Abwärme der eingebauten Steuerplatine wird an die Umgebungsluft abgegeben.

### Status-LEDs

Die Betriebsanzeige der Steuerung wird über eine Status-LED angezeigt. Die angezeigten Zustände sind:

- Versorgung ein
- Motor initialisiert
- Referenz gültig
- Fehler

Die Anzeige erfolgt über die Farben und Blinkzustände.

### Anzeige der Status-LEDs

grüne LED	rote LED	Beschreibung
ein		Achse ist initialisiert und die Referenzposition ist gültig
aus		Achse ist nicht initialisiert
blinken langsam 1 Hz		Achse ist initialisiert und die Referenzposition ist ungültig
blinken schnell 2 Hz		Achse fährt
	ein	Es ist kein Fehler bei der Achse gesetzt
	blinken langsam 1 Hz	Es ist ein Fehler bei der Achse gesetzt

Fehler bedeutet: Limit-Schalter, Bremsschalter, Endstufenfehler, Timeoutfehler

## 6.1 Anschlüsse

Die Anschlüsse der PS 10-32 befinden sich auf der Gehäusevorder- und auf der Gehäuserückseite. Dies sind Schnittstellen zur Kommunikation, Ein- und Ausgänge für Peripherie, sowie der Anschluss für den Positionierer (siehe Bild 1 und 2).

Anschluss	Funktionen	Buchse
USB-Slave	Kommunikation mit einem PC	USB-Buchse Typ B
TTL-Ein-/Ausgänge	Interaktion mit externer Hardware	D-Sub 25-poliger Stecker
Analog-Eingänge	Interaktion mit externer Hardware	D-Sub 25-poliger Stecker
SPS-Ausgänge	Interaktion mit externer Hardware	D-Sub 25-poliger Stecker
Universal-Anschlussstecker	Motorversorgung mit Motor-Haltebremse und Encoder-/Endschalteranschluss	D-Sub 37-poliger Stecker
24V Netzanschluss/Verbindungskabel	Spannungsversorgung	DC-Rundstecker 5,5 x 2,1 x 11 mm
BUS-Schnittstelle (an CAN angelehnt)	Interaktion mit weiterer Steuerung	D-Sub 9-poliger Stecker

### USB-Schnittstelle

Die PS 10-32 hat eine USB 2.0-Slave-Schnittstelle, der Anschluss befindet sich auf der Gerätevorderseite. Die Schnittstelle ist USB 1.1 und 2.0 kompatibel. Die USB-Schnittstelle der PS 10-32 ist als sogenannte COM-Brücke realisiert. Der Windows-Gerätetreiber erkennt die PS 10-32 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.

Die PS 10-32 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 Übertragungsraten der PS 10-32 mit der im Gerätetreiber eingestellten Übertragungsraten übereinstimmen, sonst ist keine Kommunikation möglich. Voreinstellung ist 9 600 Baud. (Kann aus dem Abnahmeprotokoll entnommen werden.)

### 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

Beim Einschalten misst die Steuerung den Widerstandswert und signalisiert einen Fehler, wenn der gemessene Wert nicht zu der jeweiligen Steuerplatine passt. Die Fehlermeldung der Endstufe wird über das Kommando „?ASTAT“ ausgelesen (siehe Befehlssatz ab S.18).

Die Endstufe ist mit dem Universal-Anschlussstecker verbunden, an dem der Motor mit seinen Wicklungen, der Encoder und die Schalter, die zu dieser Antriebsachse gehören, angeschlossen sind.

### End- und Referenzschalter

An die Achse können maximal vier Endschalter angeschlossen werden. Dies können Mikroschalter, TTL-Hall-Effekt-Endschalter oder TTL-Lichtschranken sein. An die Eingänge können Öffner oder Schließer, gegen 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 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 hochohmigen Spannungsteiler auf 1,4V gezogen.

### Stromversorgung

Das optionale Schaltnetzteil der PS 10-32 ist für eine Eingangsspannung von 100VAC bis 240VAC mit 50/60 Hz ausgelegt (Weitbereichseingang) und generiert 24VDC, 90W ebenso versorgt es die Ein- und Ausgänge auf der Platine.

Diese Versorgungsspannungen für Logik und Leistung sind nicht galvanisch getrennt.

### Sicherungskonzept

Für die Steuerung ist eine Schmelzsicherung vorhanden, die entsprechend dem maximal auftretenden Strom ausgelegt ist. Standardmäßige Absicherung der Steuerung: 5AT

Für die Endstufe ist eine entsprechende Übertemperatursicherung vorhanden.

### In Version PS 10-32 zusätzlich:

#### BUS-Schnittstelle

Über ein an CANopen-angetriebenes BUS-System können bis zu 32 Einheiten miteinander vernetzt werden. Die erste PS 10-32 wird via USB an einen PC angeschlossen (Master), alle weiteren Einheiten können über die BUS-Kommunikationsverbindung weiter vernetzt werden (Slaves). Bei Kommandos, die nicht für die Master-Steuerung bestimmt sind, sendet sie die Kommandos über den Bus zu den übrigen PS 10-32 weiter. Die Rückantwort der adressierten PS 10-32 wird von der ersten (Master) empfangen und über USB an den PC zurückgegeben. Die Master-Steuerung scannt den Bus, deswegen wird sie als letzte angeschaltet.

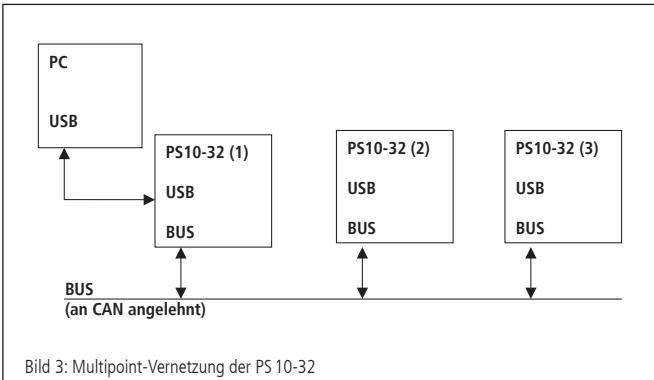


Bild 3: Multipoint-Vernetzung der PS 10-32

### Anschluss der Stromversorgung

Prinzipiell können mehrere PS 10-32 von einem externen Netzteil versorgt werden. Zum Durchschleifen der Versorgung von einer PS 10-32 zur nächsten sind auf einer Rückseite je zwei Rundstecker-Anschlüsse vorgesehen. Über ein Verbindungskabel wird die Versorgung dann an die folgende PS 10-32 weitergegeben. Man speist also an der ersten PS 10-32 mit dem 24V-Netzteil ein und schleift die Versorgung von dort auf die anderen durch. Dabei ist auf den Gesamtstromverbrauch der Positioniereinheiten zu achten. Für den Dauerbetrieb kann die Steuerung Schrittmotoren bis zu 1,8A, DC-Servomotoren bis zu 3,5A betreiben.

#### Hinweis:

Die maximale Leistung des Netzteiles darf nicht überschritten werden!

## 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 Gabellichtschränke etc. angeschlossen werden.

Mit dem TTL-Ausgang ist es möglich, digitale Hardware in der Anwendung direkt anzusteuern.

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

Eigenschaften	Pegel	Strom	Sonstiges
TTL-Eingänge	0-5V		—
Analogeingänge	0-5VDC		Auflösung 10 Bit
TTL-Ausgang	0-5V	10mA	—
SPS-Ausgänge	0-24VDC	0-100 mA	—
Leistungsausgänge	0-24VDC	1,0A	PWM

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

Die Abfragebefehle „?ANIN<uv>“ und „?INPUTS“ beziehen sich auf dieselben Eingänge der PS 10-32 (siehe Befehlssatz ab S.18). Die Auswertung der Eingänge erfolgt entweder analog oder digital.

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

Die Leistungsausgänge können als Haltebremsenansteuerung konfiguriert werden.

### 6.3 Strombereichsumschaltung der Motorenendstufe

Die PS 10-32 Endstufe besitzt zwei umschaltbare Strombereiche, um möglichst hohe Auflösung der Stromeinstellung bzw. möglichst feinen Mikroschrittbetrieb zu ermöglichen.

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

Auswahl von Strombereich 1 (niedrig), bzw. 2 (hoch) für Achse <n> erfolgt über folgende Kommandofolge:

AMPSHNT<n>=1

(Befehl zum Einstellen des Strombereiches 1 oder 2)

INIT<n>

(Befehl zum Initialisieren der Achse)

#### Vorwahl des Phasenstromes bei 2-Phasen-Schrittmotoren

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

Fahrstrom: DRICUR<n>=<uv>

(Befehl zum Einstellen des Fahrstromes in %)

Haltestrom: HOLCUR<n>=<uv>

(Befehl zum Einstellen des Haltestromes in %)

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

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

#### Hinweis:

Für den Dauerbetrieb ist ein Strom von 1,8A zulässig (entsprechen 54 % in Strombereich 2).

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

#### Einstellung der Strombegrenzung für DC-Servomotoren

Für DC-Servomotoren ist der geeignete Strombereich unter Berücksichtigung des thermisch zulässigen Dauerstroms des jeweiligen Motortyps vorzuwählen.

Maximale Strombegrenzung: DRICUR<n>=<uv>

(Befehl zum Einstellen der Strombegrenzung in %)

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

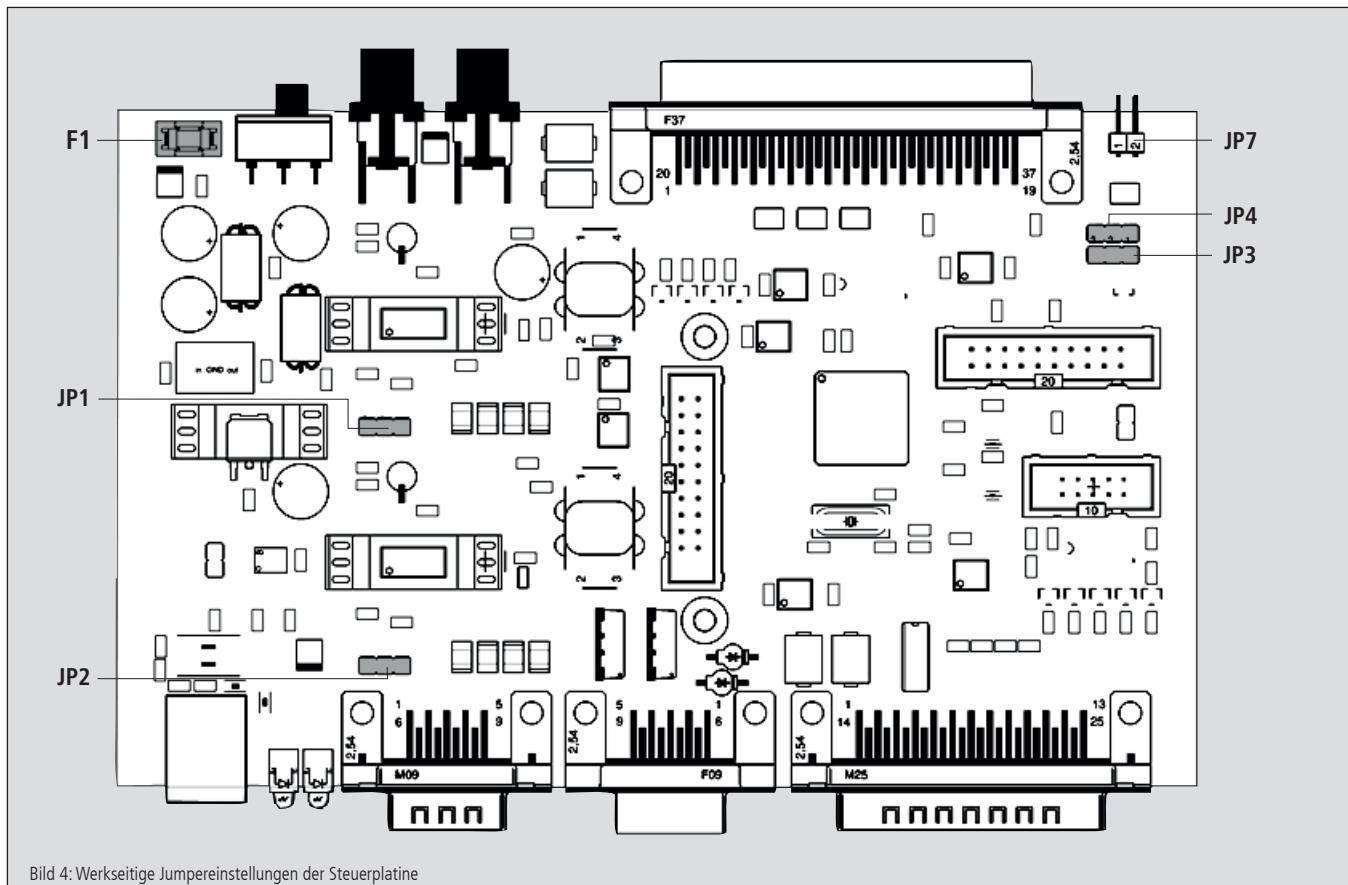
Strombereich 2 (entsprechend 100%): 6,6A

#### Hinweis:

Für den Dauerbetrieb ist ein Strom von 3,5A zulässig (entsprechen 54 % in Strombereich 2).

## 6.4 Einstellungen der Steuerplatine

### Jumpereinstellungen



Jumper	Funktion	Einstellmöglichkeiten	werksseitige Voreinstellung	Anmerkung
JP1	2-Phasen-Schrittmotor bzw. DC-Servomotor	1-2 gesteckt = 2-Phasen-Schrittmotor 2-3 gesteckt = DC-Servomotor	entsprechend gesteckt	Werkseinstellung darf nicht verändert werden
JP2	2-Phasen-Schrittmotor bzw. DC-Servomotor	1-2 gesteckt = 2-Phasen-Schrittmotor 2-3 gesteckt = DC-Servomotor	entsprechend gesteckt	Werkseinstellung darf nicht verändert werden
JP3	Modus „Firmware-Update“ oder „Betrieb“	2-3: Betrieb	gesteckt	Werkseinstellung darf nicht verändert werden
JP4	interne Verwendung	2-3: normaler Betrieb	gesteckt	Werkseinstellung darf nicht verändert werden
JP7	Endstufenfreigabe	Jumper gesteckt = Freigabe; Jumper nicht gesteckt = Freigabe extern möglich	gesteckt	

F1: 5A träge, zur Absicherung des externen 24V Versorgungseinganges der Steuerung.

## 7. Steuerungsfunktionen

### 7.1 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	(-2.147.483.647...+2.147.483.647) Counts
Geschwindigkeit	32.0	16 bit	(1...2.147.483.647) Counts / Cycle
Beschleunigung/ Verzögerung	32.0	16 bit	(1...2.147.483.647) Counts / Cycle <sup>2</sup>

Für dieses Profil errechnet der Host eine Beschleunigung, eine Verzögerung, eine Geschwindigkeit und eine Endposition. Das Profil ist nach der Kurvenform (Bild 5) benannt: die Achse beschleunigt linear (anhand des programmierten Beschleunigungswertes), bis sie die programmierte Geschwindigkeit erreicht. Die Achse bremst dann linear ab (den negativen Verzögerungswert nutzend), bis sie an der vorgegebenen Position stehen bleibt. Falls die programmierte Fahrtdistanz 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 6).

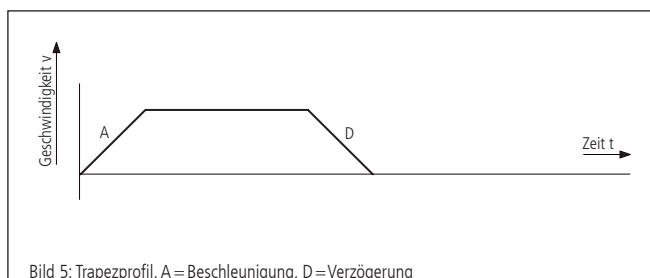


Bild 5: Trapezprofil, A = Beschleunigung, D = Verzögerung

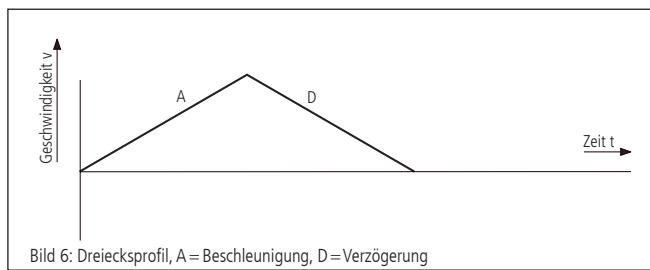


Bild 6: Dreiecksprofil, A = Beschleunigung, D = Verzögerung

Die Beschleunigungs- und Verzögerungsrampen sind symmetrisch.

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. Der Beschleunigungswert wird verwendet, bis die maximale Geschwindigkeit erreicht wurde. Der Verzögerungswert wird für die Abbremsrampe eingesetzt, bis die Geschwindigkeit auf Null sinkt.

### 7.2 Geschwindigkeitsmodus

Die folgende Tabelle fasst die Profilparameter für den Geschwindigkeitsmodus zusammen:

Profilparameter	Format	Wortlänge	Bereich
Geschwindigkeit	32.0	32 bit	(-2.147.483.647...+2.147.483.647) Counts / Cycle
Beschleunigung/ Verzögerung	32.0	16 bit	(1...2.147.483.647) Counts / Cycle <sup>2</sup>

Im Gegensatz zu dem trapezförmigen Profilmodus, bei dem 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 10-32 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 7 dargestellt.

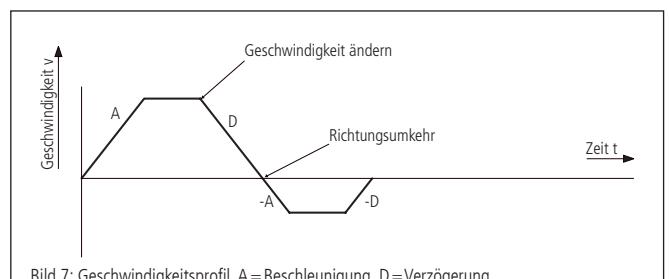


Bild 7: Geschwindigkeitsprofil, A = Beschleunigung, D = Verzögerung

#### 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.

### 7.3 Referenzierung

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

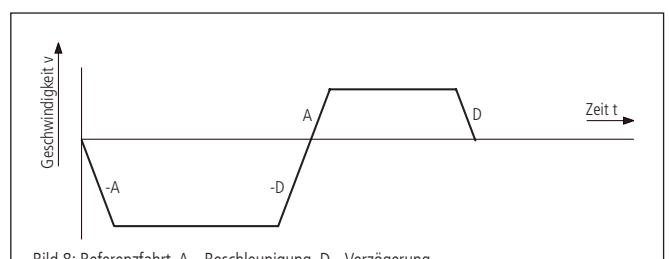


Bild 8: Referenzfahrt, A = Beschleunigung, D = Verzögerung

## 8. Wegerfassung

### Lageregelung

Für den Betrieb von DC-Servomotoren ist ein Encodereingang vorhanden. Das Signal dient der Datengewinnung für den Lage-Regelkreis (PID-Lageregelung).

### Encoder

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

Ohne Encoder ist nur der gesteuerte Betrieb (Open Loop) mit Schrittmotoren möglich. Um DC-Motoren betreiben zu können, muss ein Wegerfassungssystem angeschlossen sein. Dies kann ein Encoder sein. Üblicherweise besitzen sie 500, 1250 oder 2500 Linien pro Umdrehung. Über den Encoder erfasst der Controller die aktuelle Position der Achse und berechnet aus der zeitlichen Veränderung der Positionsdaten 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 10-32 kann als Encodersignale TTL-Pegel oder antivalente Signale (über Leitungstreiber) verarbeiten.

Die Signale werden nach einer Pegelumwandlung und Filterung direkt an den Microcontroller weitergegeben.

### Linearmesssystem

Ein Messsystem, welches direkt an die Bewegung des Aktors gekoppelt ist, nennt man Linearmesssystem. Das Wegmesssystem kann alternativ zum Encoder der Wegerfassung dienen.

## 9. PID-Regelschleifenalgorithmus

Der in der PS 10-32 benutzte Servofilter arbeitet nach einem PID-Algorithmus. Ein Integrationslimit sichert nach oben gegen einen akkumulierten Fehler ab.

Die PID-Formel lautet wie folgt:

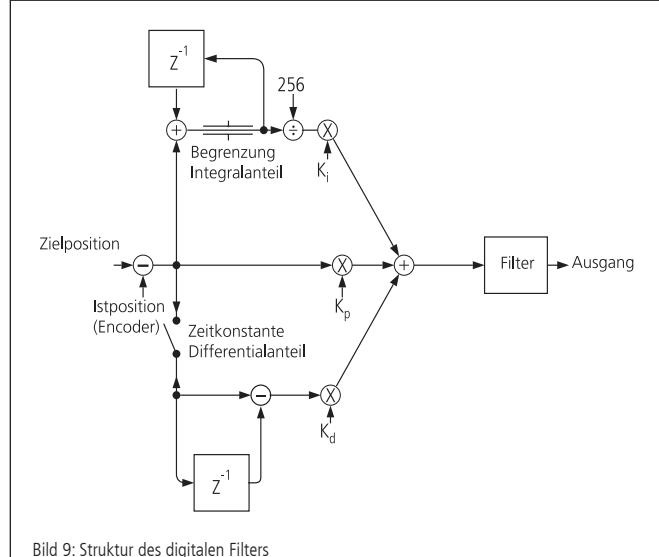
$$\text{output}_n = K_p E_n + K_d (E_n - E_{(n-1)}) + \sum_{j=0}^n E_j \frac{K_i}{256}$$

Hierbei ist:

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

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

Terminus	Name	Bereich
$I_{\text{lim}}$	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)



# 10. Positioniergeschwindigkeit und -beschleunigung, Berechnung

## 10.1 2-Phasen-Schrittmotor (Open Loop)

### Allgemeines

Jede schrittmotorgetriebene Mechanik besitzt eine insbesondere von Motortyp, Systemreibung und Last (Trägheitsmoment!) abhängige sog. Start-Stop-Frequenz. Die Start-Stop-Frequenz bezeichnet die maximale Fahr Frequenz 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 Schritt Winkel 1,8°, d.h. R=200 Vollschritte pro Motorumdrehung, der z.B. mit 400 Hz VS läuft, dreht mit einer Geschwindigkeit von 2 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 mit trapezförmigem Profil. Gegebenenfalls ist eine Dämpfung (Rotationsdämpfer, 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 Hz VS im Start-Stop-Betrieb zu folgen.

Die PS 10-32 besitzt einen digitalen Profilgenerator. Die Geschwindigkeitsprofile werden periodisch berechnet und an den 2-Phasen-Schrittmotor ausgegeben.

### Periodendauer

Die Periodendauer des digitalen Profilgenerators ist durch die Hardware festgelegt.

$T_p = 256 \mu s$

### Endgeschwindigkeit

Die Positionierung der Achsen wird im Punkt-zu-Punkt-Verfahren vorgenommen. Die Achse beschleunigt mit trapezförmigem Geschwindigkeits-Profil.

Die Endgeschwindigkeit V nach der Beschleunigungsrampe wird als 16-bit-Wort angegeben. Ihr Wertebereich reicht von 1 bis 2.147.483.647.

#### 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 (dimensionslos) und gegebenem Mikroschrittfaktor Mcstp errechnet sich die Schrittfrequenz f wie folgt:

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

bzw.

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

#### Hinweis:

Maximal ist eine Frequenz von  $f_{VS, \text{max}} = \frac{1}{T_p}$  einstellbar.

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

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

bzw.

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

Für die Umrechnung von 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 16-bit-Wort anzugeben, der Wertebereich reicht von 1 bis 2.147.483.647.

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

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

Zurückgelegte Distanz während der Trapezprofil-Beschleunigungsrampe:

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

## 10.2 DC-Servomotor

### Allgemeines

Die PS 10-32 hat einen digitalen Lage-Geschwindigkeits-Regler. Stell- und Regelgröße werden periodisch berechnet. Die Erfassung des Positions-Istwertes geschieht im einfachsten Fall mittels eines Drehgebers, der am 2. Wellenende des Motors angeflanscht ist. Wichtigste Kenngröße des Encoders ist die Encoder-Strichzahl R. Sie gibt die Anzahl der sog. Linien, d.h. Hell-Dunkel-Perioden je Motorwellenumdrehung, an. Die Signale durchlaufen eine Vierfach-Auswertung, woraus sich generell eine 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 204 µs. Sie kann bei Bedarf erhöht werden:

$$T_s = (204, \dots, 20.000) \mu s$$

Als Abtastzeit können nur ganzzahlige Werte an die PS 10-32 übergeben werden.

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

### Endgeschwindigkeit

Die Positionierung der Achse wird im Punkt-zu-Punkt-Verfahren vorgenommen. Hierbei beschleunigt die Achse mit trapezförmigem Geschwindigkeits-Profil.

Die Endgeschwindigkeit V (dimensionslos) nach der Beschleunigungsrampe wird als 16-bit-Wert angegeben. Ihr Wertebereich reicht von 1 bis 2.147.483.647.

**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:

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

bzw.

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

bzw.

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

Die letzte Formel kann auch wie folgt verstanden werden:

Der Controller verfährt  $V$  Inkremete je Sekunde.

Für die Umrechnung von 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 (entsprechen 2000 Impulsen/Umdrehung) am Motor eingesetzt.

Wie ist  $V$  zu wählen?

Berechnung:

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

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

Damit wird  $V = 60000$  für  $n = 1800 \text{ /min}$  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 16-bit Wort anzugeben, der Wertebereich reicht von 1 bis 2.147.483.647

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

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

Zurückgelegte Distanz während der Trapezprofil-Beschleunigungsrampe:

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

## 11. Inbetriebnahme der PS 10-32

### 11.1 Vorbereitung der Steuerung

#### Aufstellung

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

**Hinweis:**

Wärmestau an der Steuerung ist zu vermeiden.

### 11.2 Anschluss der Peripherie und Geräte

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

Es müssen:

- die Positioniereinheit
- die Stromversorgung
- der Computer

sowie für eine Multipoint-Vernetzung:

- das Verbindungskabel zur Signalübertragung mit Abschlusssteckern (Abschlusswiderstand)
- die Vernetzung der Stromversorgung (bzw. alternativ ein oder weitere Netzteile) angeschlossen werden.

Die Verbindung zum Computer erfolgt über die USB-Schnittstelle.

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

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

**Hinweis:**

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

### 11.3 Systemstart

Beim ersten Windows-Start mit angeschlossener PS 10-32 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 die Achse zunächst per INIT-Befehl initialisiert werden. Achsenparameter, die verändert werden, 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 Version 8.2) und zusätzliche Programmiersprachen (32/64-Bit). Damit kann die PS 10-32 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 Achse 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 Kapitel 3.7 der Bedienungsanleitung OWISoft.

Für die Inbetriebnahme mittels eigener Applikationssoftware folgt im Anschluss das Kapitel „Hinweise zum Aufbau einer eigenen Applikationssoftware“. Dort ist im Anschluss auch eine Tabelle mit den Befehlssätzen der PS 10-32 angefügt.

### CANopen-Vernetzung

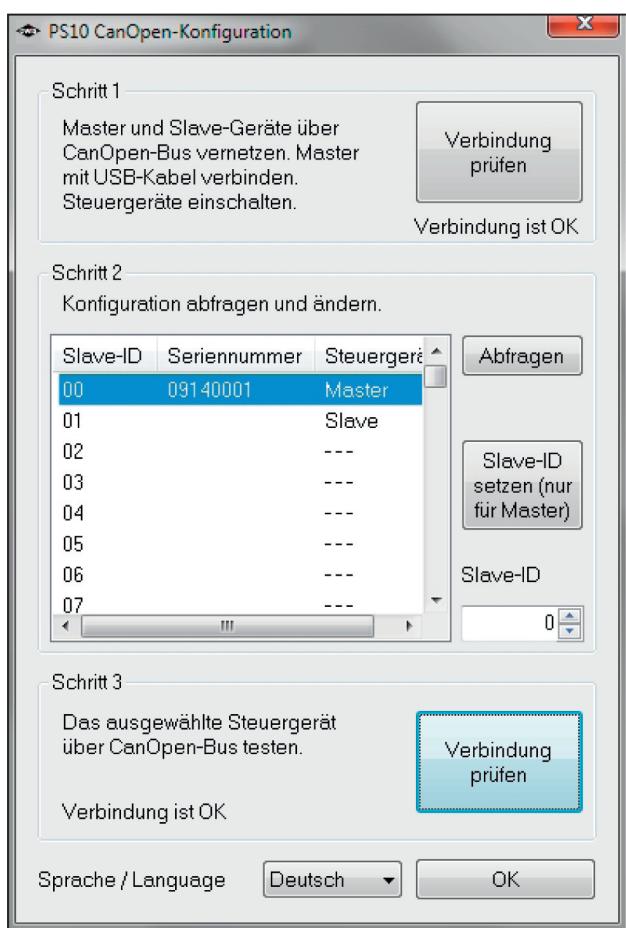
Das Ansprechen der vernetzten Steuerungen erfolgt über Master/Slave-Adressierung (ID). Die über USB vernetzte Steuerung wird automatisch zum Master. Die jeweiligen ID's sind vorkonfiguriert und können dem Abnahmeprotokoll der jeweiligen Steuerung (siehe „Slave ID“) entnommen werden.

Sollte eine Änderung der Slave-ID notwendig sein, starten Sie bitte die Anwendung „PS 10-32 CANconfig.exe“ aus dem Verzeichnis „..\OWISoft\Application\System“.

Für diese Änderung muss die entsprechende Steuerung hierfür als Master (über USB-Port verbunden) angesprochen werden.

In „Schritt 1“ werden die Slave-ID's von 0-99 gescannt und die angeschlossenen Steuerungen jeweils angezeigt.

In „Schritt 2“ kann nun durch Auswahl des Master-Slave die gewünschte ID-Nummer gewählt und gesetzt werden.



## 12. Fehlerüberwachung

### 12.1 Endschalter

Die PS 10-32 besitzt vier Endschaltereingänge, davon jeweils zwei Eingänge für Limit-Schalter (MINSTOP, MAXSTOP) und zwei für Bremschalter (MINDEC, MAXDEC), sowie Auswertemöglichkeit für einen Referenzschalter.

Einer der vier Schalter ist als Referenzschalter definiert.

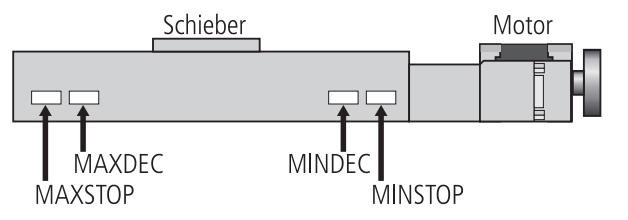
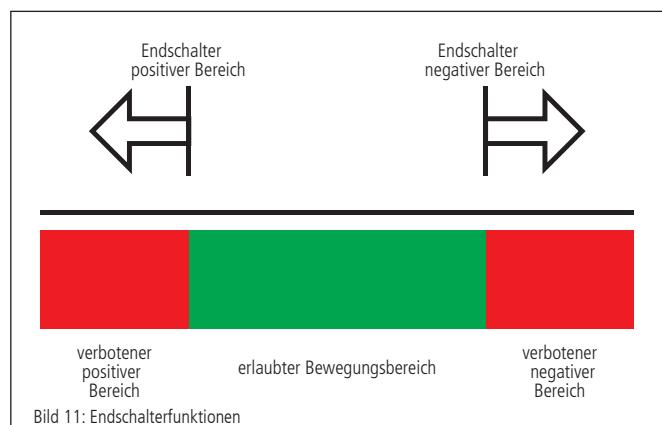


Bild 10: Endschalterkennzeichnung

Die Endschalter in negativer Fahrtrichtung (Bewegung des Schiebers zum Motor hin) werden mit MINDEC und MINSTOP bezeichnet. Die Endschalter in positiver Fahrtrichtung (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): Fals die aktuelle Fahrfrequenz, von der aus gestoppt wurde, höher gewesen ist als die Start-Stop-Frequenz des Systems, führt dies auf Grund der kinetischen Energie im System dazu, dass der Motor noch eine Bewegung ausführt. Dies kann von der Steuerung nicht erfasst werden, so dass der angezeigte Positionswert falsch ist. Eine Referenzfahrt ist nötig, um die Motorschritte wieder mit der angezeigten Position übereinstimmen zu lassen.

2. 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. Schließerkontakt nach Masse, d.h. offen in nicht betätigtem Zustand). Ein gesetztes Bit (Standardkonfiguration) bedeutet, dass der jeweilige Schalter „high“ aktiv ist (z.B. Öffnerkontakt nach Masse, d.h. geschlossen in nicht betätigtem Zustand).

Die Endschaltereingänge arbeiten standardmäßig mit 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 gerätintern vorgesehen sind.

### **Wiederinbetriebnahme nach Achsenfehler**

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

1. Initialisierung mittels Befehl:

INIT<n>

2. Freifahren des Limit-Schalters mittels Befehl:

EFREE<n>

## **12.2 Endstufen-Fehlerüberwachung**

Die Endstufe meldet ihren Status 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.

## **12.3 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) 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.19), 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.

## 13. Hinweise zum Aufbau einer eigenen Applikationssoftware

Eine PS 10-32 Applikation besteht allgemein aus einem Initialisierungssteil, welcher die erforderlichen Achsparameter für die zu verwendende Achse  $<n>$  setzt und die Achse einschaltet, einer Schleife, die eine Referenzfahrt für die Achse durchführt, und dem eigentlichen Anwenderprogramm, welches die vom Anwender gewünschte Funktionalität beinhaltet.

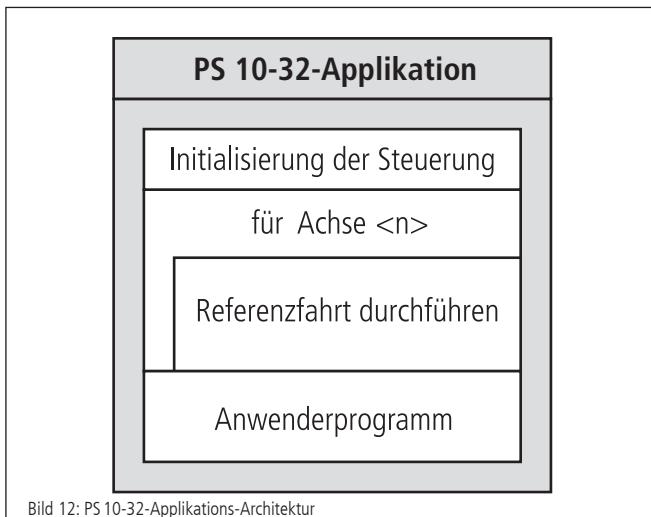


Bild 12: PS 10-32-Applikations-Architektur

Die Initialisierung der gewünschten Achse 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.

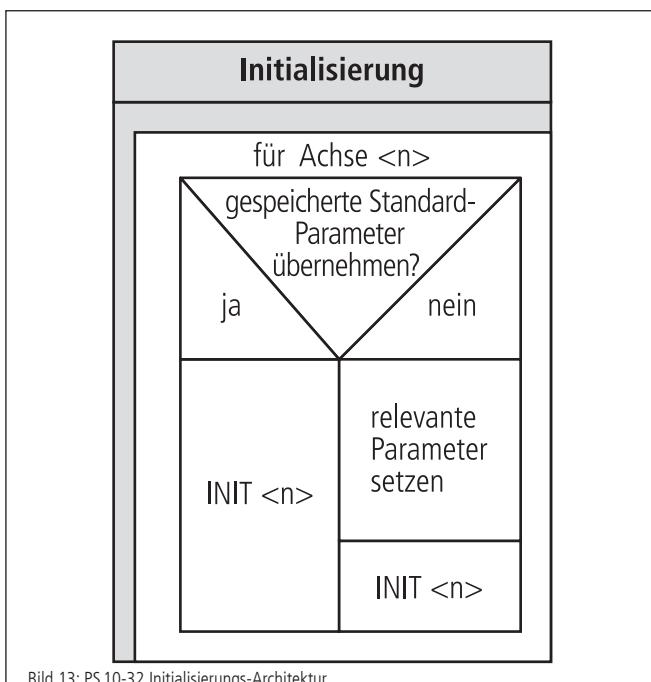


Bild 13: PS 10-32 Initialisierungs-Architektur

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

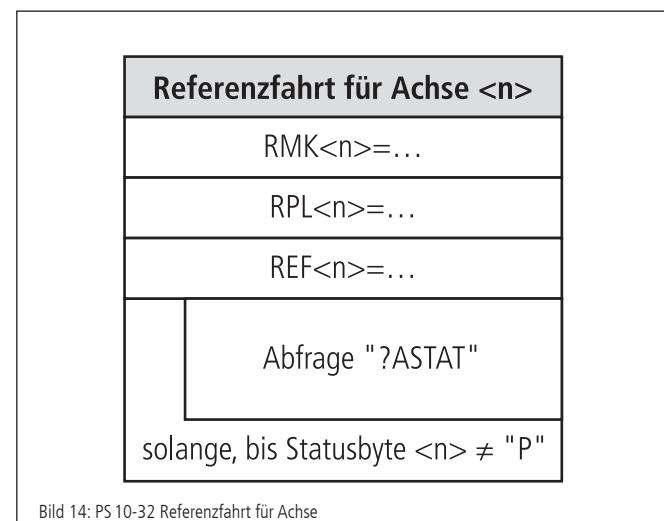


Bild 14: PS 10-32 Referenzfahrt für Achse

Zwischen zwei einzelnen Befehlen, die zur PS 10-32 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 Softwarepaket OWISoft (inklusive SDK und DLL) erleichtert die Inbetriebnahme wesentlich, da häufig verwendete Befehlsfolgen bereits als Funktionen bzw. Prozeduren zusammengefasst sind, und der erforderliche Laufzeit-abgleich ebenfalls implementiert ist.

## 14. Befehlssatz der PS 10-32

Generelles zum Format der Befehle:

Jeder Befehl wird über die USB-Schnittstelle 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 3 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.

Die Daten im Arbeitsspeicher werden mit dem Ausschalten der Steuerung gelöscht. Nach dem erneuten Einschalten werden die Parameter aus dem EEPROM geladen. Mit dem Befehl SAVEPARA kann nach dem Verändern der Konfiguration die Änderung resident abgespeichert werden.

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

<n> = Achsennummer = 1  
<uv> = Zahlenwert ohne Vorzeichen  
<sv> = Zahlenwert mit Vorzeichen

### 14.1 CANopen-Vernetzung

Das Protokoll auf der USB-Schnittstelle wurde um die Slave-Adresse erweitert. Diese wird in dezimaler Form immer mit zwei Stellen den ASCII-Befehlen vorangestellt:

64?VERSION	01?VERSION
64PVEL1 = 100000	01PVEL1=100000

Die Kommandos, die über USB empfangen werden, überprüft die PS 10-32 zunächst auf die Slave-Adresse. Ist keine Slave-Adresse vorangestellt, so wird das Kommando direkt bearbeitet. Wird eine Adresse vorangestellt, wird zuerst auf die eigene Adresse überprüft. Ist es die eigene, so wird das Kommando ebenfalls sofort ausgeführt. Ist es eine fremde Adresse, so wird das Kommando umgeformt und es erfolgt mit dem CANopen-Protokoll ein Zugriff auf das Objekt-Verzeichnis des entsprechenden PS 10-32 Slave, wodurch das Kommando dann auf der Slave-PS 10-32 ausgeführt wird. Die Antwort wird von dieser Slave-PS 10-32 wieder zurück übertragen und von der Master-PS 10-32 an den PC weitergegeben.

Als Übertragungsrate auf dem CANopen-Bus wird fest 500kBit/s verwendet.

Eine bestehende USB-Verbindung legt dabei den Master- oder den Slave-Betrieb auf der CANopen Seite fest. Besteht eine USB-Verbindung, so verhält sich diese PS 10-32 wie ein CANopen Master, besteht keine USB-Verbindung, so verhält sich die PS 10-32 wie ein CANopen Slave. Bei einer Vernetzung mehrerer PS 10-32 darf also nur eine per USB mit dem PC verbunden sein.

**Hinweis:**

Das Verbinden der PS 10-32 zum Computer oder das Trennen von dem PC sollte nur im ausgeschaltetem Zustand (der PS 10-32) erfolgen.

# Anhang

## I Befehlstabelle

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Allgemeine Statusabfragen	?ASTAT	<p>Statusabfrage der Achse. Für die 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      „V“ = Achse arbeitet im Geschwindigkeitsmodus      „P“ = Achse fährt auf Referenzposition      „F“ = Achse fährt einen Endschalter frei      „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      „?“ = Fehler, unbekannter Achsenstatus</p>	?ASTAT	IIORRTJV
	?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.</p> <p>Folgende Meldungen sind möglich:</p> <p>„00 NO MESSAGE AVAILABLE“ (wird ausgegeben, wenn der Meldungspuffer ausgelesen wird, obwohl keine Meldung verfügbar ist)</p> <p>„01 PARAMETER BEFORE EQUAL WRONG“ (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpretator den Parameter vor dem Gleichheitszeichen nicht korrekt in einen Zahlenwert umwandeln konnte)</p> <p>„02 AXIS NUMBER WRONG“ (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpretator die übergebene Achsennummer nicht auswerten konnte; zulässig 1)</p> <p>„03 PARAMETER AFTER EQUAL WRONG“ (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpretator den Parameter nach dem Gleichheitszeichen nicht korrekt in einen Zahlenwert umwandeln konnte)</p> <p>„04 PARAMETER AFTER EQUAL RANGE“ (wird in den Meldungspuffer geschrieben, wenn der Befehlsinterpretator erkannt hat, dass der Parameter hinter dem Gleichheitszeichen außerhalb des zulässigen Wertebereichs liegt)</p> <p>„05 WRONG COMMAND ERROR“ (wird in den Meldungspuffer geschrieben, wenn der gesendete Befehl syntaktisch nicht korrekt war, d.h. vom Befehlsinterpretator nicht erkannt wurde)</p> <p>„06 REPLY IMPOSSIBLE“ (wird ausgegeben, wenn die Antwort nicht gesendet werden konnte, z.B., weil der Sendepuffer noch nicht leer ist)</p> <p>„07 AXIS IS IN WRONG STATE“ (wird in den Meldungspuffer geschrieben, wenn ein Fahr- oder Konfigurierungsbefehl gesendet wurde, der nicht ausgeführt werden konnte, da sich die Achse momentan in einem anderen Fahrzustand befindet)</p>	?MSG	00 NO MESSAGE...
	?ESTAT<n>	<p>Aktuellen, logischen Zustand der Endschalter und Endstufenrückmeldung der Achse auslesen:</p> <p>Bit 0 = MINSTOP      Bit 1 = MINDEC      Bit 2 = MAXDEC      Bit 3 = MAXSTOP      Bit 4 = Rückmeldung der Endstufe</p>	?ESTAT1	10101
	?ERR	<p>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.</p>	?ERR	1211
	?EMERGINP	Gibt den aktuellen Zustand des NOT-AUS-Eingangs zurück.	?EMERGINP	1

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Allgemeine Statusabfragen	?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	10
Basis-Konfiguration	MOTYPE<n>=<uv>	Motortype für die Achse einstellen: 0 = DC-Brush 1 = Schrittmotor Open Loop	MOTYPE1=0	
	?MOTYPE<n>	Motortyp für die Achse auslesen.	?MOTYPE1	0
	AMPSHNT<n>=<uv>	Shunt-Umschaltung der Endstufe einstellen. Mit diesem Befehl wird der Ausgang TTL-OUT zur MP-UNI hin eingestellt und damit die Shunt-Umschaltung eingestellt. 0 = Strombereich 1 (niedrig) 1 = Strombereich 2 (hoch)	AMPSHNT1=0	
	?AMPSHNT<n>	Shunt-Umschaltung der Endstufe abfragen.	?AMPSHNT1	0
	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=<uv>	Baudrate der seriellen Schnittstelle einstellen, erlaubte Werte sind: 9 600, 19 200, 38 400, 57600, 115 200 Diese Einstellung wird erst nach dem nächsten Reset aktiv.	BAUDRATE=9600	
	?BAUDRATE	Aktuelle Baudrate der seriellen Schnittstelle abfragen.	?BAUDRATE	9600
	COMEND=<uv>	Befehlsenderkennung einstellen: 0 = CR 1 = CR+LF 2 = LF	COMEND=0	
	?COMEND	Befehlsendekennung abfragen.	?COMEND	0
	?SERNUM	Serien-Nummer der Steuerung abfragen.	?SERNUM	09080145
	SAVEPARA	Abspeichern der globalen und der Achsen-Parameter im EEPROM. Dieser Befehl muss nach dem Verändern der Konfiguration gesendet werden, damit die Änderung resident abgespeichert wird.	SAVEPARA	
	?VERSION	Software-Version SP-Firmware auslesen.	?VERSION	PS10-V3.0-181010
	?POSERR<n>	Auslesen des aktuellen Positionsfehlers der Achse. Zurückgegeben wird die Differenz zwischen Encoder-Position und Sollposition. Kann auch "on the fly" benutzt werden um den Schleppfehler auszulesen.	?POSERR1	-15
	?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
	?AMPST<n>	Endstufenstatus abfragen: Bit 0 = Fehler Endstufentemperatur Bit 1 = Fehler Endstufe wegen NOT-AUS abgeschaltet Bit 2 = Fehler falsche Motorkodierung gelesen Bit 3 = Fehler Kurzschlußschutz hat angesprochen	?AMPST1	0
	AMPMODE<n>	Endstufenmodus einstellen. Mit diesem Kommando wird die Art der Schnellentregung für die Endstufen eingestellt: Bit 0 und Bit 1 = Entregungsart Stromchopper 00:slow, 01:15% mixed, 10:48% mixed, 11:fast Bit 2 = Blanking-Time 0 = kurz, 1 = lang Bit 3 = ExtMode, Entregung wenn Enable auf low 0 = fast, 1 = slow Bit 4 = Enable beim Nulldurchgang des Stromes wegnehmen Bit 5 = Schrittmotor Drehrichtung invertieren	AMPMODE1	
	?AMPMODE<n>	Endstufenmodus abfragen.	?AMPMODE1	3
	SLAVEID=<uv>	Slave-ID einstellen für CANopen-Vernetzung, 00 bis 99.	SLAVEID=64	
	?SLAVEID	Slave-ID abfragen.	?SLAVEID	64
	RESETMB	Reset der Hauptplatine auslösen.	RESETMB	
	ERRCLEAR	Fehlerspeicher löschen.	ERRCLEAR	

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, Anfangs-Regelparameter. Der Kurzschlußschutz der Endstufe wird zurückgesetzt.	INIT1	
	PSET<n>=<sv>	Zielposition bzw. Relativweg (ABSOL/RELAT) für die Achse setzen. Ist absolutes Positionsangabeformat 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.	PSET1=100000	
	?PSET<n>	Zielposition bzw. Relativweg für die Achse auslesen.	?PVEL1	10000
	VVEL<n>=<sv>	Sollgeschwindigkeit der Achse für Geschwindigkeitsmodus 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 der Achse starten. Die Achse fährt die neue Zielposition im Trapez-Profil an.	PGO1	
	VGO<n>	Geschwindigkeitsmodus der Achse starten.	VGO1	
	STOP<n>	Bewegung der Achse stoppen. Jegliche aktive Bewegung der Achse wird abgebrochen. Der Antrieb stoppt mit der programmierten Bremsrampe und bleibt stehen.	STOP1	
	VSTP<n>	Geschwindigkeitsmodus der Achse stoppen. Arbeitet die Achse im Geschwindigkeitsmodus, so wird dieser mit diesem Befehl beendet und die Achse gestoppt.	VSTP1	
	EFREE<n>	Endschalter der 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 nachdem, ob ein positiver oder negativer Endschalter aktiviert ist.	EFREE1	
	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	
	CNT<n>=<sv>	Aktuellen Positionszähler für die Achse setzen.	CNT1=5000	
	?CNT<n>	Aktuellen Positionszähler für die Achse auslesen.	?CNT1	5000
Positionierparameter	CRES<n>	Aktuellen Positionszähler für die Achse nullen.	CRES1	
	?VACT<n>	Aktuelle Geschwindigkeit der Achse auslesen.	?VACT1	10000
	?ENCPOS<n>	Aktuellen Positionszähler des Encoders für die Achse auslesen. Dieser Befehl liefert bei Open-Loop-Schrittmotorachsen, die aber mit Encoder betrieben werden, die aktuelle Encoder-Position zurück.	?ENCPOS1	5000
	RELAT<n>	Positionsangaben für die Achse auf „relativ“ umschalten. (= Angabe des Weges mit Vorzeichen)	RELAT1	
	ABSOL<n>	Positionsangaben für die Achse auf „absolut“ umschalten. (= Angabe der Zielposition mit Vorzeichen)	ABSOL1	
	?MODE<n>	Abfrage des aktuell eingestellten Positionsangabeformates für die Achse.	?MODE1	ABSOL
	PVEL<n>=<uv>	Maximale Positioniergeschwindigkeit für die Achse setzen. Wird für das Trapez-Profil verwendet.	PVEL1=10000	
	?PVEL<n>	Maximale Positioniergeschwindigkeit für die Achse auslesen.	?PVEL1	10000

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Achparameter	MCSTP<n>=<uv>	Mikroschrittauflösung bei Schrittmotorachse einstellen.	MCSTP1=50	
	?MCSTP<n>	Mikroschrittauflösung bei Schrittmotorachse auslesen.	?MCSTP1	50
	DRICUR<n>=<uv>	Fahrstrom bei Schrittmotorachse als ganzzahligen Prozentwert des Maximalstromes einstellen.	DRICUR1=50	
	?DRICUR<n>	Fahrstrom bei Schrittmotorachse in Prozent auslesen.	?DRICUR1	50
	HOLCUR<n>=<uv>	Haltestrom bei Schrittmotorachse in Prozent einstellen.	HOLCUR1=30	
	?HOLCUR<n>	Haltestrom bei Schrittmotorachse 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 die Achse einstellen.	FKP1=25	
	?FKP<n>	Regelparameter KP für die Achse abfragen.	?FKP1	25
	FKD<n>=<uv>	Regelparameter KD für die Achse einstellen.	FKD1=5	
	?FKD<n>	Regelparameter KD für die Achse abfragen.	?FKD1	5
	FKI<n>=<uv>	Regelparameter KI für die Achse einstellen.	FKI1=10	
	?FKI<n>	Regelparameter KI für die Achse abfragen.	?FKI1	10
	FIL<n>=<uv>	Regelparameter Integrationslimit für die Achse einstellen.	FIL1=100000	
	?FIL<n>	Regelparameter Integrationslimit für die Achse abfragen.	?FIL1	100000
	FST<n>=<uv>	Sample-Zeit für die Achse einstellen (in Mikrosekunden).	FST1=500	
	?FST<n>	Sample-Zeit für die Achse abfragen (in Mikrosekunden).	?FST1	500
	FDT<n>=<uv>	Verzögerungszeit des D-Anteils für die Achse einstellen (in Sample-Zeit-Zyklen).	FDT1=5	
	?FDT<n>	Verzögerungszeit des D-Anteils für die Achse abfragen (in Sample-Zeit-Zyklen).	?FDT1	5
	MXPOSERR<n>=<uv>	Maximalen Positionsfehler für eine Achse setzen, wird dieser Wert überschritten, so schaltet die Achse ab. Diese Abschaltung gilt nur für den Motortyp DC-Brush.	MXPOSERR1=50	
	?MXPOSERR<n>	Maximalen Positionsfehler einer Achse abfragen.	?MXPOSERR1	50
	MAXOUT<n>=<uv>	Maximalen Ausgabewert der Servoregelschleife als ganzzahligen Prozentwert einstellen. Mit diesem Befehl kann der maximale Wert für die Achse, der an den Servo-Verstärker ausgegeben wird, eingestellt werden. <b>Maximal zulässiger Wert: 99 %.</b>	MAXOUT1=95	
	?MAXOUT<n>	Maximalen Ausgabewert in Prozent auslesen.	?MAXOUT1	95
	AMPPWMF<n>=<uv>	PWM-Frequenz für Endstufe einstellen, 20000 oder 80000 ist möglich.	AMPPWMF1=20000	
	?AMPPWMF<n>	PWM-Frequenz für Endstufe abfragen.	?AMPPWMF1	20000
	PHINTIM<n>=<uv>	Phasen-Initialisierungszeit in Sample-Zeit-Zyklen einstellen.	PHINTIM1=10	
	?PHINTIM<n>	Phasen-Initialisierungszeit in Sample-Zeit-Zyklen abfragen.	?PHINTIM1	10

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Endschalterkonfiguration und Referenzfahrt	REF<n>=<uv>	Referenzfahrt mit Angabe des Referenzfahrtmodus für die 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 aktuelle Positon auf 0 setzen Modus 4 = Modus 1, zusätzlich aktuelle Positon auf 0 setzen Modus 5 = Modus 2, zusätzlich aktuelle 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	REF1=4	
	RVELS<n>=<sv>	Referenzfahrtgeschwindigkeit „langsam“ für die Achse setzen. Mit dieser Geschwindigkeit wird der Index gesucht bzw. aus dem Referenzschalter herausgefahren (vorzeichenbehaftet).	RVELS1= 2000	
	?RVELS<n>	Referenzfahrtgeschwindigkeit „langsam“ für die Achse auslesen.	?RVELS1	2000
	RVELF<n>=<sv>	Referenzfahrtgeschwindigkeit „schnell“ für die Achse setzen. Mit dieser Geschwindigkeit fährt der Antrieb auf den Referenzschalter (vorzeichenbehaftet).	RVELF1= -20000	
	?RVELF<n>	Referenzfahrtgeschwindigkeit „schnell“ für die Achse auslesen.	?RVELF1	-20000
	RDACC<n>=<uv>	Referenzfahrt-Verzögerung für die Achse einstellen. Diese Verzögerung wird benutzt, wenn der Referenzpunkt angefahren wird.	RDACC1 = 300000	
	?RDACC<n>	Referenzfahrt-Verzögerung der Achse auslesen.	?RDACC1	300000
	SMK<n>=<uv>	Endschaltermaske für die Achse setzen. Mit diesem Befehl werden die Limit-Schalter und die Brems-Schalter aktiv bzw. inaktiv gesetzt. Wird auf einen Limit-Schalter gefahren, so wird die Bewegung abrupt gestoppt und der Motor danach stromlos geschaltet. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	SMK1=0110	
	?SMK<n>	Endschaltermaske für die Achse auslesen.	?SMK1	0110
	SPL<n>=<uv>	Endschalterpolarität für die Achse setzen. Mit diesem Befehl wird der aktive Pegel für die Limit-Schalter und Brems-Schalter festgelegt. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	SPL1=1111	
	?SPL<n>	Endschalterpolarität für die Achse auslesen.	?SPL1	1111
	RMK<n>=<uv>	Referenzschaltermaske für die Achse setzen. Mit dem Befehl wird definiert, welcher der 4 Endschalter der Achse als Referenzschalter interpretiert werden soll. Es muss eine Maske mit genau einer „1“ übergeben werden. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	RMK1=0001	
	?RMK<n>	Referenzschaltermaske für die Achse auslesen.	?RMK1	0001
	RPL<n>=<uv>	Referenzschalterpolarität für die Achse setzen. Dieser Befehl definiert den aktiven Pegel des Referenzschalters. Bit-Reihenfolge: <MAXSTOP, MAXDEC, MINDEC, MINSTOP>.	RPL1=1110	
	?RPL<n>	Referenzschalterpolarität für die Achse auslesen.	?RPL1	1110
	?HYST<n>	Referenzschalterhysterese der Achse auslesen. Nach erfolgter Referenzfahrt kann mit diesem Kommando die Hysterese des Schalters ausgelesen werden. (Der Wert ist nur korrekt, wenn die Achse nicht im Referenzschalter stand)	?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.	?REFST1	1
	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

Befehlsgruppe	Kommando	Funktionsbeschreibung	Beispiel	Antwort
Ein-/Ausgänge <sup>1)</sup>	?INPUTS	Aktuellen Zustand der Eingänge auslesen (4-Bit Binärzahl).	?INPUTS	0010
	OUTPUT<uv>=<uv>	Aktuellen Zustand eines Ausgangs ändern.	OUTPUT1=0	
	?OUTPUTS	Aktuellen Zustand aller Ausgänge auslesen (5-Bit Binärzahl).	?OUTPUTS	00101
	OUTMODE=<uv>	Ausgangsbetriebsart für die Ausgänge Out1 und Out2 einstellen. Diese Ausgänge können jeweils sowohl als digitaler Ausgang als auch alternativ als PWM-Ausgang betrieben werden. Ausgangsmodus = 0 : OUT1 und OUT2 digitale Ausgänge Ausgangsmodus = 1 : OUT1 digitaler Ausgang, OUT2 PWM-Ausgang Ausgangsmodus = 2 : OUT1 und OUT2 PWM Ausgänge	OUTMODE=1	
	?OUTMODE	Ausgangsbetriebsart für die Ausgänge Out1 und Out2 abfragen.	?OUTMODE	1
	?ANIN<uv>	Analog-Eingang abfragen. Angegeben wird die Kanal-Nummer von 1 bis 4 zurückgegeben wird der gewandelte 10-Bit Wert.	?ANIN3	234
	OPWM<uv>=<uv>	PWM-Ausgang setzen. Angegeben wird die Kanal-Nummer von 1 bis 2 und der Aussteuerungswert von 0 bis 100%.	OPWM1=55	
	?OPWM<uv>	PWM-Ausgang abfragen. Angegeben wird die Kanal-Nummer von 1 bis 2 und zurückgegeben wird der zuletzt eingestellte Aussteuerungswert von 0 bis 100%.	?OPWM1	55
	HBCH<n>=<uv>	PWM-Ausgang für Haltebremse der Achse zuordnen: <Achsennummer> = <PWM-Kanal> PWM-Kanal = 0 für Haltebremsenfunktion aus	HBCH1=1	
Haltebremsesteuerung	?HBCH<n>	Zuordnung Haltebremse PWM-Kanal der Achse abfragen.	?HBCH1	1
	HBFV<n>=<uv>	Ersten PWM-Wert (zum Anziehen) bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Prozentwert>.	HBFV1=50	
	?HBFV<n>	Ersten PWM-Wert (zum Anziehen) bei der Ansteuerung der Haltebremse abfragen.	?HBFV1	50
	HBSV<n>=<uv>	Zweiten PWM-Wert (zum Halten) bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Prozentwert>.	HBSV1=20	
	?HBSV<n>	Zweiten PWM-Wert (zum Halten) bei der Ansteuerung der Haltebremse abfragen.	?HBSV1	20
	HTBI<n>=<uv>	Zeit für ersten PWM-Wert bei der Ansteuerung der Haltebremse einstellen: <Achsennummer> = <Zeit für ersten PWM-Wert in ms>	HTBI1=300	
	?HTBI<n>	Zeit für ersten PWM-Wert bei der Ansteuerung der Haltebremse abfragen.	?HTBI1	300

<sup>1)</sup> Die mit <uv> anzusprechenden Pins auf dem I/O-Stecker können der Belegungstabelle (im Anhang III) entnommen werden.

## II Relevanz der Parameter für verschiedene Motortypen

Parameter	DC-Brush	2-Phasen-Schrittmotor Open Loop
MOTYPE	+	+
FKP	+	-
FKD	+	-
FDT	+	-
FKI	+	-
FIL	+	-
FST	+	-
MAXOUT	+	+ <sup>1)</sup>
SMK	+	+
SPL	+	+
RMK	-	+
RPL	+	+
RVELF	+	+
RVELS	+	+
ACC	+	+
PVEL	+	+
FVEL	+	+
ABSOL	+	+
RELAT	+	+
AMPPWMF	+	+
MCSTP	-	+
DRICUR	+	+
HOLCUR	-	+
AMPSHNT	+	+
PHINTIM	-	+
ATOT	+	+
HBCH	(+)	(+)
HBFV	(+)	(+)
HTBI	(+)	(+)
HBSV	(+)	(+)

+ 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 Belegungstabelle

### Ein- / Ausgänge

Pinbelegung des 25-poligen D-Sub (male)

Funktion	Pin
Analogeingang 1	6
Analogeingang 2	5
Analogeingang 3	4
Analogeingang 4	3
TTL-Eingang 1	10
TTL-Eingang 2	9
TTL-Eingang 3	8
TTL-Eingang 4	7
SPS-Output 1	16
SPS-Output 2	15
SPS-Output 3	14
SPS-Output 4	13
TTL-Output 5	17
+5V, max. 300 mA Gesamtstrom	1, 2
PWM-Output 1, max. 1 A	20
PWM-Output 2, max. 1 A	21
+24V, max. 0,5 A Gesamtstrom	18, 19
GND	11, 12, 24, 25
Freigabe-Eingang + (5V) *	22
Freigabe-Eingang - (GND)	23

SubD-25 male

\* ) Freigabe der Motorenstufe erforderlich, wenn der Freigabe-Jumper JP7 gezogen ist, z.B. über Schalter (Öffner) zwischen Pin 2 und Pin 22.

### BUS-System

Pinbelegung der 9-poligen D-Sub

Funktion	Pin
GND	3, 6
CAN-H	7
CAN-L	2

DSub-9

## Universal-Motorstecker

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

Leistung	Pin	DC-Motor	Schrittmotor OL
	19	Motor +	Phase 1 +
	18	Motor -	Phase 1 -
	17	Motor +	Phase 2 +
	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 $\bar{Index}$

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	OWISid
	28	(reserviert)
	27	(reserviert)
	26	(reserviert)
	25	(reserviert)
	24	(reserviert)
	23	(reserviert)
	22	+5V
	21	GND
	20	+24V

## EU/UE Konformitätserklärung/Declaration of conformity

Wir  
We

OWIS GmbH

Im Gaisgraben 7  
79219 Staufen / Germany  
+49(0)7633/9504-0  
+49(0)7633/9504-440  
[www.owis.eu](http://www.owis.eu)  
[info@owis.eu](mailto:info@owis.eu)

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

PS 10-32

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

EN 61000-6-1:2007 mit/with EN 61000-4-2:2009, EN 61000-4-3:2011  
EN 61000-6-3:2011 mit/with EN 55014-1:2012

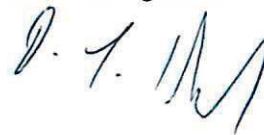
Gemäss den Bestimmungen der Richtlinie:  
Following the provisions of directive:

2014/30/EU

Ort und Datum der Ausstellung  
Place and date of issue

Staufen, 27.09.2017

Name und Unterschrift  
Name and signature



D. J. Schuh  
Leitung Vertrieb



i.A. Dr. Peter Hilgers  
Leitung Entwicklung

