MS249D

C-863.12 Mercury Controller

Benutzerhandbuch

Version: 2.1.0 Datum: 14.02.2025

Dieses Dokument beschreibt folgendes Produkt:

- **C-863.12**
  Mercury Servocontroller; 1 Achse/Kanal; Datenrekorder, ID-Chip-Erkennung; geregelte Größen: Position; 4 analoge Eingänge; 4 digitale Eingänge; 4 digitale Ausgänge

Physik Instrumente (PI) SE & Co. KG, Auf der Römerstr. 1, 76228 Karlsruhe, Deutschland
Tel. +49 721 4846-0, Fax +49 721 4846-1019, E-Mail info@pi.de, www.pi.de

Die nachfolgend aufgeführten Marken sind geistiges Eigentum der Physik Instrumente (PI) SE & Co. KG ("PI") und im Markenregister des Deutschen Patent- und Markenamts sowie teilweise auch in anderen Markenregistern unter dem Firmennamen Physik Instrumente (PI) GmbH & Co. KG eingetragen: PI®, PIC®, PICMA®, PILine®, PIFOC®, PiezoWalk®, NEXACT®, NEXLINE®, PInano®, NanoCube®, Picoactuator®, PicoCube®, PIMikroMove®, PIMag®, PIHera®

Hinweise zu Markennamen und Warenzeichen Dritter:
Microsoft® und Windows® sind eingetragene Warenzeichen oder Warenzeichen der Microsoft Corporation in den USA und/oder anderen Ländern.
EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland.
TwinCAT® ist eine eingetragene und lizenzierte Marke der Beckhoff Automation GmbH.
LabVIEW, National Instruments und NI sind Warenzeichen von National Instruments. Weder die Treibersoftware noch von PI angebotene Softwareprogramme oder andere Waren und Dienstleistungen sind verbunden mit oder gefördert von National Instruments.
Python® ist ein eingetragenes Warenzeichen der Python Software Foundation.
BiSS ist ein Warenzeichen der iC-Haus GmbH.

Bei den nachfolgend aufgeführten Bezeichnungen handelt es sich um geschützte Firmennamen, Warenzeichen oder eingetragene Warenzeichen fremder Inhaber:
Linux, MATLAB, MathWorks, FTDI
Die Verwendung dieser Bezeichnungen dient ausschließlich Identifizierungszwecken.

Von PI zur Verfügung gestellte Softwareprodukte unterliegen den Allgemeinen Softwarelizenzbestimmungen der Physik Instrumente (PI) SE & Co. KG und können Drittanbieter-Softwarekomponenten beinhalten und/oder verwenden. Weitere Informationen finden Sie in den Allgemeinen Softwarelizenzbestimmungen (https://www.physikinstrumente.com/fileadmin/user_upload/physik_instrumente/files/legal/General-Software-License-Agreement-Physik-Instrumente.pdf) und in den Drittanbieter-Softwarehinweisen (https://www.physikinstrumente.com/fileadmin/user_upload/physik_instrumente/files/legal/Third-Party-Software-Note-Physik-Instrumente.pdf) auf unserer Webseite.

© 2025 Physik Instrumente (PI) SE & Co. KG, Karlsruhe, Deutschland. Die Texte, Bilder und Zeichnungen dieses Handbuchs sind urheberrechtlich geschützt. Physik Instrumente (PI) SE & Co. KG behält insoweit sämtliche Rechte vor. Die Verwendung dieser Texte, Bilder und Zeichnungen ist nur auszugsweise und nur unter Angabe der Quelle erlaubt.

Originalbetriebsanleitung
Erstdruck: 14.02.2025
Dokumentnummer: MS249D, ASt, Version 2.1.0

Änderungen vorbehalten. Dieses Handbuch verliert seine Gültigkeit mit Erscheinen einer neuen Revision. Die jeweils aktuelle Revision ist auf unserer Webseite (https://www.physikinstrumente.de/de/) zum Herunterladen verfügbar.

Inhalt

1 Über dieses Dokument 1

1.1 Ziel und Zielgruppe dieses Benutzerhandbuchs ... 1
1.2 Symbole und Kennzeichnungen ... 1
1.3 Begriffserklärung ... 2
1.4 Abbildungen ... 3
1.5 Mitgeltende Dokumente ... 3
1.6 Handbücher herunterladen ... 4

2 Sicherheit 5

2.1 Bestimmungsgemäße Verwendung ... 5
2.2 Allgemeine Sicherheitshinweise ... 5
2.3 Organisatorische Maßnahmen ... 6

3 Produktbeschreibung 7

3.1 Produktansicht ... 7
3.1.1 Vorderwand ... 7
3.1.2 Rückwand ... 8
3.2 Typenschild ... 9
3.3 Lieferumfang ... 9
3.4 Optionales Zubehör ... 10
3.5 PC-Softwareübersicht ... 11
3.5.1 PI Software Suite ... 11
3.6 Positioniererdatenbanken ... 13
3.7 Kommunikationsschnittstellen ... 14
3.7.1 Ansteuerung von PI Systemen ... 14
3.7.2 Kommunikationssschnittstellen des C-863.12 ... 15
3.8 Funktionsprinzipien ... 16
3.8.1 Blockdiagramm ... 16
3.8.2 Motoransteuerung ... 16
3.8.3 Kommandierbare Elemente ... 17
3.8.4 Wichtige Komponenten der Firmware ... 19
3.8.5 Betriebsarten ... 20
3.8.6 Physikalische Einheiten ... 21
3.8.7 Auslösen von Bewegungen ... 22
3.8.8 Erzeugung des Dynamikprofils ... 23
3.8.9 Regelalgorithmus und weitere Stellwertkorrekturen ... 26
3.8.10 On-Target-Status ... 29
3.8.11 Referenzschaltererkennung ... 30
3.8.12 Endschaltererkennung ... 30

3.8.13 Stellweg und Verfahrbereichsgrenzen 31
3.8.14 Referenzierung 35

4 Auspacken 39

5 Installation 41

5.1 PC-Software installieren 41
5.1.1 Erstinstallation ausführen 41
5.1.2 Updates installieren 42
5.1.3 Kundenspezifische Positioniererdatenbank installieren 43
5.2 C-863.12 montieren 44
5.3 C-863.12 erden 45
5.4 Positionierer anschließen 45
5.5 PC anschließen 46
5.5.1 An RS-232-Schnittstelle anschließen 47
5.5.2 An USB-Schnittstelle anschließen 47
5.5.3 Daisy-Chain-Netzwerk aufbauen 48
5.6 Netzteil an C-863.12 anschließen 49
5.7 Analogen Joystick anschließen 49
5.8 Digitale Ein- und Ausgänge anschließen 50
5.8.1 Digitale Ausgänge anschließen 50
5.8.2 Digitale Eingänge anschließen 51
5.9 Analoge Signalquellen anschließen 52

6 Inbetriebnahme 53

6.1 Allgemeine Hinweise zur Inbetriebnahme 53
6.2 DIP-Schalter-Einstellungen anpassen 53
6.2.1 Generelle Vorgehensweise 53
6.2.2 Controlleradresse 54
6.2.3 Baudrate 55
6.3 C-863.12 einschalten 56
6.4 Kommunikation herstellen 57
6.4.1 Kommunikation über RS-232 herstellen 57
6.4.2 Kommunikation über USB herstellen 58
6.4.3 Kommunikation für vernetzten Controller herstellen 60
6.5 Bewegungen starten 64
6.6 Regelparameter optimieren 69

7 Betrieb 75

7.1 Bewegungsfehler 75
7.1.1 Verhalten bei Bewegungsfehler 75
7.1.2 Betriebsbereitschaft wiederherstellen 76

7.2 Datenrekorder ... 76
7.2.1 Datenrekorder einrichten ... 76
7.2.2 Aufzeichnung starten ... 78
7.2.3 Aufgezeichnete Daten auslesen ... 78

7.3 Digitale Ausgangssignale ... 78
7.3.1 Befehle für digitale Ausgänge ... 78
7.3.2 Triggermodus "Position Distance" einrichten ... 80
7.3.3 Triggermodus "On Target" einrichten ... 82
7.3.4 Triggermodus "Motion Error" einrichten ... 82
7.3.5 Triggermodus "In Motion" einrichten ... 83
7.3.6 Triggermodus "Position + Offset" einrichten ... 84
7.3.7 Triggermodus "Single Position" einrichten ... 85
7.3.8 Signalpolarität einstellen ... 86

7.4 Digitale Eingangssignale ... 86
7.4.1 Befehle und Parameter für digitale Eingänge ... 87
7.4.2 Digitale Eingangssignale in Makros verwenden ... 89
7.4.3 Digitale Eingangssignale als Schaltersignale verwenden ... 89

7.5 Analoge Eingangssignale ... 91
7.5.1 Befehle für analoge Eingänge ... 91
7.5.2 Analoge Eingangssignale in Makros verwenden ... 92

7.6 Joystick-Steuerung ... 92
7.6.1 Funktionsweise der Joystick-Steuerung ... 92
7.6.2 Befehle und Parameter für Joystick-Steuerung ... 93
7.6.3 Achsenbewegungen steuern ... 94
7.6.4 Joystick kalibrieren ... 95
7.6.5 Verfügbare Joystick-Geräte ... 98

7.7 Controllermakros ... 100
7.7.1 Übersicht: Makrofunktionalitäten und Beispielmakros ... 100
7.7.2 Befehle und Parameter für Makros ... 101
7.7.3 Mit Makros arbeiten ... 102
7.7.4 Controllermakros sichern und laden ... 110
7.7.5 Makrobeispiel: Synchronisation zweier Controller ... 112
7.7.6 Makrobeispiel: Bewegung per Tastendruck stoppen ... 113
7.7.7 Makrobeispiel: Joysticksteuerung mit Speicherung von Positionen ... 114
7.7.8 Makrobeispiel: Joysticksteuerung mit Änderung der Geschwindigkeit ... 117

8 GCS-Befehle 119

8.1 Schreibweise ... 119
8.2 GCS-Syntax für Syntaxversion 2.0 ... 119
8.3 Empfänger- und Senderadresse ... 122
8.4 Variablen ... 123
8.5 Befehlsübersicht ... 124
8.6 Befehlsbeschreibungen für GCS 2.0 ... 130
8.7 Fehlercodes ... 210

9 Anpassen von Einstellungen 235

9.1 Einstellungen des C-863.12 ... 235
9.2 Parameterwerte im C-863.12 ändern ... 235
9.2.1 Allgemeine Befehle für Parameter ... 236
9.2.2 Befehle für Schnellzugriff auf einzelne Parameter ... 236
9.2.3 Parameterwerte in Textdatei sichern ... 237
9.2.4 Parameterwerte ändern: Generelle Vorgehensweise ... 238
9.3 Positionierertyp anlegen oder ändern ... 240
9.4 Parameterübersicht ... 244

10 Wartung 253

10.1 C-863.12 reinigen ... 253
10.2 Firmware aktualisieren ... 253

11 Störungsbehebung 257

12 Kundendienst 263

13 Technische Daten 265

13.1 Spezifikationen ... 265
13.1.1 Datentabelle ... 265
13.1.2 Bemessungsdaten ... 267
13.1.3 Umgebungsbedingungen und Klassifizierungen ... 267
13.2 Abmessungen ... 268
13.3 Pinbelegung ... 269
13.3.1 Motor ... 269
13.3.2 I/O ... 270
13.3.3 Kabel C-170.IO für den Anschluss an die I/O-Buchse ... 271
13.3.4 Joystick ... 272
13.3.5 Kabel C-819.20Y für Joystick C-819.20 ... 273
13.3.6 RS-232 In und RS-232 Out ... 274
13.3.7 Netzteilanschluss ... 275

14 Altgerät entsorgen 277

15 Europäische Konformitätserklärungen 279

1 Über dieses Dokument

1.1 Ziel und Zielgruppe dieses Benutzerhandbuchs

Dieses Benutzerhandbuch enthält die erforderlichen Informationen für die bestimmungsgemäße Verwendung des C-863.12.

Grundsätzliches Wissen zu geregelten Systemen, zu Konzepten der Bewegungssteuerung und zu geeigneten Sicherheitsmaßnahmen wird vorausgesetzt.

1.2 Symbole und Kennzeichnungen

In diesem Benutzerhandbuch werden folgende Symbole und Kennzeichnungen verwendet:

HINWEIS

Gefährliche Situation
Bei Nichtbeachtung drohen Sachschäden.

➢ Maßnahmen, um die Gefahr zu vermeiden

INFORMATION

Informationen zur leichteren Handhabung, Tricks, Tipps, etc.

Symbol/ Kennzeichnung	Bedeutung
RS-232	Bedienelementbeschriftung auf dem Produkt (Beispiel: Buchse der RS-232 Schnittstelle)
⚠	Auf dem Produkt angebrachtes Warnzeichen, das auf ausführliche Informationen in diesem Handbuch verweist.
tart > Einstellungen	Menüpfad in der PC-Software (Beispiel: Zum Aufrufen des Menüs müssen nacheinander die Menüeinträge ***Start*** und ***Einstellungen*** gewählt werden)
`POS?`	Befehlszeile oder Befehl aus dem universellen Befehlssatz GCS von PI (Beispiel: Befehl zum Abfragen der aktuellen Achsenposition)
Device S/N	Parameterbezeichnung (Beispiel: Parameter, in dem die Seriennummer gespeichert ist)

Symbol/ Kennzeichnung	Bedeutung
5	Wert, der über die PC-Software eingegeben bzw. ausgewählt werden muss

1.3 Begriffserklärung

Begriff	Erklärung
Achse	Auch als "logische Achse" bezeichnet. Die logische Achse bildet die Bewegung der Mechanik in der Firmware des C-863.12 ab. Bei Mechaniken, die Bewegung in mehreren Richtungen erlauben (z. B. in X, Y und Z), entspricht jede Bewegungsrichtung einer logischen Achse.
Daisy Chain	Verkabelungs-Schema, bei dem ein Controller mit dem nächsten nacheinander verbunden wird (Reihenschaltungsprinzip). Dabei ist der erste Controller direkt mit dem PC verbunden. Die weiteren Controller sind jeweils mit ihren Vorgängern verbunden, so dass eine Kette entsteht. Das Signal zu und von einem Controller geht über seine Vorgänger bis zum PC hin.
Dynamikprofil	Umfasst die vom Profilgenerator des C-863.12 für jeden Zeitpunkt der Bewegung berechnete Zielposition, Geschwindigkeit und Beschleunigung der Achse. Die errechneten Werte werden "kommandierte Werte" genannt.
Firmware	Software, die auf dem Controller installiert ist.
Flüchtiger Speicher	RAM-Baustein, in dem bei eingeschaltetem Controller die Parameter gespeichert sind (Arbeitsspeicher). Die Parameterwerte im flüchtigen Speicher bestimmen das aktuelle Verhalten des Systems. In der PC-Software von PI werden die Parameterwerte im flüchtigen Speicher auch als "Active Values" bezeichnet.
GCS	PI General Command Set: Befehlssatz für Controller von PI
Inkrementeller Positionssensor	Sensor (Encoder) zur Erfassung von Lageänderungen oder Winkeländerungen. Die Signale des inkrementellen Positionssensors werden für die Rückmeldung der Achsenposition verwendet. Nach dem Einschalten des Controllers muss eine Referenzierung durchgeführt werden, bevor absolute Zielpositionen kommandiert und erreicht werden können.
PC-Software	Software, die auf dem PC installiert wird.
Permanenter Speicher	Speicherbaustein (Festspeicher, z. B. EEPROM oder Flash-Speicher), von dem beim Start des Controllers die Standardwerte der Parameter in den flüchtigen Speicher geladen werden. In der PC-Software von PI werden die Parameterwerte im permanenten Speicher auch als "Startup Values" bezeichnet.
Positionierer	An den C-863.12 angeschlossene Mechanik. Bei Positionierern mit nur einer Bewegungsachse ist die Bezeichnung "Achse" gleichbedeutend

Begriff	Erklärung
	mit "Positionierer". Positionierer, die Bewegung in mehreren Achsen erlauben, werden auch als "mehrachsige Positionierer" bezeichnet. Für diese Positionierer muss zwischen den einzelnen Achsen unterschieden werden.
Stellwert	Der Stellwert ist die Eingangsgröße für den PWM-Konverter des C-863.12. Der PWM-Konverter wandelt den Stellwert in das PWM-Signal für die Achse des Positionierers um.

1.4 Abbildungen

Zugunsten eines besseren Verständnisses können Farbgebung, Größenverhältnisse und Detaillierungsgrad in Illustrationen von den tatsächlichen Gegebenheiten abweichen. Auch fotografische Abbildungen können abweichen und stellen keine zugesicherten Eigenschaften dar.

1.5 Mitgeltende Dokumente

Alle in dieser Dokumentation erwähnten Geräte und Programme von PI sind in separaten Handbüchern beschrieben.

Beschreibung	Dokument
Kurzanleitung zur Installation und Inbetriebnahme des C-863.12	MS242EK Kurzanleitung für digitale Motorcontroller
PI GCS Treiberbibliothek zur Verwendung mit NI LabVIEW-Software	SM158E Software Manual
PI MATLAB Treiber GCS 2.0	SM155D Software-Handbuch
PI GCS 2.0 DLL	SM151E Software Manual
GCS Array Datenformatbeschreibung	SM146E Software Manual
PIMikroMove®	SM148E Software Manual
PIStages3Editor: Software zur Verwaltung der Positioniererdatenbank	SM156D Software-Handbuch
PIUpdateFinder: Aktualisierung von PI-Software	A000T0028 Benutzerhandbuch
PI Software on ARM-Based Platforms	A000T0089 Technical Note
Downloading manuals from PI: PDF-Datei mit Links auf die Handbücher für digitale Elektroniken und Software von PI. Wird zusammen mit der PI-Software ausgeliefert.	A000T0081 Technical Note

1.6 Handbücher herunterladen

INFORMATION

Wenn ein Handbuch fehlt oder Probleme beim Herunterladen auftreten:

➢ Wenden Sie sich an den Kundendienst von PI (S. 263).

Handbücher herunterladen

1. Öffnen Sie die Website **www.pi.de**.
2. Suchen Sie auf der Website nach der Produktnummer (z. B. C-863.12).
3. Um die Produktdetailseite zu öffnen, wählen Sie in den Suchergebnissen das Produkt.
4. Wählen Sie ***Downloads***.

   Die Handbücher werden unter ***Dokumentation*** angezeigt. Softwarehandbücher werden unter ***Allgemeine Software-Dokumentation*** angezeigt.
5. Wählen Sie für das gewünschte Handbuch ***HINZUFÜGEN*** und dann ***ANFORDERN***.
6. Füllen Sie das Anfrageformular aus und wählen Sie ***ANFRAGE SENDEN***.

   Der Downloadlink wird an die eingegebene E-Mail-Adresse gesendet.

2 Sicherheit

2.1 Bestimmungsgemäße Verwendung

Der C-863.12 ist ein Laborgerät im Sinne der DIN EN 61010. Er ist für die Verwendung in Innenräumen und in einer Umgebung vorgesehen, die frei von Schmutz, Öl und Schmiermitteln ist.

Entsprechend seinem Aufbau ist der C-863.12 dafür vorgesehen, Positionierer von PI, die mit DC-Motoren oder integrierten PWM-Motortreibern ausgestattet sind, zu betreiben.

Der C-863.12 ist für den geregelten Betrieb mit inkrementellen Positionssensoren vorgesehen. Außerdem kann er die Referenz- und Endschaltersignale des angeschlossenen Positionierers auslesen und weiterverarbeiten.

Der C-863.12 darf nur unter Einhaltung der technischen Spezifikationen und Anweisungen in diesem Benutzerhandbuch verwendet werden. Für die Prozessvalidierung ist der Benutzer verantwortlich.

2.2 Allgemeine Sicherheitshinweise

Der C-863.12 ist nach dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln gebaut. Bei unsachgemäßer Verwendung des C-863.12 können Benutzer gefährdet werden und/oder Schäden am C-863.12 entstehen.

- Benutzen Sie den C-863.12 nur bestimmungsgemäß und in technisch einwandfreiem Zustand.
- Lesen Sie das Benutzerhandbuch.
- Beseitigen Sie Störungen, die die Sicherheit beeinträchtigen können, umgehend.

Der Betreiber ist für den korrekten Einbau und Betrieb des C-863.12 verantwortlich.

- Installieren Sie den C-863.12 in der Nähe der Stromversorgung, damit der Netzstecker schnell und einfach vom Netz getrennt werden kann.
- Verwenden Sie zum Anschließen des C-863.12 an die Stromversorgung die mitgelieferten Komponenten (Netzteil, Netzkabel).
- Wenn eine der mitgelieferten Komponenten für das Anschließen an die Stromversorgung ersetzt werden muss, verwenden Sie eine ausreichend bemessene Komponente.

2.3 Organisatorische Maßnahmen

Benutzerhandbuch

- Halten Sie dieses Benutzerhandbuch ständig am C-863.12 verfügbar.
  Die aktuellen Versionen der Benutzerhandbücher stehen auf unserer Website zum Herunterladen (S. 4) bereit.
- Fügen Sie alle vom Hersteller bereitgestellten Informationen, z. B. Ergänzungen und Technical Notes, zum Benutzerhandbuch hinzu.
- Wenn Sie den C-863.12 an Andere weitergeben, fügen Sie dieses Handbuch und alle sonstigen vom Hersteller bereitgestellten Informationen bei.
- Führen Sie Arbeiten grundsätzlich anhand des vollständigen Benutzerhandbuchs durch. Fehlende Informationen aufgrund eines unvollständigen Benutzerhandbuchs können zu Sachschäden führen.
- Installieren und bedienen Sie den C-863.12 nur, nachdem Sie dieses Benutzerhandbuch gelesen und verstanden haben.

Personalqualifikation

Nur autorisiertes und entsprechend qualifiziertes Personal darf den C-863.12 installieren, in Betrieb nehmen, bedienen, warten und reinigen.

3 Produktbeschreibung

3.1 Produktansicht

3.1.1 Vorderwand

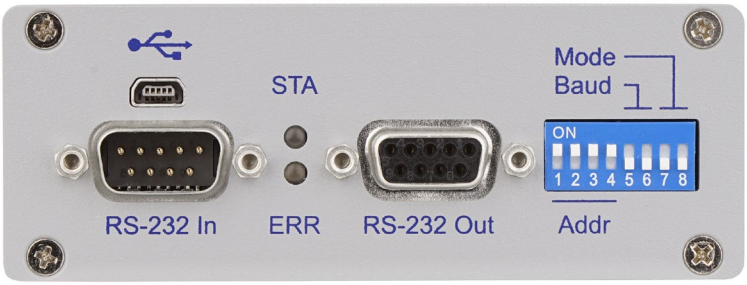


Abbildung 1: Vorderwand des C-863.12

Beschriftung	Typ	Funktion
	Mini-USB Typ B	Universal Serial Bus für Verbindung zum PC; nicht verbinden, wenn **RS-232 In** bereits verbunden ist.
RS-232 In	D-Sub 9 (m) (S. 273)	Serielle Verbindung zum PC oder zum vorhergehenden Controller in einem Daisy-Chain-Netzwerk; nicht mit dem PC verbinden, wenn die USB-Schnittstelle bereits verbunden ist.
RS-232 Out	D-Sub 9 (w) (S. 273)	Serielle Verbindung zum nachfolgenden Controller in einem Daisy-Chain-Netzwerk
STA	LED grün	Controllerstatus: ▪ Dauerhaftes Leuchten: C-863.12 ist bereit für den Normalbetrieb ▪ Blinken: C-863.12 befindet sich im Firmware-Update-Modus ▪ Aus: C-863.12 ist nicht an der Versorgungsspannung angeschlossen
ERR	LED rot	Fehleranzeige: ▪ An: Fehler (Fehlercode ≠ 0) ▪ Aus: Kein Fehler (Fehlercode = 0) Der Fehlercode kann mit dem Befehl `ERR?` abgefragt werden. Durch die Abfrage wird der Fehlercode auf null zurückgesetzt, und die LED wird ausgeschaltet.

Beschriftung	Typ	Funktion
Mode **Baud** **Addr**	8-Bit-DIP-Schalter (S. 53)	Einstellung von ▪ Geräteadresse (***Addr***) ▪ Baudrate für die Kommunikation mit dem PC (***Baud***) Die Schalter 7 und 8 (***Mode***) haben keine Funktion.

3.1.2 Rückwand

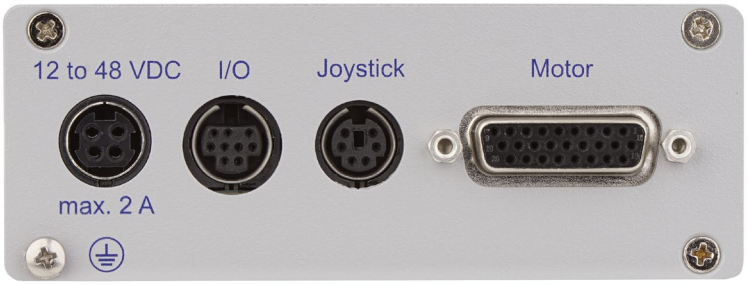


Abbildung 2: Rückwand des C-863.12

Beschriftung	Typ	Funktion
12 to 48 VDC **max. 2 A**	DC-Power-Buchse (Kycon), 4-polig (w), verriegelbar (S. 275)	Anschluss für die Versorgungsspannung
I/O	Mini-DIN, 9-polig (w) (S. 270)	Digitale Ein-/Ausgänge: ▪ Ausgänge: Triggern externer Geräte ▪ Eingänge: Verwendung in Makros oder als Schaltersignale Analoge Eingänge: ▪ Verwendung in Makros oder für Scanvorgänge
Joystick	Mini-DIN, 6-polig (w) (S. 271)	Anschluss für analogen Joystick ▪ Eingänge für Signale der Joystick-Achsen und -Tasten ▪ Ausgang für Versorgungsspannung des Joysticks
Motor	HD D-Sub 26 (w)	Anschluss des Positionierers ▪ ***Ausgang für die Ansteuersignale des Motors*** ▪ Eingang der Signale des inkrementellen Positionssensors ▪ Eingang der Signale der Endschalter und des Referenzschalters

Beschriftung	Typ	Funktion
⏚	Schraube und Zahnscheibe C-863.12 erden (S. 45)	Erdungsanschluss Wenn ein Potentialausgleich erforderlich ist, kann die Schraube mit dem Erdungssystem verbunden werden.

3.2 Typenschild

Beschriftung	Funktion
	DataMatrix-Code (Beispiel; enthält die Seriennummer)
C-863.12	Produktbezeichnung
PI	Herstellerlogo
116056789	Seriennummer (Beispiel), individuell für jeden C-863.12 Bedeutung der Stellen (Zählung von links): 1 = interne Information, 2 und 3 = Herstellungsjahr, 4 bis 9 = fortlaufende Nummer
Country of origin: Germany	Herkunftsland
⚠	Warnzeichen "Handbuch beachten!"
	Altgeräteentsorgung (S. 277)
CE	Konformitätszeichen CE
WWW.PI.WS	Herstelleradresse (Website)

3.3 Lieferumfang

Artikel	Komponente
C-863.12	Controller
C-501.24120DIN4	Weitbereichsnetzteil 24 V 120 W, Mini-DIN, 4-polig
3763	Netzkabel
000036360	USB-Kabel (Typ A auf Mini-B) zur Verbindung mit dem PC
C-815.34	RS-232-Nullmodemkabel, 3 m, 9/9-pol.
C-862.CN	Netzwerkkabel für Daisy-Chain-Netzwerk, 30 cm
000084853	4 Klebefüße für C-863.12
C-990.CD1	Datenträger mit PC-Software von PI
MS242EK	Kurzanleitung für digitale Motorcontroller

3.4 Optionales Zubehör

Artikel	Komponente
C-862.CN2	Netzwerkkabel für Daisy-Chain-Netzwerk, 3 m
C-819.20	Analoger Joystick für 2 Achsen, Details siehe "Verfügbare Joystick-Geräte" (S. 98)
C-819.20Y	Y-Kabel für 2 Controller an Joystick C-819.20
C-819.30	Analoger Joystick für 3 Achsen, Details siehe "Verfügbare Joystick-Geräte" (S. 99)
C-170.PB	Pushbutton-Box mit 4 Tasten und 4 LEDs Anschluss an Buchse **I/O** des C-863.12; liefert 4 TTL-Eingangssignale und zeigt den Status der 4 digitalen Ausgänge anhand der LEDs an.
C-170.IO	I/O-Kabel, 2 m, offenes Ende (S. 271)

Wenden Sie sich für Bestellungen an den Kundendienst (S. 263).

3.5 PC-Softwareübersicht

3.5.1 PI Software Suite

Im Lieferumfang (S. 9) des C-863.12 ist ein Datenträger mit der PI Software Suite enthalten. Einige Bestandteile der PI Software Suite werden in der nachfolgenden Tabelle beschrieben. Informationen zur Kompatibilität der Software mit PC-Betriebssystemen finden Sie in der C-990.CD1 Release News im Hauptverzeichnis des Datenträgers.

Bibliotheken, Treiber

PC-Software	Kurzbeschreibung	Empfohlene Verwendung
Dynamische Programmbibliothek für GCS	Ermöglicht die Software-Programmierung für den C-863.12 mit Programmiersprachen wie z. B. C++. Die Funktionen in der dynamischen Programmbibliothek basieren auf dem PI General Command Set (GCS).	Für Anwender, die für ihre Anwendung eine dynamische Programmbibliothek nutzen möchten. Wird für PIMikroMove® benötigt. Wird für die NI LabVIEW-Treiber benötigt.
Treiber zur Verwendung mit NI LabVIEW Software	NI LabVIEW ist eine Software für die Datenerfassung und Prozesssteuerung (von National Instruments separat zu beziehen). Die Treiberbibliothek ist eine Sammlung von Virtual-Instrument-Treibern für Elektroniken von PI. Die Treiber unterstützen das GCS.	Für Anwender, die NI LabVIEW zur Programmierung ihrer Anwendung verwenden möchten.
MATLAB-Treiber	MATLAB ist eine Entwicklungsumgebung und Programmiersprache für numerische Berechnungen (von MathWorks separat zu beziehen). Der PI MATLAB-Treiber besteht aus einer MATLAB-Klasse, die in jedes beliebige MATLAB-Skript eingebunden werden kann. Diese Klasse unterstützt das GCS. Der PI MATLAB-Treiber benötigt keine zusätzlichen MATLAB-Toolboxen.	Für Anwender, die MATLAB zur Programmierung ihrer Anwendung verwenden möchten.
USB-Treiber	Treiber für die USB-Schnittstelle	Für Anwender, die den Controller über die USB-Schnittstelle an den PC anschließen möchten.

Anwender-Software

PC-Software	Kurzbeschreibung	Empfohlene Verwendung
PIMikroMove®	Grafische Benutzerschnittstelle für Windows, mit der der C-863.12 und andere Controller von PI bedient werden können: ▪ Das System kann ohne Programmieraufwand gestartet werden ▪ Grafische Darstellung der Bewegungen im geregelten und ungeregelten Betrieb ▪ Makrofunktionalität zum Abspeichern von Befehlsfolgen auf dem PC (Hostmakros) ▪ Komplette Umgebung für die Befehlseingabe, zum Ausprobieren von verschiedenen Befehlen PIMikroMove® verwendet die dynamische Programmbibliothek zur Kommandierung des Controllers.	Für Anwender, die einfache Automatisierungsaufgaben ausführen oder ihre Ausrüstung vor oder anstelle der Programmierung einer Anwendung testen möchten. Ein Logfenster mit Anzeige der gesendeten Befehle ermöglicht auch das Erlernen der Befehlsverwendung.
PITerminal	Terminalprogramm, das für nahezu alle PI-Controller verwendet werden kann.	Für Anwender, die die Befehle des GCS direkt an den Controller senden möchten.
PIStages3Editor	Programm zum Öffnen und Editieren von Positioniererdatenbanken des Formats .db.	Für Anwender, die sich intensiv mit den Inhalten der Positioniererdatenbanken auseinandersetzen möchten.
PIUpdateFinder	Überprüft die auf dem PC installierte Software von PI. Wenn auf dem PI-Server aktuellere Versionen der PC-Software vorhanden sind, wird das Herunterladen angeboten.	Für Anwender, die die PC-Software aktualisieren möchten.
PIFirmwareManager	Programm zur Unterstützung des Anwenders bei der Aktualisierung der Firmware des C-863.12.	Für Anwender, die die Firmware aktualisieren möchten.

3.6 Positioniererdatenbanken

Sie können in der PC-Software von PI den für Ihren Positionierer geeigneten Parametersatz aus einer Positioniererdatenbank auswählen. Die PC-Software überträgt die Werte des ausgewählten Parametersatzes in den flüchtigen Speicher des Controllers.

Dateiname	Beschreibung
PISTAGES3.DB	Enthält im Auslieferungszustand Parametersätze für Standardpositionierer von PI und PI miCos; wird bei der Installation der PC-Software automatisch auf dem PC gespeichert Neue Parametersätze können angelegt, editiert und gespeichert werden (S. 240).
<Produkt>.db z. B.: M-xxxxxxx.db	Enthält den Parametersatz für einen kundenspezifischen Positionierer. Damit der Parametersatz in der PC-Software ausgewählt werden kann, muss er zuvor in PISTAGES3.DB eingefügt werden, siehe "Kundenspezifische Positioniererdatenbank installieren" (S. 43).

Parameter, die aus der Positioniererdatenbank geladen werden, sind in der Parameterübersicht (S. 244) farbig markiert.

Weitere Informationen zur Positioniererdatenbank finden Sie in den Handbüchern zum PIStages3Editor und zur PI GCS-Programmbibliothek.

INFORMATION

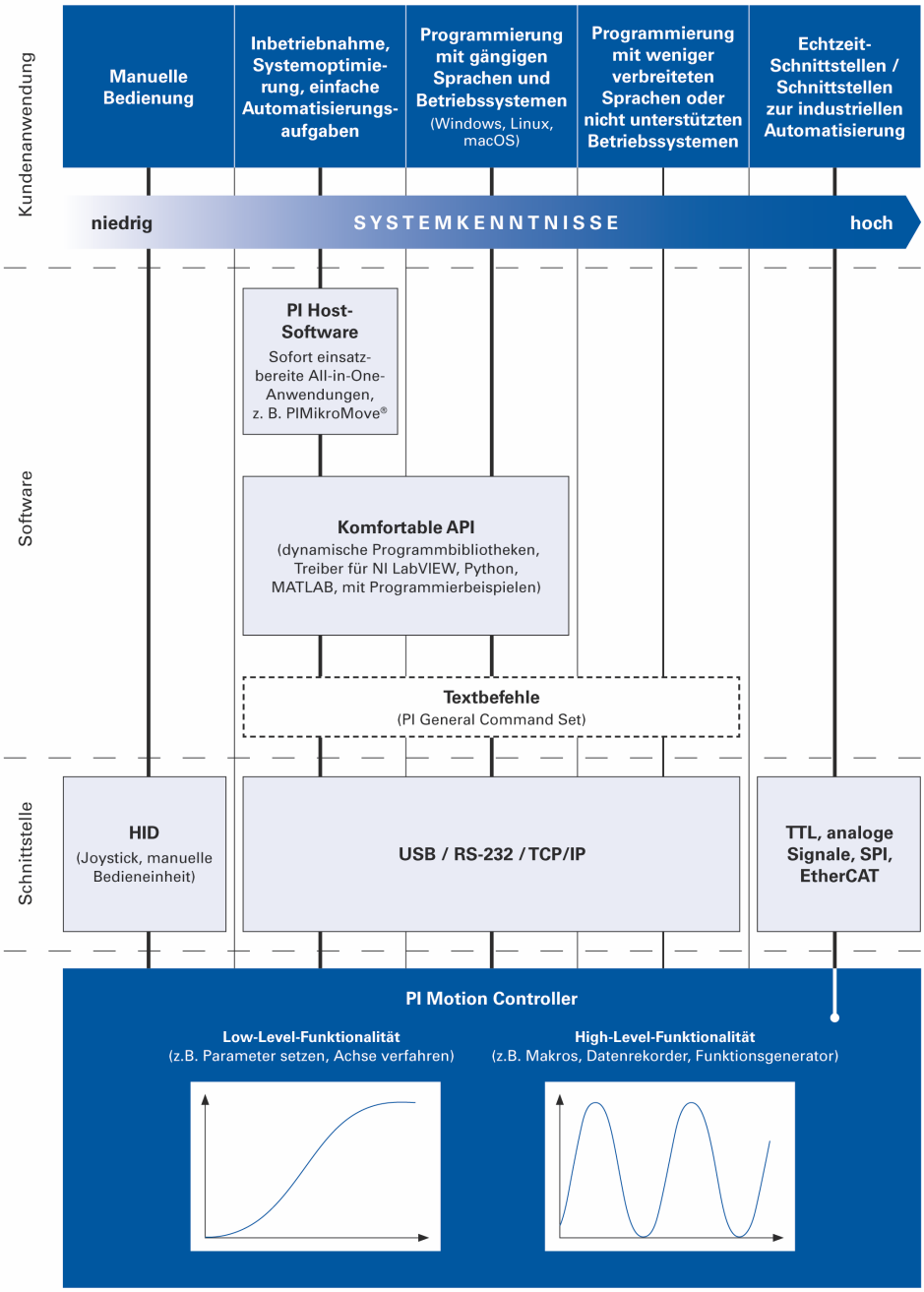
Wenn die Positioniererdatenbanken pistages2.dat und pimicosstages2.dat auf Ihrem PC vorhanden sind:

Positioniererdatenbanken im Format .dat werden nur aus Kompatibilitätsgründen installiert und für den in diesem Handbuch beschriebenen C-863.12 **nicht** verwendet.

3.7 Kommunikationsschnittstellen

3.7.1 Ansteuerung von PI Systemen

Systeme von PI können grundsätzlich wie folgt angesteuert werden:



3.7.2 Kommunikationssschnittstellen des C-863.12

Der C-863.12 kann von einem PC mit ASCII-Befehlen gesteuert werden. Der Anschluss an den PC kann über eine direkte Verbindung oder über ein Daisy-Chain-Netzwerk erfolgen. Für die direkte Verbindung zum PC können folgende Schnittstellen des C-863.12 verwendet werden:

- Serielle RS-232-Verbindung
- USB-Verbindung

Es darf immer nur eine von beiden Schnittstellen an den PC angeschlossen sein.

Standard-Kommunikationseinstellungen

Schnittstelle	Eigenschaft	Standardwert
RS-232	Baudrate	115200 Einstellungen der DIP-Schalter 5 und 6; siehe "Baudrate" (S. 55) Sonstiges: 8 Daten-Bits und 1 Stopp-Bit, ohne Parität; interne Zwischenspeicher erfordern keinen Handshake

INFORMATION

Für die USB-Schnittstelle wird im C-863.12 ein USB-UART-Modul (FTDI) verwendet. Wenn der C-863.12 über den USB-Anschluss verbunden und eingeschaltet ist, wird die USB-Schnittstelle in der PC-Software deshalb auch als COM-Port angezeigt. Der C-863.12 verwendet für diese Schnittstelle die Baudrate 115200.

Daisy-Chain-Netzwerk

Über ein Daisy-Chain-Netzwerk können bis zu 16 Controller über einen einzigen RS-232- oder USB-Anschluss an den PC angeschlossen werden. Die Vernetzung erfolgt in Reihe. Siehe auch "Begriffserklärung" (S. 2).

3.8 Funktionsprinzipien

3.8.1 Blockdiagramm

Der C-863.12 steuert die Bewegung der logischen Achse eines Positionierers. Das nachfolgende Blockdiagramm zeigt, wie der C-863.12 das Ausgangssignal für die verbundene Achse erzeugt:

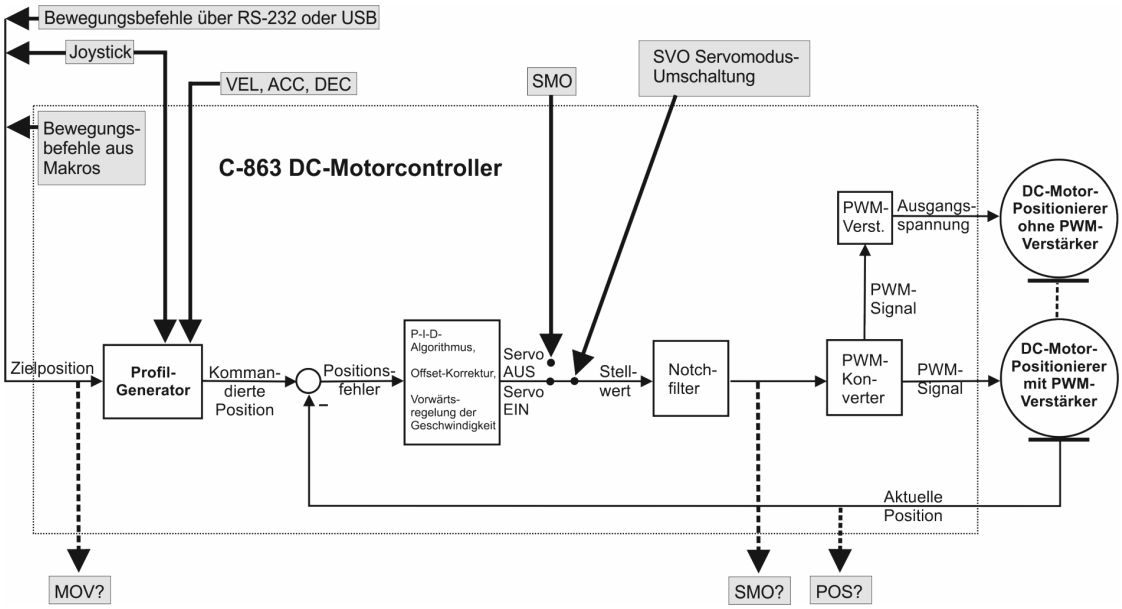


Abbildung 3: C-863.12: Erzeugung des Stellwerts

Der C-863.12 unterstützt sowohl Positionierer mit PWM-Verstärker als auch Positionierer ohne PWM-Verstärker (S. 45). Für die Rückmeldung der aktuellen Position muss der Positionierer mit einem Encoder mit A/B-Quadratur-Signalübertragung ausgestattet sein.

3.8.2 Motoransteuerung

Die Ausgangsspannung des C-863.12 ist maximal so hoch wie seine Versorgungsspannung. Um Motorschäden durch zu hohe Betriebsspannung zu vermeiden, skaliert der C-863.12 seine Ausgangsspannung auf den angeschlossenen Motor.

Details zur Skalierung:

- Der C-863.12 vergleicht seine aktuelle Versorgungsspannung (maximal 48 V) mit dem Wert des Parameters 0x7C (***Maximum Motor Output (V)***), der die maximal zulässige Betriebsspannung des Motors angibt (je nach Motortyp maximal 48 V; siehe die Dokumentation des Positionierers).
- Abhängig vom Ergebnis des Vergleichs skaliert der C-863.12 den Stellwert für den PWM-Konverter (siehe Blockdiagramm (S. 16)) und damit auch die Ausgangsspannung.

Beispiel:

Der C-863.12 wird mit einem 24-V-Netzteil betrieben. Der Motor des angeschlossenen Positionierers ist für eine maximale Betriebsspannung von 12 V ausgelegt, d.h. der Parameter

0x7C hat den Wert 12. Das PWM-Signal hat dann ein Tastverhältnis von maximal 50 %, d.h. die Ausgangsspannung beträgt maximal 12 V.

INFORMATION

Wenn der Positionierer mit einem PWM-Verstärker ausgestattet ist, der über ein separates Netzteil versorgt wird:

- Um die optimale Motorleistung zu erreichen, verwenden Sie für den C-863.12 ein Netzteil, das die gleiche Ausgangsspannung liefert wie das Netzteil des PWM-Verstärkers.

Mit den folgenden Parametern wird der C-863.12 für die maximal zulässige Betriebsspannung des Motors konfiguriert:

Parameter	Beschreibung und mögliche Werte
Maximum Motor Output (V) 0x7C	Maximal zulässige Betriebsspannung des Motors 0 bis 48 V
Maximum Motor Output 0x9	Maximal zulässiger Betrag des Stellwerts (dimensionslos) 0 bis 32767 Der Wert 32767 (Standard) entspricht der maximal zulässigen Betriebsspannung des Motors, die durch den Wert des Parameters 0x7C vorgegeben wird. Der Standardwert des Parameters 0x9 sollte nicht geändert werden.

3.8.3 Kommandierbare Elemente

Die folgende Tabelle enthält die mit den Befehlen des GCS (S. 130) kommandierbaren Elemente.

Element	Anzahl	Kennung	Beschreibung
Logische Achse	1	1 (änderbar)	Die logische Achse bildet die Bewegung des Positionierers in der Firmware des C-863.12 ab. Sie entspricht einer Achse eines linearen Koordinatensystems. Alle Befehle zur Bewegung eines Positionierers beziehen sich auf logische Achsen. In der Firmware des C-863.12 werden Bewegungen für logische Achsen kommandiert (d. h. für die Bewegungsrichtungen eines Positionierers). Im geregelten Betrieb stehen z. B. die Bewegungsbefehle `MOV` und `MVR` zur Verfügung. Der Bewegungsbefehl für den ungeregelten Betrieb lautet `SMO`. Die Achsenkennung kann mit dem Befehl `SAI?` abgefragt und mit dem Befehl `SAI` geändert werden. Sie kann aus bis zu 8 Zeichen bestehen; gültige Zeichen sind 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-_

Element	Anzahl	Kennung	Beschreibung
			Die neue Achsenkennung wird nur in den flüchtigen Speicher des C-863.12 übernommen. Eine geänderte Achsenkennung kann mit dem Befehl `WPA` (S. 209) permanent im C-863.12 gespeichert werden.
Analoge Eingänge	7	1 bis 7	Die analogen Eingangsleitungen mit den Kennungen 1 bis 4 sind die Eingänge 1 bis 4 der Buchse **I/O** (S. 270). Ihre Anzahl wird mit dem Befehl `TAC?` angezeigt, und ihre Werte können mit dem Befehl `TAV?` abgefragt werden. Beachten Sie, dass diese Leitungen auch als digitale Eingänge verwendet werden können (siehe unten). Weitere analoge Eingangsleitungen befinden sich an der Buchse **Joystick** (S. 271). Diese Leitungen werden nicht über `TAC?` und `TAV?` ausgegeben. Die Werte aller Eingänge können mittels der Aufzeichnungsoption 81 des Befehls `DRC` aufgenommen werden.
Digitale Ausgänge	4	1 bis 4	1 bis 4 kennzeichnen die digitalen Ausgangsleitungen 1 bis 4 der Buchse **I/O** (S. 270). Weitere Informationen siehe "Digitale Ausgangssignale" (S. 78).
Digitale Eingänge	4	1 bis 4	1 bis 4 kennzeichnen die digitalen Eingangsleitungen 1 bis 4 der Buchse **I/O** (S. 270), die auch als analoge Eingänge verwendet werden können (siehe oben). Weitere Informationen siehe "Digitale Eingangssignale" (S. 86).
Joystick-Gerät Joystick-Achse Joystick-Taste	1 1 1	1 1 1	An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden. Pin 4: Kommandierung als Achse 1 von Joystick-Gerät 1 Pin 6: Kommandierung als Taste 1 von Joystick-Gerät 1 Weitere Informationen siehe "Joystick-Steuerung" (S. 92). Für die Konfiguration des Datenrekorders mit dem Befehl `DRC` gelten abweichend von den Angaben oben die folgenden Datenquellen-Kennungen: 5 = Achse 1 von Joystick-Gerät 1 6 = Taste 1 von Joystick-Gerät 1
Datenrekordertabelle	4	1 bis 4	Der C-863.12 hat 4 Datenrekordertabellen (Abfrage mit `TNR?`) mit 1024 Datenpunkten pro Tabelle.
Controlleradresse	1	1 bis 16	Die Controlleradresse kann im Bereich von 1 bis 16 mit den DIP-Schaltern auf der Vorderwand des C-863.12 gesetzt werden (S. 54). In einem Daisy-Chain-Netzwerk (S. 48) muss

Element	An-zahl	Ken-nung	Beschreibung
			jeder Controller eine eindeutige Adresse haben (S. 122).
Gesamtsystem	1	1	C-863.12 als Gesamtsystem

3.8.4 Wichtige Komponenten der Firmware

Die Firmware des C-863.12 stellt die folgenden funktionalen Einheiten bereit:

Firmware-Komponente	Beschreibung
ASCII-Befehle	Die Kommunikation mit dem C-863.12 kann mit den Befehlen des PI General Command Set (GCS; Version 2.0) geführt werden. Der GCS ist von der Hardware (Controller, angeschlossene Positionierer) unabhängig. Beispiele für die Verwendung des GCS: ▪ C-863.12 konfigurieren ▪ Betriebsart einstellen ▪ Bewegungen des Positionierers starten ▪ System- und Positionswerte abfragen Eine Liste der verfügbaren Befehle finden Sie im Abschnitt "Befehlsübersicht" (S. 124).
Parameter	Parameter spiegeln die Eigenschaften des angeschlossenen Positionierers wider (z. B. Stellweg) und bestimmen das Verhalten des C-863.12 (z. B. Einstellungen für den Regelalgorithmus oder für die Verwendung der digitalen Eingänge). Parameterwerte können geändert werden, um das System an die jeweilige Anwendung anzupassen. Weitere Informationen finden Sie im Abschnitt "Anpassen von Einstellungen" (S. 235). Der Schreibzugriff auf die Parameter des C-863.12 ist durch Befehlsebenen festgelegt. Nach dem Einschalten oder Neustart des Controllers ist die aktive Befehlsebene immer 0. Auf Befehlsebenen > 1 besteht Schreibzugriff nur für PI-Servicepersonal.
Profilgenerator und Regelalgorithmus	Im geregelten Betrieb errechnet ein Profilgenerator das Dynamikprofil. Der Positionsfehler, der sich aus der Differenz zwischen dem errechneten Dynamikprofil und der tatsächlichen Position (Rückmeldung des Sensors) ergibt, durchläuft einen P-I-D-Regelalgorithmus. Weitere Informationen finden Sie in den Abschnitten "Erzeugung des Dynamikprofils" (S. 23), "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26) und "Auslösen von Bewegungen" (S. 22).

Firmware-Komponente	Beschreibung
Datenrekorder	Der C-863.12 besitzt einen Echtzeit-Datenrekorder (S. 76). Der Datenrekorder kann verschiedene Signale (z. B. Position, analoger Eingang) aus verschiedenen Datenquellen (z. B. logische Achsen oder Eingangskanäle) aufzeichnen.
Makros	Der C-863.12 kann Makros (S. 100) speichern. Über die Makrofunktion können Befehlssequenzen festgelegt und dauerhaft im nichtflüchtigen Speicher des Geräts gespeichert werden. Ein Startup-Makro kann festgelegt werden, das bei jedem Einschalten oder Neustart des C-863.12 ausgeführt wird. Das Startup-Makro vereinfacht den Stand-Alone-Betrieb (Betrieb ohne Verbindung zum PC). Weitere Informationen finden Sie im Abschnitt "Controllermakros" (S. 100).

Die Firmware kann mit einem Hilfsprogramm aktualisiert werden (S. 253).

3.8.5 Betriebsarten

Der C-863.12 unterstützt die folgenden Betriebsarten:

Betriebsart	Beschreibung
Geregelter Betrieb (Servomodus Ein)	Ein Profilgenerator berechnet das Dynamikprofil aus den vorgegebenen Werten für Zielposition, Geschwindigkeit, Beschleunigung und Abbremsung. Der Positionsfehler, der sich aus der Differenz zwischen dem errechneten Dynamikprofil und der tatsächlichen Position (Rückmeldung des Sensors) ergibt, durchläuft einen P-I-D-Regelalgorithmus (**p**roportional-**i**ntegral-**d**ifferenzial). Außerdem können weitere Korrekturen erfolgen. Das Ergebnis ist der Stellwert für den im C-863.12 integrierten PWM-Konverter. Weitere Informationen finden Sie in den Abschnitten "Erzeugung des Dynamikprofils" (S. 23) und "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
Ungeregelter Betrieb (Servomodus Aus)	Im ungeregelten Betrieb berechnet der C-863.12 kein Dynamikprofil und wertet nicht die Signale des Positionssensors aus. Daher kann der Positionierer ungebremst an das Ende des Stellwegs fahren und trotz Endschalterfunktion auf dem mechanischen Anschlag aufprallen.

INFORMATION

Der C-863.12 ist für den geregelten Betrieb mit inkrementellen Positionssensoren vorgesehen (Servomodus Ein). Nach dem Einschalten ist standardmäßig der ungeregelte Betrieb aktiviert (Servomodus Aus).

- Fragen Sie die aktuelle Betriebsart mit den Befehlen `SVO?`, `#4` oder `SRG?` ab.
- Aktivieren Sie den geregelten Betrieb mit dem Befehl `SVO`.
- Wenn nötig, programmieren Sie ein Startup-Makro, das den C-863.12 über den Befehl `SVO` im geregelten Betrieb startet; siehe "Startup-Makro einrichten" (S. 109).

➢ Vermeiden Sie Bewegungen im ungeregelten Betrieb.

3.8.6 Physikalische Einheiten

Der C-863.12 unterstützt verschiedene Längeneinheiten für Positionsangaben. Die Anpassung erfolgt durch einen Faktor, mit dem die Impulse des inkrementellen Encoders in die gewünschte physikalische Längeneinheit umgerechnet werden. Der Umrechnungsfaktor wird mit folgenden Parametern eingestellt:

Parameter	Beschreibung und mögliche Werte
Numerator Of The Counts-Per-Physical-Unit Factor 0xE	Zähler und Nenner des Faktors für Impulse pro physikalischer Längeneinheit 1 bis 1.000.000.000 für jeden Parameter. Der Faktor für die Impulse pro physikalische Längeneinheit bestimmt die Längeneinheit für Positionsabfragen und Bewegungsbefehle im geregelten Betrieb. An den eingestellten Faktor werden automatisch die Werte aller Parameter angepasst, deren Einheit entweder die physikalische Längeneinheit selbst oder eine darauf basierende Maßeinheit ist. Der Faktor für die Impulse pro physikalische Längeneinheit hat keinen Einfluss auf die Stabilität des Regelkreises, wird aber für die Eingangs- und Ausgangsskalierung von Positionswerten verwendet.
Denominator Of The Counts-Per-Physical-Unit Factor 0xF	

Das Einheitenzeichen kann für Anzeigezwecke mit folgendem Parameter angepasst werden:

Parameter	Beschreibung und mögliche Werte
Axis Unit 0x07000601	Einheitenzeichen Maximal 20 Zeichen. Das Einheitenzeichen ist z. B. "MM", wenn der Faktor für die Impulse pro physikalische Längeneinheit mit den Parametern 0xE und 0xF so eingestellt ist, dass die Encoderimpulse in Millimeter umwandelt werden. Das Einheitenzeichen für Rotationstische lautet normalerweise "deg". Der Wert des Parameters 0x07000601 wird nicht vom C-863.12 ausgewertet, sondern nur von der PC-Software für Anzeigezwecke genutzt. Beispiele: 1 Encoderimpuls = 100 nm Impulse pro physikalischer Längeneinheit: 10000:1 → Einheitenzeichen: mm 1 Encoderimpuls = 0,254 mm Impulse pro physikalischer Längeneinheit: 100:1 → Einheitenzeichen: Zoll

3.8.7 Auslösen von Bewegungen

Bewegungen im geregelten Betrieb

Auslöser der Bewegung	Befehle	Beschreibung
Bewegungsbefehle, gesendet von der Befehlszeile oder durch die PC-Software	MOV, MVR	Bewegung zu absoluter oder relativer Zielposition
	GOH	Bewegung zur Nullposition
	STE	Startet einen Schritt und zeichnet die Sprungantwort auf
	FNL, FPL, FRF	Starten von Referenzierungsfahrten
	FED	Starten von Fahrten zu Signalflanken
Controllermakros mit Bewegungsbefehlen	MAC	Ruft eine Makrofunktion auf. Erlaubt das Aufzeichnen, Löschen und Ausführen von Makros auf dem Controller. Von der Befehlszeile können sämtliche Befehle gesendet werden, während auf dem Controller ein Makro läuft. Der Makroinhalt und Bewegungsbefehle, die von der Befehlszeile empfangen werden, können sich gegenseitig überschreiben.
	Weitere Makrobefehle und Informationen siehe "Controllermakros" (S. 100).	
Joystick-Steuerung Der Joystick steuert die Geschwindigkeit der Achse (vom Profilgenerator ausgegebene kommandierte Geschwindigkeit).	JON	Aktiviert oder deaktiviert ein Joystick-Gerät, das an den Controller angeschlossen ist. Bewegungsbefehle sind nicht zulässig, wenn die Joystick-Steuerung für die Achse aktiviert ist.
	JAX	Bestimmt die Achse, die durch einen an den Controller angeschlossenen Joystick gesteuert wird.
	Weitere Joystick-Befehle siehe "Joystick-Steuerung" (S. 92).	

INFORMATION

Absolute Zielpositionen können nur kommandiert werden, wenn die Achse zuvor referenziert wurde, siehe "Referenzierung" (S. 35).

Bewegungen im ungeregelten Betrieb

Bewegungen werden mit dem Befehl `SMO` ausgelöst, der den Stellwert für den PWM-Konverter im C-863.12 direkt festlegt.

Im ungeregelten Betrieb ist keine Joystick-Steuerung möglich.

3.8.8 Erzeugung des Dynamikprofils

Im geregelten Betrieb führt der Profilgenerator Berechnungen durch, um die Zielposition, Geschwindigkeit und Beschleunigung der Achse zu jedem Zeitpunkt der Bewegung festzulegen (Dynamikprofil). Die errechneten Werte werden kommandierte Werte genannt. Das vom Profilgenerator des C-863.12 erzeugte Dynamikprofil hängt von den Bewegungsgrößen ab, die durch Befehle (S. 130), Parameter und/oder per Joystick vorgegeben werden:

Bewegungs-größe	Befehle	Parameter	Anmerkungen
Beschleunigung (A)	`ACC` `ACC?`	Beschleunigung im geregelten Betrieb (Parameter 0xB; physikalische Längeneinheit/s²); Änderung mit dem Befehl `ACC` oder mit `SPA` / `SEP`; kann mit `WPA` gespeichert werden.	Wird durch Parameter 0x4A (Maximale Beschleunigung im geregelten Betrieb) begrenzt
Abbremsung (D)	`DEC` `DEC?`	Abbremsung im geregelten Betrieb (Parameter 0xC; physikalische Längeneinheit/s²); Änderung mit dem Befehl `DEC` oder mit `SPA` / `SEP`; kann mit `WPA` gespeichert werden.	Wird durch Parameter 0x4B (Maximale Abbremsung im geregelten Betrieb) begrenzt
Geschwindigkeit (V)	`VEL` `VEL?`	Geschwindigkeit im geregelten Betrieb (Parameter 0x49; physikalische Längeneinheit/s); Änderung mit dem Befehl `VEL` oder mit `SPA` / `SEP`; kann mit `WPA` gespeichert werden.	Wird durch Parameter 0xA (Maximale Geschwindigkeit im geregelten Betrieb) begrenzt Bei Anschluss eines Joysticks an den C-863.12 und Aktivierung des Joysticks mit dem Befehl `JON` wird ein Faktor auf die aktuelle, mit dem Befehl `VEL` eingestellte Geschwindigkeit angewendet. Weitere Informationen siehe "Joystick-Steuerung" (S. 92).

Zielposition am Ende der Bewegung	`MOV`
	`MVR`
	`GOH`
	`STE`
		-	Bei Anschluss eines Joysticks an den C-863.12 und Aktivierung des Joysticks mit dem Befehl `JON` werden die Verfahrbereichsgrenzen als jeweilige Zielposition eingestellt. Bei Deaktivierung des Joystick-Geräts wird die Zielposition auf die

Bewegungs-größe	Befehle	Parameter	Anmerkungen
			aktuelle Position für joystickgesteuerte Achsen eingestellt. Weitere Informationen siehe "Joystick-Steuerung" (S. 92). Beim Einschalten des Servomodus mit dem Befehl `SVO` oder beim Anhalten der Achsenbewegung mit den Befehlen `#24`, `STP` oder `HLT` wird die Zielposition auf die aktuelle Position eingestellt.

Der Profilgenerator des C-863.12 unterstützt ausschließlich trapezförmige Geschwindigkeitsprofile: Die Achse beschleunigt linear (auf Basis des vorgegebenen Beschleunigungswerts), bis sie die vorgegebene Geschwindigkeit erreicht. Sie bewegt sich weiter mit dieser Geschwindigkeit, bis sie linear (auf Basis des vorgegebenen Abbremsungswerts) abbremst und an der vorgegebenen Zielposition anhält.

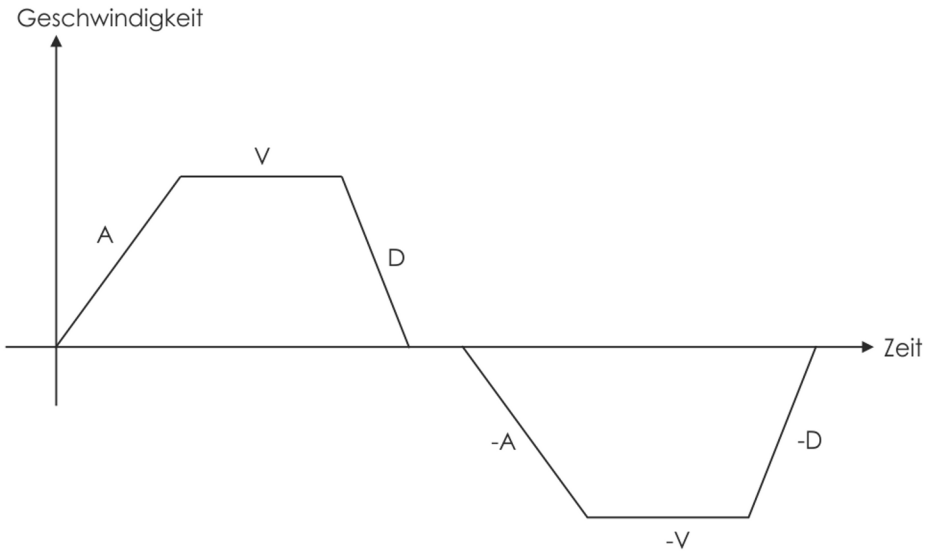


Abbildung 4: Einfaches trapezförmiges Geschwindigkeitsprofil; A = Beschleunigung, D = Abbremsung, V = Geschwindigkeit

Wenn die Abbremsung beginnen muss, bevor die Achse die vorgegebene Geschwindigkeit erreicht, wird das Profil keinen konstanten Geschwindigkeitsanteil haben, und aus dem Trapez wird ein Dreieck.

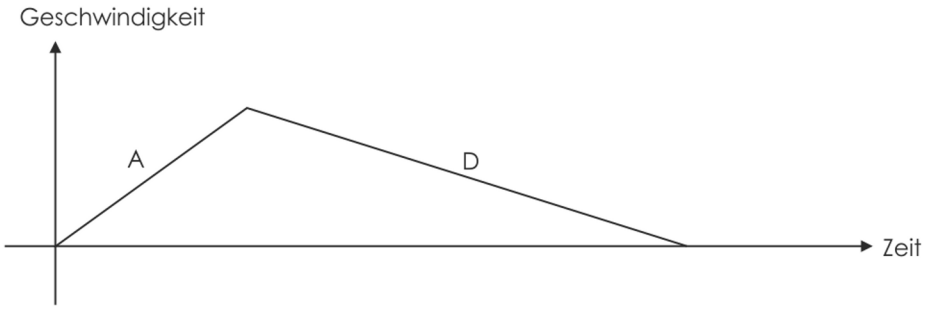


Abbildung 5: Einfaches trapezförmiges Geschwindigkeitsprofil; A = Beschleunigung, D = Abbremsung; keine konstante Geschwindigkeit

Die Flanken für Beschleunigung und Abbremsung können symmetrisch (Beschleunigung = Abbremsung) oder asymmetrisch (Beschleunigung ≠ Abbremsung) sein. Der Beschleunigungswert wird immer zu Beginn der Bewegung angewandt. Danach werden der Beschleunigungswert bei Zunahme der absoluten Geschwindigkeit und der Abbremsungswert bei Abnahme der absoluten Geschwindigkeit verwendet. Werden während der Bewegung keine Bewegungsgrößen verändert, so wird der Beschleunigungswert bis zum Erreichen der maximalen Geschwindigkeit verwendet, und der Abbremsungswert wird für die Verringerung der Geschwindigkeit bis auf null verwendet.

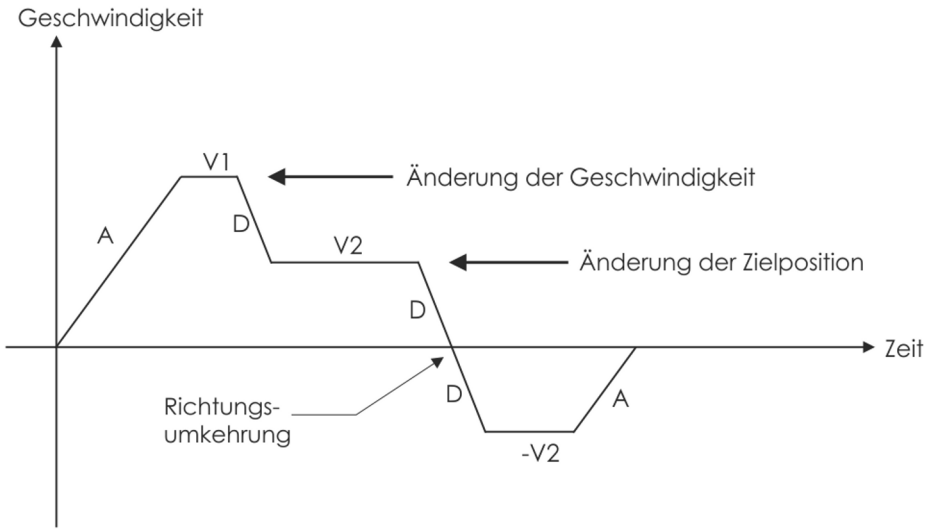


Abbildung 6: Komplexes trapezförmiges Profil mit Parameteränderungen; A = Beschleunigung; D = Abbremsung; V1, V2, -V2 = Geschwindigkeiten

Alle Bewegungsgrößen können geändert werden, während die Achse in Bewegung ist. Der Profilgenerator wird immer versuchen, innerhalb der zulässigen Bewegungsgrenzen zu bleiben, die durch die Bewegungsgrößen vorgegeben sind. Wenn die Zielposition während der Bewegung so geändert wird, dass ein Überschwingen unvermeidlich ist, wird der Profilgenerator bis zum Stillstand abbremsen und die Bewegungsrichtung umkehren, um die vorgegebene Position zu erreichen.

3.8.9 Regelalgorithmus und weitere Stellwertkorrekturen

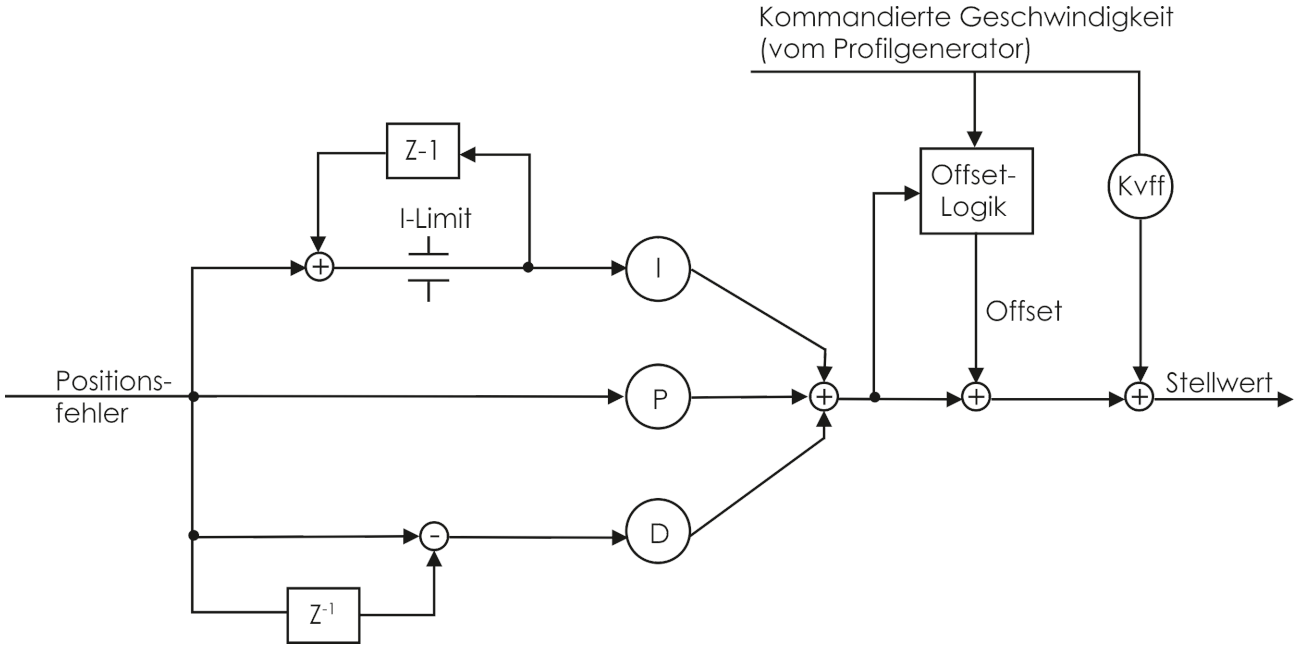


Abbildung 7: P-I-D-Algorithmus, Offset-Korrektur und Vorwärtsregelung der Geschwindigkeit (KVff); hier nicht dargestellt ist der Kerbfilter

Im geregelten Betrieb wird der Stellwert für den im C-863.12 integrierten PWM-Konverter und damit das Einschwingverhalten des Systems durch folgende Korrekturen optimiert:

- Regelalgorithmus: Der Positionsfehler, der sich aus der Differenz zwischen dem errechneten Dynamikprofil (siehe "Erzeugung des Dynamikprofils" (S. 23)) und der tatsächlichen Position (Rückmeldung des Sensors) ergibt, durchläuft einen P-I-D-Regelalgorithmus (**p**roportional-**i**ntegral-**d**ifferential).
- Dynamikprofilkorrekturen: Das errechnete Dynamikprofil kann einer Offset-Korrektur und einer Vorwärtsregelung der Geschwindigkeit unterzogen werden.

Unabhängig von der Betriebsart kann der Stellwert durch den Kerbfilter einer weiteren Korrektur unterzogen werden.

Regelalgorithmus

Der Regelalgorithmus verwendet die folgenden Regelparameter. Die optimale Einstellung der Regelparameter hängt von Ihrer Anwendung und Ihren Wünschen ab; siehe "Regelparameter optimieren" (S. 69).

Parameter	Beschreibung und mögliche Werte
P-Term 0x411	Proportionalkonstante (dimensionslos) 0 bis 32767 Ziel: Schnelle Korrektur des Positionsfehlers
I-Term	Integrationskonstante (dimensionslos)

Parameter	Beschreibung und mögliche Werte
0x412	0 bis 32767 Ziel: Reduzierung des statischen Positionsfehlers
D-Term 0x413	Differentialkonstante (dimensionslos) 0 bis 32767 Ziel: Dämpfung schneller Regelschwingungen
I-Limit 0x414	Begrenzung der Integrationskonstante (dimensionslos) 0 bis 32767
D-Term Delay (No. Of Servo Cycles) 0x71	D-Term-Verzögerung Der D-Term kann als fließender Mittelwert über mehrere Servozyklen berechnet werden. Der Parameter legt fest, wie viele Werte (d.h. Servozyklen) für die Mittelwertberechnung verwendet werden sollen.

Mit folgenden Parametern kann der Eingang des Regelalgorithmus des C-863.12 konfiguriert werden:

Parameter	Beschreibung und mögliche Werte
Numerator Of The Servo-Loop Input Factor 0x5A	Zähler und Nenner des Eingangsfaktors des Regelkreises 1 bis 1.000.000 für beide Parameter Der Eingangsfaktor des Regelkreises entkoppelt die Regelparameter von der Encoderauflösung. Der Eingangsfaktor des Regelkreises ist unabhängig vom Faktor für die Impulse pro physikalische Längeneinheit (0xE und 0xF). Zähler und Nenner des Eingangsfaktors des Regelkreises sollten nicht geändert werden.
Denominator Of The Servo-Loop Input Factor 0x5B	

Korrekturen des Dynamikprofils

Die Korrekturen des Dynamikprofils für den geregelten Betrieb können durch die nachfolgend aufgelisteten Parameter konfiguriert werden:

Parameter	Beschreibung und mögliche Werte
Motor Offset Positive 0x33	Offset für die positive Bewegungsrichtung (dimensionslos). 0 bis 32766
Motor Offset Negative 0x34	Offset für die negative Bewegungsrichtung (dimensionslos). 0 bis 32766
Motor Drive Offset 0x48	Geschwindigkeitsabhängiger Offset (dimensionslos). Wird angewendet, wenn die kommandierte Geschwindigkeit ungleich null ist (d.h. wenn das Ende des Dynamikprofils noch nicht erreicht ist). 0 bis 32766

Parameter	Beschreibung und mögliche Werte
Kvff 0x415	Vorwärtsregelung der kommandierten Geschwindigkeit 0 bis 32767 Ziel: Minimierung des Positionsfehlers

Betriebsartunabhängige Stellwertkorrekturen

Die nachfolgend aufgelisteten Parameter korrigieren den Stellwert sowohl im geregelten als auch im ungeregelten Betrieb.

Parameter	Beschreibung und mögliche Werte
Notch Filter Frequency 1 (Hz) 0x94	Frequenz des ersten Kerbfilters 40 bis 20000 Hz Der entsprechende Frequenzanteil im Stellwert wird reduziert, um unerwünschte Resonanzen in der Mechanik zu kompensieren. Eine Anpassung kann vor allem bei sehr hohen Lasten sinnvoll sein.
Notch Filter Edge 1 0x95	Anstieg der Flanke des ersten Kerbfilters (dimensionslos) 0,1 bis 10 Dieser Parameterwert sollte nicht geändert werden.

Filterung der Encodersignale

Für langsame Bewegungen des angeschlossenen Positionierers kann über den folgenden Parameter ein Filter eingestellt werden, der eventuelle Störungen im Encodersignal reduziert:

Parameter	Beschreibung und mögliche Werte
Quadrature Encoder Filter 0x03003900	Filter für das AB-Encodersignal: 0 = Filter deaktiviert (Standardeinstellung) 1 = fast: Filter für Bewegungen mit hoher Geschwindigkeit 2 = medium: Filter für Bewegungen mit mittlerer Geschwindigkeit 3 = slow: Filter für Bewegungen mit niedriger Geschwindigkeit Bei deaktiviertem Filter (0) verarbeitet der Controller die Encodersignale so schnell wie möglich. Für langsamere Bewegungen kann es sinnvoll sein, den Filter auf eine der 3 Stufen zu setzen: von 1 für schnelle Bewegungen (geringste Filterung) bis 3 für langsame Bewegungen (stärkste Filterung). Auch bei Problemen mit dem Encodersignal kann der Filter genutzt werden, um Störungen zu reduzieren. Für den normalen Betrieb muss der Filter nicht eingestellt werden.

3.8.10 On-Target-Status

Im geregelten Betrieb kann das Erreichen der Zielposition anhand des On-Target-Status überprüft werden:

- On-Target-Status = wahr (1): die Zielposition gilt als erreicht
- On-Target-Status = falsch (0): die Zielposition ist nicht erreicht

Der C-863.12 ermittelt den On-Target-Status anhand folgender Kriterien:

- Einschwingfenster um die Zielposition (Parameter 0x36)
- Verzögerungszeit für Setzen des On-Target-Status (Parameter 0x3F)

Der On-Target-Status nimmt in folgenden Fällen den Wert **wahr** an:

- Die aktuelle Position ist im Einschwingfenster und bleibt dort mindestens für die Dauer der Verzögerungszeit.
- Wenn der Wert für die Verzögerungszeit auf 0 gesetzt ist: Das Ende des Dynamikprofils ist erreicht.

Der On-Target-Status kann mit den Befehlen `ONT?`, `#4` und `SRG?` ausgelesen werden.

Im Triggermodus *On Target* (S. 82) wird der On-Target-Status der gewählten Achse am gewählten Triggerausgang ausgegeben.

Parameter	Beschreibung und mögliche Werte
Settling Time (s) 0x3F	Verzögerungszeit für das Setzen des On-Target-Status 0 bis 1,000 s
Settling Window (encoder counts) 0x36	Einschwingfenster um die Zielposition 0 bis 2^{31} Impulse des Encoders Gibt die Fenstergrenzen vor. Wenn die aktuelle Position das Einschwingfenster verlässt, gilt die Zielposition nicht mehr als erreicht. Der Parameterwert entspricht der Hälfte der Fensterbreite. Er kann nur geändert werden, wenn der Servomodus ausgeschaltet ist.

3.8.11 Referenzschaltererkennung

Der C-863.12 nimmt auf der Buchse **Motor (S. 269)** das Signal des Referenzschalters entgegen.

Mit folgenden Parametern kann die Referenzschaltererkennung durch den C-863.12 konfiguriert werden:

Parameter	Beschreibung und mögliche Werte
Invert Reference? 0x31	Soll das Referenzsignal invertiert werden? 0 = Referenzsignal nicht invertiert 1 = Referenzsignal invertiert Dieser Parameter dient zur Invertierung des Referenzsignals, dessen Quelle entweder der Referenzschalter oder ein digitaler Eingang sein kann, der anstelle des Referenzschalters verwendet wird (S. 89).
Has Reference? 0x14	Hat der Positionierer einen Referenzschalter? 0 = Kein Referenzschalter eingebaut 1 = Referenzschalter vorhanden (Signaleingang am Motoranschluss) Dieser Parameter aktiviert bzw. deaktiviert Referenzierungsfahrten zum eingebauten Referenzschalter.
Reference Signal Type 0x70	Art des Referenzsignals 0 = richtungserkennender Referenzschalter (Standard-Einstellung). Der Signalpegel ändert sich beim Überfahren des Referenzschalters. 1 = Pulssignal mit einer Pulsweite von mehreren Nanosekunden (Parameter 0x47 muss korrekt gesetzt sein). 2 = Indexpuls. Das Anfahren erfolgt über den negativen Endschalter.

Das Signal des Referenzschalters des Positionierers kann für Referenzierungsfahrten verwendet werden. Nach einer Referenzierungsfahrt zum Referenzschalter kennt der Controller die absolute Achsenposition; siehe "Referenzierung" (S. 35).

3.8.12 Endschaltererkennung

Der C-863.12 nimmt auf der Buchse **Motor (S. 269)** Endschaltersignale entgegen:

- Pin 12: positiver Endschalter
- Pin 11: negativer Endschalter

Mit folgenden Parametern kann konfiguriert werden, wie der C-863.12 die Endschalter erkennt:

Parameter	Beschreibung und mögliche Werte
Limit Mode 0x18	Signallogik der Endschalter 0 = Positiver Endschalter high-aktiv (pos-HI), Negativer Endschalter high-aktiv (neg-HI) 1 = Positiver Endschalter low-aktiv (pos-LO), neg-HI 2 = pos-HI, neg-LO 3 = pos-LO, neg-LO

Parameter	Beschreibung und mögliche Werte
Has No Limit Switches? 0x32	Hat der Positionierer Endschalter? 0 = Positionierer hat Endschalter (Signaleingänge am Motoranschluss) 1 = Positionierer hat keine Endschalter Dieser Parameter aktiviert bzw. deaktiviert das Anhalten der Bewegung an den eingebauten Endschaltern.
Use Limit Switches Only For Reference Moves? 0x77	Sollen die Endschalter nur für Referenzierungsfahrten verwendet werden? 0 = Endschalter zum Anhalten am Ende des Stellwegs und für Referenzierungsfahrten verwenden (Standard) 1 = Endschalter nur für Referenzierungsfahrten verwenden Dieser Parameter ist für die Verwendung mit Rotationstischen vorgesehen. Dieser Parameter wird nur ausgewertet, wenn Parameter 0x32 den Wert 0 hat.

Die Signale der Endschalter (auch Endlagensensoren) eines Linearpositionierers werden verwendet, um die Bewegung vor dem mechanischen Anschlag an beiden Enden des Stellwegs anzuhalten. Da die eingestellte Abbremsung dabei nicht berücksichtigt wird, besteht bei hohen Geschwindigkeiten die Gefahr, dass der Positionierer trotzdem auf dem mechanischen Anschlag aufprallt. Um dies zu vermeiden, können Verfahrbereichsgrenzen (S. 31) über Parameter des C-863.12 eingestellt werden.

Die Endschaltersignale können auch für Referenzierungsfahrten verwendet werden. Nach einer Referenzierungsfahrt zu einem Endschalter kennt der Controller die absolute Achsenposition; siehe "Referenzierung" (S. 35).

3.8.13 Stellweg und Verfahrbereichsgrenzen

Folgende Parameter des C-863.12 spiegeln den physikalischen Stellweg des Positionierers wider und definieren Verfahrbereichsgrenzen:

Parameter	Beschreibung und mögliche Werte
Maximum Travel In Positive Direction (Phys. Unit) 0x15	Verfahrbereichsgrenze in positiver Richtung (physikalische Einheit) Bezogen auf die Nullposition. Wenn dieser Wert kleiner als der Positionswert für den positiven Endschalter ist (welcher sich aus der Summe der Parameter 0x16 und 0x2F ergibt), kann der positive Endschalter nicht für Referenzierungsfahrten verwendet werden. Der Wert kann negativ sein.
Value At Reference Position (Phys. Unit) 0x16	Positionswert am Referenzschalter (physikalische Einheit) Die aktuelle Position wird auf diesen Wert gesetzt, wenn die Achse eine Referenzierungsfahrt zum Referenzschalter ausgeführt hat. Der Parameterwert wird außerdem zur Berechnung der Positionswerte verwendet, die nach Referenzierungsfahrten zu den Endschaltern gesetzt werden; dies gilt auch, wenn die Mechanik keinen Referenzschalter hat.

Parameter	Beschreibung und mögliche Werte
Distance From Negative Limit To Reference Position (Phys. Unit) 0x17	Abstand zwischen Referenzschalter und negativem Endschalter (physikalische Einheit) Wenn die Achse eine Referenzierungsfahrt zum negativen Endschalter ausgeführt hat, wird die aktuelle Position auf die Differenz zwischen den Werten der Parameter 0x16 und 0x17 gesetzt.
Distance From Reference Position To Positive Limit (Phys. Unit) 0x2F	Abstand zwischen Referenzschalter und positivem Endschalter (physikalische Einheit) Wenn die Achse eine Referenzierungsfahrt zum positiven Endschalter ausgeführt hat, wird die aktuelle Position auf die Summe der Werte der Parameter 0x16 und 0x2F gesetzt.
Maximum Travel In Negative Direction (Phys. Unit) 0x30	Verfahrbereichsgrenze in negativer Richtung (physikalische Einheit) Bezogen auf die Nullposition. Wenn dieser Wert größer als der Positionswert für den negativen Endschalter ist (welcher sich aus der Differenz der Parameter 0x16 und 0x17 ergibt), kann der negative Endschalter nicht für Referenzierungsfahrten verwendet werden. Der Wert kann negativ sein.

INFORMATION

Der C-863.12 legt Verfahrbereichsgrenzen anhand der Parameter 0x15 (***Maximum Travel In Positive Direction (Phys. Unit)***) und 0x30 (***Maximum Travel In Negative Direction (Phys. Unit)***) fest:

- Die Grenzen legen den erlaubten Verfahrbereich im geregelten Betrieb fest.
- Bewegungsbefehle werden nur ausgeführt, wenn die kommandierte Position innerhalb dieser Verfahrbereichsgrenzen liegt.
- Die Grenzen beziehen sich immer auf die aktuelle Nullposition.
- Passende Werte werden bei der Auswahl des Positionierertyps aus der Positioniererdatenbank geladen.

Beispiele

Die nachfolgenden Beispiele beziehen sich auf eine Achse eines Positionierers mit inkrementellem Sensor, Referenzschalter und Endschaltern.

Der Abstand zwischen negativem und positivem Endschalter der Achse beträgt 20 mm. Der Referenzschalter hat zum negativen Endschalter 8 mm Abstand und zum positiven Endschalter 12 mm Abstand.

Diese Schalteranordnung der Achse spiegelt sich in folgenden Parametern wider:

- Parameter 0x17: Abstand zwischen negativem Endschalter und Referenzschalter = 8 mm
- Parameter 0x2F: Abstand zwischen Referenzschalter und positivem Endschalter = 12 mm

INFORMATION

Der Schalteranordnung der Achse kann mit den Befehlen `FED` und `POS?` ermittelt werden.

Beispiel 1: Maximaler Stellweg verfügbar

Nach Referenzierungsfahrten (S. 35) soll die aktuelle Position folgende Werte haben:

- Fahrt zum negativen Endschalter (Start mit `FNL`): aktuelle Position = 0
- Fahrt zum Referenzschalter (Start mit `FRF`): aktuelle Position = 8
- Fahrt zum positiven Endschalter (Start mit `FPL`): aktuelle Position = 20

Deshalb hat Parameter 0x16, der bei Referenzierungsfahrten den Positionswert am Referenzschalter bestimmt und in die Berechnung der Positionswerte an den Endschaltern eingeht, den Wert 8.

Der Stellweg soll nicht durch Verfahrbereichsgrenzen beschränkt werden. Deshalb werden die entsprechenden Parameter wie folgt gesetzt:

- Parameter 0x15 = 20
- Parameter 0x30 = 0

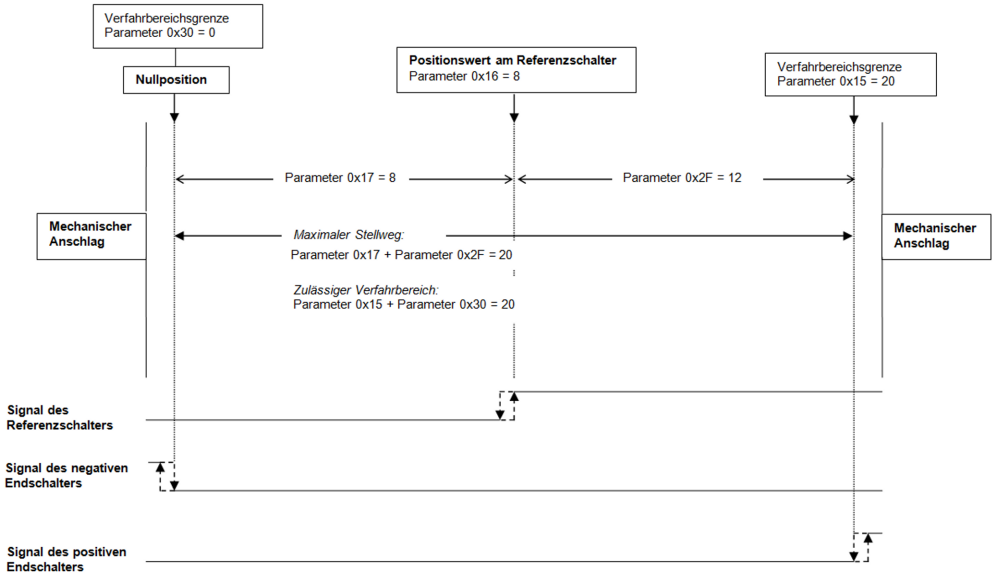


Abbildung 8: Der Stellweg der Achse wird nicht durch Verfahrbereichsgrenzen beschränkt.

Nach einer Referenzierungsfahrt der Achse zum Referenzschalter (Befehl `FRF`) liefern Abfragebefehle folgende Antworten:

- `TMN?` liefert den Wert 0
- `TMX?` liefert den Wert 20
- `POS?` liefert den Wert 8

Beispiel 2: Stellweg durch Verfahrbereichsgrenzen beschränkt

Die Nullposition soll sich ungefähr bei einem Drittel des Abstands zwischen dem negativen Endschalter und dem Referenzschalter befinden. Parameter 0x16 hat deshalb nun den Wert 5,4.

An beiden Enden des Stellwegs soll durch Festlegung von Verfahrbereichsgrenzen ein Sicherheitsabstand eingerichtet werden. Die Verfahrbereichsgrenzen werden deshalb wie folgt eingestellt:

- Parameter 0x15 = 16,4
- Parameter 0x30 = -2,1

Demnach kann die Achse jeweils ab der Nullposition 16,4 mm in positiver Richtung und 2,1 mm in negativer Richtung fahren. Die Endschalter können nicht mehr für Referenzierungsfahrten verwendet werden.

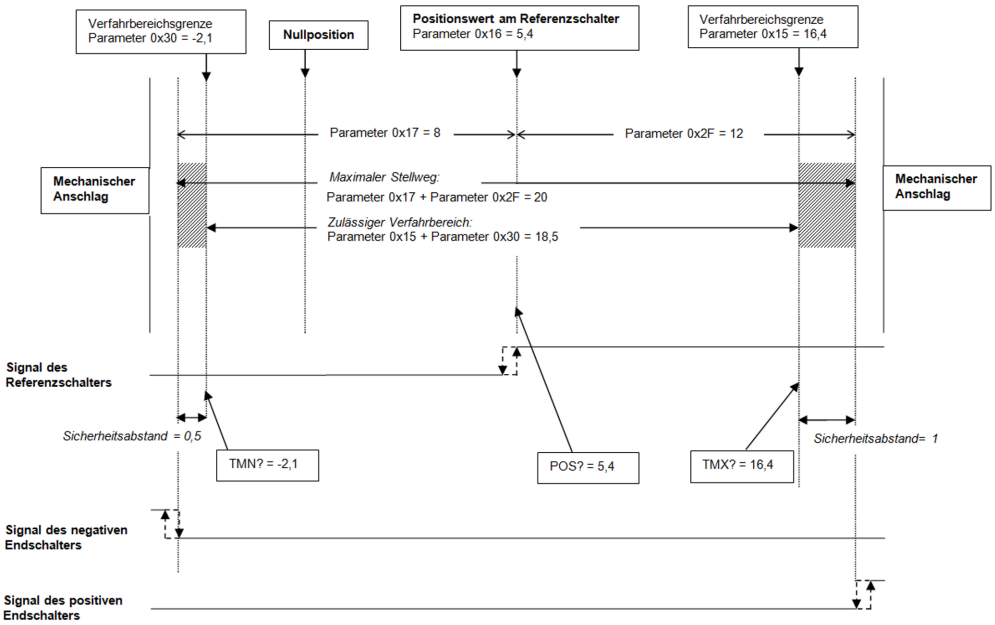


Abbildung 9: Der Stellweg der Achse wird durch Verfahrbereichsgrenzen beschränkt.

Nach einer Referenzierungsfahrt der Achse zum Referenzschalter (Befehl `FRF`) liefern Abfragebefehle folgende Antworten:

- `TMN?` liefert den Wert -2,1
- `TMX?` liefert den Wert 16,4
- `POS?` liefert den Wert 5,4

3.8.14 Referenzierung

Beim Einschalten oder Neustart kennt der Controller die absolute Position der Achse nicht. Bevor absolute Zielpositionen kommandiert und erreicht werden können, muss die Achse deshalb referenziert werden.

Die Referenzierung kann auf unterschiedliche Weise durchgeführt werden:

- **Referenzierungsfahrt** (Standard): Eine Referenzierungsfahrt bewegt die Achse zu einem fest definierten Punkt, z. B. zum Referenzschalter oder zu einem Endschalter. An diesem Punkt wird die aktuelle Position auf einen definierten Wert gesetzt. Der Controller kennt nun die absolute Achsenposition.
- **Manuelle Festlegung der absoluten Position**: Wenn diese Referenzierungsmethode mit dem Befehl `RON` (S. 186) gewählt wurde, können Sie mit dem Befehl `POS` (S. 184) die aktuelle Position der Achse an einem beliebigen Punkt auf einen beliebigen Wert setzen. Dabei wird die Achse nicht bewegt. Der Controller kennt anschließend die absolute Achsenposition.

INFORMATION

Bei der Inbetriebnahme mit PIMikroMove® erfolgt die Referenzierung standardmäßig durch eine Referenzierungsfahrt. Die Kenntnis der hier beschriebenen Befehle und Parameter ist für die Referenzierung mit PIMikroMove® nicht erforderlich.

INFORMATION

Um eine möglichst hohe Wiederholgenauigkeit der Referenzierung zu erreichen, besteht jede Referenzierungsfahrt aus folgenden Abschnitten:

1. Erste Fahrt zum ausgewählten Schalter. Die maximale Geschwindigkeit wird durch Parameter 0x49 vorgegeben (***Closed-Loop Velocity (Phys. Unit/s)***, entspricht der Einstellung mit dem Befehl VEL).
2. Stopp beim Erreichen der Schalterflanke. Je höher die Geschwindigkeit beim Anfahren war, desto weiter überfährt die Achse die Schalterflanke (Überschwingen).
3. Fahrt in die entgegengesetzte Richtung zum Ausgleichen des Überschwingens.
4. Zweite Fahrt zum ausgewählten Schalter. Die maximale Geschwindigkeit wird durch Parameter 0x50 vorgegeben (***Velocity For Reference Moves (Phys. Unit/s)***, spezielle Geschwindigkeitsvorgabe nur für Referenzierungsfahrten).
5. Stopp beim Erreichen der Schalterflanke.
6. Fahrt in die entgegengesetzte Richtung zum Ausgleichen des Überschwingens.
7. Setzen der aktuellen Position auf einen definierten Wert, die Referenzierung ist beendet.

Je geringer die Geschwindigkeit beim Anfahren des Schalters ist, desto geringer ist das Überschwingen und desto höher die Wiederholgenauigkeit. Deshalb sollte der Wert des Parameters 0x50 maximal so groß sein wie der Wert des Parameters 0x49, idealerweise jedoch wesentlich kleiner.

Die tatsächlichen Geschwindigkeiten während der Referenzierungsfahrt werden aus den Werten folgender Parameter berechnet und können niedriger ausfallen als die Maximalwerte:

- Parameter 0x49 oder 0x50
- Parameter 0x63 (***Distance Between Limit And Hard Stop (Phys. Unit)***)
- Parameter 0xC (***Closed-Loop Deceleration (Phys. Unit/s²)***)

Befehle

Folgende Befehle stehen für die Referenzierung zur Verfügung:

Befehl	Syntax	Funktion
`RON`	RON {<AxisID> <ReferenceOn>}	Wählt die Referenzierungsmethode: ▪ <ReferenceOn> = 0: Ein absoluter Positionswert kann mit POS zugewiesen werden, oder eine Referenzierungsfahrt kann mit FRF, FNL oder FPL gestartet werden. ▪ <ReferenceOn> = 1 (Standard): Eine Referenzierungsfahrt muss mit FRF, FNL oder FPL gestartet werden. Die Verwendung von POS ist nicht zulässig.
`RON?`	RON? [{<AxisID>}]	Fragt die Referenzierungsmethode ab.
`FRF`	FRF [{<AxisID>}]	Startet eine Referenzierungsfahrt zum Referenzschalter. Das Anfahren hängt vom Wert des Parameters ***Reference Signal Type*** (0x70) ab: ▪ 0 oder 1: Das Anfahren erfolgt immer von derselben Seite, unabhängig von der Achsenposition beim Senden des Befehls. ▪ 2: Das Anfahren erfolgt über den negativen Endschalter.
`FRF?`	FRF? [{<AxisID>}]	Fragt ab, ob die Achse bereits referenziert wurde. 1 = Achse ist referenziert 0 = Achse ist nicht referenziert
`FNL`	FNL [{<AxisID>}]	Startet eine Referenzierungsfahrt zum negativen Endschalter.
`FPL`	FPL [{<AxisID>}]	Startet eine Referenzierungsfahrt zum positiven Endschalter.
`POS`	POS {<AxisID> <Position>}	Setzt die aktuelle Achsenposition (löst keine Bewegung aus) und referenziert damit die Achse.

Parameter

Mit folgenden Parametern können Referenzierungsfahrten konfiguriert werden:

Parameter	Beschreibung und mögliche Werte
Closed-Loop Deceleration (Phys. Unit/s²) 0xC	Abbremsung im geregelten Betrieb Details siehe "Erzeugung des Dynamikprofils" (S. 23).
Reference Travel Direction 0x47	Standardrichtung für die Referenzierungsfahrt 0 = automatische Erkennung 1 = negative Richtung 2 = positive Richtung
Closed-Loop Velocity (Phys. Unit/s) 0x49	Geschwindigkeit im geregelten Betrieb Details siehe "Erzeugung des Dynamikprofils" (S. 23).
Velocity For Reference Moves (Phys. Unit/s) 0x50	Geschwindigkeit für Referenzierungsfahrt Gibt die maximale Geschwindigkeit für das zweite Anfahren des ausgewählten Schalters während einer Referenzierungsfahrt an. Für hohe Wiederholgenauigkeit der Referenzierung sollte dieser Wert maximal so groß sein wie der Wert des Parameters 0x49. Wenn der Wert des Parameters 0x50 auf 0 gesetzt wird, sind Referenzierungsfahrten nicht möglich.
Distance Between Limit And Hard Stop (Phys. Unit) 0x63	Abstand zwischen eingebautem Endschalter und mechanischem Anschlag Legt den maximalen Bremsweg bei Referenzierungsfahrten fest. Die tatsächlichen Geschwindigkeiten während einer Referenzierungsfahrt werden auf der Grundlage dieses Werts, der eingestellten Abbremsung (0xC) und der eingestellten Geschwindigkeiten (0x49 und 0x50) berechnet.
Distance From Limit To Start Of Ref. Search (Phys. Unit) 0x78	Abstand zwischen Endschalter und Startposition für die Bewegung zum Indexpuls.
Distance For Reference Search (Phys. Unit) 0x79	Maximale Strecke für die Bewegung zum Indexpuls Die Parameter 0x78 und 0x79 werden für Referenzierungsfahrten verwendet, wenn die beiden folgenden Bedingungen erfüllt sind: ▪ Die Referenzierungsfahrt wird mit FRF gestartet. ▪ Der Parameter ***Reference Signal Type*** (0x70) hat den Wert 2. Ablauf der Referenzierungsfahrt: 1. Die Achse bewegt sich zum negativen Endschalter. 2. Die Achse bewegt sich um die mit dem Parameter 0x78

Parameter	Beschreibung und mögliche Werte
	angegebene Strecke weg vom Endschalter. 3. Die Achse bewegt sich zum Indexpuls und legt maximal die mit dem Parameter 0x79 vorgegebene Strecke zurück.

INFORMATION

➢ Führen Sie für größtmögliche Wiederholgenauigkeit die Referenzierungsfahrt immer auf dieselbe Weise aus.

INFORMATION

Die Endschalter können nur für Referenzierungsfahrten verwendet werden, wenn der Stellweg nicht durch Verfahrbereichsgrenzen (S. 31) beschränkt wird.

INFORMATION

Sie können für Referenzierungsfahrten auch die digitalen Eingänge des C-863.12 als Quelle für das Referenzsignal, das negative Endschaltersignal oder das positive Endschaltersignal verwenden. Siehe "Digitale Eingangssignale als Schaltersignale verwenden" (S. 89) für weitere Informationen.

INFORMATION

Wenn die absolute Position der Achse mit dem Befehl `POS` manuell festgelegt wird, können Konflikte mit den Einstellungen für die Verfahrbereichsgrenzen entstehen (Parameter 0x15, Abfrage mit `TMX?`, und 0x30, Abfrage mit `TMN?`).

➢ Setzen Sie die absolute Position der Achse nur manuell, wenn die Referenzierung nicht anders möglich ist.

INFORMATION

Wenn die aktuellen Parametereinstellungen des C-863.12 in PIMikroMove® oder durch Eingabe des Befehls WPA in den permanenten Speicher geschrieben werden, gilt die Achse anschließend nicht mehr als "referenziert" (die Antwort auf FRF? ist 0).

4 Auspacken

1. Packen Sie den C-863.12 vorsichtig aus.
2. Vergleichen Sie die erhaltene Lieferung mit dem Lieferumfang laut Vertrag und mit dem Lieferschein.
3. Überprüfen Sie den Inhalt auf Anzeichen von Schäden. Bei Schäden oder fehlenden Teilen wenden Sie sich sofort an unseren Kundendienst (S. 263).
4. Bewahren Sie das komplette Verpackungsmaterial auf für den Fall, dass das Produkt zurückgeschickt werden muss.

5 Installation

5.1 PC-Software installieren

Die Kommunikation zwischen dem C-863.12 und einem PC ist zur Konfiguration des C-863.12 und zur Bewegungskommandierung mit den Befehlen des GCS notwendig. Dafür stehen verschiedene PC-Softwareanwendungen zur Verfügung.

5.1.1 Erstinstallation ausführen

Zubehör

- PC mit Betriebssystem Windows oder Linux und mindestens 30 MB freiem Speicherplatz
- Datenträger mit der PI Software Suite (im Lieferumfang)

  Informationen zur Kompatibilität der Software mit PC-Betriebssystemen finden Sie in der C-990.CD1 Release News im Hauptverzeichnis des Datenträgers.

PC-Software auf Windows installieren

1. Starten Sie den Installationsassistenten, indem Sie im Installationsverzeichnis (Hauptverzeichnis des Datenträgers) Datei ***PISoftwareSuite.exe*** aufrufen.

   Das Fenster ***InstallShield Wizard*** für die Installation der PI Software Suite wird geöffnet.
2. Folgen Sie den Anweisungen am Bildschirm.

   Die PI Software Suite umfasst unter anderem folgende Komponenten:

   – Treiber zur Verwendung mit NI LabVIEW-Software

   – Dynamische Programmbibliothek für GCS

   – PIMikroMove®

   – PC-Software zum Aktualisieren der Firmware des C-863.12

   – PIUpdateFinder zum Aktualisieren der PI Software Suite

   – USB-Treiber

PC-Software auf Linux installieren

1. Entpacken Sie das tar-Archiv aus dem Verzeichnis /Linux des Datenträgers in ein Verzeichnis auf Ihrem PC.
2. Öffnen Sie ein Terminal und wechseln Sie in das Verzeichnis, in das Sie das tar-Archiv entpackt haben.

3. Melden Sie sich als Superuser (Root-Rechte) an.
4. Um die Installation zu starten, geben Sie ./INSTALL ein.
   Achten Sie beim Eingeben des Befehls auf Groß-/Kleinschreibung.
5. Folgen Sie den Anweisungen am Bildschirm.

Sie können einzelne Komponenten zur Installation auswählen.

5.1.2 Updates installieren

Die PI Software Suite wird von PI ständig verbessert.

➢ Installieren Sie immer die neueste Version der PI Software Suite und der Positioniererdatenbank.

Voraussetzungen

✓ Aktive Verbindung zum Internet

✓ Wenn Ihr PC ein Windows-Betriebssystem verwendet:

– Sie haben das Handbuch für den PIUpdateFinder (A000T0028) von der PI Website heruntergeladen. Sie finden den Link in der Datei "A000T0081-Downloading Manuals from PI.pdf" im Ordner \Manuals auf dem Datenträger mit der PI Software Suite.

PC-Software und PISTAGES3.DB auf Windows aktualisieren

➢ Verwenden Sie den PIUpdateFinder:

– Folgen Sie den Anweisungen im Handbuch für den PIUpdateFinder (A000T0028).

PC-Software auf Linux aktualisieren

1. Öffnen Sie die Webseite https://www.physikinstrumente.de/de/produkte/software-suite (https://www.physikinstrumente.de/de/produkte/software-suite).
2. Scrollen Sie nach unten zu ***Downloads***.
3. Für ***PI Software Suite C-990.CD1***: Wählen Sie ***HINZUFÜGEN+***
4. Wählen Sie ***ANFORDERN***
5. Füllen Sie das Anfrageformular aus und senden Sie die Anfrage ab.

   Der Download-Link wird an die eingegebene E-Mail-Adresse gesendet.
6. Entpacken Sie die Archivdatei auf Ihrem PC in ein separates Installationsverzeichnis.
7. Wechseln Sie im Verzeichnis mit den entpackten Dateien in das Unterverzeichnis ***linux***.
8. Entpacken Sie die Archivdatei im Verzeichnis ***linux***, indem Sie in der Konsole den Befehl tar -xvpf <Name der Archivdatei> eingeben.
9. Melden Sie sich am PC als Superuser (Root-Rechte) an.
10. Installieren Sie das Update.

INFORMATION

Wenn Software im Bereich ***Downloads*** fehlt oder Probleme beim Herunterladen auftreten:

➢ Wenden Sie sich an den Kundendienst von PI (S. 263).

PISTAGES3.DB auf Linux aktualisieren

1. Wenden Sie sich an den Kundendienst (S. 263), um die neueste Version der Positioniererdatenbank PISTAGES3.DB zu erhalten.
2. Melden Sie sich am PC als Superuser (Root-Rechte) an.
3. Installieren Sie das Update, das Sie von unserem Kundendienst erhalten haben, auf dem PC.

5.1.3 Kundenspezifische Positioniererdatenbank installieren

Mit einem kundenspezifischen Positionierer erhalten Sie von PI einen Datenträger mit folgendem Inhalt:

- Programm Import PI CustomStage
- Kundenspezifische Positioniererdatenbank mit dem Parametersatz für den Positionierer

Damit der Parametersatz in der PC-Software ausgewählt werden kann, muss er zuvor mit dem Programm Import PI Custom Stage in die Positioniererdatenbank PIStages3 eingefügt werden.

➢ Installieren Sie die kundenspezifische Positioniererdatenbank, indem Sie im Hauptverzeichnis des Datenträgers auf die Datei ***Import_PI_CustomStage.exe*** doppelklicken.

Der Parametersatz aus der kundenspezifischen Positioniererdatenbank wird in PIStages3 eingefügt.

Wenn eine Meldung erscheint, dass die Installation der kundenspezifischen Positioniererdatenbank fehlgeschlagen ist:

a) Aktualisieren Sie die Positioniererdatenbank PIStages3 auf Ihrem PC, siehe "Updates installieren" (S. 42).

b) Wiederholen Sie die Installation der kundenspezifischen Positioniererdatenbank.

5.2 C-863.12 montieren

Der C-863.12 kann als Tischgerät verwendet oder in beliebiger Ausrichtung auf einer Unterlage montiert werden.

Der C-863.12 ist stapelbar und kann in einen Schaltschrank eingebaut werden.

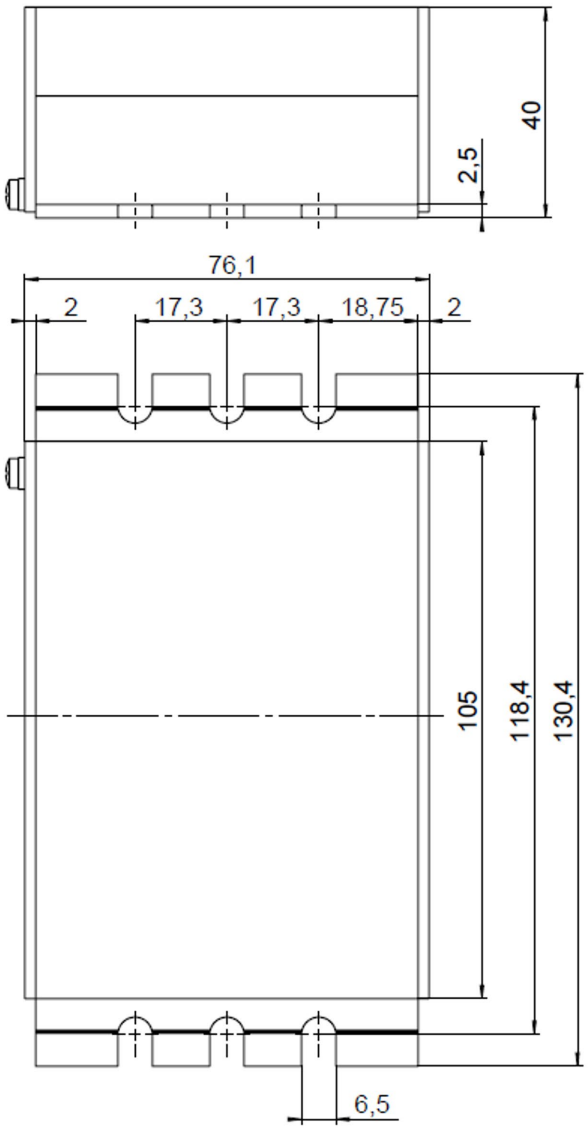


Abbildung 10: C-863.12, Abmessungen in mm

Werkzeug und Zubehör

- Geeignete Schrauben
- Geeigneter Schraubendreher

C-863.12 auf einer Unterlage befestigen

1. Bringen Sie in die Unterlage die erforderlichen Bohrungen ein.

   Die Anordnung der Aussparungen in den Montageleisten des C-863.12 können Sie der Abbildung oben entnehmen.

2. Befestigen Sie den C-863.12 an den Aussparungen in den Montageleisten mit jeweils zwei geeigneten Schrauben pro Seite.

5.3 C-863.12 erden

Der C-863.12 ist nicht über den Netzteilanschluss geerdet.

Wenn ein Potentialausgleich erforderlich ist:

➢ Schließen Sie die mit dem Schutzerde-Symbol gekennzeichnete Schraube (siehe Abbildung) an der Gehäuserückwand des C-863.12 an das Erdungssystem an.

5.4 Positionierer anschließen

HINWEIS

Schäden bei Anschluss eines falschen Motors!

Das Anschließen eines Positionierers mit Schrittmotor an einen DC-Motorcontroller kann irreparable Schäden verursachen.

➢ Schließen Sie an den C-863.12 nur einen Positionierer mit DC-Motor oder integriertem PWM-Motortreiber an.

INFORMATION

Der C-863.12 unterstützt sowohl Positionierer mit PWM-Verstärker als auch Positionierer ohne PWM-Verstärker. An der Buchse **Motor (S. 269)** stehen separate Leitungen für beide Positionierervarianten zur Verfügung. Die Auswahl der passenden Leitungen muss über den Anschlussstecker des Positionierers gewährleistet werden.

INFORMATION

Wenn Positionierer, Kabel und C-863.12 als zusammengehörendes System gekennzeichnet sind:

➢ Halten Sie Rücksprache mit unserem Kundendienst (S. 263), bevor Sie Systemkomponenten austauschen.

Voraussetzungen

- ✓ Der C-863.12 ist ausgeschaltet, d. h. das Netzteil ist **nicht** über das Netzkabel an der Steckdose angeschlossen.
- ✓ Sie haben das Benutzerhandbuch des Positionierers gelesen und verstanden.

Werkzeug und Zubehör

- Positionierer mit DC-Motor und Encoder mit A/B-Quadratur-Signalübertragung
- Geeignetes Kabel aus dem Lieferumfang des Positionierers

Positionierer anschließen

1. Schließen Sie den Positionierer an der Buchse **Motor** des C-863.12 an.
2. Sichern Sie die Steckverbindungen mit den integrierten Schrauben gegen unbeabsichtigtes Abziehen.

5.5 PC anschließen

Die Kommunikation zwischen dem C-863.12 und einem PC ist zur Konfiguration des C-863.12 und zur Bewegungskommandierung mit den Befehlen des GCS notwendig. Der C-863.12 verfügt dazu über folgende Schnittstellen:

- RS-232-Schnittstelle
- USB-Schnittstelle

In diesem Abschnitt erfahren Sie, wie Sie die entsprechenden Kabelverbindungen zwischen C-863.12 und PC sowie in einem Daisy-Chain-Netzwerk herstellen. Die Schritte zur Herstellung der Kommunikation zwischen C-863.12 und PC sind im Kapitel "Inbetriebnahme" beschrieben:

- "Kommunikation über RS-232 herstellen" (S. 57)
- "Kommunikation über USB herstellen" (S. 58)
- "Kommunikation für vernetzten Controller herstellen" (S. 60)

INFORMATION

Über ein Daisy-Chain-Netzwerk können bis zu 16 Controller über einen einzigen RS-232- oder USB-Anschluss an den PC angeschlossen werden.

5.5.1 An RS-232-Schnittstelle anschließen

HINWEIS

Falsche Verkabelung!
Das gleichzeitige Anschließen der USB- und der RS-232-Schnittstelle des Controllers an den PC kann den PC oder den Controller beschädigen.

➢ Schließen Sie entweder die USB- oder die RS-232-Schnittstelle an den PC an.

Voraussetzungen

✓ Der PC verfügt über eine freie RS-232-Schnittstelle (auch als "serielle Schnittstelle" oder "COM-Port" bezeichnet, z. B. COM1 oder COM2).

Werkzeug und Zubehör

- RS-232-Nullmodemkabel (C-815.34 im Lieferumfang)

C-863.12 an den PC anschließen

➢ Verbinden Sie die Buchse **RS-232 In** an der Vorderwand des C-863.12 und die RS-232-Schnittstelle des PC (ein Einbaustecker D-Sub 9(m)) mit dem Nullmodemkabel.

5.5.2 An USB-Schnittstelle anschließen

HINWEIS

Falsche Verkabelung!
Das gleichzeitige Anschließen der USB- und der RS-232-Schnittstelle des Controllers an den PC kann den PC oder den Controller beschädigen.

➢ Schließen Sie entweder die USB- oder die RS-232-Schnittstelle an den PC an.

Voraussetzungen

✓ Der PC verfügt über eine freie USB-Schnittstelle.

Werkzeug und Zubehör

- USB-Kabel (Typ A auf Mini-B) zur Verbindung mit dem PC, im Lieferumfang (S. 9)

C-863.12 an den PC anschließen

➢ Verbinden Sie die USB-Buchse des C-863.12 und die USB-Schnittstelle des PC mit dem USB-Kabel.

5.5.3 Daisy-Chain-Netzwerk aufbauen

INFORMATION

Die Vernetzung in einem Daisy-Chain-Netzwerk erfolgt in Reihe. Siehe auch "Begriffserklärung" (S. 2). Dabei ist der erste Controller direkt mit dem PC verbunden.

INFORMATION

Die DIP-Schalter des C-863.12 müssen passend eingestellt sein:

- Stellen Sie für jeden Controller in einem Daisy-Chain-Netzwerk eine eindeutige Adresse ein. Dabei muss einer der Controller die Adresse 1 haben. Dieser Controller braucht nicht derjenige zu sein, der direkt an den PC angeschlossen ist. Details siehe "Controlleradresse" (S. 54).
- Stellen Sie für alle Controller in einem Daisy-Chain-Netzwerk die gleiche Baudrate ein. Details siehe "Baudrate" (S. 55).

Werkzeug und Zubehör

- Ein Netzwerkkabel für jeden an das Netzwerk anzuschließenden Controller. Verfügbar sind:
  - C-862.CN, 30 cm, im Lieferumfang
  - C-862.CN2, 180 cm, erhältlich als optionales Zubehör (S. 10)

Controller vernetzen

- Bauen Sie die Controller-Reihe auf. Verbinden Sie dazu jeweils den Anschluss **RS-232 Out** des vorhergehenden Controllers über das Netzwerkkabel mit dem Anschluss **RS-232 In** des nachfolgenden Controllers.
- Schließen Sie den ersten Controller der Reihe an den PC an:
  - Verwenden Sie die RS-232-Schnittstelle (S. 47).

  oder

  - Verwenden Sie die USB-Schnittstelle (S. 47).

INFORMATION

Ein C-863.12 kann mit folgenden Controllern in einem gemeinsamen Daisy-Chain-Netzwerk betrieben werden:

- Mercury Controller C-863.11, C-663.11, C-663.12
- PILine® Motion Controller der Reihen C-867 und C-877
- PiezoWalk® NEXACT® Controller E-861

5.6 Netzteil an C-863.12 anschließen

INFORMATION

Wenn der Positionierer mit einem PWM-Verstärker ausgestattet ist, der über ein separates Netzteil versorgt wird:

➢ Um die optimale Motorleistung zu erreichen, verwenden Sie für den C-863.12 ein Netzteil, das die gleiche Ausgangsspannung liefert wie das Netzteil des PWM-Verstärkers.

Voraussetzungen

✓ Das Netzkabel ist **nicht** an der Steckdose angeschlossen.

Werkzeug und Zubehör

- Mitgeliefertes Weitbereichsnetzteil (S. 9), für Netzspannungen zwischen 100 und 240 Volt Wechselspannung bei 50 oder 60 Hz
- Alternativ: ausreichend bemessenes Netzteil
- Mitgeliefertes Netzkabel
- Alternativ: ausreichend bemessenes Netzkabel

Netzteil an den C-863.12 anschließen

➢ Verbinden Sie den 4-poligen Stecker des Netzteils mit der Buchse **48 V 2 A** des C-863.12.

– Stellen Sie sicher, dass der Stecker in der Buchse eingerastet ist.

➢ Verbinden Sie das Netzkabel mit dem Netzteil.

5.7 Analogen Joystick anschließen

INFORMATION

An der Buchse **Joystick** können Sie eine Achse und eine Taste eines analogen Joysticks anschließen:

- Pin 4: Achse 1 von Joystick-Gerät 1
- Pin 6: Taste 1 von Joystick-Gerät 1

Mit der Joystick-Achse können Sie die Geschwindigkeit des am C-863.12 angeschlossenen Positionierers steuern.

INFORMATION

Die als optionales Zubehör erhältlichen Joysticks C-819.20 und C-819.30 nutzen die Pins 4 und 6 der Buchse **Joystick**. Über Pin 3 dieser Buchse erfolgt die Spannungsversorgung des Joysticks.

Sie können ein Y-Kabel C-819.20Y verwenden, um zwei C-863.12 an einen Joystick C-819.20 anzuschließen. In diesem Fall erfolgt die Spannungsversorgung des Joysticks über den C-863.12, der am X-Zweig des Kabels angeschlossen ist.

Werkzeug und Zubehör

- Analoger Joystick von PI für den Betrieb mit 0 bis 3,3 V, erhältlich als optionales Zubehör (S. 10):
  - C-819.20 analoger Joystick für 2 Achsen
  - Wenn ein Joystick C-819.20 an zwei Controller angeschlossen werden soll: C-819.20Y Y-Kabel

  oder

  - C-819.30 analoger Joystick für 3 Achsen

Analogen Joystick anschließen

➢ Schließen Sie den Joystick an der Buchse **Joystick** des C-863.12 an:

- Wenn Sie einen Joystick C-819.20 nur mit diesem Controller betreiben wollen, verbinden Sie ihn direkt mit dem Controller.
- Wenn Sie einen Joystick C-819.20 mit zwei Controllern (d.h. zwei Achsen) betreiben wollen, verbinden Sie den Joystick mit dem Y-Kabel C-819.20Y und schließen die beiden Controller an die X- und Y-Zweige des Kabels an. Die Spannungsversorgung des Joysticks erfolgt über den X-Zweig. Deshalb muss der X-Zweig auch dann an einen Controller angeschlossen sein, wenn für diesen Controller die Joystick-Steuerung nicht aktiviert werden soll.
- Wenn Sie eine Achse des Joysticks C-819.30 anschließen wollen, verbinden Sie das entsprechende Kabel des Joysticks mit dem Controller.

5.8 Digitale Ein- und Ausgänge anschließen

Die digitalen Ein- und Ausgänge auf der Buchse **I/O** des C-863.12 können wie folgt verwendet werden:

- Ausgänge: Triggern von externen Geräten; siehe "Digitale Ausgangssignale" (S. 78).
- Eingänge: Verwendung in Makros (S. 89) und/oder als Quelle für die Referenz- und Endschaltersignale der Achse (S. 89)

5.8.1 Digitale Ausgänge anschließen

INFORMATION

Digitale Ausgangssignale sind auf den Pins 5, 6, 7 und 8 der Buchse **I/O** verfügbar.

INFORMATION

Wenn die Pushbutton-Box C-170.PB von PI an der Buchse **I/O** angeschlossen ist, zeigt sie über LEDs den Status der digitalen Ausgangsleitungen an.

Werkzeug und Zubehör

- Geeignetes Kabel, z. B. C-170.IO IO-Kabel mit offenem Ende, erhältlich als optionales Zubehör (S. 10)
- Zu triggerndes Gerät mit digitalem Eingang für TTL-Signale

Zu triggerndes Gerät anschließen

➢ Schließen Sie ein geeignetes Gerät an einen der Pins 5, 6, 7 oder 8 der Buchse **I/O** des C-863.12 an.

5.8.2 Digitale Eingänge anschließen

INFORMATION

Digitale Eingangssignale können über die Pins 1, 2, 3 und 4 der Buchse **I/O** in den C-863.12 eingespeist werden.

INFORMATION

Die digitalen Eingänge (Pins 1 bis 4) auf der Buchse **I/O** können auch als analoge Eingänge genutzt werden.

- Digital: TTL
- Analog: 0 bis +5 V

Werkzeug und Zubehör

- Geeignete Signalquelle:
  - Wenn die digitalen Eingänge in Makros verwendet werden sollen, kann z.B. die Pushbutton-Box C-170.PB angeschlossen werden, erhältlich als optionales Zubehör (S. 10).
  - Wenn die digitalen Eingänge als Quelle für die Referenz- und Endschaltersignale der Achse verwendet werden sollen, darf der Signalpegel nur einmal über den gesamten Stellweg wechseln.
- Wenn notwendig: Geeignetes Kabel, z. B. C-170.IO IO-Kabel mit offenem Ende, erhältlich als optionales Zubehör (S. 10).

Digitale Signalquelle anschließen

➢ Schließen Sie eine geeignete Signalquelle an einen der Pins 1, 2, 3, oder 4 der Buchse **I/O** des C-863.12 an.

5.9 Analoge Signalquellen anschließen

Die analogen Eingänge auf der Buchse **I/O** des C-863.12 können wie folgt verwendet werden:

- Verwendung in Makros (S. 92): Details und Beispiele zu Makros finden Sie unter "Controllermakros" (S. 100).
- Scan-Anwendungen mit PIMikroMove® (siehe PIMikroMove® Handbuch)

INFORMATION

Analoge Eingangssignale können über die Pins 1, 2, 3 und 4 der Buchse **I/O** in den C-863.12 eingespeist werden.

INFORMATION

Die analogen Eingänge (Pins 1 bis 4) auf der Buchse **I/O** können auch als digitale Eingänge genutzt werden.

- Analog: 0 bis +5 V
- Digital: TTL

Werkzeug und Zubehör

- Geeignete Signalquelle
- Wenn notwendig: Geeignetes Kabel, z. B. C-170.IO IO-Kabel mit offenem Ende, erhältlich als optionales Zubehör (S. 10).

Analoge Signalquelle anschließen

➢ Schließen Sie eine geeignete Signalquelle an einen der Pins 1, 2, 3 oder 4 der Buchse **I/O** des C-863.12 an.

6 Inbetriebnahme

6.1 Allgemeine Hinweise zur Inbetriebnahme

HINWEIS

Schäden durch deaktivierte Endschalterauswertung!
Der Aufprall eines bewegten Teils am Ende des Stellwegs oder auf einem Hindernis sowie hohe Beschleunigungen können Schäden oder erheblichen Verschleiß an der Mechanik verursachen.

- Vermeiden Sie Bewegungen im ungeregelten Betrieb.
- Wenn Bewegungen im ungeregelten Betrieb notwendig sind:
  - Setzen Sie den Stellwert mit dem Befehl SMO so, dass sich die Achse mit niedriger Geschwindigkeit bewegt.
  - Stoppen Sie die Achse rechtzeitig. Verwenden Sie dazu die Befehle #24, STP oder HLT, oder setzen Sie den Stellwert mit dem Befehl SMO auf null.
- Deaktivieren Sie **nicht** per Parametereinstellung die Endschalterauswertung durch den C-863.12.
- Prüfen Sie die Funktion der Endschalter bei etwa 10 % bis 20 % der Maximalgeschwindigkeit.
- Halten Sie bei einer Fehlfunktion der Endschalter die Bewegung sofort an.

6.2 DIP-Schalter-Einstellungen anpassen

6.2.1 Generelle Vorgehensweise

INFORMATION

Geänderte DIP-Schalter-Einstellungen werden nach dem Einschalten des C-863.12 wirksam.

- Wenn Sie DIP-Schalter-Einstellungen bei eingeschaltetem C-863.12 geändert haben, schalten Sie den C-863.12 aus und wieder ein, um die neuen Einstellungen zu aktivieren.

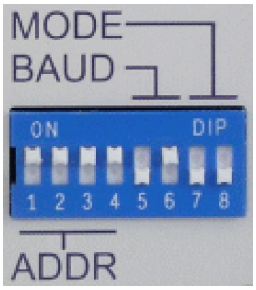


Abbildung 11: DIP-Schalter: Schalter oben = EIN; Schalter unten = AUS

Schalter	Funktion
1 bis 4	Controlleradresse (S. 54)
5 und 6	Baudrate (S. 55)
7 und 8	ohne Funktion

Voraussetzung

✓ Der C-863.12 ist ausgeschaltet, d. h. das Netzteil ist **nicht** über das Netzkabel an der Steckdose angeschlossen.

DIP-Schalter-Einstellungen anpassen

➢ Bringen Sie die einzelnen DIP-Schalter in die für Ihre Anwendung passende Stellung. Details finden Sie in den nachfolgenden Tabellen.

6.2.2 Controlleradresse

Adresse*	S1	S2	S3	S4
1	**EIN**	**EIN**	**EIN**	**EIN**
2	EIN	EIN	EIN	AUS
3	EIN	EIN	AUS	EIN
4	EIN	EIN	AUS	AUS
5	EIN	AUS	EIN	EIN
6	EIN	AUS	EIN	AUS
7	EIN	AUS	AUS	EIN
8	EIN	AUS	AUS	AUS
9	AUS	EIN	EIN	EIN
10	AUS	EIN	EIN	AUS
11	AUS	EIN	AUS	EIN

Adresse*	S1	S2	S3	S4
12	AUS	EIN	AUS	AUS
13	AUS	AUS	EIN	EIN
14	AUS	AUS	EIN	AUS
15	AUS	AUS	AUS	EIN
16	AUS	AUS	AUS	AUS

*Werkseinstellungen sind fett gedruckt.

INFORMATION

Für jeden Controller in einem Daisy-Chain-Netzwerk muss eine eindeutige Adresse eingestellt sein. Dabei muss einer der Controller die Adresse 1 haben. Dieser Controller braucht nicht derjenige zu sein, der direkt an den PC angeschlossen ist.

INFORMATION

Ein nicht vernetzter Controller muss die Adresse 1 haben, wenn er

- in PIMikroMove® verwendet werden soll.
- in Treibern für NI LabVIEW-Software verwendet werden soll.
- mit dem PITerminal ohne Angabe der Empfängeradresse angesprochen werden soll; in den Antworten des C-863.12 entfallen dann auch Empfänger- und Senderadresse (S. 122).

6.2.3 Baudrate

Baudrate*	S5	S6
9600	EIN	EIN
19200	EIN	AUS
38400	AUS	EIN
115200	**AUS**	**AUS**

*Werkseinstellungen sind fett gedruckt.

INFORMATION

Für alle Controller in einem Daisy-Chain-Netzwerk muss die gleiche Baudrate eingestellt sein.

6.3 C-863.12 einschalten

INFORMATION

Der C-863.12 ist für den geregelten Betrieb mit inkrementellen Positionssensoren vorgesehen (Servomodus Ein). Nach dem Einschalten ist standardmäßig der ungeregelte Betrieb aktiviert (Servomodus Aus).

- Fragen Sie die aktuelle Betriebsart mit den Befehlen `SVO?`, `#4` oder `SRG?` ab.
- Aktivieren Sie den geregelten Betrieb mit dem Befehl `SVO`.
- Wenn nötig, programmieren Sie ein Startup-Makro, das den C-863.12 über den Befehl `SVO` im geregelten Betrieb startet; siehe "Startup-Makro einrichten" (S. 109).
- Vermeiden Sie Bewegungen im ungeregelten Betrieb.

Voraussetzungen

- ✓ Sie haben die allgemeinen Hinweise zur Inbetriebnahme gelesen und verstanden (S. 53).
- ✓ Der C-863.12 wurde ordnungsgemäß installiert (S. 41).
- ✓ Sie haben die DIP-Schalter des C-863.12 passend zu Ihrer Anwendung eingestellt (S. 53).

C-863.12 einschalten

- Verbinden Sie das Netzkabel des Netzteils mit der Steckdose.

  Der C-863.12 lädt die Parameterwerte aus dem permanenten Speicher in den flüchtigen Speicher.

  Die LED **STA** an der Vorderwand des C-863.12 zeigt den Status des C-863.12 an:

  - Dauerhaftes Leuchten: C-863.12 ist bereit für den Normalbetrieb
  - Blinken: C-863.12 befindet sich im Firmware-Update-Modus
  - Aus: C-863.12 ist nicht an der Stromversorgung angeschlossen oder könnte defekt sein
- Wenn sich der C-863.12 im Firmware-Update-Modus befindet, aktualisieren Sie die Firmware (S. 253).
- Wenn der C-863.12 korrekt mit dem Netzteil verbunden ist (S. 49) und die LED **STA** nach dem Einschalten nicht leuchtet, kontaktieren Sie unseren Kundendienst (S. 263).

6.4 Kommunikation herstellen

INFORMATION

Für die USB-Schnittstelle wird im C-863.12 ein USB-UART-Modul (FTDI) verwendet. Wenn der C-863.12 über den USB-Anschluss verbunden und eingeschaltet ist, wird die USB-Schnittstelle in der PC-Software deshalb auch als COM-Port angezeigt. Der C-863.12 verwendet für diese Schnittstelle die Baudrate 115200.

INFORMATION

Verwenden Sie die Registerkarten ***USB Daisy Chain*** und ***RS-232 Daisy Chain*** in der PC-Software nur dann zum Aufbau der Kommunikation, wenn Sie tatsächlich ein Daisy-Chain-Netzwerk an den PC angeschlossen haben.

INFORMATION

Ein nicht vernetzter Controller muss die Adresse 1 haben, wenn er in PIMikroMove® verwendet werden soll. Details siehe "Controlleradresse" (S. 54).

Im Folgenden ist das Vorgehen für PIMikroMove® beschrieben.

Informationen zur Herstellung der Kommunikation auf Linux-Systemen finden Sie in der Technical Note "PI Software on ARM-Based Platforms", A000T0089 (S. 3).

6.4.1 Kommunikation über RS-232 herstellen

Voraussetzungen

- ✓ Sie haben die allgemeinen Hinweise zur Inbetriebnahme gelesen und verstanden (S. 53).
- ✓ Der C-863.12 ist an die RS-232-Schnittstelle des PC angeschlossen (S. 47).
- ✓ Sie haben vor dem Einschalten des C-863.12 folgende Einstellungen mit den entsprechenden DIP-Schaltern vorgenommen (S. 53):
  - Controlleradresse = 1
  - passende Baudrate
- ✓ Der C-863.12 ist eingeschaltet (S. 55).
- ✓ Der PC ist eingeschaltet.
- ✓ Die benötigte Software ist auf dem PC installiert (S. 41).
- ✓ Sie haben das Handbuch der verwendeten PC-Software gelesen und verstanden. Links auf die Software-Handbücher finden Sie in der Datei A000T0081 auf dem Datenträger mit der PI-Software.

Kommunikation herstellen

1. Starten Sie PIMikroMove®.

   Das Fenster ***Start up controller*** öffnet sich mit dem Schritt ***Connect controller***.

   – Wenn sich das Fenster ***Start up controller*** nicht automatisch öffnet, wählen Sie im Hauptfenster den Menüeintrag ***Connections > New...***.

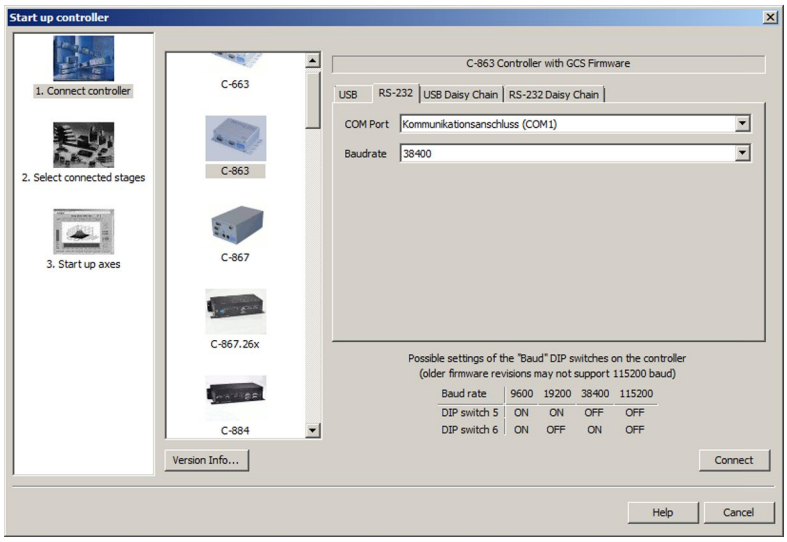


Abbildung 12: Start up controller – Connect controller

2. Wählen Sie im Feld für die Controllerauswahl ***C-863*** aus.
3. Wählen Sie auf der rechten Seite des Fensters die Registerkarte ***RS-232*** aus.
4. Wählen Sie im Feld ***COM Port*** den COM-Port des PC aus, an dem Sie den C-863.12 angeschlossen haben.
5. Stellen Sie im Feld ***Baudrate*** den Wert ein, der mit den DIP-Schaltern 5 und 6 des C-863.12 eingestellt ist.

   Damit passen Sie die Baudrate des PC an die Baudrate des C-863.12 an.
6. Klicken Sie auf ***Connect***, um die Kommunikation herzustellen.

Wenn die Kommunikation erfolgreich hergestellt wurde, leitet PIMikroMove® Sie durch die Konfiguration des C-863.12 für den angeschlossenen Positionierer, siehe "Bewegungen starten" (S. 64).

6.4.2 Kommunikation über USB herstellen

INFORMATION

Wenn der Controller über den USB-Anschluss verbunden und eingeschaltet ist, wird die USB-Schnittstelle in der PC-Software auch als COM-Port angezeigt.

Voraussetzungen

- ✓ Sie haben die allgemeinen Hinweise zur Inbetriebnahme gelesen und verstanden (S. 53).
- ✓ Der C-863.12 ist an die USB-Schnittstelle des PC angeschlossen.
- ✓ Sie haben vor dem Einschalten des C-863.12 die DIP-Schalter für die Controlleradresse auf die Adresse 1 eingestellt (S. 53).
- ✓ Der C-863.12 ist eingeschaltet (S. 55).
- ✓ Der PC ist eingeschaltet.
- ✓ Die benötigte Software sowie die USB-Treiber sind auf dem PC installiert.
- ✓ Sie haben das Handbuch der verwendeten PC-Software gelesen und verstanden. Links auf die Software-Handbücher finden Sie in der Datei A000T0081 auf dem Datenträger mit der PI-Software.

Kommunikation herstellen

1. Starten Sie PIMikroMove®.

   Das Fenster ***Start up controller*** öffnet sich mit dem Schritt ***Connect controller***.

   – Wenn sich das Fenster ***Start up controller*** nicht automatisch öffnet, wählen Sie im Hauptfenster den Menüeintrag ***Connections > New...***.

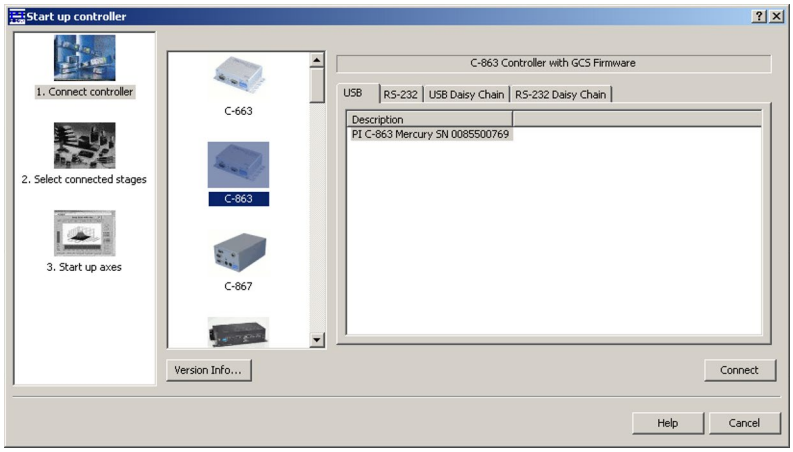


   Abbildung 13: Start up controller – Connect controller

2. Wählen Sie im Feld für die Controllerauswahl ***C-863*** aus.
3. Wählen Sie auf der rechten Seite des Fensters die Registerkarte ***USB*** aus.
4. Wählen Sie auf der Registerkarte ***USB*** den angeschlossenen C-863.12 aus.
5. Klicken Sie auf ***Connect***, um die Kommunikation herzustellen.

Wenn die Kommunikation erfolgreich hergestellt wurde, leitet PIMikroMove® Sie durch die Konfiguration des C-863.12 für den angeschlossenen Positionierer, siehe "Bewegungen starten" (S. 64).

➢ Wenn die Kommunikation nicht hergestellt werden konnte, suchen Sie in "Störungsbehebung" (S. 257) nach einer Lösung des Problems.

6.4.3 Kommunikation für vernetzten Controller herstellen

Im Folgenden ist das Vorgehen für PIMikroMove® und für PITerminal beschrieben.

INFORMATION

Wenn Sie die Kommunikation mit einem vernetzten Controller über PITerminal herstellen, ist die Adresse des anzusprechenden Controllers in jeder Befehlszeile erforderlich. Details siehe "Empfänger- und Senderadresse" (S. 122).

➢ Verwenden Sie PITerminal, um die Kommunikation mit vernetzten Controllern zu testen.

INFORMATION

Die RS-232-Ausgangsleitungen mancher PCs sind nicht für die Maximalanzahl von 16 Controllern in einem Netzwerk geeignet. Wenn Sie ein Daisy-Chain-Netzwerk über die RS-232-Schnittstelle an einen solchen PC angeschlossen haben, können Kommunikationsstörungen auftreten (z. B. Timeout). Bei Kommunikationsstörungen:

1. Entfernen Sie das Nullmodemkabel von der Buchse **RS-232 In** des Controllers, der mit dem PC verbunden ist.
2. Schließen Sie das Daisy-Chain-Netzwerk über die USB-Schnittstelle dieses Controllers an den PC an.

Voraussetzungen

✓ Sie haben die allgemeinen Hinweise zur Inbetriebnahme gelesen und verstanden (S. 53).

✓ Sie haben das Daisy-Chain-Netzwerk aufgebaut (S. 48).

✓ Sie haben jedem der vernetzten Controller eine eindeutige Controlleradresse zugewiesen (S. 54).

✓ Sie haben für alle Controller dieselbe Baudrate eingestellt (S. 55).

✓ Alle Controller im Daisy-Chain-Netzwerk sind eingeschaltet (S. 55).

✓ Der PC ist eingeschaltet.

✓ Die benötigte Software ist auf dem PC installiert (S. 41).

✓ Wenn Sie den ersten Controller in der Reihe über die USB-Schnittstelle an den PC angeschlossen haben: Die USB-Treiber sind auf dem PC installiert (S. 41).

✓ Sie haben das Handbuch der verwendeten PC-Software gelesen und verstanden. Links auf die Software-Handbücher finden Sie in der Datei A000T0081 auf dem Datenträger mit der PI-Software.

Kommunikation mit PIMikroMove® herstellen

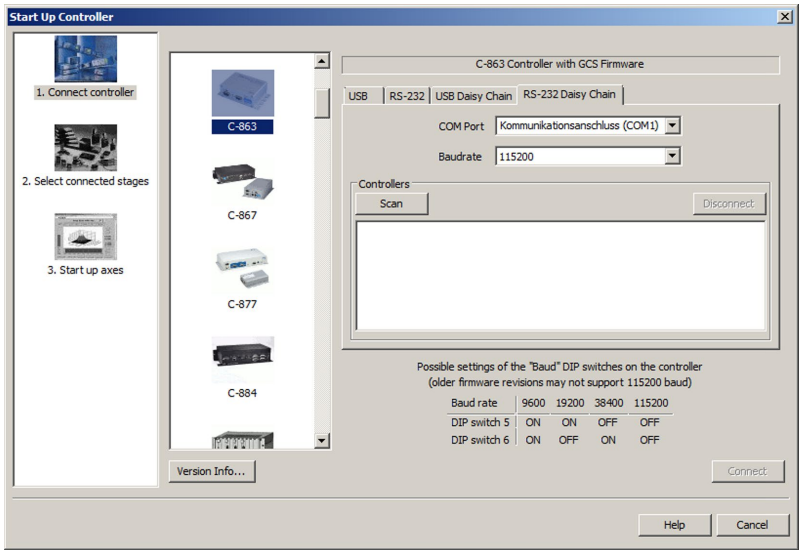
1. Starten Sie PIMikroMove®.

   Das Fenster ***Start up controller*** öffnet sich mit dem Schritt ***Connect controller***.

   – Wenn sich das Fenster ***Start up controller*** nicht automatisch öffnet, wählen Sie im Hauptfenster den Menüeintrag ***Connections > New...***.

2. Wählen Sie im Feld für die Controllerauswahl den passenden Controllertyp aus.

   Im Beispiel in den nachfolgenden Abbildungen besteht das Daisy-Chain-Netzwerk aus einem C-863.12 mit der Controlleradresse 1 und einem C-663.12 mit der Controlleradresse 2. Wenn Sie zuerst den C-863.12 verbinden möchten, wählen Sie ***C-863***.



3. Wählen Sie auf der rechten Seite des Fensters die passende Registerkarte aus:

   – Wenn Sie den ersten Controller der Reihe über die RS-232-Schnittstelle an den PC angeschlossen haben, wählen Sie die Registerkarte ***RS-232 Daisy Chain***.

   – Wenn Sie den ersten Controller der Reihe über die USB-Schnittstellen an den PC angeschlossen haben, wählen Sie die Registerkarte ***USB Daisy Chain***.

4. Nehmen Sie auf der ausgewählten Registerkarte die Einstellungen für die Schnittstelle vor:

   – Registerkarte ***RS-232 Daisy Chain***:

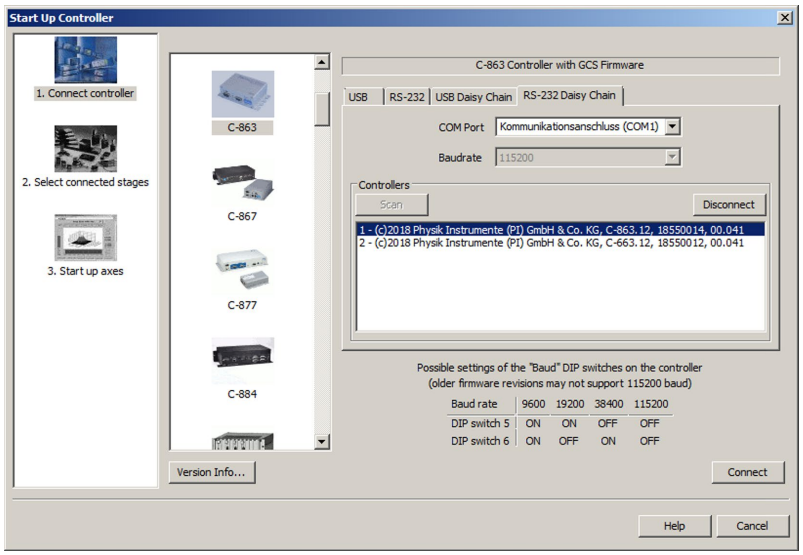
     ▫ Wählen Sie im Feld ***COM Port*** den COM-Port des PC aus, an dem Sie den ersten Controller der Reihe angeschlossen haben.

     ▫ Stellen Sie im Feld ***Baudrate*** den Wert ein, der für alle Controller der Reihe eingestellt ist.

   – Registerkarte ***USB Daisy Chain***:

     ▫ Wählen Sie im oberen Bereich der Registerkarte den am PC angeschlossenen Controller aus.

5. Klicken Sie im unteren Bereich der Registerkarte auf die Schaltfläche ***Scan***, um alle Controller des Daisy-Chain-Netzwerks aufzulisten.

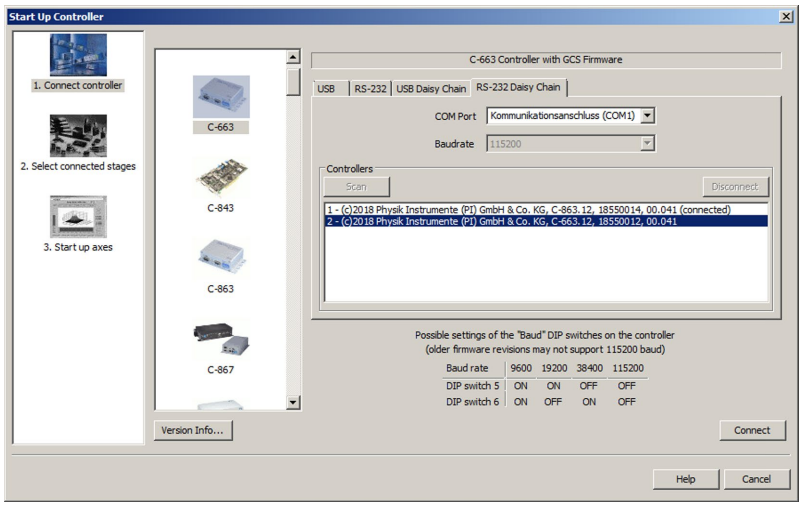


6. Wählen Sie einen Controller aus der Liste aus. Die Auswahl muss zum Controllertyp passen, den Sie in Schritt 2 ausgewählt haben.
7. Klicken Sie auf ***Connect***, um die Kommunikation mit dem ausgewählten Controller herzustellen.

   Wenn die Kommunikation erfolgreich hergestellt wurde, leitet PIMikroMove® Sie durch die Konfiguration des C-863.12 für den angeschlossenen Positionierer.

   – Gehen Sie weiter vor wie in "Bewegungen starten" (S. 64) beschrieben.
8. Wenn Sie einen weiteren Controller des Daisy-Chain-Netzwerks verbinden möchten, wählen Sie im Hauptfenster den Menüeintrag ***Connections > New...***.
9. Führen Sie erneut die Schritte 2, 6 und 7 in der angegebenen Reihenfolge aus.

   In der nachfolgenden Abbildung soll auch der ***C-663*** verbunden werden.



10. Wiederholen Sie die Schritte 8, 2, 6 und 7 für jeden weiteren Controller des Daisy-Chain-Netzwerks, den Sie verbinden möchten.

Wenn Sie die Kommunikation mit einem der Controller aus dem Daisy-Chain-Netzwerk beenden möchten:

➢ Wählen Sie im Hauptfenster den Menüeintrag ***Connections > Close*** für den entsprechenden Controller.

Kommunikation mit PITerminal herstellen

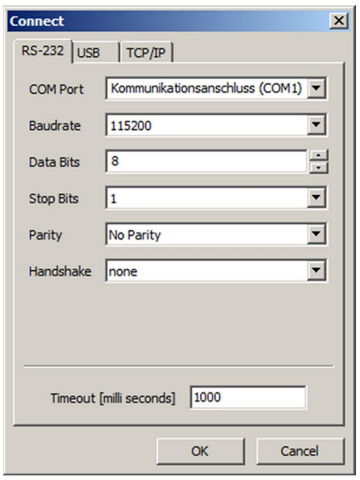
INFORMATION

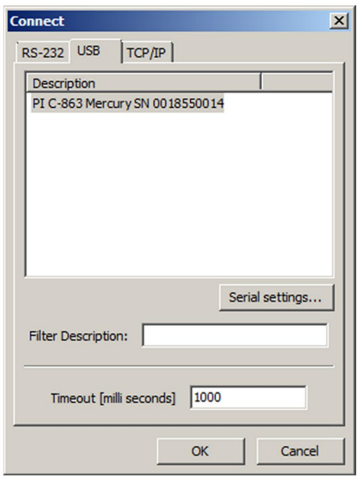
PITerminal unterstützt durch die Schaltfläche ***Mercury*** Controller mit älteren, nicht GCS-kompatiblen Firmware-Versionen.

➢ Stellen Sie im PITerminal sicher, dass **nicht** die Schaltfläche ***Mercury*** aktiviert ist.

1. Starten Sie PITerminal.
2. Klicken Sie auf ***Connect…***.

   Das Fenster ***Connect*** öffnet sich.
3. Wählen Sie im Fenster ***Connect*** die Registerkarte ***RS-232*** oder ***USB*** aus, je nachdem, über welche Schnittstelle Sie den ersten Controller in der Reihe an den PC angeschlossen haben.





4. Nehmen Sie auf der ausgewählten Registerkarte die Einstellungen für die Schnittstelle vor:
   – Registerkarte ***RS-232***:
     - Wählen Sie im Feld ***COM Port*** den COM-Port des PC aus, an dem Sie den C-863.12 angeschlossen haben.
     - Stellen Sie im Feld ***Baudrate*** den Wert ein, der mit den DIP-Schaltern 5 und 6 des C-863.12 eingestellt ist.

– Registerkarte ***USB***:

  ▫ Wählen Sie den angeschlossenen C-863.12 aus.

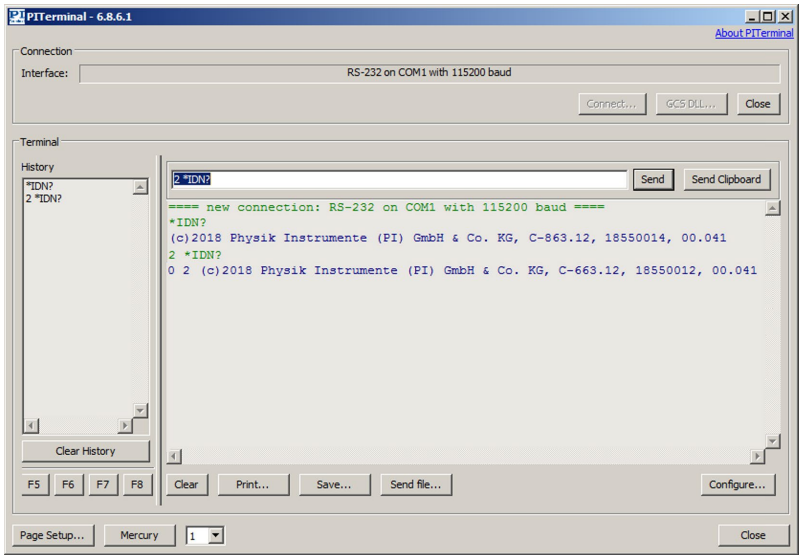
5. Klicken Sie auf ***OK***, um die Kommunikation herzustellen.
6. Senden Sie den Befehl `*IDN?` für alle Controller im Daisy-Chain-Netzwerk, um die Kommunikation zu prüfen.

   Im Beispiel in der nachfolgenden Abbildung besteht das Daisy-Chain-Netzwerk aus einem C-863.12 mit der Controlleradresse 1 und einem C-663.12 mit der Controlleradresse 2. Sie senden:

   – `*IDN?`, um die Ident-Bezeichnung des Controllers mit der Adresse 1 abzufragen; die Controlleradresse ist nicht erforderlich (da = 1)

   – `2 *IDN?`, um die Ident-Bezeichnung des Controllers mit der Adresse 2 abzufragen.

   Weitere Informationen siehe "Empfänger- und Senderadresse" (S. 122).



6.5 Bewegungen starten

Im Folgenden wird PIMikroMove® verwendet, um den Positionierer zu bewegen. Das Programm leitet Sie dabei durch die folgenden Schritte, so dass Sie sich nicht mit den entsprechenden GCS-Befehlen auseinandersetzen müssen:

- Anpassung der Parametereinstellungen des C-863.12 an den angeschlossenen Positionierer durch Laden eines Parametersatzes aus einer Positioniererdatenbank
- Einschalten des Servomodus (geregelter Betrieb)
- Ausführen einer Referenzierungsfahrt; Details siehe "Referenzierung" (S. 35).

HINWEIS

Auswahl eines falschen Positionierertyps

Die Auswahl eines falschen Positionierertyps in der PC-Software kann Schäden am Positionierer verursachen.

- Stellen Sie sicher, dass der in der PC-Software ausgewählte Positionierertyp mit dem angeschlossenen Positionierer übereinstimmt.

HINWEIS

Schwingungen!

Ungeeignete Einstellungen der Regelparameter des C-863.12 können den Positionierer zum Schwingen bringen. Schwingungen können den Positionierer und/oder die auf ihm angebrachte Last beschädigen.

- Befestigen Sie den Positionierer und alle Lasten ausreichend.
- Wenn der angeschlossene Positionierer schwingt (ungewöhnliches Laufgeräusch), schalten Sie den Servomodus sofort aus oder trennen Sie den C-863.12 von der Stromversorgung.
- Schalten Sie den Servomodus erst wieder ein, nachdem Sie die Einstellungen der Regelparameter des C-863.12 geändert haben; siehe "Regelparameter optimieren" (S. 69).
- Wenn aufgrund einer sehr hohen Last Schwingungen bereits während der Referenzierungsfahrt auftreten, folgen Sie für die Referenzierungsfahrt den Anweisungen in "Störungsbehebung" (S. 257).

INFORMATION

Der C-863.12 verfügt über einen permanenten Speicher für Parameterwerte. Deshalb können nach dem Einschalten bereits die korrekten Parametereinstellungen für den angeschlossenen Positionierer geladen sein.

- Wenn Sie trotzdem einen Parametersatz aus der Positioniererdatenbank geladen und damit die ursprünglichen Einstellungen des C-863.12 im flüchtigen Speicher überschrieben haben, vermeiden Sie das Speichern der neuen Einstellungen im permanenten Speicher des C-863.12. Nach dem Aus- und Wiedereinschalten oder einem Neustart des C-863.12 sind wieder die ursprünglichen Einstellungen aktiv.
- Wenn Sie für die Positioniererauswahl das Fenster ***Start up controller*** in PIMikroMove® verwenden und gefragt werden, wie Sie die neuen Einstellungen speichern möchten, klicken Sie die Schaltfläche ***Save all settings permanently on controller*** nur dann, wenn Sie sicher sind, dass der C-863.12 mit den Einstellungen korrekt funktioniert.

INFORMATION

Wenn in PIMikroMove® nicht der Schritt ***Select connected stages*** angezeigt wird, hat der Controller wahrscheinlich schon die korrekten Parametereinstellungen für den angeschlossenen Positionierertyp geladen.

1. Prüfen Sie im Schritt ***Start up axes***, ob in der Spalte ***Stage*** im mittleren Bereich des Fensters der korrekte Positionierertyp steht.

2. Wenn der Positionierertyp nicht korrekt ist, klicken Sie im linken Bereich des Fensters ***Start up controller*** auf ***Select connected stages***, um die Auswahl des Positionierertyp ändern zu können.

Voraussetzungen

- ✓ Sie haben die allgemeinen Hinweise zur Inbetriebnahme gelesen und verstanden (S. 53).
- ✓ PIMikroMove® ist auf dem PC installiert (S. 41).
- ✓ Sie haben das Handbuch für PIMikroMove® gelesen und verstanden. Links auf die Softwarehandbücher finden Sie in der Datei A000T0081 auf dem Datenträger mit der PI-Software.
- ✓ Sie haben die neueste Version der Positioniererdatenbank PISTAGES3.DB auf dem PC installiert (S. 41).
- ✓ Wenn Sie für Ihren Positionierer eine kundenspezifische Positioniererdatenbank von PI erhalten haben, dann haben Sie diesen Datensatz in PIStages3 importiert (S. 43).
- ✓ Sie haben den Positionierer so installiert, wie er in Ihrer Anwendung eingesetzt wird (entsprechende Last, Ausrichtung und Befestigung).
- ✓ Sie haben den Positionierer an den C-863.12 angeschlossen (S. 46).
- ✓ Sie haben die Kommunikation zwischen dem C-863.12 und dem PC mit PIMikroMove® hergestellt (S. 57).

Bewegungen starten mit PIMikroMove®

1. Wenn in PIMikroMove® der Schritt ***Select connected stages*** angezeigt wird, wählen Sie den Positionierertyp des angeschlossenen Positionierers aus:

   Wenn rechts im Fenster in der Liste ***Controller axes*** bereits der korrekte Positionierertyp in der Spalte ***Current stage type*** eingetragen ist:

   – Klicken Sie auf ***OK***.

   Wenn der eingetragene Positionierertyp nicht korrekt ist:

   a) Markieren Sie den Positionierertyp in der Liste ***Stage database entries***.

   b) Klicken Sie auf ***Assign***.

c) Bestätigen Sie die Auswahl mit ***OK***.

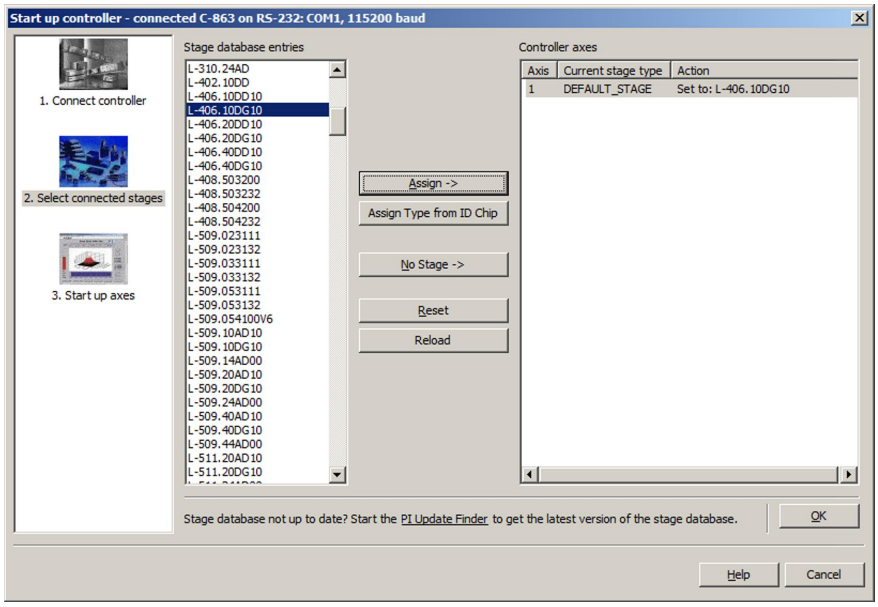


Abbildung 14: Start up controller – Select connected stages

2. Geben Sie im Dialog ***Save all changes permanently?*** an, wie Sie die Parametereinstellungen in den C-863.12 laden wollen:

   – Temporär laden: Klicken Sie auf ***Keep the changes temporarily***, um die Parametereinstellungen in den flüchtigen Speicher des C-863.12 zu laden. Die Einstellungen gehen beim Ausschalten oder Neustart des C-863.12 verloren.

   – Als Standardwerte laden: Klicken Sie auf ***Save all settings permanently on controller***, um die Parametereinstellungen in den permanenten Speicher des C-863.12 zu laden. Die Einstellungen sind nach dem Einschalten oder Neustart des C-863.12 sofort vorhanden und müssen nicht erneut geladen werden.

   Das Fenster ***Start up controller*** wechselt zum Schritt ***Start up axes***.

3. Führen Sie im Schritt ***Start up axes*** die Referenzierungsfahrt für die Achse aus, damit der Controller die absolute Achsenposition kennt:

   – Wenn Sie die Referenzierungsfahrt zum Referenzschalter starten wollen, klicken Sie auf ***Ref. switch***.

   – Wenn Sie die Referenzierungsfahrt zum negativen Endschalter starten wollen, klicken Sie auf ***Neg. limit***.

   – Wenn Sie die Referenzierungsfahrt zu positiven Endschalter starten wollen, klicken Sie auf ***Pos. limit***.

   Wenn eine Warnmeldung erscheint, dass der Servomodus ausgeschaltet ist:

   – Schalten Sie den Servomodus durch einen Klick auf die Schaltfläche ***Switch on servo*** ein (geregelter Betrieb).

Die Achse führt die Referenzierungsfahrt aus.

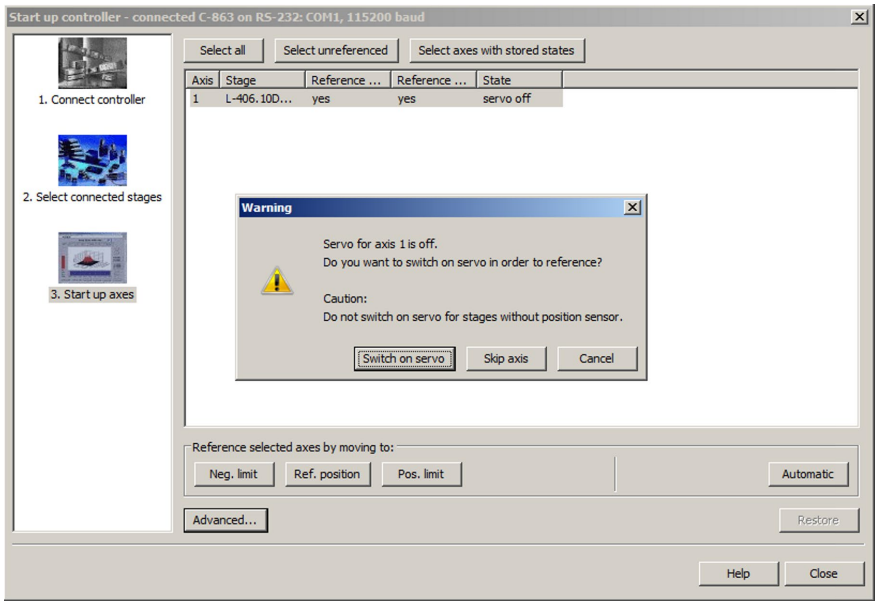


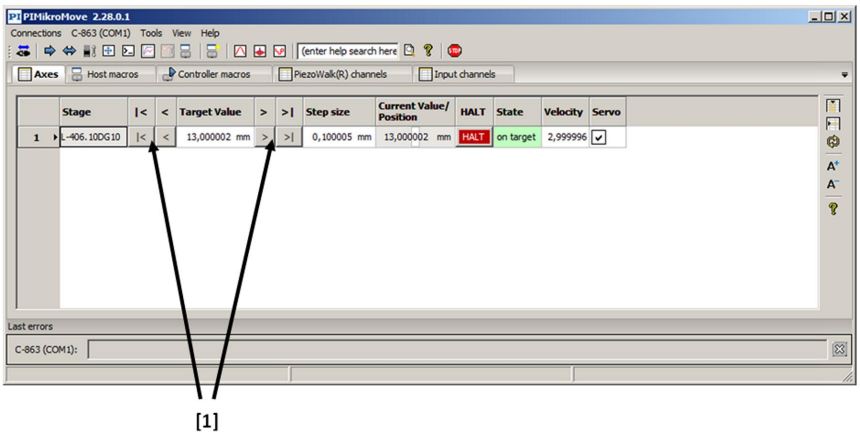
Abbildung 15: Start up controller – Start up axes

4. Nach erfolgreicher Referenzierungsfahrt schließen Sie das Fenster ***Start up controller*** durch Klicken auf ***OK > Close***.

   Das Hauptfenster von PIMikroMove® öffnet sich.

5. Starten Sie einige Testbewegungen der Achse.

   Im Hauptfenster von PIMikroMove® können Sie z. B. Bewegungen um eine bestimmte Strecke (Vorgabe in Spalte ***Step size***) oder zu den Grenzen des Stellwegs ausführen, indem Sie auf die entsprechenden Pfeiltasten für die Achse klicken.



[1] Pfeiltasten für Bewegungen; Hauptfenster von PIMikroMove®

6.6 Regelparameter optimieren

Durch die Einstellung des P-I-D-Reglers werden die dynamischen Eigenschaften des Systems (Überschwingen und Einschwingzeit) optimiert. Die optimale Einstellung des P-I-D-Reglers hängt von Ihrer Anwendung und Ihren Wünschen ab.

Typischerweise erfolgt die Optimierung empirisch und umfasst die folgenden Parameter. Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26):

- ***P-Term*** (0x411)
- ***I-Term*** (0x412)
- ***D-Term*** (0x413)
- ***I-Limit*** (0x414)

Dabei wird das Verhalten des Positionierers bei verschiedenen Werten im geregelten Betrieb beobachtet.

Im Folgenden wird PIMikroMove® für die Optimierung der Regelparameter verwendet.

Voraussetzung

✓ Sie haben den Positionierer so installiert, wie er in Ihrer Anwendung eingesetzt wird (entsprechende Last, Ausrichtung und Befestigung).

✓ Sie haben mit PIMikroMove® erste Bewegungen gestartet (S. 64).

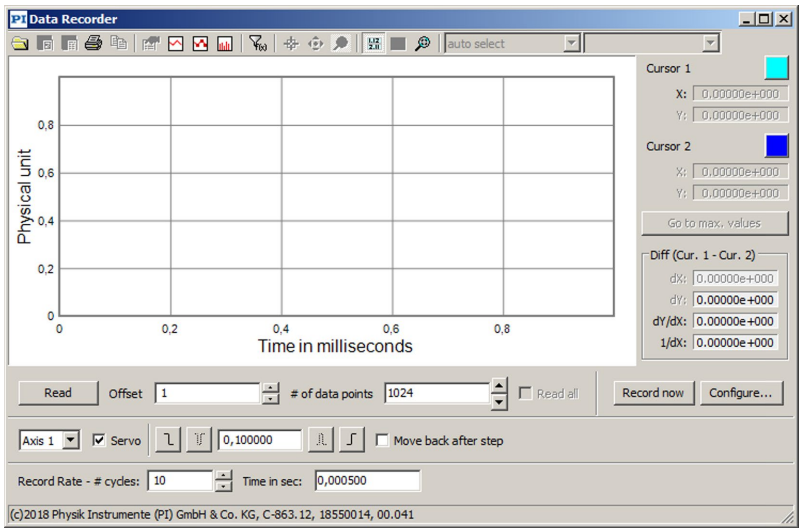
✓ Alle Geräte sind noch betriebsbereit.

Regelparameter prüfen: Sprungantwort aufzeichnen

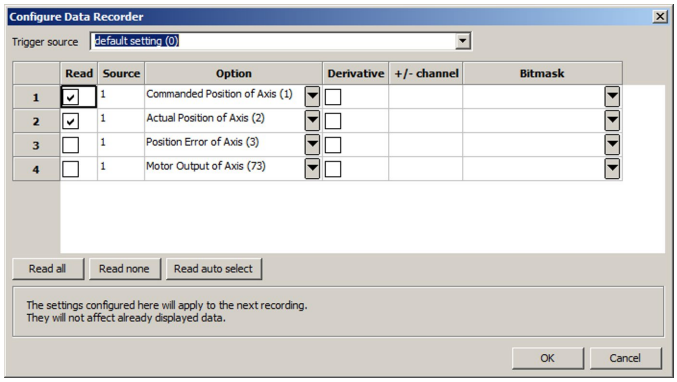
Mit der Aufzeichnung der Sprungantwort ermitteln Sie das Einschwingverhalten des Positionierers im geregelten Betrieb.

1. Öffnen Sie im Hauptfenster von PIMikroMove® das Fenster ***Data Recorder*** über den Menüeintrag ***C-863 > Show data recorder***.
2. Stellen Sie mit dem Kontrollkästchen ***Servo*** sicher, dass der Servomodus eingeschaltet ist.
   - Wenn das Kontrollkästchen ***Servo*** nicht markiert ist, ist der Servomodus ausgeschaltet. Markieren Sie das Kontrollkästchen, um den Servomodus anzuschalten.
3. Konfigurieren Sie den Datenrekorder.
   a) Stellen Sie als Größe des auszuführenden Sprungs einen Wert ein, der für Ihre Anwendung typisch ist, z. B. 0,100000 (Angabe in physikalischen Einheiten).
   b) Stellen Sie für die Aufzeichnungsrate des Datenrekorders im Feld ***Record Rate - # cycles*** den Wert 10 ein.

c) Stellen Sie für die Anzahl der für die grafische Darstellung auszulesenden Datenpunkte im Feld ***# of data points*** den Wert 1024 (oder weniger) ein.



d) Klicken Sie auf die Schaltfläche ***Configure…*** und stellen Sie im Fenster ***Configure Data Recorder*** sicher, dass als aufzuzeichnende Größen "Commanded Position of Axis" und "Actual Position of Axis" ausgewählt sind. Schließen Sie das Fenster mit ***OK***.



4. Starten Sie im Fenster ***Data Recorder*** den Sprung in positive Richtung sowie die Aufzeichnung durch Anklicken der Schaltfläche .

   Die Achse führt den Sprung aus, und die Sprungantwort wird aufgezeichnet und grafisch dargestellt.

5. Überprüfen Sie die dargestellte Sprungantwort (siehe untenstehende Beispiele).

   – Wenn nötig, vergrößern Sie die Darstellung, indem Sie auf die Schaltfläche klicken und mit gedrückter linker Maustaste den zur Lupe umgewandelten Mauszeiger über einen Bereich der grafischen Darstellung ziehen (ein Klick mit der rechten Maustaste in das Grafikfeld verkleinert die Darstellung wieder auf die ursprüngliche Größe).

Beispiele für Sprungantworten:

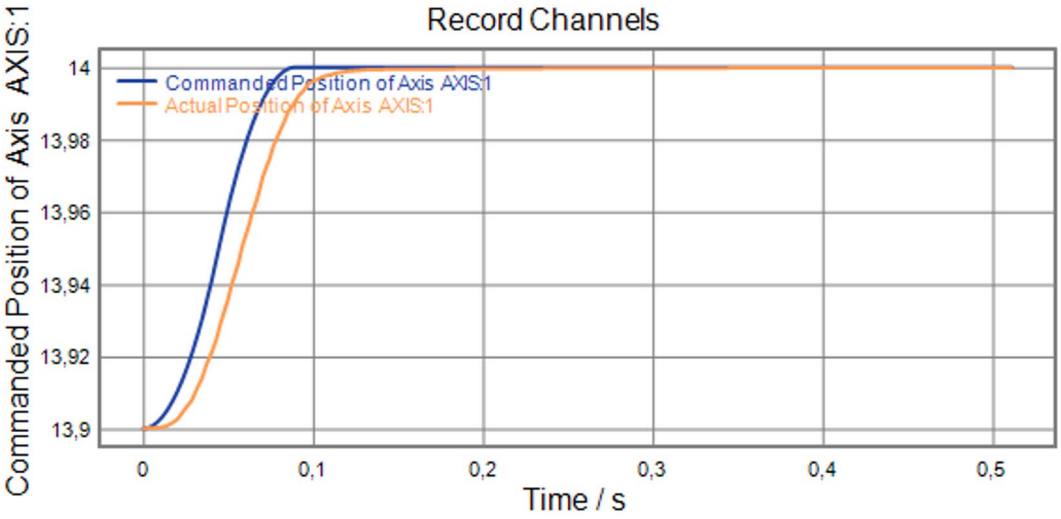


Abbildung 16: Zu langsames Einschwingen

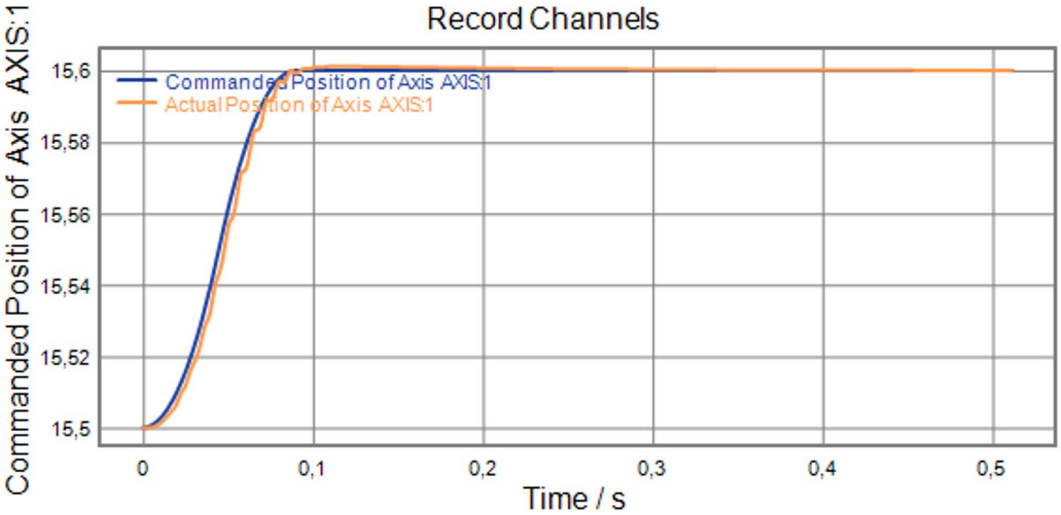


Abbildung 17: Schwingungen

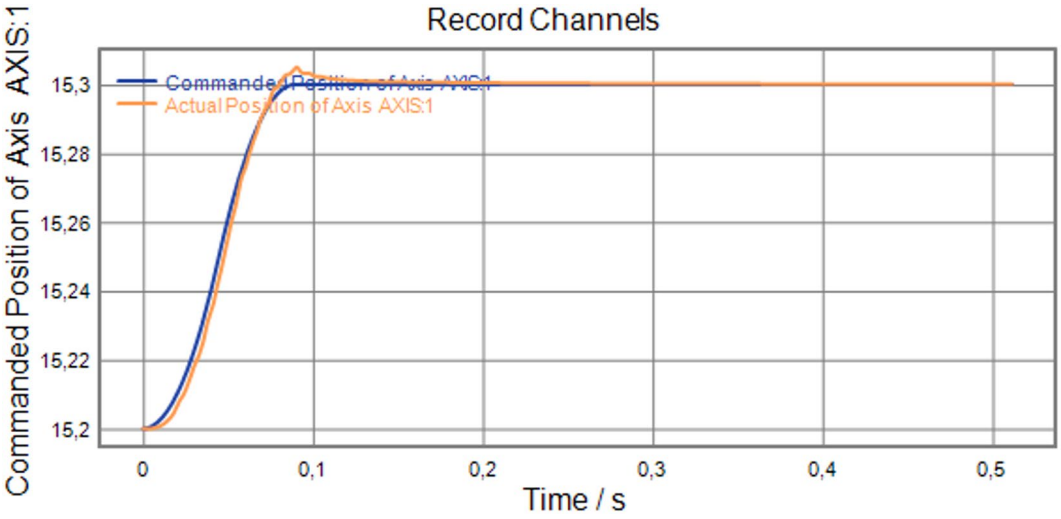


Abbildung 18: Starkes Überschwingen

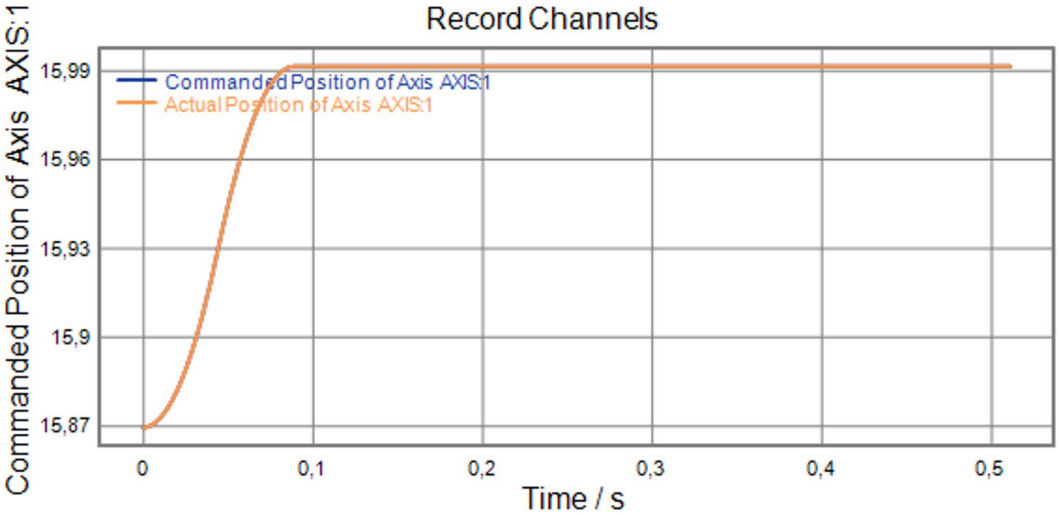


Abbildung 19: Optimales Einschwingverhalten (kommandierte und tatsächliche Position deckungsgleich)

Wenn das Ergebnis zufriedenstellend ist (d. h. geringstmögliches Überschwingen, Einschwingzeit nicht zu lang):

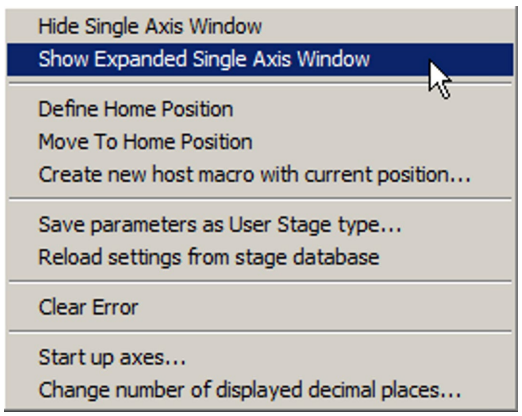
– Sie verfügen bereits über optimale Parametereinstellungen und brauchen nichts weiter zu unternehmen.

Wenn das Ergebnis nicht zufriedenstellend ist:

– Optimieren Sie die Regelparameter, siehe unten.

Regelparameter optimieren

1. Öffnen Sie im Hauptfenster von PIMikroMove® das erweiterte Einzelachsen-Fenster für den angeschlossenen Positionierer, indem Sie mit der rechten Maustaste auf die entsprechende Zeile der Registerkarte ***Axes*** klicken und im Kontextmenü ***Show Expanded Single Axis Window*** auswählen.

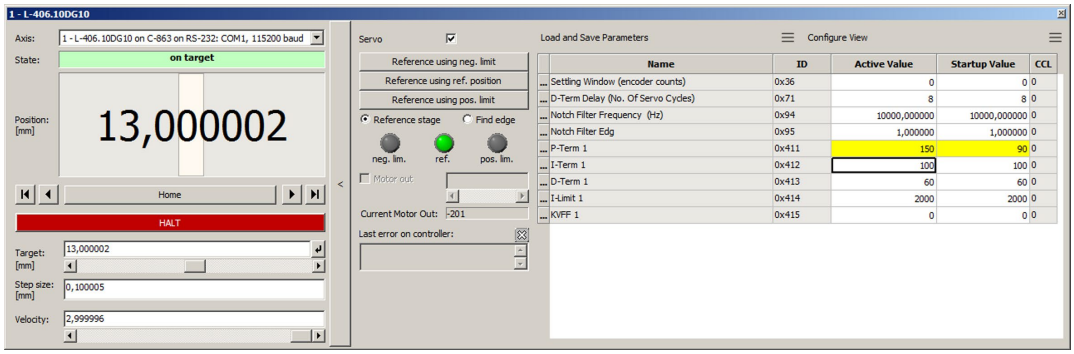


2. Geben Sie neue Werte für die anzupassenden Parameter ein:

   a) Wenn die zu ändernden Parameter nicht in der Liste auf der rechten Seite des Fensters enthalten sind, klicken Sie auf ***Configure View -> C-863 - CONTROL ALGORITHM***.

   b) Tippen Sie den neuen Parameterwert in das entsprechende Eingabefeld in der Spalte ***Active Value*** der Liste ein.

   c) Drücken Sie auf der Tastatur des PC die Enter-Taste oder klicken Sie mit der Maus außerhalb des Eingabefeldes, um den Parameterwert in den flüchtigen Speicher des Controllers zu übertragen. Anmerkung: Wenn ein Parameterwert im flüchtigen Speicher (Spalte ***Active Value***) vom Parameterwert im permanenten Speicher (Spalte ***Startup Value***) abweicht, ist die Zeile in der Liste farbig markiert.



3. Zeichnen Sie im Fenster ***Data Recorder*** erneut die Sprungantwort des Positionierers auf.

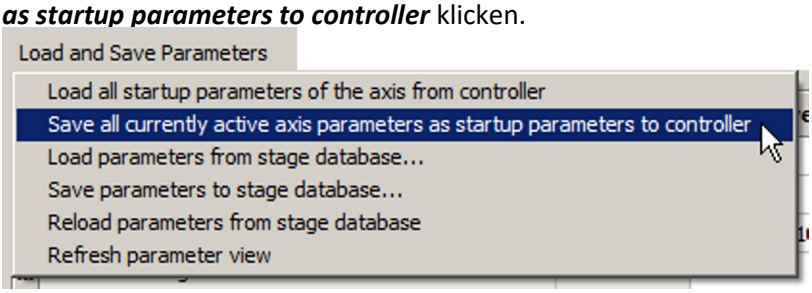
   Wenn das Ergebnis nicht zufriedenstellend ist:

   – Geben Sie im erweiterten Einzelachsen-Fenster andere Werte für die Regelparameter ein und zeichnen Sie die Sprungantwort erneut auf.

Wenn Sie mit dem Ergebnis zufrieden sind und die neuen Einstellungen der Regelparameter beibehalten wollen, speichern Sie die neuen Einstellungen. Sie haben folgende Möglichkeiten:

– Speichern Sie einen Parametersatz in der Positioniererdatenbank auf dem PC, indem Sie im erweiterten Einzelachsen-Fenster auf ***Load and Save Parameters -> Save parameters to stage database...*** klicken, siehe "Positionierertyp anlegen oder ändern" (S. 240).

– Übertragen Sie die aktuellen Werte der aufgelisteten Parameter aus dem flüchtigen in den permanenten Speicher des C-863.12, indem Sie im erweiterten Einzelachsen-Fenster auf ***Load and Save Parameters -> Save all currently active axis parameters as startup parameters to controller*** klicken.



7 Betrieb

7.1 Bewegungsfehler

7.1.1 Verhalten bei Bewegungsfehler

Bewegungsfehler können z. B. durch Störungen des Antriebs oder des Positionssensors des Positionierers verursacht werden.

Ein Bewegungsfehler liegt vor, wenn der Positionsfehler (d.h. der absolute Wert der Differenz zwischen der aktuellen Position und der kommandierten Position) im geregelten Betrieb den vorgegebenen Maximalwert überschreitet. Der Bereich, in dem die Abweichung liegen darf, ist durch den Parameter ***Maximum Position Error (Phys. Unit)*** (ID 0x8) festgelegt.

Wenn ein Bewegungsfehler auftritt, reagiert der C-863.12 wie folgt, um das System vor Schäden zu schützen:

- Der Servomodus wird für die betroffene Achse ausgeschaltet.
- Wenn vorhanden, wird die Bremse für die betroffene Achse aktiviert.
- Alle Bewegungen werden gestoppt.
- Der Fehlercode -1024 wird gesetzt.

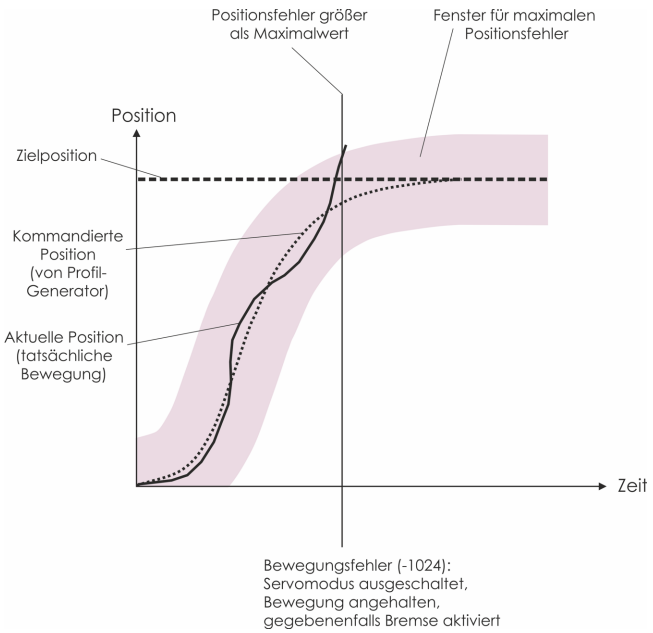


Abbildung 20: Verhalten bei Bewegungsfehlern

7.1.2 Betriebsbereitschaft wiederherstellen

HINWEIS

Unbeabsichtigte Bewegungen nach Bremsendeaktivierung!

Wenn der Servomodus ausgeschaltet ist, z. B. nach Auftreten eines Bewegungsfehlers, kann die Bremse des Positionierers per Befehl deaktiviert werden. Das Deaktivieren der Bremse kann unbeabsichtigte Bewegungen des Positionierers verursachen.

➢ Sichern Sie den Positionierer gegen unbeabsichtigte Bewegungen, bevor Sie die Bremse per Befehl deaktivieren!

Betriebsbereitschaft wiederherstellen

1. Senden Sie den Befehl `ERR?`, um den Fehlercode auszulesen.

   Wenn ein Bewegungsfehler vorliegt, wird der Fehlercode -1024 ausgegeben. `ERR?` setzt den Fehlercode bei der Abfrage auf null zurück.

2. Überprüfen Sie Ihr System und vergewissern Sie sich, dass alle Achsen gefahrlos bewegt werden können.

3. Schalten Sie mit dem Befehl `SVO` (S. 198) den Servomodus für die betroffene Achse ein.

   Beim Einschalten des Servomodus wird die Zielposition auf die aktuelle Achsenposition gesetzt und gegebenenfalls die Bremse deaktiviert. Die Achse kann nun wieder bewegt werden, und Sie können eine neue Zielposition kommandieren.

INFORMATION

Mit den Befehlen `CTO` (S. 140) und `TRO` (S. 203) können Sie die digitalen Ausgangsleitungen des C-863.12 so programmieren, dass sie bei Bewegungsfehlern aktiviert werden. Die programmierten Ausgangsleitungen bleiben aktiv, bis der Fehlercode auf 0 zurückgesetzt wird. Details siehe "Triggermodus "Motion Error" einrichten" (S. 82).

7.2 Datenrekorder

7.2.1 Datenrekorder einrichten

Der C-863.12 enthält einen Echtzeit-Datenrekorder. Der Datenrekorder kann z. B. die aktuelle Position der Achse aufzeichnen.

Die aufgezeichneten Daten werden temporär in 4 Datenrekordertabellen mit jeweils 1024 Punkten gespeichert. Jede Datenrekordertabelle enthält die Daten einer Datenquelle.

Sie können den Datenrekorder konfigurieren, indem Sie z. B. den aufzuzeichnenden Datentyp und die Datenquellen bestimmen und festlegen, wie die Aufzeichnung gestartet werden soll.

INFORMATION

Die Einstellungen zur Einrichtung des Datenrekorders lassen sich nur im flüchtigen Speicher des C-863.12 ändern. Nach dem Einschalten oder dem Neustart des C-863.12 sind Standardeinstellungen aktiv, wenn nicht durch ein Startup-Makro bereits eine Konfiguration erfolgt.

Allgemeine Informationen über den Datenrekorder auslesen

- Senden Sie den Befehl `HDR?` (S. 161).

  Die verfügbaren Aufzeichnungs- und Triggeroptionen sowie Informationen über zusätzliche Parameter und Befehle für die Datenaufzeichnung werden angezeigt.

Aufzuzeichnende Daten konfigurieren

Sie können den Datenrekordertabellen die Datenquellen und die Aufzeichnungsoptionen zuordnen.

- Senden Sie den Befehl `DRC?` (S. 151), um die aktuelle Konfiguration auszulesen. Datenrekordertabellen mit Aufzeichnungsoption 0 sind deaktiviert, d. h. es wird nichts aufgezeichnet. In der Standardeinstellung zeichnen die Datenrekordertabellen des C-863.12 Folgendes auf:
  - Datenrekordertabelle 1: Aufzeichnungsoption 1: kommandierte Position der Achse
  - Datenrekordertabelle 2: Aufzeichnungsoption 2: aktuelle Position der Achse
  - Datenrekordertabelle 3: Aufzeichnungsoption 3: Positionsfehler der Achse
  - Datenrekordertabelle 4: Aufzeichnungsoption 73: Stellwert der Achse
- Konfigurieren Sie den Datenrekorder mit dem Befehl `DRC` (S. 149).

Auslösen der Aufzeichnung konfigurieren

Sie können festlegen, wie die Aufzeichnung ausgelöst werden soll.

- Fragen Sie mit `DRT?` (S. 154) die aktuelle Triggeroption ab.
- Ändern Sie die Triggeroption mit dem Befehl `DRT` (S. 153). Die Triggeroption gilt für alle Datenrekordertabellen, deren Aufzeichnungsoption nicht auf 0 eingestellt ist.

Aufzeichnungsrate einstellen

- Senden Sie den Befehl `RTR?` (S. 188), um die Aufzeichnungsrate des Datenrekorders auszulesen.

  Die Aufzeichnungsrate gibt an, nach wie vielen Servozyklen jeweils ein Datenpunkt aufgezeichnet wird. Der Standardwert beträgt 10 Servozyklen. Die Servozykluszeit des C-863.12 beträgt 50 µs.
- Ändern Sie die Aufzeichnungsrate mit dem Befehl `RTR` (S. 188).

  Mit zunehmender Aufzeichnungsrate erhöhen Sie die maximale Dauer der Datenaufzeichnung.

7.2.2 Aufzeichnung starten

- Starten Sie die Aufzeichnung durch die mit `DRT` eingestellte Triggeroption.

Unabhängig von der eingestellten Triggeroption wird die Datenaufzeichnung immer ausgelöst, wenn eine Sprungantwortmessung mit `STE` (S. 197) gestartet wird.

Die Datenaufzeichnung erfolgt immer für alle Datenrekordertabellen, deren Aufzeichnungsoption nicht auf 0 eingestellt ist. Sie endet, wenn die Datenrekordertabellen voll sind.

7.2.3 Aufgezeichnete Daten auslesen

INFORMATION

Das Auslesen der aufgezeichneten Daten kann abhängig von der Anzahl der Datenpunkte einige Zeit dauern.
Die Daten können auch bei laufender Datenaufzeichnung ausgelesen werden.

- Lesen Sie die zuletzt aufgezeichneten Daten mit dem Befehl `DRR?` (S. 152) aus.

  Die Daten werden im GCS-Array-Format ausgegeben (siehe Benutzerhandbuch SM146E).
- Fragen Sie die Anzahl der in der letzten Aufzeichnung enthaltenen Punkte mit dem Befehl `DRL?` (S. 151) ab.

7.3 Digitale Ausgangssignale

Die digitalen Ausgänge des C-863.12 sind auf der Buchse **I/O** (S. 270) verfügbar.

- Fragen Sie die Anzahl der am C-863.12 verfügbaren Ausgangsleitungen mit dem Befehl `TIO?` (S. 201) ab.

Über die digitalen Ausgänge des C-863.12 können externe Geräte getriggert werden. Anwendungsmöglichkeiten:

- Koppeln der Triggerausgabe an die Bewegung der Achse. Details und Beispiele finden Sie in diesem Abschnitt.
- Direktes Schalten der Ausgangsleitungen, z. B. in Makros. Details und Beispiele zu Makros finden Sie unter "Controllermakros" (S. 100).

7.3.1 Befehle für digitale Ausgänge

Folgende Befehle stehen für die Verwendung digitaler Ausgänge zur Verfügung:

Befehl	Syntax	Funktion
`CTO`	CTO {<TrigOutID> <CTOPam> <Value>}	Konfiguriert die Bedingungen für die Triggerausgabe. Koppelt die Triggerausgabe an die Achsenbewegung.
`DIO`	DIO {<DIOID> <OutputOn>}	Schaltet digitale Ausgangsleitungen direkt in den Zustand low oder high, entweder einzeln oder alle Leitungen auf einmal. Sollte nicht für Ausgangsleitungen verwendet werden, bei denen die Triggerausgabe mit `TRO` aktiviert ist.
`TRO`	TRO {<TrigOutID> <TrigMode>}	Aktiviert oder deaktiviert die mit `CTO` gesetzten Bedingungen der Triggerausgabe. Standard: Triggerausgabe deaktiviert.

Pro Befehl `CTO` kann eine Konfigurationseinstellung vorgenommen werden:

`CTO <TrigOutID> <CTOPam> <Value>`

- `<TrigOutID>` ist eine digitale Ausgangsleitung des Controllers.
- `<CTOPam>` ist die CTO-Parameter-ID im Dezimalformat.
- `<Value>` ist der Wert, auf den der CTO-Parameter eingestellt ist.

Folgende Triggermodi (<Value>) können für <CTOPam> = 3 eingestellt werden:

<Value>	Triggermodus	Kurzbeschreibung
0 (Standard)	Position Distance	Sobald sich die Achse um eine vorgegebene Strecke bewegt hat, wird ein Triggerpuls ausgegeben (S. 80). Optional können Start- und Stoppwerte definiert werden, um die Triggerung auf einen Positionsbereich und eine bestimmte Bewegungsrichtung (negativ oder positiv) zu beschränken.
2	On Target	Der On-Target-Status der gewählten Achse wird am gewählten Triggerausgang ausgegeben (S. 82).
5	Motion Error	Die gewählte digitale Ausgangsleitung wird aktiv, wenn ein Bewegungsfehler auftritt (S. 82). Die Leitung bleibt aktiv, bis der Fehlercode auf 0 zurückgesetzt wird (durch eine Abfrage mit `ERR?`).
6	In Motion	Die gewählte digitale Ausgangsleitung ist solange aktiv, wie die gewählte Achse in Bewegung ist (S. 83).
7	Position+Offset	Der erste Triggerpuls wird ausgegeben, wenn die Achse eine vorgegebene Position erreicht hat. Die nächsten Triggerpulse werden jeweils ausgegeben, wenn die Achsenposition gleich der Summe der letzten gültigen Triggerposition und einer vorgegebenen Strecke ist. Die Triggerausgabe wird beim Erreichen eines Stoppwertes beendet. Das Vorzeichen des Streckenwertes bestimmt, für welche Bewegungsrichtung Triggerpulse ausgegeben werden sollen. Details siehe "Triggermodus "Position +

<Value>	Triggermodus	Kurzbeschreibung
		Offset" einrichten" (S. 84).
8	Single Position	Die gewählte digitale Ausgangsleitung ist aktiv, wenn die Achsenposition eine vorgegebene Position erreicht hat oder überschreitet (S. 85).

Darüber hinaus kann die Polarität (high-aktiv / low-aktiv) des Signals am digitalen Ausgang eingestellt werden (S. 86).

INFORMATION

Die Einstellungen zur Konfiguration der digitalen Ausgangsleitungen lassen sich nur im flüchtigen Speicher des C-863.12 ändern. Nach dem Einschalten oder dem Neustart des C-863.12 sind werkseitige Standardeinstellungen aktiv, sofern nicht durch ein Startup-Makro bereits eine Konfiguration erfolgt.

7.3.2 Triggermodus "Position Distance" einrichten

Der Triggermodus *Position Distance* eignet sich für Scananwendungen. Sobald sich die Achse die Strecke weiterbewegt hat, die mit der CTO-Parameter-ID = 1 (TriggerStep) eingestellt wurde, wird ein Triggerpuls ausgegeben. Die Pulsweite beträgt einen Servozyklus.

Die Einheit der Strecke (TriggerStep) hängt von den Einstellungen der Parameter 0xE und 0xF ab. Standard ist mm.

1. Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:
   - Senden Sie `CTO <TrigOutID> 2 A`, wobei *A* die zu bewegende Achse bezeichnet.
   - Senden Sie `CTO <TrigOutID> 3 0`, wobei 0 den Triggermodus *Position Distance* bestimmt.
   - Senden Sie `CTO <TrigOutID> 1 S`, wobei *S* die Strecke bezeichnet.
2. Wenn Sie die Bedingungen für die Triggerausgabe aktivieren wollen, senden Sie `TRO <TrigOutID> 1`.

Beispiel:

Auf der digitalen Ausgangsleitung 1 wird immer dann ein Puls ausgegeben, wenn die Achse 1 des Positionierers eine Strecke von 0,1 µm zurückgelegt hat.

➢ Senden Sie:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
TRO 1 1
```

Triggermodus "Position Distance" mit Start- und Stoppwerten für positive Achsenbewegungsrichtung

Optional können Sie für die Triggerung Start- und Stoppwerte zur Begrenzung des Bereichs und zur Bestimmung der Achsenbewegungsrichtung (positiv oder negativ) festlegen.

INFORMATION

Wenn Start- und Stoppwert denselben Wert haben, werden sie ignoriert.
Wenn sich die Bewegungsrichtung umkehrt, bevor die Achsenposition den Stoppwert erreicht hat, werden weiterhin Triggerpulse ausgegeben.

1. Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:
   - Senden Sie `CTO <TrigOutID> 2 A`, wobei *A* die zu bewegende Achse bezeichnet.
   - Senden Sie `CTO <TrigOutID> 3 0`, wobei 0 den Triggermodus *Position Distance* bestimmt.
   - Senden Sie `CTO <TrigOutID> 1 S`, wobei *S* die Strecke bezeichnet.
   - Senden Sie `CTO <TrigOutID> 8 Start`, wobei *Start* den Startwert bezeichnet.
   - Senden Sie `CTO <TrigOutID> 9 Stopp`, wobei *Stopp* den Stoppwert bezeichnet.
2. Wenn Sie die Bedingungen für die Triggerausgabe aktivieren wollen, senden Sie `TRO <TrigOutID> 1`.

Beispiel

Auf der digitalen Ausgangsleitung 1 wird immer dann ein Puls ausgegeben, wenn die Achse 1 des Positionierers eine Strecke von 0,1 µm zurückgelegt hat, solange sich Achse 1 in positiver Bewegungsrichtung im Bereich von 0,2 µm bis 0,55 µm bewegt (Startwert < Stoppwert).

➢ Senden Sie:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
CTO 1 8 0.0002
CTO 1 9 0.00055
TRO 1 1
```

Triggermodus "Position Distance" mit Start- und Stoppwerten für negative Achsenbewegungsrichtung

Im Folgenden ist das Beispiel von oben mit vertauschten Start- und Stoppwerten gezeigt. Die Triggerung erfolgt in negativer Achsenbewegungsrichtung (Stoppwert < Startwert) im Bereich zwischen 0,55 µm und 0,2 µm.

Beispiel:

➢ Senden Sie:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
CTO 1 8 0.00055
CTO 1 9 0.0002
TRO 1 1
```

7.3.3 Triggermodus "On Target" einrichten

Im Triggermodus *On Target* wird der On-Target-Status der gewählten Achse (S. 29) am gewählten Triggerausgang ausgegeben.

1. Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:
   – Senden Sie `CTO <TrigOutID> 2 A`, wobei *A* die zu bewegende Achse bezeichnet.
   – Senden Sie `CTO <TrigOutID> 3 2`, wobei *2* den Triggermodus *On Target* bestimmt.
2. Wenn Sie die Bedingungen für die Triggerausgabe aktivieren wollen, senden Sie `TRO <TrigOutID> 1`.

Beispiel:

Der On-Target-Status von Achse 1 soll auf der digitalen Ausgangsleitung 1 ausgegeben werden.

➢ Senden Sie:

```
CTO 1 2 1
CTO 1 3 2
TRO 1 1
```

7.3.4 Triggermodus "Motion Error" einrichten

Der Triggermodus *Motion Error* eignet sich für die Überwachung von Bewegungen. Die gewählte digitale Ausgangsleitung wird aktiv, wenn auf einer der angeschlossenen Achsen ein Bewegungsfehler auftritt. Die Leitung bleibt aktiv, bis der Fehlercode auf 0 zurückgesetzt wird (durch eine Abfrage mit `ERR?`).

INFORMATION

Ein Bewegungsfehler liegt vor, wenn die aktuelle Position während der Bewegung zur sehr von der kommandierten Position abweicht.
Weitere Informationen siehe "Bewegungsfehler" (S. 75).

1. Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:
   – Senden Sie `CTO <TrigOutID> 3 5`, wobei 5 den Triggermodus *Motion Error* bestimmt.
2. Wenn Sie die Bedingungen für die Triggerausgabe aktivieren wollen, senden Sie `TRO <TrigOutID> 1`.

7.3.5 Triggermodus "In Motion" einrichten

Im Triggermodus *In Motion* wird der Bewegungsstatus der gewählten Achse am gewählten Triggerausgang ausgegeben. Die Leitung ist aktiv, solange die gewählte Achse in Bewegung ist.

Der Bewegungsstatus kann auch mit den Befehlen `#5` (S. 131), `#4` (S. 130) und `SRG?` (S. 196) ausgelesen werden.

INFORMATION

Wenn die Achse in Bewegung ist, dann ist Bit 14 des Statusregisters 1 der Achse gesetzt.

1. Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:
   – Senden Sie `CTO <TrigOutID> 2 A`, wobei *A* die zu bewegende Achse bezeichnet.
   – Senden Sie `CTO <TrigOutID> 3 6`, wobei 6 den Triggermodus *In Motion* bestimmt.
2. Wenn Sie die Bedingungen für die Triggerausgabe aktivieren wollen, senden Sie `TRO <TrigOutID> 1`.

Beispiel:

Die digitale Ausgangsleitung 1 soll aktiv sein, wenn die Achse 1 des Positionierers in Bewegung ist.

➢ Senden Sie:

```
CTO 1 2 1
CTO 1 3 6
TRO 1 1
```

7.3.6 Triggermodus "Position + Offset" einrichten

Der Triggermodus *Position+Offset* eignet sich für Scananwendungen. Der erste Triggerpuls wird ausgegeben, wenn die Achse eine vorgegebene Position erreicht hat (TriggerPosition). Die nächsten Triggerpulse werden jeweils ausgegeben, wenn die Achsenposition gleich der Summe der letzten gültigen Triggerposition und einer vorgegebenen Strecke (TriggerStep) ist. Die Triggerausgabe wird beim Erreichen eines Stoppwertes beendet. Das Vorzeichen des Streckenwertes bestimmt, für welche Bewegungsrichtung Triggerpulse ausgegeben werden sollen.

Die Pulsweite beträgt einen Servozyklus.

Die Einheit für TriggerPosition, TriggerStep und Stoppwert hängt von den Einstellungen der Parameter 0xE und 0xF ab. Standard ist mm.

1. Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:
   - Senden Sie `CTO <TrigOutID> 2 A`, wobei *A* die zu bewegende Achse bezeichnet.
   - Senden Sie `CTO <TrigOutID> 3 7`, wobei 7 den Triggermodus *Position+Offset* bestimmt.
   - Senden Sie `CTO <TrigOutID> 1 S`, wobei *S* die Strecke bezeichnet.
   - Senden Sie `CTO <TrigOutID> 10 TriPos`, wobei *TriPos* die Position für die Ausgabe des ersten Triggerpulses bezeichnet.
   - Senden Sie `CTO <TrigOutID> 9 Stopp`, wobei *Stopp* den Stoppwert bezeichnet.
2. Wenn Sie die Bedingungen für die Triggerausgabe aktivieren wollen, senden Sie `TRO <TrigOutID> 1`.

Beispiel 1:

Auf der digitalen Ausgangsleitung 1 soll der erste Triggerpuls ausgegeben werden, wenn die absolute Position von Achse 1 1,5 mm beträgt. Danach soll auf dieser Leitung immer dann ein Puls ausgegeben werden, wenn Achse 1 eine Strecke von 0,1 µm in positiver Richtung zurückgelegt hat. Der letzte Triggerpuls soll ausgegeben werden, wenn die absolute Achsenposition 2,5 mm beträgt.

➢ Senden Sie:

```
CTO 1 2 1
CTO 1 3 7
CTO 1 1 0.0001
CTO 1 10 1.5
CTO 1 9 2.5
TRO 1 1
```

Beispiel 2:

Auf der digitalen Ausgangsleitung 2 soll der erste Triggerpuls ausgegeben werden, wenn die absolute Position von Achse B 0,4 mm beträgt. Danach soll auf dieser Leitung immer dann ein Puls ausgegeben werden, wenn Achse B eine Strecke von 1 µm in negativer Richtung zurückgelegt hat. Der letzte Triggerpuls soll ausgegeben werden, wenn die absolute Achsenposition 0,1 mm beträgt.

➢ Senden Sie:

```
CTO 2 2 B
CTO 2 3 7
CTO 2 1 -0.001
CTO 2 10 0.4
CTO 2 9 0.1
```

INFORMATION

Die Geschwindigkeitseinstellung der Achse muss für die mit dem Befehl `CTO` kommandierte Streckeneinstellung (TriggerStep) geeignet sein. Empfohlener Wert:
Maximale Geschwindigkeit = Strecke * 20 kHz / 2
wobei 20 kHz die Frequenz des Servozyklus des C-863.12 ist.

7.3.7 Triggermodus "Single Position" einrichten

Im Triggermodus *Single Position* ist die gewählte digitale Ausgangsleitung aktiv, wenn die Achsenposition eine vorgegebene Position (TriggerPosition) erreicht hat oder überschreitet.

Die Einheit für TriggerPosition hängt von den Einstellungen der Parameter 0xE und 0xF ab. Standard ist mm.

1. Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:
   - Senden Sie `CTO <TrigOutID> 2 A`, wobei *A* die zu bewegende Achse bezeichnet.
   - Senden Sie `CTO <TrigOutID> 3 8`, wobei 8 den Triggermodus *Single Position* bestimmt.
   - Senden Sie `CTO <TrigOutID> 10 TriPos`, wobei *TriPos* die Position bezeichnet, ab der die Ausgangsleitung aktiv sein soll.
2. Wenn Sie die Bedingungen für die Triggerausgabe aktivieren wollen, senden Sie `TRO <TrigOutID> 1`.

Beispiel:

Die digitale Ausgangsleitung 1 soll aktiv sein, wenn die absolute Position von Achse 1 mindestens 1,5 mm beträgt.

➢ Senden Sie:

```
CTO 1 2 1
```

```
CTO 1 3 8
CTO 1 10 1.5
```

7.3.8 Signalpolarität einstellen

Mit dem CTO-Parameter *Polarity* kann die Polarität des Signals am digitalen Ausgang, der zur Triggerung verwendet wird, gewählt werden. Die Polarität kann folgende Werte haben:

- high-aktiv = 1 (Standardeinstellung)
- low-aktiv = 0

➢ Konfigurieren Sie die digitale Ausgangsleitung (<TrigOutID>), die als Triggerausgang verwendet werden soll:

  – Senden Sie `CTO <TrigOutID> 7 P`, wobei *P* die Polarität bezeichnet.

Beispiel:

Die Signalpolarität für die digitale Ausgangsleitung 1 soll auf low-aktiv eingestellt werden.

➢ Senden Sie:

```
CTO 1 7 0
```

7.4 Digitale Eingangssignale

Die digitalen Eingänge des C-863.12 sind auf der Buchse **I/O** (S. 270) verfügbar.

➢ Fragen Sie die Anzahl der am C-863.12 verfügbaren Eingangsleitungen mit dem Befehl `TIO?` (S. 201) ab.

➢ Fragen Sie den Status der Eingangsleitungen mit dem Befehl `DIO?` (S. 149) ab.

Anwendungsmöglichkeiten:

- Verwendung in Makros (S. 89). Details und Beispiele zu Makros finden Sie unter "Controllermakros" (S. 100).
- Verwendung als Schaltersignale (S. 89)

INFORMATION

Die digitalen Eingänge (Pins 1 bis 4) auf der Buchse **I/O** können auch als analoge Eingänge genutzt werden.

- Digital: TTL
- Analog: 0 bis +5 V

7.4.1 Befehle und Parameter für digitale Eingänge

Befehle

Folgende Befehle stehen für die Verwendung digitaler Eingänge zur Verfügung:

Befehl	Syntax	Funktion
CPY	CPY <Variable> <CMD?>	Kopiert den Status einer digitalen Eingangsleitung in eine Variable, wenn in Kombination mit dem Abfragebefehl DIO? verwendet. Verwendung in Makros zum Setzen lokaler Variablen (S. 123).
DIO?	DIO? [{<DIOID>}]	Fragt den Status digitaler Eingangsleitungen ab.
FED	FED {<AxisID> <EdgeID> <Param>}	Startet eine Fahrt zu einer Signalflanke. Die Quelle des Signals kann eine digitale Eingangsleitung sein.
FNL	FNL [{<AxisID>}]	Startet eine Referenzierungsfahrt zur negativen physikalischen Grenze des Stellwegs. Eine digitale Eingangsleitung kann anstelle des negativen Endschalters als Quelle des negativen Endschaltersignals verwendet werden.
FPL	FPL [{<AxisID>}]	Startet eine Referenzierungsfahrt zur positiven physikalischen Grenze des Stellwegs. Endschalter. Eine digitale Eingangsleitung kann anstelle des positiven Endschalters als Quelle des positiven Endschaltersignals verwendet werden.
FRF	FRF [{<AxisID>}]	Startet eine Referenzierungsfahrt zum Referenzschalter. Eine digitale Eingangsleitung kann anstelle des Referenzschalters als Quelle des Referenzschaltersignals verwendet werden.
JRC	JRC <Jump> <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Löst einen relativen Sprung des Makroausführungszeigers in Abhängigkeit vom Status einer digitalen Eingangsleitung aus, wenn in Kombination mit dem Abfragebefehl DIO? verwendet.
MEX	MEX <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Stoppt Makroausführung in Abhängigkeit vom Status einer digitalen Eingangsleitung, wenn in Kombination mit dem Abfragebefehl DIO? verwendet.
WAC	WAC <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Wartet, bis eine digitale Eingangsleitung einen bestimmten Status erreicht, wenn in Kombination mit dem Abfragebefehl DIO? verwendet.

Parameter

Folgende Parameter stehen für die Konfiguration digitaler Eingänge zur Verfügung:

Parameter	Beschreibung und mögliche Werte
Source Of Reference Signal 0x5C	Bestimmt die Quelle des Referenzsignals für die Befehle `FRF` und `FED`: 0 = Referenzschalter 1 = Digitaler Eingang 1 2 = Digitaler Eingang 2 3 = Digitaler Eingang 3 4 = Digitaler Eingang 4
Source Of Negative Limit Signal 0x5D	Bestimmt die Quelle(n) des negativen Endschaltersignals für die Befehle `FNL` und `FED` über eine Bitmaske: 0 = Negativer Endschalter (Standard-Einstellung) 1 = Digitaler Eingang 1 (Bit 0) 2 = Digitaler Eingang 2 (Bit 1) 4 = Digitaler Eingang 3 (Bit 2) 8 = Digitaler Eingang 4 (Bit 3)
Source Of Positive Limit Signal 0x5E	Bestimmt die Quelle(n) des positiven Endschaltersignals für die Befehle `FPL` und `FED` über eine Bitmaske: 0 = Positiver Endschalter (Standard-Einstellung) 1 = Digitaler Eingang 1 (Bit 0) 2 = Digitaler Eingang 2 (Bit 1) 4 = Digitaler Eingang 3 (Bit 2) 8 = Digitaler Eingang 4 (Bit 3)
Invert Digital Input Used For Negative Limit 0x5F	Invertiert die Polarität der digitalen Eingänge, die als Quellen des negativen Endschaltersignals dienen, über eine Bitmaske: 0 = Kein digitaler Eingang invertiert (Standard-Einstellung). 1 = Digitaler Eingang 1 invertiert (Bit 0) 2 = Digitaler Eingang 2 invertiert (Bit 1) 4 = Digitaler Eingang 3 invertiert (Bit 2) 8 = Digitaler Eingang 4 invertiert (Bit 3)
Invert Digital Input Used For Positive Limit 0x60	Invertiert die Polarität der digitalen Eingänge, die als Quellen des positiven Endschaltersignals dienen, über eine Bitmaske: 0 = Kein digitaler Eingang invertiert (Standard-Einstellung). 1 = Digitaler Eingang 1 invertiert (Bit 0) 2 = Digitaler Eingang 2 invertiert (Bit 1) 4 = Digitaler Eingang 3 invertiert (Bit 2) 8 = Digitaler Eingang 4 invertiert (Bit 3)

7.4.2 Digitale Eingangssignale in Makros verwenden

Die digitalen Eingänge auf der Buchse **I/O** können in Makros wie folgt verwendet werden:

- Bedingte Ausführung des Makros
- Bedingtes Stoppen der Makroausführung
- Bedingter Sprung des Makroausführungs-Zeigers
- Kopieren des Eingangszustands in eine Variable

Weitere Informationen und Beispiele finden Sie unter "Controllermakros" (S. 100).

INFORMATION

Um die digitalen Eingangssignale für die Verwendung in Makros zu erzeugen, können Sie die Pushbutton-Box C-170.PB von PI an die Buchse **I/O** (S. 270) anschließen. Sie zeigt über LEDs auch den Status der digitalen Ausgangsleitungen an.

7.4.3 Digitale Eingangssignale als Schaltersignale verwenden

Die digitalen Eingänge auf der Buchse **I/O** können als Quelle der Referenz- und Endschaltersignale (z. B. für Referenzierungsfahrten (S. 35)) für eine Achse verwendet werden.

Digitalen Eingang als Referenzsignal verwenden

INFORMATION

Der Pegel des digitalen Eingangssignals, das Sie anstelle des Referenzschalters verwenden, darf sich über den gesamten Stellweg hinweg nur einmal ändern.

- ➢ Verwenden Sie eine geeignete Signalquelle.
- ➢ Wenn nötig, invertieren Sie die Signallogik der digitalen Eingangsleitung durch entsprechendes Setzen des Parameters ***Invert Reference?*** (ID 0x31).

INFORMATION

Der Parameter ***Has Reference?*** (ID 0x14) hat keinen Einfluss auf die Verwendung einer digitalen Eingangsleitung als Quelle des Referenzsignals.

- ➢ Wählen Sie die Quelle des Referenzsignals für die Achse aus, indem Sie den Parameter ***Source Of Reference Signal*** (ID 0x5C) ändern.

Genaue Informationen zum Ändern von Parametern finden Sie in "Anpassen von Einstellungen" (S. 235).

Digitale Eingänge als Quelle der Endschaltersignale verwenden

INFORMATION

Mehrere digitale Eingänge können als Quelle für ein Endschaltersignal ausgewählt sein.

Wenn ein Endschaltersignal für Referenzierungsfahrten verwendet wird, darf nur eine digitale Eingangsleitung als Quelle des Endschaltersignals ausgewählt sein.

INFORMATION

Der Pegel des digitalen Eingangssignals, das Sie anstelle eines eingebauten Endschalters verwenden, darf sich über den gesamten Stellweg hinweg nur einmal ändern.

- Verwenden Sie geeignete Signalquellen.
- Wenn nötig, invertieren Sie die Signallogik der digitalen Eingangsleitungen durch entsprechendes Setzen der Parameter ***Invert Digital Input Used For Negative Limit*** (ID 0x5F) und ***Invert Digital Input Used For Positive Limit*** (ID 0x60).

INFORMATION

Der Parameter ***Has No Limit Switches?*** (ID 0x32) legt fest, ob der C-863.12 die Signale der eingebauten Endschalter des Positionierers auswertet. Auf die Verwendung von digitalen Eingangsleitungen als Quelle des Endschaltersignals hat dieser Parameter keinen Einfluss.

- Wählen Sie die Quelle(n) des negativen Endschaltersignals für die Achse aus, indem Sie den Parameter ***Source Of Negative Limit Signal*** (ID 0x5D) ändern.
- Wählen Sie die Quelle(n) des positiven Endschaltersignals für die Achse aus, indem Sie den Parameter ***Source Of Positive Limit Signal*** (ID 0x5E) ändern.

Genaue Informationen zum Ändern von Parametern finden Sie in "Anpassen von Einstellungen" (S. 235).

Beispiel:

Die digitalen Eingangsleitungen 1, 3 und 4 sollen für Achse 1 als Quellen des positiven Endschaltersignals verwendet werden. Außerdem soll für Achse 1 die Signalpolarität der Leitungen 1 und 3 invertiert werden. Alle Anpassungen werden nur im flüchtigen Speicher des C-863.12 vorgenommen.

- Senden Sie:

  `SPA 1 0x5E 13`, um die Leitungen 1, 3 und 4 auszuwählen.

  `SPA 1 0x60 5`, um die Signalpolarität der Leitungen 1 und 3 zu invertieren.

7.5 Analoge Eingangssignale

Die analogen Eingänge des C-863.12 sind auf der Buchse **I/O** (S. 270) verfügbar.

- Fragen Sie die Anzahl der am C-863.12 verfügbaren analogen Eingangsleitungen mit dem Befehl `TAC?` (S. 200) ab.
- Fragen Sie die Spannung an den Analogeingängen mit dem Befehl `TAV?` (S. 200) ab.
- Verwenden Sie den Datenrekorder (S. 76), um die Signale der analogen Eingänge aufzuzeichnen.

Anwendungsmöglichkeiten:

- Verwendung in Makros (S. 92): Details und Beispiele zu Makros finden Sie unter "Controllermakros" (S. 100).
- Scan-Anwendungen mit PIMikroMove® (siehe PIMikroMove® Handbuch)

INFORMATION

Die analogen Eingänge (Pins 1 bis 4) auf der Buchse **I/O** können auch als digitale Eingänge genutzt werden.

- Analog: 0 bis +5 V
- Digital: TTL

7.5.1 Befehle für analoge Eingänge

Folgende Befehle stehen für die Verwendung analoger Eingänge zur Verfügung:

Befehl	Syntax	Funktion
`CPY`	CPY <Variable> <CMD?>	Kopiert den Spannungswert einer analogen Eingangsleitung in eine Variable, wenn in Kombination mit dem Abfragebefehl TAV? verwendet. Verwendung in Makros zum Setzen lokaler Variablen (S. 123).
`DRC`	DRC {<RecTableID> <Source> <RecOption>}	Konfiguriert den Datenrekorder. Mit der Aufzeichnungsoption 81 können die Werte der analogen Eingänge aufgenommen werden.
`JRC`	JRC <Jump> <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Löst einen relativen Sprung des Makroausführungszeigers in Abhängigkeit von der Spannung an einer analogen Eingangsleitung aus, wenn in Kombination mit dem Abfragebefehl TAV? verwendet.
`MEX`	MEX <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Stoppt Makroausführung in Abhängigkeit von der Spannung an einer analogen Eingangsleitung, wenn in Kombination mit dem Abfragebefehl TAV? verwendet.
`TAC?`	TAC?	Fragt die Anzahl installierter Analogleitungen ab.

Befehl	Syntax	Funktion
`TAV?`	TAV? [{<AnalogInputID>}]	Fragt die Spannung am Analogeingang ab.
`WAC`	WAC <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Wartet, bis eine analoge Eingangsleitung eine bestimmte Spannung erreicht, wenn in Kombination mit dem Abfragebefehl TAV? verwendet.

7.5.2 Analoge Eingangssignale in Makros verwenden

Die analogen Eingänge auf der Buchse **I/O** können in Makros wie folgt verwendet werden:

- Bedingte Ausführung des Makros
- Bedingtes Stoppen der Makroausführung
- Bedingter Sprung des Makroausführungs-Zeigers
- Kopieren des Eingangszustands in eine Variable

Weitere Informationen und Beispiele finden Sie unter "Controllermakros" (S. 100).

7.6 Joystick-Steuerung

7.6.1 Funktionsweise der Joystick-Steuerung

Die Joystick-Achse steuert die Geschwindigkeit der am C-863.12 angeschlossenen Positionierer-Achse (vom Profilgenerator ausgegebene kommandierte Geschwindigkeit).

Der Zusammenhang zwischen der Auslenkung der Joystick-Achse und der Geschwindigkeit der Positionierer-Achse wird vom C-863.12 durch eine Lookup-Tabelle hergestellt. Die 256 Werte in der Lookup-Tabelle sind Faktoren, die während der Joysticksteuerung auf die mit dem Befehl `VEL` (S. 206) gesetzte Geschwindigkeit angewendet werden. Der Wertebereich reicht von -1.0000 bis 1.0000.

Die Firmware des Controllers bietet zwei vordefinierte Lookup-Tabellentypen (linear und parabolisch) zur Auswahl an und erlaubt das Füllen der Lookup-Tabelle mit individuellen Werten. Der Inhalt der Lookup-Tabelle ist automatisch im permanenten Speicher des C-863.12 gespeichert.

Während der Joystick-Steuerung wird als Zielposition die Verfahrbereichsgrenze verwendet, die durch den Parameter 0x15 bzw. 0x30 vorgegeben ist. Details siehe "Stellweg und Verfahrbereichsgrenzen" (S. 31). Bei Deaktivierung des Joysticks wird als neue Zielposition die aktuelle Position der joystickgesteuerten Achse gesetzt.

INFORMATION

Bewegungsbefehle sind nicht zulässig, wenn ein Joystick für die Achse aktiv ist.
Im ungeregelten Betrieb (Servomodus aus) ist keine Joystick-Steuerung möglich.

7.6.2 Befehle und Parameter für Joystick-Steuerung

Befehle

Folgende Befehle stehen für die Verwendung des Joysticks zur Verfügung:

Befehl	Syntax	Funktion
`JON`	JON {<JoystickID> <uint>}	Aktiviert oder deaktiviert ein Joystick-Gerät, das an den Controller angeschlossen ist.
`JON?`	JON? [{<JoystickID>}]	Fragt den Status der Aktivierung des Joystick-Geräts ab.
`JAX`	JAX <JoystickID> <JoystickAxis> <AxisID>	Bestimmt die Achse, die durch einen an den Controller angeschlossenen Joystick gesteuert wird.
`JAX?`	JAX? [{<JoystickID> <JoystickAxis>}]	Fragt die Achse ab, die durch einen an den Controller angeschlossenen Joystick gesteuert wird.
`JAS?`	JAS? [{<JoystickID> <JoystickAxis>}]	Fragt den aktuellen Status einer Joystick-Achse ab (Auslenkung).
`JBS?`	JBS? [{<JoystickID> <JoystickButton>}]	Fragt den aktuellen Status einer Joystick-Taste ab (gedrückt oder nicht gedrückt)
`JDT`	JDT {<JoystickID> <JoystickAxis> <uint>}	Bestimmt einen Standard-Lookup-Tabellentyp für eine Joystick-Achse.
`JLT`	JLT <JoystickID> <JoystickAxis> <Addr> <floatn>	Füllt die Lookup-Tabelle für eine Joystick-Achse mit individuellen Werten.
`JLT?`	JLT? [<StartPoint> <NumberOfPoints> [{<JoystickID> <JoystickAxis>}]]	Fragt die aktuell gültigen Werte der Lookup-Tabelle ab.

Parameter

Folgende Parameter stehen für die Verwendung des Joysticks zur Verfügung:

Parameter	Beschreibung und mögliche Werte
Invert Direction Of Motion For Joystick-Controlled Axis? 0x61	Bestimmt die Bewegungsrichtung für joystickgesteuerte Achsen. 0 = Bewegungsrichtung nicht invertiert (Standard-Einstellung) 1 = Bewegungsrichtung invertiert

7.6.3 Achsenbewegungen steuern

Im Folgenden wird PIMikroMove® verwendet, um die Joystick-Steuerung für den Positionierer zu aktivieren. Die Kenntnis der entsprechenden GCS-Befehle ist hierfür nicht erforderlich.

HINWEIS

Unbeabsichtigte Bewegungen bei Joystick-Aktivierung!
Wenn kein Joystick an den C-863.12 angeschlossen ist, kann die Aktivierung des Joysticks in der Software unbeabsichtigte Bewegungen der angeschlossenen Achse verursachen.

➢ Aktivieren Sie den Joystick in der Software nur, wenn tatsächlich ein Joystick an den C-863.12 angeschlossen ist.

INFORMATION

Durch die Verwendung von Makros bieten sich vielfältige Einsatzmöglichkeiten für die Joystick-Steuerung. Insbesondere die Joystick-Taste kann in Makros für verschiedenste Zwecke genutzt werden. Details und Beispiele zu Makros finden Sie in "Controllermakros" (S. 100).

INFORMATION

Der C-863.12 unterstützt eine logische Achse und wird daher im Normalfall mit Positionierern eingesetzt, die nur eine Bewegungsachse haben. In diesem Fall ist die Bezeichnung "Achse" gleichbedeutend mit "Positionierer". In den nachfolgenden Handlungsanleitungen wird daher nicht zwischen "Positionierer" und "Achse" unterschieden.

Voraussetzungen

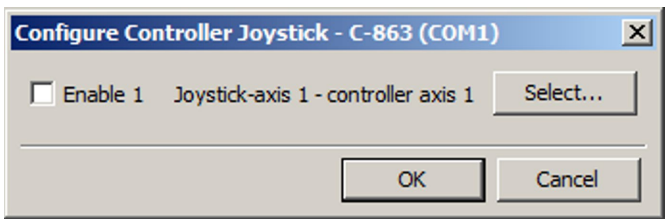
✓ Sie haben den Positionierer mit PIMikroMove® in Betrieb genommen und erste Bewegungen gestartet (S. 64).

✓ Sie haben einen Joystick an den C-863.12 angeschlossen (S. 49).

Achsenbewegung über einen Joystick steuern

1. Öffnen Sie im Hauptfenster von PIMikroMove® das Fenster ***Configure Controller Joystick*** über den Menüeintrag ***C-863.12 > Configure controller joystick(s)...***.

   Das Fenster ***Configure Controller Joystick*** listet das vom C-863.12 unterstützte Joystick-Gerät und dessen Joystick-Achse auf.



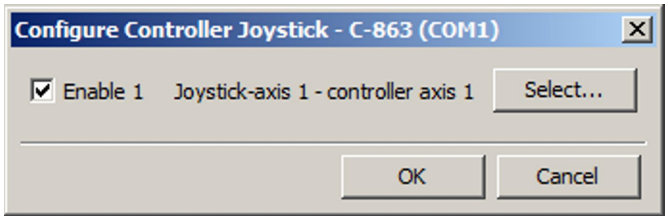
2. Weisen Sie der Joystick-Achse des angeschlossenen Joystick-Geräts die zu bewegende Achse zu:

a) Klicken Sie im Fenster ***Configure Controller Joystick*** auf ***Select***.

b) Markieren Sie im Fenster ***Select controller axis*** den korrekten Positionierernamen.

c) Klicken Sie im Fenster ***Select controller axis*** auf ***OK***, um die Auswahl zu bestätigen und das Fenster zu schließen.

3. Aktivieren Sie im Fenster ***Configure Controller Joystick*** das angeschlossene Joystick-Gerät, indem Sie das entsprechende Kontrollkästchen ***Enable*** markieren.



Wenn die Joystick-Steuerung nicht zufriedenstellend funktioniert oder der Positionierer sich bewegt, obwohl Sie den Joystick nicht betätigen:

– Prüfen Sie, ob der Joystick mechanisch arretiert ist.

– Kalibrieren Sie den Joystick (S. 95).

4. Steuern Sie die Geschwindigkeit des Positionierers über den Joystick.

5. Wenn Sie die Joystick-Steuerung deaktivieren möchten, entfernen Sie im Fenster ***Configure Controller Joystick*** das Häkchen aus dem entsprechenden Kontrollkästchen ***Enable***.

7.6.4 Joystick kalibrieren

Nach dem erstmaligen Anschließen eines Joysticks an den C-863.12 wird die Kalibration der einzelnen Joystick-Achsen empfohlen.

Die Kalibration umfasst die folgenden Schritte:

- Wenn entsprechende Bedienelemente am Joystick-Gerät vorhanden sind: mechanisches Justieren der Joystick-Achsen.
- Kalibration der Joystick-Achsen in PIMikroMove®

INFORMATION

Bei der Joystick-Kalibration in PIMikroMove® wird der zu verwendende Lookup-Tabellentyp ausgewählt bzw. die Lookup-Tabelle mit individuellen Werten gefüllt. Dazu muss kein Positionierer am C-863.12 angeschlossen sein.

In folgenden Fällen ist die Kalibration für eine ordnungsgemäße Joystick-Steuerung notwendig:

- Nach dem Aktivieren der Joystick-Steuerung bewegt sich der Positionierer, obwohl Sie den Joystick nicht betätigen.
- Das Ansprechverhalten des Joysticks entspricht nicht Ihren Anforderungen.
- Sie verwenden die Z-Achse eines Joysticks C-819.30.

INFORMATION

Die Anzahl der Schreibzyklen im permanenten Speicher ist durch die begrenzte Lebensdauer des Speicherchips (EEPROM) beschränkt.

- Kalibrieren Sie die Joystick-Achsen in PIMikroMove® nur, wenn es notwendig ist.
- Wenden Sie sich an unseren Kundendienst (S. 263), wenn der C-863.12 unerwartetes Verhalten zeigt.

INFORMATION

Für die Z-Achse des Joysticks C-819.30 ist keine mechanische Justierung möglich, und sie kann nicht mit den Standard-Lookup-Tabellentypen (linear oder parabolisch) des C-863.12 betrieben werden.

- Kalibrieren Sie die Z-Achse des Joysticks nach dem Anschließen an den C-863.12 mit PIMikroMove®.
- Verwenden Sie zur Kalibration der Z-Achse die Methode ***Measure Joystick Parameters and Use Custom Lookup Table***.
- Wiederholen Sie die Kalibration der Z-Achse, wenn Sie die Z-Achse an einen anderen Controller anschließen.

INFORMATION

Der parabolische Lookup-Tabellentyp ermöglicht mehr Feinfühligkeit bei langsamer Fahrt.

Joystick-Achse mechanisch justieren

➢ Prüfen Sie bei aktivierter Joystick-Steuerung, ob sich der Positionierer bereits bewegt, wenn Sie den Joystick nicht betätigen.

Wenn ja:

➢ Prüfen Sie, ob der Joystick mechanisch arretiert ist, und lösen Sie gegebenenfalls die Arretierung.

➢ Belassen Sie die betroffene Joystick-Achse (d. h. z. B. den Steuerhebel) in Mittelstellung und justieren Sie sie mit den entsprechenden Bedienelementen, bis sich der Positionierer nicht mehr bewegt. Bei den Joysticks C-819.20 und C-819.30 drehen Sie dazu den entsprechenden Drehknopf für die Justierung (S. 98).

Wenn nein:

➢ Prüfen Sie, ob das Ansprechverhalten des Joysticks Ihren Anforderungen entspricht.

Wenn ja:

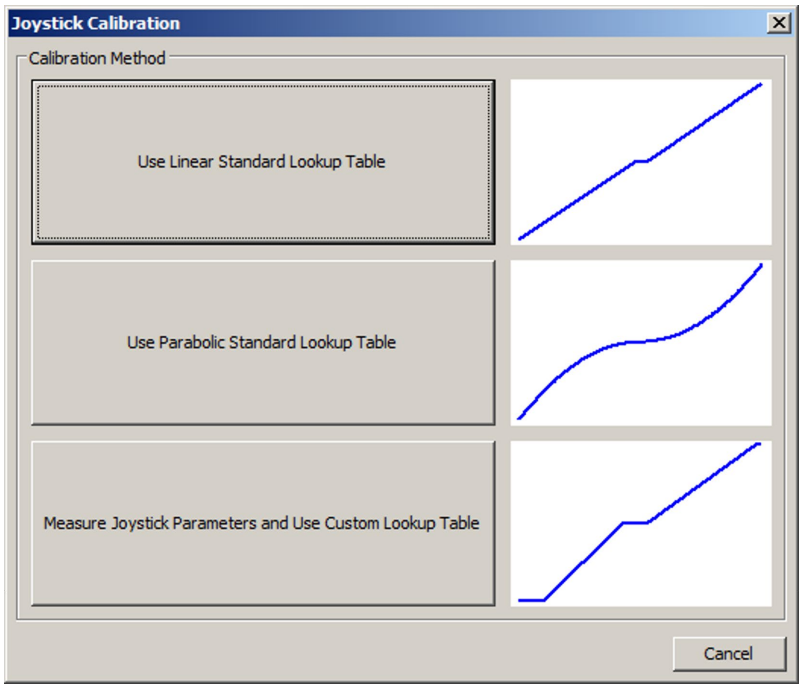
– Die Kalibration ist beendet.

Wenn nein:

– Kalibrieren Sie die Joystick-Achse in PIMikroMove®.

Joystick-Achse in PIMikroMove® kalibrieren

1. Öffnen Sie im Hauptfenster von PIMikroMove® das Fenster ***Joystick Calibration*** über den Menüeintrag ***C-863.12 > Calibrate controller joystick…***.



2. Wählen Sie die Methode der Kalibration aus, indem Sie auf die entsprechende Schaltfläche klicken:
   - Wenn Sie den linearen Lookup-Tabellentyp für die Joystick-Achse verwenden wollen, klicken Sie auf ***Use Linear Standard Lookup Table***. Damit wird der entsprechende Lookup-Tabellentyp geladen, und die Kalibration ist beendet.
   - Wenn Sie den parabolischen Lookup-Tabellentyp für die Joystick-Achse verwenden wollen, klicken Sie auf ***Use Parabolic Standard Lookup Table***. Damit wird der entsprechende Lookup-Tabellentyp geladen, und die Kalibration ist beendet.
   - Wenn Sie die Z-Achse eines Joysticks C-819.30 angeschlossen haben oder generell das Verhalten des Joysticks in einer individuellen Lookup-Tabelle abbilden wollen, klicken Sie auf ***Measure Joystick Parameters and Use Custom Lookup Table***. Damit öffnet sich das Fenster ***Controller Joystick Calibration***.
3. Wenn sich das Fenster ***Controller Joystick Calibration*** geöffnet hat, folgen Sie den Anweisungen in diesem Fenster.

   Auf diese Weise werden individuelle Lookup-Tabellenwerte ermittelt.

   Durch Klicken auf ***OK*** laden Sie die ermittelten Lookup-Tabellenwerte in den permanenten Speicher des C-863.12. Die Kalibration ist damit beendet.

7.6.5 Verfügbare Joystick-Geräte

PI bietet die nachfolgend beschriebenen Joystick-Geräte als optionales Zubehör (S. 10) an.

Analoger Joystick C-819.20, 2 Achsen

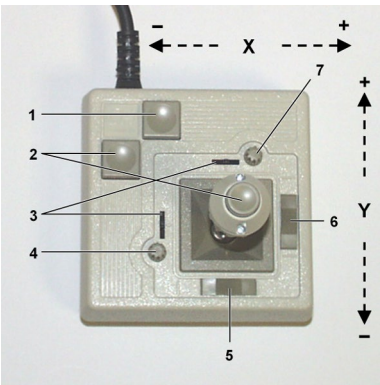


Abbildung 21: Joystick C-819.20

1 Drucktaste für die X-Achse
2 Drucktaste für die Y-Achse
3 Justageanzeiger
4 Drehknopf für Justierung der Y-Achse (Kalibrierung)
5 Arretierung der X-Achse
6 Arretierung der Y-Achse
7 Drehknopf für Justierung der X-Achse (Kalibrierung)

Analoger Joystick C-819.30, 3 Achsen

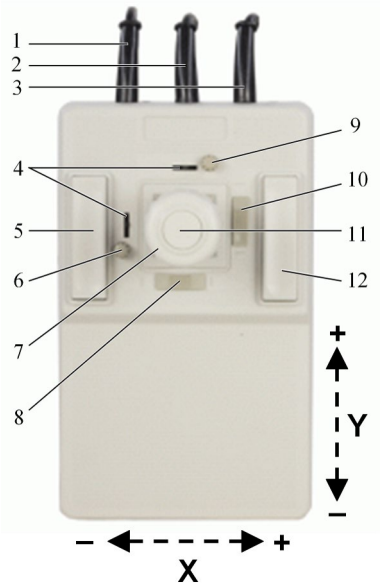


Abbildung 22: Joystick C-819.30

1 Kabel für die Z-Achse
2 Kabel für die Y-Achse
3 Kabel für die X-Achse
4 Justageanzeiger
5 Drucktaste für die Y-Achse
6 Drehknopf für Justierung der Y-Achse (Kalibrierung)
7 XY-Steuerhebel mit Drehknopf für Z-Achse
8 Arretierung der X-Achse
9 Drehknopf für Justierung der X-Achse (Kalibrierung)
10 Arretierung der Y-Achse
11 Drucktaste für die Z-Achse
12 Drucktaste für die X-Achse

Abbildung 23: Joystick C-819.30, Drehknopf für die Z-Achse

7.7 Controllermakros

7.7.1 Übersicht: Makrofunktionalitäten und Beispielmakros

Der C-863.12 kann Befehlsfolgen als Makros speichern und abarbeiten.

Die folgenden Funktionalitäten machen Makros zu einem wichtigen Werkzeug in vielen Anwendungsgebieten:

- Mehrere Makros können gleichzeitig gespeichert werden.
- Ein beliebiges Makro kann als Startup-Makro festgelegt werden. Das Startup-Makro wird bei jedem Einschalten oder Neustart des C-863.12 ausgeführt.
- Die Abarbeitung des Makros und das Stoppen der Makroausführung lassen sich an Bedingungen knüpfen. So können auch Schleifen realisiert werden.
- Makros können sich selbst oder andere Makros aufrufen.
- Variablen (S. 123) können für das Makro und im Makro selbst gesetzt und in verschiedenen Operationen verwendet werden.
- Eingangssignale können für Bedingungen und Variablen ausgewertet werden.

In diesem Handbuch finden Sie Beispielmakros für folgende Aufgaben:

- Achse hin und her bewegen (S. 104)
- Makro für Controller aufzeichnen, dessen Adresse verschieden von 1 ist (S. 105)
- Achse mit variablem Verfahrweg hin und her bewegen (S. 107)
- Mehrfachaufruf eines Makros durch Schleife realisieren (S. 108)
- Achse durch Startup-Makro für den Betrieb vorbereiten (S. 109)
- Synchronisation zweier Controller (S. 111)
- Bewegung per Tastendruck stoppen (S. 113)
- Joystick-Steuerung mit Speicherung von Positionen (S. 114)
- Joystick-Steuerung mit Änderung der Geschwindigkeit (S. 117)

7.7.2 Befehle und Parameter für Makros

Befehle

Folgende Befehle stehen speziell für die Handhabung von Makros oder für die Verwendung in Makros zur Verfügung:

Befehl	Syntax	Funktion
ADD (S. 134)	ADD <Variable> <FLOAT1> <FLOAT2>	Addiert zwei Werte und speichert das Ergebnis als Variable (S. 123). Für lokale Variablen nur innerhalb von Makros verwendbar.
CPY (S. 138)	CPY <Variable> <CMD?>	Kopiert eine Antwort auf einen Befehl in eine Variable (S. 123). Für lokale Variablen nur innerhalb von Makros verwendbar.
DEL (S. 145)	DEL <uint>	Kann nur in Makros verwendet werden. Verzögert um <uint> Millisekunden.
JRC (S. 172)	JRC <Jump> <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Löst einen relativen Sprung des Makroausführungszeigers in Abhängigkeit von einer Bedingung aus.
MAC (S. 174)	MAC BEG <macroname>	Startet die Aufzeichnung eines Makros mit dem Namen *macroname* auf dem Controller. *macroname* kann aus bis zu 8 Zeichen bestehen.
	MAC DEF <macroname>	Legt das angegebene Makro als Startup-Makro fest.
	MAC DEF?	Fragt das Startup-Makro ab.
	MAC DEL <macroname>	Löscht das angegebene Makro.
	MAC END	Stoppt die Makroaufzeichnung.
	MAC ERR?	Meldet den letzten Fehler, der während der Ausführung eines Makros auftrat.
	MAC NSTART <macroname> <uint> [<String1> [<String2>]]	Startet das angegebene Makro n-mal hintereinander (n = Anzahl der Ausführungen). Mit <String1> und <String2> können die Werte lokaler Variablen für das Makro gesetzt werden.
	MAC START <macroname> [<String1> [<String2>]]	Startet eine Ausführung des angegebenen Makros. Mit <String1> und <String2> können die Werte lokaler Variablen für das Makro gesetzt werden.
MAC? (S. 177)	MAC? [<macroname>]	Listet alle Makros oder den Inhalt des angegebenen Makros auf.
MEX (S. 179)	MEX <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Stoppt die Makroausführung in Abhängigkeit von einer Bedingung.
RMC? (S. 186)	RMC?	Listet die aktuell laufenden Makros auf.

Befehl	Syntax	Funktion
VAR (S. 205)	VAR <Variable> <String>	Setzt eine Variable (S. 123) auf einen bestimmten Wert oder löscht sie. Für lokale Variablen nur innerhalb von Makros verwendbar.
VAR? (S. 206)	VAR? [{<Variable>}]	Gibt Variablenwerte zurück.
WAC (S. 208)	WAC <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Wartet, bis eine Bedingung erfüllt ist.
#8 (S. 132)	-	Prüft, ob ein Makro auf dem Controller ausgeführt wird.

Parameter

Folgender Parameter steht für die Arbeit mit Makros zur Verfügung:

Parameter	Beschreibung und mögliche Werte
Ignore Macro Error? 0x72	Legt fest, ob das Controllermakro gestoppt wird, wenn bei dessen Ausführung ein Fehler auftritt. ▪ 0 = Bei Fehler Makro anhalten (Standard) ▪ 1 = Fehler ignorieren

7.7.3 Mit Makros arbeiten

Die Arbeit mit Makros umfasst Folgendes:

- Aufzeichnen von Makros (S. 102)
- Starten der Makroausführung (S. 106)
- Stoppen der Makroausführung (S. 108)
- Einrichten eines Startup-Makros (S. 109)
- Löschen von Makros (S. 110)

INFORMATION

Für die Arbeit mit Controllermakros wird die Verwendung der Registerkarte ***Controller macros*** in PIMikroMove® empfohlen. Dort können Sie Controllermakros komfortabel aufzeichnen, starten und verwalten. Details finden Sie im PIMikroMove® Handbuch.

Makro aufzeichnen

INFORMATION

Der C-863.12 kann bis zu 32 Makros gleichzeitig speichern. Maximal 5 Verschachtelungsebenen sind in Makros möglich.

INFORMATION

Grundsätzlich können alle GCS-Befehle (S. 119) Bestandteil eines Makros werden. Ausnahmen:

- `RBT` für den Neustart des C-863.12
- `MAC BEG` und `MAC END` für die Makroaufzeichnung
- `MAC DEL` zum Löschen eines Makros

Abfragebefehle können in Makros in Kombination mit den Befehlen `CPY`, `JRC`, `MEX` und `WAC` verwendet werden. Andernfalls bleiben sie wirkungslos, da Makros keine Antworten an Schnittstellen senden.

INFORMATION

Wenn Sie ein Makro auf einem C-863.12 aufzeichnen, dessen Controlleradresse von 1 abweicht, beachten Sie Folgendes beim Eingeben der Befehle, die Bestandteil des Makros sein sollen:

- Wenn Sie mit PITerminal arbeiten und die Kommunikation mit der Schaltfläche ***Connect…*** hergestellt haben, muss die Empfängeradresse in jeder Befehlszeile eingetippt werden.
- Wenn Sie mit PIMikroMove® arbeiten oder mit PITerminal die Kommunikation mit der Schaltfläche ***GCS DLL…*** hergestellt haben, wird die Empfängeradresse automatisch mitgesendet und darf nicht eingetippt werden.

INFORMATION

Um die Anwendung von Makros flexibler zu gestalten, können Sie in Makros lokale und globale Variablen verwenden. Weitere Informationen siehe "Variablen" (S. 123).

INFORMATION

Die Anzahl der Schreibzyklen im permanenten Speicher ist durch die begrenzte Lebensdauer des Speicherchips beschränkt.

- ➢ Zeichnen Sie Makros nur auf, wenn es notwendig ist.
- ➢ Verwenden Sie Variablen (S. 123) in Makros, und geben Sie beim Starten der Makroausführung die entsprechenden Variablenwerte an.
- ➢ Wenden Sie sich an unseren Kundendienst (S. 263), wenn der C-863.12 unerwartetes Verhalten zeigt.

INFORMATION

Ein Makro wird überschrieben, wenn erneut ein Makro mit demselben Namen aufgezeichnet wird.

1. Starten Sie die Makroaufzeichnung.

- Wenn Sie mit PITerminal oder im Fenster ***Command entry*** von PIMikroMove® arbeiten: Senden Sie den Befehl `MAC BEG macroname`, wobei *macroname* den Namen des Makros bezeichnet.
- Wenn Sie in PIMikroMove® auf der Registerkarte ***Controller macros*** arbeiten: Klicken Sie auf das Symbol ***Create new empty macro***, um eine Registerkarte zum Eingeben eines neuen Makros zu erzeugen. Geben Sie **nicht** den Befehl `MAC BEG macroname` ein.

2. Geben Sie unter Verwendung der normalen Befehlssyntax Zeile für Zeile die Befehle ein, die Bestandteil des Makros *macroname* sein sollen.

   Makros können sich selbst oder andere Makros in mehreren Verschachtelungsebenen aufrufen.

3. Beenden Sie die Makroaufzeichnung.
   - Wenn Sie mit PITerminal oder im Fenster ***Command entry*** von PIMikroMove® arbeiten: Senden Sie den Befehl `MAC END`.
   - Wenn Sie in PIMikroMove® auf der Registerkarte ***Controller macros*** arbeiten: Geben Sie **nicht** den Befehl `MAC END` ein. Klicken auf das Symbol ***Send macro to controller*** und geben Sie in einem separaten Dialogfenster den Namen des Makros ein.

   Das Makro wurde im permanenten Speicher des C-863.12 abgelegt.

4. Wenn Sie prüfen wollen, ob das Makro korrekt aufgezeichnet wurde:

   Wenn Sie mit PITerminal oder im Fenster ***Command entry*** von PIMikroMove® arbeiten:
   - Fragen Sie ab, welche Makros im C-863.12 gespeichert sind, indem Sie den Befehl `MAC?` senden.
   - Fragen Sie den Inhalt des Makros *macroname* ab, indem Sie den Befehl `MAC? macroname` senden.

   Wenn Sie in PIMikroMove® auf der Registerkarte ***Controller macros*** arbeiten:
   - Klicken Sie auf das Symbol ***Read list of macros from controller***.
   - Markieren Sie das zu prüfende Makro in der Liste auf der linken Seite, und klicken Sie auf das Symbol ***Load selected macro from controller***.

Beispiel: Achse hin und her bewegen

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Die Achse 1 soll sich hin und her bewegen. Dazu werden 3 Makros aufgezeichnet. Makro 1 startet die Bewegung in die positive Richtung und wartet, bis die Achse die Zielposition erreicht hat. Makro 2 erfüllt diese Aufgabe für die negative Bewegungsrichtung. Makro 3 ruft die Makros 1 und 2 auf.

➢ Zeichnen Sie die Makros auf, indem Sie senden:

```
MAC BEG macro1
MVR 1 12.5
WAC ONT? 1 = 1
MAC END
MAC BEG macro2
MVR 1 -12.5
WAC ONT? 1 = 1
MAC END
MAC BEG macro3
MAC START macro1
MAC START macro2
MAC END
```

Beispiel: Makro für Controller aufzeichnen, dessen Adresse verschieden von 1 ist

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Über die DIP-Schalter ist die Controlleradresse auf 2 eingestellt. In diesem Beispiel erfolgt die Makroaufzeichnung mit PITerminal, wobei die Kommunikation mit der Taste ***Connect…*** hergestellt wurde (daher muss die Empfängeradresse in jeder Befehlszeile eingetippt werden).

Durch das Makro ref soll für Achse 1 der Servomodus eingeschaltet und eine Referenzierungsfahrt zum Referenzschalter gestartet werden.

1. Zeichnen Sie das Makro auf, indem Sie senden:

   ```
   2 MAC BEG ref
   2 SVO 1 1
   2 DEL 1000
   2 FRF 1
   2 MAC END
   ```

2. Prüfen Sie den Inhalt des Makros ref, indem Sie senden:

   ```
   2 MAC? ref
   ```

   Die Antwort lautet:

   ```
   0 2 SVO 1 1
   DEL 1000
   FRF 1
   ```

Die erste Zeile der Antwort enthält die Empfänger- und Senderadresse gemäß der GCS-Syntax für mehrzeilige Antworten. Die Empfängeradresse ist jedoch nicht in das Makro aufgenommen worden.

Makroausführung starten

INFORMATION

Von der Befehlszeile können sämtliche Befehle gesendet werden, während auf dem Controller ein Makro läuft. Der Makroinhalt und Bewegungsbefehle, die von der Befehlszeile empfangen werden, können sich gegenseitig überschreiben.

INFORMATION

Zeitgleiche Ausführung mehrerer Makros ist nicht möglich. Es kann jeweils nur ein Makro ausgeführt werden.

INFORMATION

Sie können die Makroausführung mit den Befehlen `JRC` und `WAC` an Bedingungen knüpfen. Die Befehle müssen im Makro enthalten sein.

Im Folgenden wird PITerminal oder das Fenster ***Command entry*** von PIMikroMove® verwendet, um Befehle einzugeben. Details zur Arbeit mit der Registerkarte ***Controller macros*** in PIMikroMove® finden Sie im PIMikroMove® Handbuch.

1. Wenn die Makroausführung trotz Auftretens eines Fehlers fortgesetzt werden soll:
   - Stellen Sie den Parameter ***Ignore Macro Error?*** (ID 0x72) entsprechend ein: Senden Sie den Befehl `SPA 1 0x72 Status`, wobei *Status* die Werte 0 oder 1 annehmen kann (0 = Bei Fehler Makro anhalten (Standard); 1 = Makrofehler ignorieren).

   Weitere Informationen zum Ändern von Parametern finden Sie in "Anpassen von Einstellungen" (S. 235).
2. Starten Sie die Makroausführung:
   - Wenn das Makro einmal ausgeführt werden soll, senden Sie den Befehl `MAC START macroname string`, wobei *macroname* den Namen des Makros bezeichnet.
   - Wenn das Makro n-mal ausgeführt werden soll, senden Sie den Befehl `MAC NSTART macroname n string`, wobei *macroname* den Namen des Makros bezeichnet und *n* die Anzahl der Ausführungen angibt.

   string steht für die Werte lokaler Variablen. Die Werte sind nur dann anzugeben, wenn das Makro entsprechende lokale Variablen enthält. Die Reihenfolge der Werte bei der Eingabe muss der Nummerierung der zugehörigen lokalen Variablen entsprechen, beginnend mit dem Wert der lokalen Variablen 1. Die einzelnen Werte müssen durch Leerzeichen voneinander getrennt werden.

3. Wenn Sie die Makroausführung prüfen wollen:
   - Fragen Sie ab, ob ein Makro auf dem Controller ausgeführt wird, indem Sie den Befehl `#8` senden.
   - Fragen Sie den Namen des Makros ab, das gerade auf dem Controller ausgeführt wird, indem Sie den Befehl `RMC?` senden.

Beispiel: Achse mit variablem Verfahrweg hin und her bewegen

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Die Achse 1 soll sich hin und her bewegen. Der Verfahrweg nach links und rechts soll variabel einstellbar sein, ohne dass dazu die verwendeten Makros geändert werden müssen. Deshalb werden lokale und globale Variablen verwendet.

1. Legen Sie die globalen Variablen LEFT und RIGHT an, indem Sie senden:

   ```
   VAR LEFT 5
   VAR RIGHT 15
   ```

   LEFT hat damit den Wert 5, und RIGHT hat den Wert 15. Diese Werte können Sie jederzeit ändern, indem Sie z. B. den Befehl `VAR` erneut senden.

   - Legen Sie die globalen Variablen nach jedem Einschalten oder Neustart des C-863.12 erneut an, da sie nur in den flüchtigen Speicher des C-863.12 geschrieben werden.

2. Zeichnen Sie das Makro MOVLR auf, indem Sie senden:

   ```
   MAC BEG movlr
   MAC START movwai ${LEFT}
   MAC START movwai ${RIGHT}
   MAC END
   ```

   MOVLR startet das (noch aufzuzeichnende) Makro MOVWAI nacheinander für beide Bewegungsrichtungen. Die Werte der globalen Variablen LEFT und RIGHT werden beim Start von MOVWAI verwendet, um den Wert der in MOVWAI enthaltenen lokalen Variable 1 zu setzen (Dollarzeichen und geschweifte Klammern sind erforderlich, damit die lokale Variable 1 im Makro tatsächlich mit dem *Wert* der globalen Variable und nicht mit ihrem *Namen* ersetzt wird).

3. Zeichnen Sie das Makro MOVWAI auf, indem Sie senden:

   ```
   MAC BEG movwai
   MOV 1 $1
   WAC ONT? 1 = 1
   MAC END
   ```

MOVWAI bewegt die Achse 1 zur Zielposition, die durch den Wert der lokalen Variablen 1 vorgegeben ist, und wartet, bis die Achse die Zielposition erreicht hat.

4. Starten Sie die Ausführung des Makros MOVLR, indem Sie senden:

   `MAC NSTART movlr 5`

   Das Makro MOVLR wird fünfmal hintereinander ausgeführt, d. h. die Achse 1 bewegt sich fünfmal im Wechsel zu den Positionen 5 und 15. Für die Anzahl der Ausführungen können Sie auch einen beliebigen anderen Wert wählen.

Beispiel: Mehrfachaufruf eines Makros durch Schleife realisieren

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Das Makro TESTDION prüft den Zustand der digitalen Eingangsleitungen auf der Buchse **I/O**. Es verwendet eine lokale Variable, um die digitale Eingangsleitung zu bezeichnen (1 bis 4). Damit das Makro TESTDION nicht für jede Eingangsleitung separat aufgerufen werden muss, wird ein weiteres Makro mit einer Schleife aufgezeichnet.

➢ Zeichnen Sie das Makro LOOPDION auf, indem Sie senden:

```
MAC BEG loopdion
VAR COUNTER 1
MAC START TESTDION ${COUNTER}
ADD COUNTER ${COUNTER} 1
JRC -2 VAR? COUNTER < 5
MAC END
```

Die Variable COUNTER wird mit dem Wert 1 angelegt. Anschließend wird das Makro TESTDION für die Eingangsleitung gestartet, deren Kennung durch die Variable COUNTER vorgegeben ist. Danach wird der Wert von COUNTER um 1 hochgesetzt. Solange der Wert von COUNTER kleiner als 5 ist, springt der Makroausführungszeiger anschließend 2 Zeilen zurück, so dass TESTDION nun für die nächste digitale Eingangsleitung gestartet wird.

Makroausführung stoppen

INFORMATION

Sie können das Stoppen der Makroausführung mit dem Befehl `MEX` an eine Bedingung knüpfen. Der Befehl muss im Makro enthalten sein.

Im Folgenden wird PITerminal oder das Fenster ***Command entry*** von PIMikroMove® verwendet, um Befehle einzugeben. Details zur Arbeit mit der Registerkarte ***Controller macros*** in PIMikroMove® finden Sie im PIMikroMove®-Handbuch.

➢ Stoppen Sie die Makroausführung mit den Befehlen `#24` oder `STP`.

- Wenn Sie prüfen wollen, ob während der Makroausführung ein Fehler aufgetreten ist, senden Sie den Befehl `MAC ERR?`. Die Antwort zeigt den letzten Fehler an, der aufgetreten ist.

Startup-Makro einrichten

Ein beliebiges Makro kann als Startup-Makro festgelegt werden. Das Startup-Makro wird bei jedem Einschalten oder Neustart des C-863.12 ausgeführt.

INFORMATION

Das Löschen eines Makros löscht nicht seine Auswahl als Startup-Makro.

Im Folgenden wird PITerminal oder das Fenster ***Command entry*** von PIMikroMove® verwendet, um Befehle einzugeben. Details zur Arbeit mit der Registerkarte ***Controller macros*** in PIMikroMove® finden Sie im PIMikroMove® Handbuch.

- Legen Sie mit dem Befehl `MAC DEF macroname` ein Makro als Startup-Makro fest, wobei *macroname* den Namen des Makros bezeichnet.
- Wenn Sie die Auswahl des Startup-Makros aufheben und kein anderes Makro als Startup-Makro festlegen wollen, senden Sie nur `MAC DEF`.
- Fragen Sie den Namen des aktuell festgelegten Startup-Makros ab, indem Sie den Befehl `MAC DEF?` senden.

Beispiel: Achse durch Startup-Makro für den Betrieb vorbereiten

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Das Makro STARTCL schaltet für Achse 1 die Joystick-Steuerung aus und den Servomodus ein und startet eine Referenzierungsfahrt zum negativen Endschalter. Indem STARTCL als Startup-Makro festgelegt wird, ist die Achse 1 sofort nach dem Einschalten für den geregelten Betrieb bereit.

- Senden Sie:

```
MAC BEG startcl
JON 1 0
SVO 1 1
DEL 1000
FNL 1
MAC END
MAC DEF startcl
```

INFORMATION

Bei Verwendung dieses Makros sollten die Parametereinstellungen des C-863.12 im permanenten Speicher an den angeschlossenen Positionierer angepasst sein. Alternativ können die Parametereinstellungen im flüchtigen Speicher auch durch das Startup-Makro gesetzt werden. Weitere Informationen siehe "Anpassen von Einstellungen" (S. 235).

Makro löschen

INFORMATION

Ein laufendes Makro kann nicht gelöscht werden.

Im Folgenden wird PITerminal oder das Fenster ***Command entry*** von PIMikroMove® verwendet, um Befehle einzugeben. Details zur Arbeit mit der Registerkarte ***Controller macros*** in PIMikroMove® finden Sie im PIMikroMove® Handbuch.

➢ Löschen Sie ein Makro mit dem Befehl `MAC DEL macroname`, wobei *macroname* den Namen des Makros bezeichnet.

7.7.4 Controllermakros sichern und laden

Das Sichern von Controllermakros auf dem PC kann z. B. vor der Aktualisierung der Firmware (S. 253) sinnvoll sein.

INFORMATION

Für das Sichern und Laden von Controllermakros wird die Verwendung der Registerkarte ***Controller macros*** in PIMikroMove® empfohlen. Eine detaillierte Beschreibung der Registerkarte finden Sie imPIMikroMove® Handbuch.

Controllermakros mit PIMikroMove® auf dem PC sichern

1. Wählen Sie im Hauptfenster von PIMikroMove® die Registerkarte ***Controller macros***.
2. Wählen Sie in der Liste ***Macros on controller*** die Makros aus, die Sie auf den PC sichern wollen:
   - Zur Auswahl eines einzelnen Makros klicken Sie den gewünschten Listeneintrag an.
   - Zur Auswahl mehrerer Makros klicken Sie mit gedrückter Umschalt-Taste (Shift) alle gewünschten Listeneinträge an.
   - Um die Auswahl aufzuheben, klicken Sie auf eine freie Fläche in der Liste.

   Durch die Auswahl eines oder mehrerer Makros wird die Schaltfäche (*Save selected macros to PC*) aktiv.
3. Speichern Sie die ausgewählten Makros auf dem PC:
   a) Klicken Sie auf die Schaltfläche , um ein Fenster zur Verzeichnisauswahl zu öffnen.

b) Wählen Sie das Verzeichnis auf dem PC aus, in dem Sie die Makros speichern wollen.

c) Klicken Sie auf ***Speichern***.

Die Makros werden als Textdateien (<Makroname>.txt) im ausgewählten Verzeichnis des PC gespeichert.

Controllermakros mit PIMikroMove® vom PC in den C-863.12 laden

1. Wählen Sie im Hauptfenster von PIMikroMove® die Registerkarte ***Controller macros***.
2. Laden Sie Makros vom PC in den C-863.12:

   a) Klicken Sie auf die Schaltfläche , um ein Dateiauswahlfenster zu öffnen.

   b) Wählen Sie im Dateiauswahlfenster die Textdateien (<Makroname>.txt) aus, deren Inhalt Sie als Makro vom PC in den C-863.12 laden wollen.

   c) Klicken Sie auf ***Öffnen***.

   Für jede ausgewählte Textdatei (<Makroname>.txt) wird der Inhalt als Makro <Makroname> in den C-863.12 geladen.

7.7.5 Makrobeispiel: Synchronisation zweier Controller

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Aktion	Befehl	Ergebnis
Digitale Ausgangsleitung 1 auf der Buchse **I/O** des Master-Controllers mit der digitalen Eingangsleitung 1 auf der Buchse **I/O** des Slave-Controllers verbinden.	- Geeignetes Kabel verwenden. Pinbelegung siehe "I/O" (S. 270).	Das digitale Ausgangssignal des Master-Controllers kann als Trigger für die Bewegung der Achse verwendet werden, die an den Slave-Controller angeschlossen ist.

Auf dem Master-Controller und auf dem Slave-Controller die Bewegung vorbereiten.	`SVO 1 1`
	`FRF 1 1`
	`VEL 1 0`
	`MOV 1 5.5`
		Für beide Controller: Der Servomodus ist eingeschaltet und die Achse hat eine Referenzierungsfahrt ausgeführt – hier zum Referenzschalter. Die Geschwindigkeit ist auf null gesetzt. Somit bewegt sich die Achse vorerst nicht, obwohl das Bewegungskommando für die Fahrt zur absoluten Position 5,5 schon gesendet wurde.

Makro MASTER auf dem Master-Controller aufzeichnen.	`MAC BEG master`
	`DIO 1 1`
	`VEL 1 100`
	`MAC END`
		Das Makro hat folgende Aufgaben:
		▪ Digitale Ausgangsleitung 1 des Mastercontrollers in den Zustand high schalten, um den Slave-Controller zu triggern
		▪ Geschwindigkeit auf 100 setzen, um die Bewegung zu starten

Makro SLAVE auf dem Slave-Controller aufzeichnen.	`MAC BEG slave`
	`WAC DIO? 1 = 1`
	`VEL 1 100`
	`MAC END`
		Das Makro hat folgende Aufgaben:
		▪ Bedingung setzen: Das Makro wird erst dann weiter ausgeführt, wenn die digitale Eingangsleitung 1 den Zustand high hat (d.h. wenn der Mastercontroller das Triggersignal ausgibt).
		▪ Geschwindigkeit auf 100 setzen, um die Bewegung zu starten

Makro SLAVE auf dem Slave-Controller starten.	`MAC START slave`	Die Achse am Slave-Controller bewegt sich noch nicht, da die Bedingung zur weiteren Makroausführung noch nicht erfüllt ist.
Makro MASTER auf dem Master-Controller starten.	`MAC START master`	Beide Achsen bewegen sich, da ihre Geschwindigkeit jetzt jeweils von Null verschieden ist. Die Bewegung erfolgt synchron.

7.7.6 Makrobeispiel: Bewegung per Tastendruck stoppen

INFORMATION

Um die digitalen Eingangssignale für die Verwendung in Makros zu erzeugen, können Sie die Pushbutton-Box C-170.PB von PI an die Buchse **I/O** anschließen. Sie zeigt über LEDs auch den Status der digitalen Ausgangsleitungen an.

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Aktion	Befehl	Ergebnis
Digitale Eingangsleitung 1 auf der Buchse **I/O** mit einer geeigneten Signalquelle verbinden.	- Pinbelegung siehe "I/O" (S. 270).	Das digitale Eingangssignal kann z.B. für einen bedingten Sprung des Makroausführungs-Zeigers verwendet werden.

Makro HALT auf dem Controller aufzeichnen.	`MAC BEG halt`
	`MVR 1 5`
	`JRC 2 DIO? 1 = 1`
	`JRC -1 ONT? 1 = 0`
	`HLT 1`
	`MAC END`
		Das Makro hat folgende Aufgaben:
		▪ Relative Bewegung der Achse 1 starten
		▪ Bedingung setzen: Wenn die digitale Eingangsleitung 1 den Zustand high hat (bei Verwendung der Pushbutton-Box: Taste 1 ist gedrückt), springt der Makroausführungs-Zeiger zwei Zeilen nach vorn. Damit wird die Achse angehalten. Andernfalls wird die Makroausführung mit der nächsten Zeile fortgesetzt.
		▪ Bedingung setzen: Solange die Achse 1 die Zielposition noch nicht erreicht hat, springt der Makroausführungs-Zeiger eine Zeile zurück. Damit wird eine Schleife eingerichtet.

Makro HALT auf dem Controller starten.	`MAC START halt`	Die Bewegung der Achse 1 startet. Sie wird durch Schalten der digitalen Eingangsleitung 1 in den Zustand high (z.B. durch Tastendruck) angehalten. Unabhängig davon, ob die Achse die Zielposition erreicht hat oder vorher angehalten wurde, wird der Fehlercode durch den Befehl `HLT` auf 10 gesetzt.

Wenn der Fehlercode 10 stört: alternatives Makro HALTVAR aufzeichnen, das eine Variable verwendet. Details siehe "Variablen" (S. 123).	`MAC BEG haltvar`
	`MVR 1 5`
	`JRC 2 DIO? 1 = 1`
	`JRC -1 ONT? 1 = 0`
	`CPY TARGET POS? 1`
	`MOV 1 ${TARGET}`
		Das Makro hat die selben Aufgaben wie das Makro HALT. Jedoch wird Achse 1 bei Tastendruck nicht durch den Befehl `HLT` angehalten, sondern das Ergebnis der Abfrage `POS? 1` wird in die Variable TARGET kopiert. Diese Variable wird dann als Zielposition für den Befehl `MOV` verwendet. Somit bleibt die Achse, wo sie gerade war. Zur Bereinigung

Aktion	Befehl	Ergebnis
	`VAR TARGET` `MAC END`	wird TARGET mit dem Befehl `VAR` als leer definiert, wodurch die Variable gelöscht wird.
Makro HALTVAR auf dem Controller starten.	`MAC START haltvar`	Die Bewegung der Achse 1 startet. Sie wird durch Schalten der digitalen Eingangsleitung 1 in den Zustand high (z.B. durch Tastendruck) angehalten. Fehlercode 10 wird nicht gesetzt, weil kein Halte- oder Stoppbefehl verwendet wird.

7.7.7 Makrobeispiel: Joysticksteuerung mit Speicherung von Positionen

Aufgabe:

Achse 1 soll mit einem Joystick gesteuert werden. Die Joystick-Steuerung soll nur aktiv sein, wenn gleichzeitig der Joystick-Button gedrückt ist. Durch die Tasten einer angeschlossenen Pushbutton-Box sollen außerdem bis zu vier Positionen im Controller gespeichert bzw. von der Achse angefahren werden können. Die LEDs der Pushbutton-Box sollen anzeigen, ob der Controller zum Speichern der aktuellen Position bereit ist und ob die Speicherung erfolgt ist.

Lösungsansatz:

Die Makros STARTUP, MAINLOOP, TESTJOYB, TESTDION und MVAX2ST werden auf dem Controller aufgezeichnet. Sie verwenden die globalen Variablen STORE1, STORE2, STORE3, STORE4, COUNTER sowie die lokalen Variablen 1 und 2.

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Aktion	Befehl	Ergebnis
Pushbutton-Box C-170.PB von PI an der Buchse **I/O** anschließen.	-	Die digitalen Eingangsleitungen 1 bis 4 sind in den Zustand high geschaltet, solange die entsprechende Taste gedrückt ist. Die Zustände der digitalen Ausgangsleitungen 1 bis 4 werden durch die LEDs dargestellt, die in den Tasten integriert sind.
Joystick C-819.20 oder C-819.30 an der Buchse **Joystick** anschließen.	-	Für Befehle ist die angeschlossene Joystick-Achse zugänglich als Achse 1 von Joystick-Gerät 1, und die Joystick-Taste ist zugänglich als Taste 1 von Joystick-Gerät 1.
Servomodus für Achse 1 einschalten.	`SVO 1 1`	Der Servomodus muss eingeschaltet sein, damit Achse 1 über einen Joystick gesteuert werden kann.
Referenzierungsfahrt für Achse 1 starten.	`FRF 1`	Die Achse startet eine Referenzierungsfahrt – hier zum Referenzschalter. Danach können absolute Achsenpositionen kommandiert werden.

Aktion	Befehl	Ergebnis
Bestimmen, welche Achse über die Joystick-Achse gesteuert werden soll.	`JAX 1 1 1`	Achse 1 wird der Joystick-Achse 1 des Joystickgeräts 1 zugewiesen. Die Joystick-Steuerung ist noch nicht aktiv.

Makro STARTUP auf dem Controller aufzeichnen.	`MAC BEG startup`
	`CPY STORE1 POS? 1`
	`CPY STORE2 POS? 1`
	`CPY STORE3 POS? 1`
	`CPY STORE4 POS? 1`
	`MAC START MAINLOOP`
	`MAC END`
		Das Makro hat folgende Aufgaben:
		▪ Variablen zur Speicherung der Position initialisieren
		▪ Makro MAINLOOP für die Hauptschleife starten

Makro MAINLOOP auf dem Controller aufzeichnen.	`MAC BEG mainloop`
	`MAC START TESTJOYB`
	`VAR COUNTER 1`
	`MAC START TESTDION ${COUNTER}`
	`ADD COUNTER ${COUNTER} 1`
	`JRC -2 VAR? COUNTER < 5`
	`MAC START MAINLOOP`
	`MAC END`
		Das Makro hat folgende Aufgaben:
		▪ Makro TESTJOYB für Joystick-Steuerung starten
		▪ Makro TESTDION nacheinander für alle digitalen Eingänge (d.h. alle Tasten der Pushbutton-Box) starten, unter Verwendung einer Schleife
		▪ Sich selbst aufrufen, um die Hauptschleife zu bilden

Makro TESTJOYB auf dem Controller aufzeichnen.	`MAC BEG testjoyb`	Das Makro hat folgende Aufgaben:
	`MEX JBS? 1 1 = 0`	▪ Makroausführung stoppen, wenn Joystick-Taste 1 nicht gedrückt ist
	`JON 1 1`	▪ Joystick-Gerät 1 aktivieren
	`DIO 0 15`	▪ Alle LEDs an der Pushbutton-Box einschalten
	`JRC 6 JBS? 1 1 = 0`	▪ 6 Zeilen nach vorn springen (zu JON 1 0), wenn Joystick-Taste 1 nicht mehr gedrückt ist
	`DEL 50`	▪ 50 ms warten
	`DIO 0 0`	▪ Alle LEDs an der Pushbutton-Box ausschalten
	`JRC 3 JBS? 1 1 = 0`	▪ 3 Zeilen nach vorn springen (zu JON 1 0), wenn Joystick-Taste 1 nicht mehr gedrückt ist
	`DEL 50`	▪ Weitere 50 ms warten
	`JRC -6 JBS? 1 1 = 1`	▪ 6 Zeilen zurück springen (zu DIO 0 15), wenn Joystick-Taste 1 immer noch gedrückt ist
	`JON 1 0`	▪ Joystick-Gerät 1 deaktivieren
	`DIO 0 0`	▪ Alle LEDs an der Pushbutton-Box ausschalten

Aktion	Befehl	Ergebnis
	`MAC END`	
Makro TESTDION auf dem Controller aufzeichnen.	`MAC BEG testdion`	Das Makro hat folgende Aufgaben:
	`MEX VAR? 0 != 1`	▪ Makroausführung stoppen, wenn die Anzahl der beim Start von TESTDION angegebenen lokalen Variablen nicht 1 ist
	`MEX DIO? $1 = 0`	▪ Makroausführung stoppen, wenn die durch die lokale Variable 1 angegebene Taste der Pushbutton-Box nicht mehr gedrückt ist (entsprechende Eingangsleitung hat den Zustand low)
	`DEL 300`	▪ 300 ms warten
	`JRC 3 DIO? $1 = 1`	▪ Wenn die Taste immer noch gedrückt ist, 3 Zeilen nach vorn springen (zu `DEL 400`)
	`MAC START MVAX2ST $1`	▪ Makro MVAX2ST starten, da die Taste nur kurz gedrückt wurde. Der Wert der lokalen Variablen 1 wird auch für die lokale Variable 1 in MVAX2ST verwendet. MVAX2ST bewegt Achse 1 zur Position, die für die Taste hinterlegt ist.
	`MEX DIO? $1 = 0`	▪ Makroausführung stoppen, wenn die Taste nicht mehr gedrückt ist
	`DEL 400`	▪ 400 ms warten
	`MEX DIO? $1 = 0`	▪ Makroausführung stoppen, wenn die Taste nicht mehr gedrückt ist
	`DIO $1 1`	▪ Die zur gedrückten Taste gehörende LED der Pushbutton-Box einschalten, um die Speicherung der aktuellen Position anzuzeigen
	`WAC DIO? $1 = 0`	▪ Das Makro wird erst dann weiter ausgeführt, wenn die Taste nicht mehr gedrückt ist
	`DIO $1 0`	▪ LED ausschalten
	`CPY STORE$1 POS? 1`	▪ Aktuelle Position der Achse 1 in der durch die lokale Variable 1 bezeichneten globalen Variable speichern
	`MAC END`	
Makro MVAX2ST auf dem Controller aufzeichnen.	`MAC BEG MVAX2ST`	Das Makro hat folgende Aufgaben:
	`CPY 2 VAR? STORE$1`	▪ Fragt die durch die lokale Variable 1 bezeichnete Speichervariable ab und kopiert ihren Wert in die lokale Variable 2
	`MOV 1 $2`	▪ Bewegung der Achse 1 zur Zielposition starten,

Aktion	Befehl	Ergebnis
		die durch die lokale Variable 2 vorgegeben ist
	`MAC END`	
Makro STARTUP auf dem Controller starten. Alternativ: Wenn die Variablen zur Speicherung von Positionen nicht initialisiert werden sollen, stattdessen das Makro MAINLOOP auf dem Controller starten.	`MAC START startup`	Die Joystick-Steuerung wird durch Drücken der Joystick-Taste aktiviert. Bei aktivierter Joystick-Steuerung blinken die LEDs der Pushbutton-Box schnell und zeigen damit an, dass die Tasten der Box nicht betätigt werden sollen. Nach dem Loslassen der Joystick-Taste ist die Joystick-Steuerung deaktiviert und die LEDs schalten sich ab. Die Pushbutton-Box kann jetzt zum Anfahren von gespeicherten Positionen oder zur Speicherung der aktuellen Position verwendet werden. Um den Positionierer zu einer gespeicherten Position zu bewegen, wird die entsprechende Taste der Pushbutton-Box kurz gedrückt. Um die aktuelle Position des Positionierers zu speichern, wird eine Taste an der Pushbutton-Box so lange gedrückt, bis die LED der Taste aufleuchtet.

7.7.8 Makrobeispiel: Joysticksteuerung mit Änderung der Geschwindigkeit

INFORMATION

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Aktion	Befehl	Ergebnis
Joystick C-819.20 oder C-819.30 an der Buchse **Joystick** anschließen.	-	Für Befehle ist die angeschlossene Joystick-Achse zugänglich als Achse 1 von Joystick-Gerät 1, und die Joystick-Taste ist zugänglich als Taste 1 von Joystick-Gerät 1.
Servomodus für Achse 1 einschalten.	`SVO 1 1`	Der Servomodus muss eingeschaltet sein, damit Achse 1 über einen Joystick gesteuert werden kann.
Referenzierungsfahrt für Achse 1 starten.	`FRF 1`	Die Achse startet eine Referenzierungsfahrt – hier zum Referenzschalter. Danach können absolute Achsenpositionen kommandiert werden.
Bestimmen, welche Achse über die Joystick-Achse gesteuert werden soll.	`JAX 1 1 1`	Achse 1 wird der Joystick-Achse 1 des Joystickgeräts 1 zugewiesen. Die Joystick-Steuerung ist noch nicht aktiv.
Makro JOYVEL auf dem Controller aufzeichnen.	`MAC BEG joyvel`	Das Makro hat folgende Aufgaben:
	`JON 1 1`	▪ Joystick-Gerät 1 aktivieren

Aktion	Befehl	Ergebnis
	`JRC 3 JBS? 1 1 = 1`	▪ Wenn die Joystick-Taste 1 gedrückt ist, 3 Zeilen nach vorn springen (zu `VEL 1 1`).
	`VEL 1 0.5`	▪ Die Maximalgeschwindigkeit während der Joystick-Steuerung beträgt 0,5.
	`JRC -2 JBS? 1 1 = 0`	▪ Wenn die Joystick-Taste 1 nicht gedrückt ist, 2 Zeilen zurück springen, um eine Schleife zu bilden.
	`VEL 1 1`	▪ Die Maximalgeschwindigkeit während der Joystick-Steuerung beträgt 1.
	`JRC -4 JON? 1 = 1`	▪ Wenn Joystick-Gerät 1 noch aktiv ist, 4 Zeilen zurück springen, um eine Schleife zu bilden.
	`MAC END`	
Makro JOYVEL auf dem Controller starten.	`MAC START joyvel`	Langsame Fahrt: Bewegen Sie den Steuerhebel des Joysticks. Schnelle Fahrt: Halten Sie die Drucktaste 1 des Joysticks gedrückt und bewegen Sie den Steuerhebel.

8 GCS-Befehle

8.1 Schreibweise

Für die Festlegung der GCS-Syntax und die Beschreibung der Befehle wird folgende Schreibweise verwendet:

<...>	Spitze Klammern kennzeichnen ein Befehlsargument, das die Kennung eines Elements oder ein befehlsspezifischer Parameter sein kann.
[…]	Eckige Klammern kennzeichnen eine optionale Angabe.
{…}	Geschweifte Klammern kennzeichnen die Wiederholung von Angaben, d. h. es kann auf mehr als ein Element (z. B. mehrere Achsen) in einer Befehlszeile zugegriffen werden.
LF	LineFeed (Zeilenvorschub, ASCII-Zeichen 10) ist das Standardabschlusszeichen (Zeichen am Ende einer Befehlszeile).
SP	Space (ASCII-Zeichen 32) steht für ein Leerzeichen.
"..."	Anführungszeichen zeigen an, dass die von ihnen eingeschlossenen Zeichen ausgegeben werden oder einzugeben sind.

8.2 GCS-Syntax für Syntaxversion 2.0

Ein GCS-Befehl besteht aus 3 Buchstaben, z. B. CMD. Dem dazugehörigen Abfragebefehl wird am Ende ein Fragezeichen hinzugefügt, z. B. CMD?.

Befehlskürzel:

CMD ::= Buchstabe1 Buchstabe2 Buchstabe3 [?]

Ausnahmen:

- Einzeichenbefehle, wie z. B. Befehle für schnelles Abfragen, bestehen aus nur einem ASCII-Zeichen. Geschrieben wird das ASCII-Zeichen als Kombination aus # und dem Code des Zeichens in Dezimalschreibweise, z. B. als `#24`.
- `*IDN?` (für GPIB-Kompatibilität).

Beim Befehlskürzel wird nicht zwischen Groß- und Kleinschreibung unterschieden. Das Befehlskürzel und alle Argumente (z. B. Achsen- und Kanalkennungen, Parameter etc.) müssen mit einem Leerzeichen (SP) voneinander getrennt werden. Die Befehlszeile muss mit Zeilenvorschub (LF) beendet werden.

CMD[{{SP}<Argument>}]LF

CMD?[{{SP}<Argument>}]LF

Ausnahme:

- Auf Einzeichenbefehle folgt kein Abschlusszeichen. Die Antwort auf einen Einzeichenbefehl enthält hingegen ein Abschlusszeichen.

Das Argument <AxisID> wird für die logischen Achsen des Controllers verwendet. Je nach Controller kann die Achsenkennung aus bis zu 16 Zeichen bestehen. Alle alphanumerischen Zeichen und der Unterstrich sind erlaubt. Die vom C-863.12 unterstützten Kennungen sind im Abschnitt "Kommandierbare Elemente" (S. 17) beschrieben.

Beispiel 1:

Achse 1 soll zur Position 10.0 bewegt werden. Die Einheit hängt vom Controller ab (z. B. µm oder mm).

Senden: `MOV`SP`1`SP`10.0`LF

Mehr als ein Befehlskürzel pro Zeile ist nicht erlaubt. Mehrere Gruppen von Argumenten sind nach einem Befehlskürzel erlaubt.

Beispiel 2:

Zwei Achsen, die mit demselben Controller verbunden sind, sollen bewegt werden:

Senden: `MOV`SP`1`SP`17.3`SP`2`SP`2.05`LF

Wenn ein Teil der Befehlszeile nicht ausgeführt werden kann, wird die gesamte Zeile nicht ausgeführt.

Wenn alle Argumente optional sind und weggelassen werden, wird der Befehl für alle möglichen Werte der Argumente ausgeführt.

Beispiel 3:

Alle Parameter im flüchtigen Speicher sollen zurückgesetzt werden.

Senden: `RPA`LF

Beispiel 4:

Die Position aller Achsen soll abgefragt werden.

Senden: `POS?`LF

Die Antwort-Syntax lautet wie folgt:

[<Argument>[{SP<Argument>}]"="]<Wert>LF

In mehrzeiligen Antworten wird in der letzten Zeile das Leerzeichen von dem Abschlusszeichen weggelassen:

{[<Argument>[{SP<Argument>}]"="]<Wert>SP LF}

[<Argument>[{SP<Argument>}]"="]<Wert>LF für die letzte Zeile!

In der Antwort werden die Argumente in derselben Reihenfolge aufgelistet wie im Abfragebefehl.

Abfragebefehl:

CMD?SP<Arg3>SP<Arg1>SP<Arg2>LF

Antwort auf diesen Befehl:

<Arg3>"="<Wert3>SP LF

<Arg1>"="<Wert1>SP LF

<Arg2>"="<Wert2> LF

Beispiel 5:

Senden: `TSP?SP2SP1LF`

Empfangen: `2=-1158.4405SP LF`

`1=+0000.0000LF`

INFORMATION

Beim C-863.12 kann nur ein einzelnes Element pro Befehlszeile adressiert werden (z. B. Achse oder Parameter).

Beispiel:

Durch Senden der Befehlszeile

`SEP 100 1 0x32 0`

wird ein neuer Wert des Parameters 0x32 für Achse 1 im permanenten Speicher gespeichert, das Senden der Befehlszeile

`SEP 100 1 0x32 0 1 0x14 1`

ist jedoch nicht möglich, weil zwei Parameter geändert werden sollen.

Wenn der Befehl dies unterstützt, können auch alle Elemente durch Weglassen der Elementkennung adressiert werden.

Beispiel:

Durch Senden der Befehlszeile

`SEP?`

werden die Werte aller Parameter aus dem permanenten Speicher abgefragt.

8.3 Empfänger- und Senderadresse

Grundsätzlich sind die Adressen des anzusprechenden Controllers (Empfänger) und des Senders in jeder Befehlszeile erforderlich. Dies gilt selbst für Einzeichenbefehle (z. B. #4), oder für die Aufzeichnung von Makros. Weil aber nur der PC Befehlszeilen an die Controller senden darf, kann seine Adresse (0) weggelassen werden. Jedoch sind sowohl die Empfänger- als auch die Senderadresse in jeder Controllerantwort enthalten. Mehrzeilige Antworten enthalten die Empfänger- und Senderadresse nur in der ersten Zeile.

Beispiel:

In einem Terminal-Programm wie z. B. PITerminal wird mit dem Befehl *IDN? die Ident-Bezeichnung eines Controllers mit der Adresse 2 (hier: ein C-863.11) abgefragt.

Senden: `2 0 *IDN?`

oder

Senden: `2 *IDN?`

Die Antwort lautet in beiden Fällen:

```
0 2 (c)2011 Physik Instrumente(PI) Karlsruhe, C-863.11,0,1.2.0.0
```

Ausnahme:

Die Empfängeradresse kann weggelassen werden, wenn der anzusprechende Controller die Adresse 1 hat, selbst wenn dieser Controller Bestandteil eines Daisy-Chain-Netzwerks ist. Wenn die Empfängeradresse beim Ansprechen des Controllers weggelassen wird, werden Empfänger- und Senderadresse auch in der Antwort des Controllers weggelassen.

Beispiel:

Senden: `*IDN?`

Der Controller mit der Adresse 1 (hier: ein C-863.11) antwortet:

```
©2011 Physik Instrumente(PI) Karlsruhe, C-863.11,0,1.2.0.0
```

Senden: `1 *IDN?`

Derselbe Controller antwortet:

```
0 1 (c)2011 Physik Instrumente(PI) Karlsruhe, C-863.11,0,1.2.0.0
```

Informationen zur Einstellung der Controlleradresse finden Sie im Abschnitt "DIP-Schalter-Einstellungen anpassen" (S. 53). Die Controlleradresse kann im Bereich von 1 bis 16 liegen; die Standardadresse ist 1. Der PC hat immer die Adresse 0. Mit der Sammeladresse 255 können alle Controller in einem Daisy-Chain-Netzwerk gleichzeitig angesprochen werden, wobei dann jedoch keine Antworten an den PC gesendet werden.

8.4 Variablen

Für eine flexiblere Programmierung unterstützt der C-863.12 Variablen. Während globale Variablen immer verfügbar sind, gelten lokale Variablen immer nur für ein bestimmtes Makro. Typischerweise werden Variablen in Makros verwendet.

Variablen sind nur im flüchtigen Speicher (RAM) vorhanden. Die Variablenwerte haben den Datentyp STRING.

Für Variablennamen gelten folgende Konventionen:

- Variablennamen dürfen keine Sonderzeichen enthalten (insbesondere kein "$").
- Höchstens 8 Zeichen sind erlaubt.
- Die Namen von globalen Variablen können aus den Zeichen A bis Z und 0 bis 9 bestehen. Sie müssen mit einem Buchstaben beginnen.
- Die Namen von lokalen Variablen dürfen keine Buchstaben enthalten. Mögliche Zeichen sind 0 bis 9.
- Der Variablenname kann auch über den Wert einer anderen Variablen angegeben werden.

Wenn der Wert einer Variablen verwendet werden soll, muss folgende Schreibweise angewandt werden:

- Dem Variablennamen muss ein "$" vorangestellt werden.
- Variablennamen, die aus mehreren Zeichen bestehen, müssen in geschweifte Klammern gesetzt werden.

Wenn der Variablenname aus nur einem Zeichen besteht, können die geschweiften Klammern weggelassen werden.

Wenn die geschweiften Klammern bei Variablennamen weggelassen werden, die aus mehreren Zeichen bestehen, wird das erste Zeichen nach dem "$" als der Variablenname interpretiert.

Lokale Variablen:

- Lokale Variablen können nur in Makros verwendet werden.
- Derzeit unterstützt die Controllerfirmware drei lokale Variablen: 0, 1 und 2.
- Die Werte der lokalen Variablen 1 und 2 werden als Argumente der Befehle `MAC START` oder `MAC NSTART` beim Start des Makros angegeben.

  Befehlsformate:

  `MAC START <macroname> [<String1> [<String2>]]`

  `MAC NSTART <macroname> <uint> [<String1> [<String2>]]`

  <STRING1> und <STRING2> geben die Werte für die im Makro verwendeten lokalen Variablen 1 und 2 an. <STRING1> und <STRING2> können direkt oder über Variablenwerte angegeben werden. <uint> bestimmt, wievielmal das Makro ausgeführt werden soll. Weitere Informationen finden Sie in der Beschreibung des Befehls `MAC` (S. 174).

- Die lokale Variable 0 kann nur gelesen werden. Ihr Wert gibt an, wieviele Argumente (d.h. Werte von lokalen Variablen) beim Start des Makros angegeben wurden.
- Innerhalb eines Makros können die Werte lokaler Variablen mit den Befehlen `ADD` (S. 134), `CPY` (S. 138) und `VAR` (S. 205) verändert und mit dem Befehl `VAR` gelöscht werden (Ausnahme: lokale Variable 0).
- Solange das Makro ausgeführt wird, können die Werte der lokalen Variablen abgefragt werden mit:

  `VAR? 0`

  `VAR? 1`

  `VAR? 2`

  Die Abfragen können innerhalb oder außerhalb des Makros gesendet werden.

Globale Variablen:

- Globale Variablen können innerhalb und außerhalb von Makros verwendet werden.
- Die maximale Anzahl globaler Variablen beträgt 10.
- Globale Variablen werden mit den Befehlen `ADD`, `CPY` oder `VAR` angelegt und verändert. Sie können mit dem Befehl `VAR` gelöscht werden.
- Die Variablenwerte können mit `VAR?` abgefragt werden.

8.5 Befehlsübersicht

Befehl	Format	Beschreibung
#4 (S. 130)	#4	Request Status Register
#5 (S. 131)	#5	Request Motion Status
#7 (S. 131)	#7	Request Controller Ready Status
#8 (S. 132)	#8	Query If Macro Is Running
#24 (S. 132)	#24	Stop All Axes
*IDN? (S. 132)	*IDN?	Get Device Identification
ACC (S. 133)	ACC {<AxisID> <Acceleration>}	Set Closed-Loop Acceleration
ACC? (S. 133)	ACC? [{<AxisID>}]	Get Closed-Loop Acceleration
ADD (S. 134)	ADD <Variable> <FLOAT1> <FLOAT2>	Add and Save To Variable

Befehl	Format	Beschreibung
BRA (S. 136)	BRA {<AxisID> <BrakeState>}	Set Brake Activation State
BRA? (S. 137)	BRA? [{<AxisID>}]	Get Brake Activation State
CCL (S. 137)	CCL? <Level> [<PSWD>]	Set Command Level
CCL? (S. 138)	CCL?	Get Command Level
CPY (S. 138)	CPY <Variable> <CMD?>	Copy Into Variable
CST? (S. 139)	CST? [{<AxisID>}]	Get Assignment Of Stages To Axes
CSV? (S. 140)	CSV?	Get Current Syntax Version
CTO (S. 140)	CTO {<TrigOutID> <CTOPam> <Value>}	Set Configuration Of Trigger Output
CTO? (S. 144)	CTO? [{<TrigOutID> <CTOPam>}]	Get Configuration Of Trigger Output
DEC (S. 144)	DEC {<AxisID> <Deceleration>}	Set Closed-Loop Deceleration
DEC? (S. 145)	DEC? [{<AxisID>}]	Get Closed-Loop Deceleration
DEL (S. 145)	DEL <uint>	Delay The Command Interpreter
DFH (S. 146)	DFH [{<AxisID>}]	Define Home Position
DFH? (S. 147)	DFH? [{<AxisID>}]	Get Home Position Definition
DIO (S. 148)	DIO {<DIOID> <OutputOn>}	Set Digital Output Lines
DIO? (S. 149)	DIO? [{<DIOID>}]	Get Digital Input Lines
DRC (S. 149)	DRC {<RecTableID> <Source> <RecOption>}	Set Data Recorder Configuration
DRC? (S. 151)	DRC? [{<RecTableID>}]	Get Data Recorder Configuration
DRL? (S. 151)	DRL? [{<RecTableID>}]	Get Number Of Recorded Points

Befehl	Format	Beschreibung
DRR? (S. 152)	DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]	Get Recorded Data Values
DRT (S. 153)	DRT {<RecTableID> <TriggerSource> <Value>}	Set Data Recorder Trigger Source
DRT? (S. 154)	DRT? [{<RecTableID>}]	Get Data Recorder Trigger Source
ERR? (S. 155)	ERR?	Get Error Number
FED (S. 156)	FED {<AxisID> <EdgeID> <Param>}	Find Edge
FNL (S. 157)	FNL [{<AxisID>}]	Fast Reference Move To Negative Limit
FPL (S. 158)	FPL [{<AxisID>}]	Fast Reference Move To Positive Limit
FRF (S. 159)	FRF [{<AxisID>}]	Fast Reference Move To Reference Switch
FRF? (S. 160)	FRF? [{<AxisID>}]	Get Referencing Result
GOH (S. 161)	GOH [{<AxisID>}]	Go To Home Position
HDR? (S. 161)	HDR?	Get All Data Recorder Options
HLP? (S. 162)	HLP?	Get List of Available Commands
HLT (S. 162)	HLT [{<AxisID>}]	Halt Motion Smoothly
HPA? (S. 163)	HPA?	Get List Of Available Parameters
HPV? (S. 164)	HPV?	Get List Of Possible Parameter Values
JAS? (S. 165)	JAS? [{<JoystickID> <JoystickAxis>}]	Query Joystick Axis Status
JAX (S. 166)	JAX <JoystickID> <JoystickAxis> <AxisID>	Set Axis Controlled By Joystick
JAX? (S. 167)	JAX? [{<JoystickID> <JoystickAxis>}]	Get Axis Controlled By Joystick
JBS? (S. 167)	JBS? [{<JoystickID> <JoystickButton>}]	Query Joystick Button Status

Befehl	Format	Beschreibung
JDT (S. 168)	JDT {<JoystickID> <JoystickAxis> <uint>}	Set Joystick Default Lookup Table
JLT (S. 168)	JLT <JoystickID> <JoystickAxis> <Addr> <floatn>	Fill Joystick Lookup Table
JLT? (S. 170)	JLT? [<StartPoint> <NumberOfPoints> [{<JoystickID> <JoystickAxis>}]]	Get Joystick Lookup Table Values
JON (S. 171)	JON {<JoystickID> <uint>}	Set Joystick Activation Status
JON? (S. 172)	JON? [{<JoystickID>}]	Get Joystick Activation Status
JRC (S. 172)	JRC <Jump> <CMD?> <OP> <Value>	Jump Relatively Depending On Condition
LIM? (S. 173)	LIM? [{<AxisID>}]	Indicate Limit Switches
MAC (S. 174)	MAC <keyword> {<parameter>} insbesondere: MAC BEG <macroname> MAC DEF <macroname> MAC DEF? MAC DEL <macroname> MAC END MAC ERR? MAC NSTART <macroname> <uint> [<String1> [<String2>]] MAC START <macroname> [<String1> [<String2>]]	Call Macro Function
MAC? (S. 177)	MAC? [<macroname>]	List Macros
MAN? (S. 177)	MAN? <CMD>	Get Help String For Command
MAT (S. 178)	MAT <Variable> <=> <FLOAT1> <OP> <FLOAT2>	Calculate And Save To Variable
MEX (S. 179)	MEX <CMD?> <OP> <Value>	Stop Macro Execution Due To Condition
MOV (S. 181)	MOV {<AxisID> <Position>}	Set Target Position
MOV? (S. 182)	MOV? [{<AxisID>}]	Get Target Position
MVR (S. 183)	MVR {<AxisID> <Distance>}	Set Target Relative To Current Position

Befehl	Format	Beschreibung
ONT? (S. 184)	ONT? [{<AxisID>}]	Get On-Target State
POS (S. 184)	POS {<AxisID> <Position>}	Set Real Position
POS? (S. 185)	POS? [{<AxisID>}]	Get Real Position
RBT (S. 185)	RBT	Reboot System
RMC? (S. 186)	RMC?	List Running Macros
RON (S. 186)	RON {<AxisID> <ReferenceOn>}	Set Reference Mode
RON? (S. 187)	RON? [{<AxisID>}]	Get Reference Mode
RPA (S. 187)	RPA [{<ItemID> <PamID>}]	Reset Volatile Memory Parameters
RTR (S. 188)	RTR <RecordTableRate>	Set Record Table Rate
RTR? (S. 188)	RTR?	Get Record Table Rate
SAI (S. 189)	SAI {<AxisID> <NewIdentifier>}	Set Current Axis Identifiers
SAI? (S. 189)	SAI? [ALL]	Get List Of Current Axis Identifiers
SEP (S. 190)	SEP <Pswd> {<ItemID> <PamID> <PamValue>}	Set Non-Volatile Memory Parameters
SEP? (S. 191)	SEP? [{<ItemID> <PamID>}]	Get Non-Volatile Memory Parameters
SMO (S. 191)	SMO {<AxisID> <ControlValue>}	Set Open-Loop Control Value
SMO? (S. 192)	SMO? [{<AxisID>}]	Get Control Value
SPA (S. 193)	SPA {<ItemID> <PamID> <PamValue>}	Set Volatile Memory Parameters
SPA? (S. 195)	SPA? [{<ItemID> <PamID>}]	Get Volatile Memory Parameters
SRG? (S. 196)	SRG? {<AxisID> <RegisterID>}	Query Status Register Value

Befehl	Format	Beschreibung
STE (S. 197)	STE <AxisID> <Amplitude>	Start Step And Response Measurement
STP (S. 198)	STP	Stop All Axes
SVO (S. 198)	SVO {<AxisID> <ServoState>}	Set Servo Mode
SVO? (S. 199)	SVO? [{<AxisID>}]	Get Servo Mode
TAC? (S. 200)	TAC?	Tell Analog Channels
TAV? (S. 200)	TAV? [{<AnalogInputID>}]	Get Analog Input Voltage
TCV? (S. 201)	TCV? [{AxisID}]	Get Commanded Closed-Loop Velocity
TIO? (S. 201)	TIO?	Tell Digital I/O Lines
TMN? (S. 201)	TMN? [{<AxisID>}]	Get Minimum Commandable Position
TMX? (S. 202)	TMX? [{<AxisID>}]	Get Maximum Commandable Position
TNR? (S. 202)	TNR?	Get Number Of Record Tables
TRO (S. 203)	TRO {<TrigOutID> <TrigMode>}	Set Trigger Output State
TRO? (S. 203)	TRO? [{<TrigOutID>}]	Get Trigger Output State
TRS? (S. 204)	TRS? [{<AxisID>}]	Indicate Reference Switch
TVI? (S. 204)	TVI?	Tell Valid Character Set For Axis Identifiers
VAR (S. 205)	VAR <Variable> <String>	Set Variable Value
VAR? (S. 206)	VAR? [{<Variable>}]	Get Variable Value
VEL (S. 206)	VEL {<AxisID> <Velocity>}	Set Closed-Loop Velocity
VEL? (S. 207)	VEL? [{<AxisID>}]	Get Closed-Loop Velocity

Befehl	Format	Beschreibung
VER? (S. 207)	VER?	Get Versions Of Firmware And Drivers
WAC (S. 208)	WAC <CMD?> <OP> <Value>	Wait For Condition
WPA (S. 209)	WPA <Pswd> [{<ItemID> <PamID>}]	Save Parameters To Non-Volatile Memory

8.6 Befehlsbeschreibungen für GCS 2.0

#4 (Request Status Register)

Beschreibung: Fragt die Systemstatusinformation ab.

Format: #4

Argumente: Keine

Antwort: Die Antwort ist bit-codiert. Für die individuellen Codes siehe unten.

Hinweise: Dieser Befehl ist funktionsgleich mit SRG? (S. 196), aber es wird nur ein Zeichen über die Schnittstelle gesendet. Deshalb kann #4 auch verwendet werden, wenn der Controller zeitaufwändige Aufgaben ausführt.

Für den C-863.12 ist die Antwort die Summe der folgenden Codes in Hexadezimalformat:

Bit	15	14	13	12	11	10	9	8
Beschreibung	On-Target-Status	Führt Referenzierung aus	In Bewegung	Servomodus Ein	-	-	-	Fehlerflag

Bit	7	6	5	4	3	2	1	0
Beschreibung	Digitale Eingangsleitung 4	Digitale Eingangsleitung 3	Digitale Eingangsleitung 2	Digitale Eingangsleitung 1	-	Pos. Endschalter	Referenzschalter	Neg. Endschalter

Beispiel: Senden: `#4`
Empfangen: `0x9005`
Hinweis: Die Antwort wird im Hexadezimalformat angegeben. Sie besagt: die Achse ist an der Zielposition (On-Target-Status = wahr), der Servomodus ist eingeschaltet, es ist kein Fehler aufgetreten, der Status der digitalen Eingangsleitungen 1 bis 4 ist low, und der Positionierer befindet sich auf der positiven Seite des Referenzschalters (Endschalter sind nicht aktiv, beachten Sie, dass die Logik der Signale in diesem Beispiel invertiert ist).

#5 (Request Motion Status)

Beschreibung: Fragt den Bewegungsstatus der Achsen ab.

Format: #5

Argumente: Keine

Antwort: Die Antwort <uint> ist bit-codiert und wird als hexadezimale Summe der folgenden Codes zurückgegeben:

1 = erste Achse bewegt sich
2 = zweite Achse bewegt sich
4 = dritte Achse bewegt sich
...
0 gibt an, dass die Bewegung aller Achsen abgeschlossen ist.

#7 (Request Controller Ready Status)

Beschreibung: Fragt den Bereitschaftsstatus des Controllers ab (prüft, ob Controller zum Ausführen eines neuen Befehls bereit ist).

Hinweis: Verwenden Sie #5 (S. 131) anstelle von #7, um zu verifizieren, ob die Bewegung beendet ist.

Format: #7

Argumente: Keine

Antwort: B1h (ASCII Zeichen 177 = "±" in Windows) wenn Controller bereit ist

B0h (ASCII Zeichen 176 = "°" in Windows) wenn Controller nicht bereit ist
(z. B. führt eine Referenzierungsfahrt aus)

Fehlersuche: Die Antwortzeichen können in nicht-westeuropäischen Zeichensätzen oder anderen Betriebssystemen unterschiedlich angezeigt werden.

#8 (Query if Macro Is Running)

Beschreibung: Prüft, ob ein Makro auf dem Controller ausgeführt wird.

Format: #8

Argumente: Keine

Antwort: <uint>=0 wenn kein Makro ausgeführt wird
<uint>=1 wenn ein Makro aktuell ausgeführt wird

#24 (Stop All Axes)

Beschreibung: Stoppt alle Achsen abrupt. Nähere Angaben siehe Hinweise unten.

Setzt den Fehlercode auf 10.

Dieser Befehl ist funktionsgleich mit STP (S. 198), aber es wird nur ein Zeichen über die Schnittstelle gesendet.

Format: #24

Argumente: Keine

Antwort: Keine

Hinweise: #24 stoppt jede Bewegung, die durch Bewegungsbefehle (z. B. MOV (S. 181), MVR (S. 183), GOH (S. 161), STE (S. 197), SMO (S. 191)), Befehle zur Referenzierung (FNL (S. 157), FPL (S. 158), FRF (S. 159)) und Makros (MAC (S. 174)) verursacht wird. Stoppt auch die Makroausführung.

Nachdem die Achsen gestoppt wurden, werden ihre Zielpositionen auf ihre aktuellen Positionen gesetzt.

HLT (S. 162) stoppt im Gegensatz zu #24 die Bewegung mit angegebener Abbremsung im Hinblick auf die Systemträgheit.

***IDN? (Get Device Identification)**

Beschreibung: Fragt die Ident-Bezeichnung des Geräts ab.

Format: *IDN?

Argumente: Keine

Antwort: Mit dem Abschlusszeichen (line feed) beendeter einzeiliger Text mit Controllername, Seriennummer und Firmwareversion

ACC (Set Closed-Loop Acceleration)

Beschreibung: Setzt die Beschleunigung für die angegebenen Achsen.

ACC kann verändert werden, während die Achse sich bewegt.

Format: ACC {<AxisID> <Acceleration>}

Argumente: <AxisID> ist eine Achse des Controllers

<Acceleration> ist der Wert der Beschleunigung in physikalischen Einheiten pro s^2.

Antwort: Keine

Fehlersuche: Unzulässige Achsenkennungen

Hinweise: Die ACC-Einstellung wird nur wirksam, wenn sich die angegebene Achse im geregelten Betrieb befindet (Servomodus EIN).

Der kleinstmögliche Wert für <Acceleration> ist 0.

ACC ändert den Wert des Parameters ***Closed-Loop Acceleration (Phys. Unit/s^2)*** (ID 0xB) im flüchtigen Speicher des C-863.12. Der Parameterwert kann mit WPA (S. 209) als Standard gespeichert werden, Details siehe "Anpassen von Einstellungen" (S. 235).

Der maximale Wert, der mit dem Befehl ACC gesetzt werden kann, wird durch den Parameter ***Maximum Closed-Loop Acceleration (Phys. Unit/s^2)*** (ID 0x4A) angegeben.

ACC? (Get Closed-Loop Acceleration)

Beschreibung: Fragt den mit ACC (S. 133) gesetzten Wert der Beschleunigung ab.

Werden alle Argumente weggelassen, wird der mit ACC gesetzte Wert aller Achsen abgefragt.

Format: ACC? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers

Antwort: {<AxisID>"="<float> LF}

wobei

<float> der mit ACC gesetzte Beschleunigungswert ist, in physikalischen Einheiten pro s^2.

ADD (Add And Save To Variable)

Beschreibung: Addiert zwei Werte und speichert das Ergebnis als Variable (S. 123).

Die Variable ist nur im flüchtigen Speicher (RAM) vorhanden.

Format: ADD <Variable> <FLOAT1> <FLOAT2>

Argumente: <Variable> ist der Name der Variable, in der das Ergebnis gespeichert werden soll.

<FLOAT1> ist der erste Summand.

<FLOAT2> ist der zweite Summand.

Für die Summanden werden Gleitkommazahlen erwartet. Sie können direkt angegeben werden oder über den Wert einer Variablen.

Antwort: Keine

Hinweise: Lokale Variablen können mit ADD nur in Makros gesetzt werden.

Beispiel 1: Wert $B wird zu Wert $A addiert und das Ergebnis wird als Variable C gespeichert:

```
ADD C $A $B
```

Beispiel 2: Der Name der Variablen, in die das Ergebnis kopiert werden soll, wird über den Wert einer anderen Variablen angegeben:

Senden: `VAR?`
Empfangen:
```
A=468
B=123
3Z=WORKS
```

Senden: `ADD A${3Z} $A $B`
Senden: `VAR?`

Empfangen:
```
A=468
B=123
AWORKS=591
3Z=WORKS
```

Senden: `ADD ${3Z} $A $B`
Senden: `VAR?`
Empfangen:
```
A=468
B=123
AWORKS=591
WORKS=591
3Z=WORKS
```

Beispiel 3:

Unter Verwendung der nachstehenden Makros ist es möglich, mit LEDs, die an die digitalen Ausgangsleitungen des Controllers angeschlossen sind, ein "Blinklicht" zu erzeugen. $1 und $2 sind Werte lokaler Variablen und müssen als Argumente des Befehls MAC START oder MAC NSTART beim Start der Makros angegeben werden (siehe unten).

DIO 0 <bitmask>: Setzt die Ausgangskanäle gemäß <bitmask>. "DIO 0 5" aktiviert beispielsweise die Kanäle 1 und 3 und deaktiviert alle anderen Kanäle (5 ist 0000 0101 in Binärschreibweise).

Führen Sie folgende Schritte zur Implementierung des "Blinklichts" durch:

1. Schreiben Sie das Makro "STEPS":
```
MAC BEG STEPS
DIO 0 $1
ADD 1 $1 1
DEL $2
JRC -3 VAR? 1 <= 15
ADD 1 $1 -1
DIO 0 $1
DEL $2
JRC -3 VAR? 1 > 0
MAC END
```
2. Schreiben Sie das Makro "TEST":
```
MAC BEG TEST
MAC START STEPS 0 $1
ADD 1 $1 10
```

```
JRC -2 VAR? 1 < 110
VAR 1 10
ADD 2 $2 -1
JRC -5 VAR? 2 > 0
MAC END
```

3. Starten Sie das Makro TEST mit Argumenten, die die Variablenwerte $1 und $2 definieren:

```
MAC START Test 10 50
```

Bedeutung der Variablenwerte dabei:

$1: Verzögerung in ms zwischen jedem Schritt im Makro STEPS. Der Wert wird durch das Makro TEST jeweils um 10 erhöht, bis er 110 erreicht hat.

$2: Anzahl der Wiederholungen der gesamten "Blinklicht"-Prozedur.

BRA (Set Brake Activation State)

Beschreibung:	Aktiviert/deaktiviert Bremse für angegebene Achsen.
Format:	BRA {<AxisID> <BrakeState>}
Argumente:	<AxisID> ist eine Achse des Controllers <BrakeState> kann die folgenden Werte haben: 0 = Bremse deaktiviert 1 = Bremse aktiviert
Antwort:	Keine
Fehlersuche:	Unzulässige Achsenkennung
Hinweise:	Die Bremse kann nur verwendet werden, wenn der Parameter 0x1A (***Has Brake?***) den Wert 1 ("ja") hat. Wenn der Parameter 0x1A (***Has Brake?***) den Wert 1 ("ja") hat, gilt Folgendes: ▪ Die Bremse kann mit BRA nur aktiviert oder deaktiviert werden, wenn der Servomodus ausgeschaltet ist. Sichern Sie den Positionierer gegen unbeabsichtigte Bewegungen, bevor Sie die Bremse mit BRA deaktivieren! ▪ Das Setzen des Servomodus mit SVO (S. 198) beeinflusst den Aktivierungszustand der Bremse: – Einschalten des Servomodus deaktiviert die Bremse. – Ausschalten des Servomodus aktiviert die Bremse. ▪ Beim Auftreten eines Bewegungsfehlers (S. 75) wird der Servomodus ausgeschaltet, und die Bremse wird

aktiviert.

Wenn der integrierte Bremsentreiber des C-863.12 verwendet werden soll, muss zusätzlich der Parameter 0x3094 (***Internal Brake***) den Wert 1 haben. Weitere Informationen siehe die Beschreibungen in "Parameterübersicht" (S. 244).

BRA? (Get Brake Activation State)

Beschreibung:	Fragt den Status der Bremsenaktivierung für die angegebenen Achsen ab. Werden keine Argumente angegeben, wird der Status aller Achsen abgefragt.
Format:	BRA? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers
Antwort:	{<AxisID>"="<BrakeState> LF} wobei <BrakeState> der aktuelle Status der Bremsenaktivierung der Achse ist: 0 = Bremse deaktiviert 1 = Bremse aktiviert
Fehlersuche:	Unzulässige Achsenkennung

CCL (Set Command Level)

Beschreibung:	Ändert die aktive "Befehlsebene" und bestimmt somit die Verfügbarkeit von Befehlen und von Schreibzugriff auf Systemparameter.
Format:	CCL <Level> [<PSWD>]
Argumente:	<Level> ist eine Befehlsebene des Controllers <PSWD> ist das Kennwort, das für den Wechsel in die entsprechende Befehlsebene erforderlich ist Es gelten folgende Befehlsebenen und Kennwörter: Level = 0 ist die Standardeinstellung, alle Befehle, die dem "normalen" Benutzer zur Verfügung gestellt werden, sowie der Lesezugriff auf alle Parameter sind zugänglich, kein Kennwort erforderlich.

Level = 1 fügt zusätzliche Befehle und den Schreibzugriff für Parameter der Ebene 1 hinzu (Befehle und Parameter der Ebene 0 sind inbegriffen). Das erforderliche Kennwort lautet "advanced".

Level > 1 ist nur für Servicepersonal von PI vorgesehen. Die Benutzer können nicht zu einer Ebene > 1 wechseln. Wenn Sie Probleme mit Parametern der Ebene 2 oder höher haben sollten, wenden Sie sich an den Kundendienst (S. 263).

Antwort: Keine

Fehlersuche: Ungültiges Kennwort

Hinweise: Beim C-863.12 legen die Befehlsebenen nur das Schreibrecht auf die Parameter fest. Die Verfügbarkeit der Befehle des C-863.12 ist unabhängig von der aktiven Befehlsebene.

HPA? (S. 163) listet die Parameter einschließlich der Information darüber, welche Befehlsebene Schreibzugriff auf sie erlaubt. Weitere Informationen zur Parameterverwendung siehe "Anpassen von Einstellungen" (S. 235).

Nach dem Einschalten oder Neustart des Controllers ist die aktive Befehlsebene immer 0.

CCL? (Get Command Level)

Beschreibung: Fragt die aktive "Befehlsebene" ab.

Format: CCL?

Argumente: Keine

Antwort: <Level> ist die aktuell aktive Befehlsebene; uint.

Hinweise: <Level> sollte 0 oder 1 sein.

<Level> = 0 ist die Werkseinstellung, auf Parameter der Ebene 0 besteht Schreibzugriff, auf alle Parameter besteht Lesezugriff

<Level> = 1 lässt Schreibzugriff für Parameter der Ebene 1 zu (Parameter von Ebene 0 sind inbegriffen)

CPY (Copy Into Variable)

Beschreibung: Kopiert eine Antwort auf einen Befehl in eine Variable (S. 123).

Die Variable ist nur im flüchtigen Speicher (RAM) vorhanden.

Format: CPY <Variable> <CMD?>

Argumente: <Variable> ist der Name der Variablen, in die die Befehlsantwort kopiert werden soll.

<CMD?> ist ein Abfragebefehl in seiner üblichen Schreibweise. Die Antwort muss ein einzelner Wert sein (und nicht mehr).

Antwort: Keine

Hinweise: Lokale Variablen können mit CPY nur in Makros gesetzt werden.

Beispiel 1: Unter Verwendung des nachstehenden Makros ist es möglich, die digitalen Eingangs- und Ausgangsleitungen des Controllers durchzukontaktieren. 1 ist eine lokale Variable, deren Wert als Argument des Befehls MAC START oder MAC NSTART beim Start der Makros angegeben werden muss.

Schreiben Sie das Makro "connect":

```
MAC BEG connect
CPY 1 DIO? 0
DIO 0 $1
MAC START CONNECT
MAC END
```

Beispiel 2: Es ist möglich, den Wert einer Variable (z. B. SOURCE) in eine andere Variable (z. B. TARGET) zu kopieren:

```
CPY TARGET VAR? SOURCE
```

CST? (Get Assignment Of Stages To Axes)

Beschreibung: Fragt den Namen des Positionierertyps ab, der an die angegebene Achse angeschlossen ist.

Format: CST? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers

Antwort: {<AxisID>"="<string> LF}

wobei

	<string> der Name des Positionierertyps ist, der der Achse zugewiesen ist.
Hinweise:	Der Positionierername wird vom Parameter ***Stage Name*** gelesen (ID 0x3C), dessen Werkseinstellungswert "DEFAULT_STAGE" ist. Sie können mit SPA (S. 193) oder SEP (S. 190) den Parameterwert auf den Namen Ihres Positionierers setzen. Einzelheiten finden Sie in der Parameterübersicht (S. 244).

CSV? (Get Current Syntax Version)

Beschreibung:	Fragt die GCS-Syntaxversion ab, die in der Firmware verwendet wird.
Format:	CSV?
Argumente:	Keine
Antwort:	Die aktuelle GCS-Syntaxversion
Hinweise:	1.0 (für GCS 1.0) oder 2.0 (für GCS 2.0) sind mögliche Antworten.

CTO (Set Configuration Of Trigger Output)

Beschreibung:	Konfiguriert die Bedingungen für die Triggerausgabe für die angegebene digitale Ausgangsleitung.
Format:	CTO {<TrigOutID> <CTOPam> <Value>}
Argumente:	<TrigOutID> ist eine digitale Ausgangsleitung des Controllers; weitere Angaben siehe unten. <CTOPam> ist die ID des CTO-Parameters im Dezimalformat; vorhandene IDs siehe unten. <Value> ist der Wert, auf den der CTO-Parameter gesetzt wird; siehe unten.
Antwort:	Keine
Hinweise:	Die Bedingungen für die Triggerausgabe werden aktiv, wenn sie mit TRO (S. 203) aktiviert werden. Verwenden Sie DIO (S. 148) nicht für digitale Ausgangsleitungen, bei denen die Triggerausgabe mit TRO aktiviert ist. Die CTO-Einstellungen gehen verloren, wenn C-863.12 ausgeschaltet oder neugestartet wird. Durch Speichern in einem Makro können sie auf einfache Weise beibehalten werden.
Vorhandene	<TrigOutID> entspricht den digitalen Ausgangsleitungen 1

Ausgangs-leitungen und Triggerbe-dingungen:

bis 4, IDs = 1 bis 4; siehe "I/O" (S. 270).

<CTOPam> Parameter-IDs, verfügbar für C-863.12:
1 = TriggerStep
2 = Axis
3 = TriggerMode
7 = Polarity
8 = StartThreshold
9 = StopThreshold
10 = TriggerPosition

<Value> verfügbar für die entsprechende <CTOPam> ID:

für TriggerStep: Strecke

für Axis: die Kennung der Achse, die mit der digitalen Ausgangsleitung verbunden werden soll. Irrelevant für den Triggermodus MotionError.

für TriggerMode (Standardwert ist 0):

- 0 = PositionDistance;
  es wird jedes Mal ein Triggerpuls geschrieben, wenn die Achse die Strecke TriggerStep zurückgelegt hat (<CTOPam> ID 1). Optional können Werte für StartThreshold und StopThreshold (<CTOPam> IDs 8 und 9) definiert werden, um die Triggerausgabe nur für einen begrenzten Positionsbereich und eine bestimmte Bewegungsrichtung zu aktivieren (negativ oder positiv; Hinweis: Falls sich die Bewegungsrichtung umkehrt, bevor die Achsenposition den Stop-Schwellenwert erreicht hat, werden weiterhin Triggerpulse erzeugt). Werden StartThreshold und StopThreshold auf den gleichen Wert gesetzt, werden sie nicht verwendet.
- 2 = OnTarget;
  der On-Target-Status der gewählten Achse wird an die gewählte digitale Ausgangsleitung übertragen (dieser Status kann auch mit dem Befehl ONT? gelesen werden).
- 5 = MotionError;
  die gewählte digitale Ausgangsleitung wird aktiv, wenn ein Bewegungsfehler auftritt. Die Leitung bleibt aktiv, bis der Fehlercode auf 0 zurückgesetzt wird (durch eine Abfrage mit ERR?).
- 6 = InMotion;
  die gewählte digitale Ausgangsleitung ist solange aktiv, wie die gewählte Achse in Bewegung ist (der Bewegungszustand kann auch mit Befehlen, z. B. SRG?

oder #5, gelesen werden).

- 7 = Position+Offset;
  der erste Triggerpuls wird geschrieben, wenn die Achse die durch TriggerPosition (<CTOPam> ID 10) angegebene Position erreicht hat. Die nächsten Triggerpulse werden jeweils geschrieben, wenn die Achsenposition gleich der Summe der letzten gültigen Triggerposition und der durch TriggerStep (<CTOPam> ID 1) angegebenen Strecke ist. Die Triggerausgabe wird beendet, wenn die Achsenposition den durch StopThreshold (<CTOPam> ID 9) angegebenen Wert übersteigt. Das Vorzeichen des Wertes TriggerStep bestimmt, für welche Bewegungsrichtung Triggerpulse ausgegeben werden sollen. Die Triggerverarbeitung erfolgt durch den DSP des C-863.12.
- 8 = SinglePosition;
  die gewählte digitale Ausgangsleitung ist aktiv, wenn die Achsenposition die durch TriggerPosition (<CTOPam> ID 10) angegebene Position erreicht hat oder überschreitet.

für Polarity (Standardwert ist 1): setzt die Signalpolarität für die digitale Ausgangsleitung
0 = low-aktiv
1 = high-aktiv

für StartThreshold/StopThreshold: Positionswert;
bei Verwendung für den Triggermodus PositionDistance müssen beide Schwellenwerte gesetzt werden, um den Positionsbereich und die Bewegungsrichtung für die Triggerausgabe zu bestimmen;
StopThreshold wird als Stopp-Bedingung für den Triggermodus Position+Offset verwendet

für TriggerPosition: Positionswert;
bei Verwendung im Triggermodus Position+Offset wird an dieser Position der erste Triggerpuls ausgeben;
bei Verwendung im Triggermodus SinglePosition ist die Ausgangsleitung aktiv, wenn diese Position erreicht oder überschritten ist

Anwendungsbeispiele und weitere Angaben siehe "Digitale Ausgangssignale" (S. 78) und nachstehende Zeilen.

Beispiel 1: Ein Puls soll an der digitalen Ausgangsleitung 1 (ID 1) erzeugt werden, wenn Achse 1 eine Distanz von 0,05 µm zurückgelegt hat. Folgende Parameter müssen gesetzt werden:

TrigOutID = 1

Axis = 1
TriggerMode = 0
TriggerStep = 0.05
Senden: `CTO 1 2 1`
Senden: `CTO 1 3 0`
Senden: `CTO 1 1 0.00005`

Beispiel 2: In diesem Beispiel soll die digitale Ausgangsleitung 1 von low auf high gesetzt werden, wenn Achse A ihre Bewegung beginnt. Folgende Parameter müssen gesetzt werden:
TrigOutID = 1
Axis = A (Achsenkennung wurde mit `SAI` geändert)
TriggerMode = 6
Polarity = high-aktiv
Sie müssen also Folgendes senden:

```
CTO 1 2 A
CTO 1 3 6
CTO 1 7 1
```

Beispiel 3: L-509.1xxxxx (Stellweg: 26 mm) wird mit Achse 1 verbunden. Die Referenzposition von L-509.1xxxxx ist 13 mm. Von ihrer Referenzposition ausgehend soll die Achse abwechselnd vorwärts und rückwärts bewegt werden; Triggerpulse sollen für beide Bewegungsrichtungen in einem Bereich von 1 mm mit dem Triggermodus Position+Offset ausgegeben werden. Zu diesem Zweck werden zwei Makros in den Controller geschrieben. Das Makro TRIGREF initialisiert den Controller und kann auch als Startup-Makro festgelegt werden, während das Makro TRIGGER die Bewegung und somit die Triggerausgabe startet. Schreiben Sie die Makros wie nachstehend beschrieben. Nähere Angaben zu Makros siehe "Mit Makros arbeiten" (S. 102).

Stellen Sie sicher, dass die Geschwindigkeitseinstellung für die Achse zur CTO-Einstellung für die Strecke passt.
Empfohlener Wert:
Höchstgeschwindigkeit = Strecke * 20 kHz / 2
wobei 20 kHz die Frequenz des Regelzyklus des C-863.12 ist.

Bei einer Geschwindigkeit von 20 mm/s ergibt sich eine Frequenz des Triggersignals von 1 kHz.

➢ Zeichnen Sie ein Makro namens TRIGREF mit folgendem Inhalt auf:

```
CTO 1 3 7
SVO 1 1
```

```
FRF
TRO 1 1
MAC START TRIGGER
```

➢ Zeichnen Sie ein Makro namens TRIGGER mit folgendem Inhalt auf:

```
CTO 1 1 0.02
CTO 1 9 15
CTO 1 10 14
DEL 1000
DRT 0 2 0
MOV 1 15.01
WAC POS? 1 > 14.8
MEX CTO? 1 10 < 13.9
CTO 1 1 -0.02
CTO 1 9 14
CTO 1 10 15
DEL 1000
MOV 1 13.99
WAC POS? 1 < 14
MEX CTO? 1 10 > 14.1
MAC START TRIGGER
```

CTO? (Get Configuration Of Trigger Output)

Beschreibung:	Fragt die Werte ab, die für die angegebenen Trigger-Ausgangsleitungen und Parameter gesetzt wurden.
Format:	CTO? [{<TrigOutID> <CTOPam>}]
Argumente:	<TrigOutID>: ist eine digitale Ausgangsleitung des Controllers; siehe CTO. <CTOPam>: Parameter-ID; siehe CTO. Werden alle Argumente weggelassen, enthält die Antwort die Werte für alle Parameter und alle Ausgangsleitungen.
Antwort:	{<TrigOutID> <CTOPam>"="<Value> LF} Für <Value> siehe CTO.

DEC (Set Closed-Loop Deceleration)

Beschreibung:	Setzt die Abbremsung für die angegebenen Achsen. DEC kann verändert werden, während die Achse sich

bewegt.

Format: DEC {<AxisID> <Deceleration>}

Argumente: <AxisID> ist eine Achse des Controllers.

<Deceleration> ist der Wert der Abbremsung in physikalischen Einheiten pro s^2.

Antwort: Keine

Fehlersuche: Unzulässige Achsenkennungen

Hinweise: Die DEC-Einstellung wird nur wirksam, wenn sich die angegebene Achse im geregelten Betrieb befindet (Servomodus EIN).

Der kleinstmögliche Wert für <Deceleration> ist 0.

DEC ändert den Wert des Parameters ***Closed Loop Deceleration (Phys. Unit/s^2)*** (ID 0xC) im flüchtigen Speicher des C-863.12. Der Parameterwert kann mit WPA (S. 209) als Standard gespeichert werden, Details siehe "Anpassen von Einstellungen" (S. 235).

Der maximale Wert, der mit dem Befehl DEC gesetzt werden kann, wird durch den Parameter ***Maximum Closed-Loop Deceleration (Phys. Unit/s^2)*** (ID 0x4B) angegeben.

DEC? (Get Closed-Loop Deceleration)

Beschreibung: Fragt den mit DEC (S. 144) gesetzten Wert der Abbremsung ab.

Werden alle Argumente weggelassen, wird der mit DEC gesetzte Wert aller Achsen abgefragt.

Format: DEC? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers.

Antwort: {<AxisID>"="<float> LF}

wobei

<float> der mit DEC gesetzte Wert der Abbremsung ist, in physikalischen Einheiten pro s^2.

DEL (Delay the Command Interpreter)

Beschreibung:	Verzögert um <uint> Millisekunden.
Format:	DEL <uint>
Argumente:	<uint> ist der Verzögerungswert in Millisekunden.
Antwort:	Keine
Hinweise:	DEL kann nur in Makros verwendet werden. Verwechseln Sie nicht MAC DEL (löscht Makros) mit DEL (verzögert). Weitere Informationen finden Sie in der Beschreibung des Befehls MAC (S. 174) und im Abschnitt "Controllermakros" (S. 100).

DFH (Define Home Position)

Beschreibung:	Definiert die Nullposition für die angegebene Achse neu, indem der Positionswert an der aktuellen Position auf null gesetzt wird. Werden alle Argumente weggelassen, definiert DHF die Nullposition aller Achsen neu.
Format:	DFH [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers
Antwort:	keine
Fehlersuche:	Unzulässige Achsenkennung
Hinweise:	DFH setzt die aktuelle Position der Achse auf null und speichert den Positionswert, der beim Befehlsaufruf gültig war, als Offset im flüchtigen Speicher. Durch Addition dieses Offsets zur Antwort werden die Ausgabewerte folgender Befehle an die neue Nullposition angepasst: ▪ POS? (S. 185) (Abfrage der aktuellen Position) ▪ TMN? (S. 201) (Abfrage der kleinsten kommandierbaren Position) ▪ TMX? (S. 202) (Abfrage der größten kommandierbaren Position) DFH ändert **nicht** die Werte der Parameter für die Definition von Stellweg und Verfahrbereichsgrenzen (S. 31). Der Offset wird in folgenden Fällen auf null zurückgesetzt: ▪ Beim Einschalten und Neustart des C-863.12: für alle Achsen

- Bei der Referenzierung: für die betroffene Achse

Beispiel:

Senden: `MOV 1 9.87`
Senden: `POS? 1`
Empfangen: `1=9.8700005`
Senden: `DFH? 1`
Empfangen: `1=0.0000000`
Senden: `TMN? 1`
Empfangen: `1=0.0000000`
Senden: `TMX? 1`
Empfangen: `1=14.9999982`
Hinweis: Achse 1 wird zur absoluten Position 9,87 mm bewegt. Anschließend werden die aktuelle Achsenposition (mit POS?), der aktuelle Offsetwert (mit DFH?) sowie die kleinste und größte kommandierbare Position (mit TMN? und TMX?) abgefragt.
Senden: `DFH 1`
Senden: `POS? 1`
Empfangen: `1=0.0000000`
Senden: `DFH? 1`
Empfangen: `1=9.8700005`
Senden: `TMN? 1`
Empfangen: `1=-9.8700005`
Senden: `TMX? 1`
Empfangen: `1=5.1299978`
Hinweis: Die Achse hat sich nicht bewegt. Mit DFH wurde die aktuelle Achsenposition als neue Nullposition festgelegt. Der Offsetwert für Achse 1 beträgt deshalb nun 9,87 mm. Die Werte für die kleinste und größte kommandierbare Position wurden durch Addition des Offsets an die neue Nullposition angepasst.

DFH? (Get Home Position Definition)

Beschreibung: Fragt den Positionswert ab, der für die angegebene Achse aktuell als Offset zum Verschieben der Nullposition verwendet wird.

Werden alle Argumente weggelassen, wird der Positionswert aller Achsen abgefragt.

Format: DFH? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers

Antwort: {<AxisID>"="<PositionOffset> LF}

wobei

<PositionOffset> ist die Achsenposition, die zum Zeitpunkt der Verarbeitung des letzten DFH-Befehls gültig war. Dieser Positionswert wird intern als Offset für die Berechnung der aktuellen Achsenposition verwendet.

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Die Achsenposition, die zum Zeitpunkt der Verarbeitung des letzten DFH-Befehls gültig war, ist als Offset im flüchtigen Speicher vorhanden. Der Offset wird in folgenden Fällen auf null zurückgesetzt:

- Beim Einschalten und Neustart des C-863.12: für alle Achsen
- Bei der Referenzierung: für die betroffene Achse

Siehe DFH für ein Beispiel.

DIO (Set Digital Output Lines)

Beschreibung: Schaltet die angegebene(n) digitale(n) Ausgangsleitung(en) in den angegebenen Status.

Verwenden Sie TIO? (S. 201), um die Anzahl installierter digitaler I/O-Leitungen abzufragen.

Format: DIO {<DIOID> <OutputOn>}

Argumente: <DIOID> ist eine digitale Ausgangsleitung des Controllers; weitere Angaben siehe unten.

<OutputOn> ist der Status der digitalen Ausgangsleitung; nähere Angaben siehe unten.

Antwort: Keine

Hinweise: Mit dem Befehl DIO können die digitalen Ausgangsleitungen 1 bis 4, die sich auf der Buchse **I/O** (S. 270) befinden, aktiviert/deaktiviert werden. Mit dem C-863.12 können Sie pro Befehl DIO sowohl eine einzelne Leitung setzen als auch alle Leitungen auf einmal.

Die für die Leitungen zu verwendenden Kennungen <DIOID> sind 1 bis 4. Mit der Kennung 0 werden alle Leitungen gemäß einem durch <OutputOn> angegebenen Bitmuster gesetzt.

Wenn <OutputOn>=1 wird die Leitung auf HIGH/ON gesetzt, wenn <OutputOn>=0 wird sie auf LOW/OFF

gesetzt.

Verwenden Sie DIO nicht für Ausgangsleitungen, bei denen die Triggerausgabe mit TRO (S. 203) aktiviert ist.

DIO? (Get Digital Input Lines)

Beschreibung:	Fragt den Status der angegebenen digitalen Eingangsleitungen ab. Verwenden Sie TIO? (S. 201), um die Anzahl verfügbarer digitaler I/O-Leitungen abzufragen.
Format:	DIO? [{<DIOID>}]
Argumente:	<DIOID> ist die Kennung der digitalen Eingangsleitung; nähere Angaben siehe unten.
Antwort:	{<DIOID>"="<InputOn> LF} wobei <InputOn> den Status der digitalen Eingangsleitung angibt; nähere Angaben siehe unten.
Hinweise:	Mit dem Befehl DIO? können die digitalen Eingangsleitungen 1 bis 4, die sich auf der Buchse **I/O** (S. 270) befinden, direkt gelesen werden. Die für die Leitungen zu verwendenden Kennungen <DIOID> sind 1 bis 4. Wenn die Kennung weggelassen wird oder 0 ist, werden alle Leitungen abgefragt. Wenn <InputOn>=0, ist das digitale Eingangssignal LOW/OFF, wenn <InputOn>=1, ist das digitale Eingangssignal HIGH/ON. Wenn <DIOID> 0 ist, ist <InputOn> ein Bitmuster, das den Status aller Leitungen im Hexadezimalformat angibt.

DRC (Set Data Recorder Configuration)

Beschreibung:	Bestimmt für die angegebene Datenrekordertabelle die zu verwendende Datenquelle und die aufzunehmende Datenart (Aufzeichnungsoption).
Format:	DRC {<RecTableID> <Source> <RecOption>}
Argumente:	<RecTableID> ist eine Datenrekordertabelle des Controllers, siehe unten.

<Source> ist die ID der Datenquelle, zum Beispiel einer Achse oder eines Kanals des Controllers. Die erforderliche Quelle ist von der ausgewählten Aufzeichnungsoption abhängig.

<RecOption> bezeichnet die aufzuzeichnende Datenart (Aufzeichnungsoption).

Für Details siehe die nachfolgende Liste der verfügbaren Aufzeichnungsoptionen und der entsprechenden Datenquellen

Antwort: Keine

Hinweise: Der C-863.12 hat 4 Datenrekordertabellen mit 1024 Punkten pro Tabelle.

Mit HDR? (S. 161) erhalten Sie eine Liste aller verfügbaren Aufzeichnungs- und Triggeroptionen sowie zusätzliche Informationen über die Datenaufzeichnung. Die Anzahl der verfügbaren Datenrekordertabellen kann mit TNR? (S. 202) gelesen werden.

Nähere Informationen siehe "Datenrekorder" (S. 76).

Verfügbare Aufzeichnungs-optionen mit den entsprechenden Datenquellen:

- 0=Nothing is recorded

Datenquelle ist die Achse:

- 1=Commanded position of axis
- 2=Actual position of axis
- 3=Position error of axis
- 70=Commanded velocity of axis
- 71=Commanded acceleration of axis
- 73=Motor output of axis (dimensionsloser Stellwert)
- 74=Kp of axis
- 75=Ki of axis
- 76=Kd of axis
- 77=Kv of axis
- 80=Signal status register of axis
- 90=active parameterset (nur relevant, wenn der Controller mehrere Gruppen von Regelparametern unterstützt)

Datenquelle ist ein analoger Eingang:

- 81=Analog input (channel = 1 - 4)

Datenquelle ist ein Motorausgang (siehe Pinbelegung (S. 269)):

- 100=Current Phase A [mA]

- 101=Current Phase B [mA]

Hinweis: Die analogen Eingänge für die Aufzeichnungsoption 81 können die Leitungen Input 1 bis Input 4 der Buchse **I/O** (S. 270) sein. Verwenden Sie für diese Datenquellen die Kennungen 1 bis 4.

Die Datenquellen-Kennungen 5 und 6 bezeichnen die Eingänge für die Achse und Taste des Joystick-Geräts:

5 = Achse 1 des Joystick-Geräts

6 = Taste 1 des Joystick-Geräts

DRC? (Get Data Recorder Configuration)

Beschreibung:	Fragt die Einstellungen für die aufzuzeichnenden Daten ab.
Format:	DRC? [{<RecTableID>}]
Argumente:	<RecTableID>: ist eine Datenrekordertabelle des Controllers; wird die Angabe weggelassen, enthält die Antwort die Einstellungen für alle Tabellen.
Antwort:	Die aktuellen DRC-Einstellungen: {<RecTableID>"="<Source> <RecOption> LF} wobei <Source>: die Datenquelle ist, zum Beispiel eine Achse oder ein Kanal des Controllers. Der Quelltyp ist von der Aufzeichnungsoption abhängig. <RecOption>: bezeichnet die aufzuzeichnende Datenart (Aufzeichnungsoption). Mit HDR? (S. 161) können die verfügbaren Aufzeichnungsoptionen abgefragt werden.

DRL? (Get Number of Recorded Points)

Beschreibung:	Fragt die Anzahl der in der letzten Aufzeichnung enthaltenen Punkte ab.
Format:	DRL? [{<RecTableID>}]
Argumente:	<RecTableID> ist eine Datenrekordertabelle des Controllers
Antwort:	{<RecTableID>"="<uint> LF} wobei

<uint> die Anzahl der in der letzten Aufzeichnung enthaltenen Punkte ist.

Hinweise: Die Anzahl der Punkte wird für die Datenrekordertabelle auf Null zurückgesetzt, wenn ihre Konfiguration mit DRC (S. 149) geändert wird.

DRR? (Get Recorded Data Values)

Beschreibung: Fragt die zuletzt aufgezeichneten Daten ab.

In Abhängigkeit von der Anzahl der zu lesenden Punkte kann das Abfragen einige Zeit in Anspruch nehmen!

Es ist möglich, die Daten zu lesen, während die Aufzeichnung noch läuft.

Format: DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]

Argumente: <StartPoint> ist der erste in der Datenrekordertabelle zu lesende Punkt, beginnt mit Index 1.

<NumberOfPoints> bezeichnet die Anzahl der je Tabelle zu lesenden Punkte.

<RecTableID> ist eine Datenrekordertabelle des Controllers.

Antwort: Die aufgezeichneten Daten im GCS-Array-Format siehe separates Handbuch für GCS Array, SM146E, und untenstehendes Beispiel.

Hinweise: Wenn <RecTableID> weggelassen wird, werden die Daten von allen Tabellen gelesen, deren Aufzeichnungsoption von Null verschieden ist.

Mit HDR? (S. 161) erhalten Sie eine Liste aller verfügbaren Aufzeichnungs- und Triggeroptionen sowie zusätzliche Informationen über die Datenaufzeichnung.

Weitere Informationen siehe die Beschreibung des Befehls DRC (S. 149) sowie "Datenrekorder" (S. 76).

Beispiel:

```
rtr?
10
drr? 1 20
# REM C-863.12
#
# VERSION = 1
# TYPE = 1
# SEPARATOR = 32
```

```
# DIM = 2
# SAMPLE_TIME = 0.000500
# NDATA = 20
#
# NAME0 = Actual Position of Axis
AXIS:1
# NAME1 = Position Error of Axis  AXIS:1
#
# END_HEADER
5.00000 0.00000
4.99998 0.00002
5.00000 0.00000
5.00000 0.00000
5.00000 0.00000
5.00000 0.00000
5.00000 0.00000
4.99998 0.00002
5.00000 0.00000
4.99998 0.00002
5.00000 0.00000
5.00000 0.00000
5.00000 0.00000
5.00000 0.00000
4.99998 0.00002
5.00000 0.00000
4.99998 0.00002
4.99998 0.00002
5.00000 0.00002
4.99998 0.00004
```

DRT (Set Data Recorder Trigger Source)

Beschreibung:	Definiert eine Trigger-Quelle für die angegebene Datenrekordertabelle.
Format:	DRT <RecTableID> <TriggerSource> <Value>
Argumente:	<RecTableID> ist eine Datenrekordertabelle des Controllers. Nähere Angaben siehe unten. <TriggerSource> ID der Trigger-Quelle, Liste verfügbarer Optionen siehe unten. <Value> ist abhängig von der Trigger-Quelle, kann ein

	Dummy sein; siehe unten.
Antwort:	Keine
Hinweise:	Zurzeit ist nur 0 für <RecTableID>gültig; dies bedeutet, dass die angegebene Triggerquelle für alle Datenrekordertabellen gesetzt wird, deren Aufzeichnungsption von Null verschieden ist. Unabhängig von der eingestellten Triggeroption wird die Datenaufzeichnung immer ausgelöst, wenn eine Sprungantwortmessung mit STE (S. 197) durchgeführt wird. Mit HDR? (S. 161) erhalten Sie eine Liste aller verfügbaren Aufzeichnungs- und Triggeroptionen sowie zusätzliche Informationen über die Datenaufzeichnung. Weitere Informationen siehe die Beschreibung des Befehls DRC (S. 149) sowie "Datenrekorder" (S. 76).
Verfügbare Triggeroptionen:	0 = default setting Datenaufzeichnung wird mit STE ausgelöst; <Value> muss ein Dummy sein. 1 = any command changing target position z. B. MVR (S. 183), MOV (S. 181); <Value> muss ein Dummy sein. 2 = next command setzt Trigger nach Ausführung zurück; <Value> muss ein Dummy sein. 6 = any command changing target position, reset trigger after execution z. B. MVR, MOV; setzt Trigger nach Ausführung zurück; <Value> muss ein Dummy sein.

DRT? (Get Data Recorder Trigger Source)

Beschreibung:	Fragt die Triggerquelle für die Datenrekordertabellen ab.
Format:	DRT? [{<RecTableID>}]
Argumente:	<RecTableID> ist eine Datenrekordertabelle des Controllers.
Antwort:	{<RecTableID>"="<TriggerSource> <Value> LF} wobei

<TriggerSource> die ID der Triggerquelle ist.

<Value> ist abhängig von der Triggerquelle.

Weitere Informationen finden Sie in der Beschreibung des Befehls DRT (S. 153).

Hinweise: Da alle Datenrekordertabellen des C-863.12 dieselbe Triggerquelle haben, wird die Antwort auf DRT? als einzelne Zeile wie folgt angegeben

0=<TriggerSource> <Value>

ERR? (Get Error Number)

Beschreibung: Fragt den Fehlercode <int> des zuletzt aufgetretenen Fehlers ab und setzt den Fehler auf 0 zurück.

Es wird nur der letzte Fehler zwischengespeichert. Deshalb sollten Sie ERR? nach jedem Befehl aufrufen.

Eine Auflistung der Fehlercodes und ihrer Beschreibungen ist unter "Fehlercodes" (S. 210) zu finden.

Format: ERR?

Argumente: Keine

Antwort: Der Fehlercode des zuletzt aufgetretenen Fehlers (Integer).

Fehlersuche: Kommunikationsstörung

Hinweise: Bei gleichzeitigem Zugriff mehrerer Instanzen auf den Controller bekommt nur die erste Instanz, die den Befehl ERR? sendet, den Fehlercode geliefert. Da der Fehlercode durch die Abfrage auf 0 zurückgesetzt wird, ist der Fehler für jede weitere abfragende Instanz nicht sichtbar.

- Wenn möglich, greifen Sie immer nur mit einer Instanz auf den Controller zu.
- Wenn der Controller bei fehlerhaftem Systemverhalten keinen Fehlercode sendet, prüfen Sie, ob der Fehlercode durch ein Makro oder Skript oder durch PC-Software (z. B. PIMikroMove®) regelmäßig im Hintergrund abgefragt wird.

Wenn die Ursache eines Fehlers weiterhin besteht, wird der entsprechende Fehlercode sofort nach der Abfrage mit ERR? wieder neu gesetzt.

FED (Find Edge)

Beschreibung:	Bewegt die angegebene Achse zu einer angegebenen Signalflanke. FED setzt keinen bestimmten Positionswert an der ausgewählten Flanke (im Gegensatz zu den Befehlen FNL (S. 157), FPL (S. 158) und FRF (S. 159) zur Referenzierung), d. h. die Achse ist nach Verwendung von FED nicht "referenziert". Enthält der Befehl mehrere Achsen, werden sie synchron bewegt.
Format:	FED {<AxisID> <EdgeID> <Param>}
Argumente:	<AxisID> ist eine Achse des Controllers. <EdgeID> ist der Flankentyp, zu dem sich die Achse bewegen soll. Nähere Angaben zu vorhandenen Flankentypen siehe unten. <Param> ist von der ausgewählten Flanke abhängig und bestimmt sie näher. Nähere Angaben siehe unten.
Antwort:	Keine
Fehlersuche:	Unzulässige Achsenkennung; Endschalter und/oder Referenzschalter sind deaktiviert (siehe unten); SVO? (S. 199) antwortet mit dem Wert 0.
Hinweise:	Der Servomodus muss mit SVO (S. 198) für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb). Anhand von Parametern (ID 0x14 für Referenzschalter; ID 0x32 für Endschalter) ermittelt die Firmware des C-863.12 das Vorhandensein oder Fehlen von Referenz- und Endschaltern. Gemäß den Werten dieser Parameter aktiviert oder deaktiviert der C-863.12 FED-Bewegungen zu den entsprechenden Signalflanken. Passen Sie die Parameterwerte mit SPA (S. 193) oder SEP (S. 190) Ihrer Hardware entsprechend an. Weitere Informationen finden Sie in der Parameterübersicht (S. 244). Sie können die digitalen Eingangsleitungen anstelle der Schalter als Quellen der Schaltersignale für FED verwenden. Weitere Informationen siehe "Digitale Eingangssignale" (S. 86). FED kann verwendet werden, um den physikalischen Stellweg einer neuen Mechanik zu messen und so die Werte für die entsprechenden Parameter zu ermitteln: Den

	Abstand vom negativen zum positiven Endschalter, den Abstand zwischen dem negativen Endschalter und dem Referenzschalter (Parameter ID 0x17) und den Abstand zwischen Referenzschalter und positivem Endschalter (Parameter ID 0x2F). Weitere Informationen siehe "Stellweg und Verfahrbereichsgrenzen" (S. 31). Die Bewegung kann durch #24 (S. 132), STP (S. 198) und HLT (S. 162) gestoppt werden. Bewegungsbefehle wie FED sind nicht zulässig, wenn der Joystick für die Achse aktiv ist. Weitere Informationen siehe "Joystick-Steuerung" (S. 92).
Vorhandene Flankenarten und Parameter:	Die folgenden Flankenarten und deren Parametereinstellungen sind vorhanden: 1 = negativer Endschalter, <Param> muss 0 sein 2 = positiver Endschalter, <Param> muss 0 sein 3 = Referenzschalter, <Param> muss 0 sein

FNL (Fast Reference Move To Negative Limit)

Beschreibung:	Startet eine Referenzierungsfahrt Bewegt die angegebene Achse zur negativen physikalischen Grenze ihres Stellwegs und setzt die aktuelle Position auf einen definierten Wert. Nähere Angaben siehe unten. Enthält der Befehl mehrere Achsen, werden sie synchron bewegt.
Format:	FNL [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers; wenn die Angabe weggelassen wird, sind alle Achsen betroffen.
Antwort:	Keine
Fehlersuche:	Unzulässige Achsenkennung
Hinweise:	Der Servomodus muss mit SVO (S. 198) für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb). Wenn die Referenzierungsfahrt erfolgreich war, ist anschließend eine absolute Bewegung im geregelten Betrieb möglich.

Die negative physikalische Grenze des Stellwegs wird durch den negativen Endschalter des Positionierers verkörpert. Die Differenz der Werte der Parameter 0x16 und 0x17 wird als aktuelle Position gesetzt, wenn die Achse am negativen Endschalter ist.

Sie können ein digitales Eingangssignal anstelle des negativen Endschalters als Quelle des negativen Endschaltersignals für FNL verwenden. Weitere Informationen siehe "Digitale Eingangssignale" (S. 86).

Die Bewegung kann durch #24 (S. 132), STP (S. 198) und HLT (S. 162) gestoppt werden.

Verwenden Sie FRF? (S. 160), um zu prüfen, ob die Referenzierungsfahrt erfolgreich war.

Um die beste Wiederholgenauigkeit zu erreichen, führen Sie die Referenzierung immer auf die gleiche Weise durch.

Wenn Verfahrbereichsgrenzen (Parameter 0x15 und 0x30) für die Verringerung des Stellwegs verwendet werden, können die Endschalter nicht für Referenzierungsfahrten verwendet werden.

Weitere Informationen siehe "Referenzierung" (S. 35) und "Stellweg und Verfahrbereichsgrenzen" (S. 31).

FPL (Fast Reference Move To Positive Limit)

Beschreibung:	Startet eine Referenzierungsfahrt Bewegt die angegebene Achse zur positiven physikalischen Grenze ihres Stellwegs und setzt die aktuelle Position auf einen definierten Wert. Nähere Angaben siehe unten. Enthält der Befehl mehrere Achsen, werden sie synchron bewegt.
Format:	FPL [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers; wenn die Angabe weggelassen wird, sind alle Achsen betroffen.
Antwort:	Keine
Fehlersuche:	Unzulässige Achsenkennung
Hinweise:	Der Servomodus muss mit SVO (S. 198) für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb).

Wenn die Referenzierungsfahrt erfolgreich war, ist anschließend eine absolute Bewegung im geregelten Betrieb möglich.

Die positive physikalische Grenze des Stellwegs wird durch den positiven Endschalter des Positionierers verkörpert. Die Summe der Werte der Parameter 0x16 und 0x2F wird als aktuelle Position gesetzt, wenn die Achse am positiven Endschalter ist.

Sie können ein digitales Eingangssignal anstelle des positiven Endschalters als Quelle des positiven Endschaltersignals für FPL verwenden. Weitere Informationen siehe "Digitale Eingangssignale" (S. 86).

Die Bewegung kann durch #24 (S. 132), STP (S. 198) und HLT (S. 162) gestoppt werden.

Verwenden Sie FRF? (S. 160), um zu prüfen, ob die Referenzierungsfahrt erfolgreich war.

Um die beste Wiederholgenauigkeit zu erreichen, führen Sie die Referenzierung immer auf die gleiche Weise durch.

Wenn Verfahrbereichsgrenzen (Parameter 0x15 und 0x30) für die Verringerung des Stellwegs verwendet werden, können die Endschalter nicht für Referenzierungsfahrten verwendet werden.

Weitere Informationen siehe "Referenzierung" (S. 35) und "Stellweg und Verfahrbereichsgrenzen" (S. 31).

FRF (Fast Reference Move To Reference Switch)

Beschreibung:	Startet eine Referenzierungsfahrt. Bewegt die angegebene Achse zum Referenzschalter und setzt die aktuelle Position auf einen definierten Wert. Nähere Angaben siehe unten. Enthält der Befehl mehrere Achsen, werden sie gleichzeitig gestartet.
Format:	FRF [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers; wenn die Angabe weggelassen wird, sind alle Achsen betroffen.
Antwort:	Keine
Fehlersuche:	Unzulässige Achsenkennung

Hinweise: Der Servomodus muss mit SVO (S. 198) für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb).

Wenn die Referenzierungsfahrt erfolgreich war, ist anschließend eine absolute Bewegung im geregelten Betrieb möglich.

Der Wert des Parameters 0x16 wird als die aktuelle Position gesetzt, wenn die Achse am Referenzschalter ist.

Sie können ein digitales Eingangssignal anstelle des Referenzschalters als Quelle des Referenzsignals für den Befehl FRF verwenden. Weitere Informationen siehe "Digitale Eingangssignale" (S. 86).

Die Bewegung kann durch #24 (S. 132), STP (S. 198) und HLT (S. 162) gestoppt werden.

Verwenden Sie FRF? (S. 160), um zu prüfen, ob die Referenzierungsfahrt erfolgreich war.

Verwenden Sie FNL (S. 157) oder FPL (S. 158) anstelle von FRF (S. 159), um eine Referenzierungsfahrt für eine Achse auszuführen, die keinen Referenzschalter, sondern Endschalter hat.

Um die beste Wiederholgenauigkeit zu erreichen, führen Sie die Referenzierung immer auf die gleiche Weise durch. Der Befehl FRF fährt den Referenzschalter immer von der gleichen Seite an, unabhängig davon, wo sich die Achse beim Befehlsaufruf befindet.

Weitere Informationen siehe "Stellweg und Verfahrbereichsgrenzen" (S. 31).

FRF? (Get Referencing Result)

Beschreibung: Fragt ab, ob die angegebene Achse referenziert ist oder nicht.

Format: FRF? [{<AxisID>}]

Argumente: <AxisID>: ist eine Achse des Controllers.

Antwort: {<AxisID>"="<uint> LF}

wobei

<uint> angibt, ob die Achse erfolgreich referenziert wurde (=1) oder nicht (=0).

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Eine Achse gilt als "referenziert", wenn der aktuelle Positionswert auf eine bekannte Position gesetzt ist. Dies ist der Fall, wenn eine Referenzierungsfahrt erfolgreich durchgeführt wurde mit FNL (S. 157), FPL (S. 158) bzw. FRF (S. 159) oder wenn die Position direkt mit POS (S. 184) eingestellt wurde (abhängig von der mit RON (S. 186) ausgewählten Referenzierungsmethode).

GOH (Go To Home Position)

Beschreibung: Bewegt die angegebene Achse zur Nullposition.

GOH [{<AxisID>}]
ist identisch mit
MOV {<AxisID> 0}

Die Bewegung kann durch #24 (S. 132), STP (S. 198) und HLT (S. 162) gestoppt werden.

Format: GOH [{<AxisID>}]

Argumente: <AxisID>: ist eine Achse des Controllers; wenn die Angabe weggelassen wird, sind alle Achsen betroffen.

Antwort: Keine

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Der Servomodus muss für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb).

HDR? (Get All Data Recorder Options)

Beschreibung: Zeigt einen Hilfetext an, der alle verfügbaren Informationen zur Datenaufzeichnung enthält (Aufzeichnungsoptionen und Triggeroptionen, Information über zusätzliche Parameter und Befehle für die Datenaufzeichnung).

Format: HDR?

Argumente: Keine

Antwort #RecordOptions
{<RecOption>"="<DescriptionString>[of <Channel>]}

#TriggerOptions
[{<TriggerOption>"="<DescriptionString>}]

#Parameters to be set with SPA
[{<ParameterID>"="<DescriptionString>}]

#Additional information
[{<Command description>"("<Command>")"}]

#Sources for Record Options
[{<RecOption>"="<Source>}]

end of help

Hinweis: TriggerOptions = 0 (default) heißt, dass die Aufzeichnung durch den Befehl STE (S. 197) ausgelöst wird.

HLP? (Get List Of Available Commands)

Beschreibung: Zeigt einen Hilfetext an, der alle verfügbaren Befehle enthält.

Format: HLP?

Argumente: Keine

Antwort: Liste der verfügbaren Befehle

Fehlersuche: Kommunikationsstörung

HLT (Halt Motion Smoothly)

Beschreibung: Stoppt die Bewegung der angegebenen Achsen sanft. Nähere Angaben siehe Hinweise unten.

Fehlercode 10 wird gesetzt.

#24 (S. 132) und STP (S. 198) stoppen die aktuelle Bewegung hingegen so schnell wie für den Controller möglich, ohne Berücksichtigung von maximaler Geschwindigkeit und Beschleunigung.

Format: HLT [{<AxisID>}]

Argumente: <AxisID>: ist eine Achse des Controllers; wenn die Angabe weggelassen wird, werden alle Achsen angehalten.

Antwort: Keine

Fehlersuche: Unzulässige Achsenkennung

Hinweise: HLT stoppt die Bewegung mit angegebener Systemabbremsung im Hinblick auf die Systemträgheit.

HLT stoppt jede Bewegung, die durch Bewegungsbefehle (z. B. MOV (S. 181), MVR (S. 183), GOH (S. 161), STE (S.

197)), Befehle zur Referenzierung (FNL (S. 157), FPL (S. 158), FRF (S. 159)) und Makros (MAC (S. 174)) verursacht wird.

Nachdem die Achsen gestoppt wurden, werden ihre Zielpositionen auf ihre aktuellen Positionen gesetzt.

HPA? (Get List Of Available Parameters)

Beschreibung:	Antwortet mit einem Hilfe-String, der alle verfügbaren Parameter mit Kurzbeschreibungen enthält. Weitere Informationen siehe "Parameterübersicht" (S. 244).
Format:	HPA?
Argumente:	Keine
Antwort	{<PamID>"="<string> LF} wobei <PamID> die ID eines Parameters im Hexadezimalformat ist <string> ein String ist, der den entsprechenden Parameter beschreibt. Der String hat folgendes Format: <CmdLevel>TAB<MaxItem>TAB<DataType>TAB<FunctionGroupDescription>TAB<ParameterDescription>[{TAB<PossibleValue>"="<ValueDescription>}] wobei <CmdLevel> die Befehlsebene ist, die Schreibzugriff auf den Parameterwert erlaubt. <MaxItem> ist die maximale Anzahl von Elementen des gleichen Typs, die von dem Parameter betroffen sind. Bei C-863.12 ist ein "Element" eine Achse oder das gesamte System. <DataType> ist der Datentyp des Parameterwertes, er kann INT, FLOAT oder CHAR sein. <FunctionGroupDescription> ist der Name der Funktionsgruppe, zu der der Parameter gehört. <ParameterDescription> ist der Name des Parameters.

<PossibleValue> ist ein Wert aus dem zulässigen Datenbereich.

<ValueDescription> ist die Bedeutung des entsprechenden Wertes.

Die mit HPA? aufgelisteten Parameter können anhand der folgenden Befehle geändert und/oder gespeichert werden:

SPA (S. 193) beeinflusst die Parametereinstellungen im flüchtigen Speicher (RAM).

WPA (S. 209) kopiert Parametereinstellungen vom flüchtigen in den permanenten Speicher.

SEP (S. 190) schreibt die Parametereinstellungen direkt in den permanenten Speicher (ohne die Einstellungen im flüchtigen Speicher zu ändern).

RPA (S. 187) setzt den flüchtigen Speicher auf die Werte aus dem permanenten Speicher zurück.

HPV? (Get Parameter Value Description)

Beschreibung:	Antwortet mit einem Hilfe-String, der mögliche Parameterwerte enthält. Wenn Sie stattdessen HPA? verwenden, erhalten Sie einen Hilfe-String, der alle verfügbaren Parameter mit Kurzbeschreibungen enthält.
Format:	HPV?
Argumente:	Keine
Antwort:	<string> hat folgendes Format:

“#Possible parameter values are:
{<PamID> <ItemID> “=” <ListType>
[{TAB <PossibleValue> “=” <ValueDescription>}] }
#CCL levels are:
{<PamID> <ItemID> “=”<CmdLevel> }
#HPA_Category enabled
end of help”

wobei

<PamID> die ID eines Parameters im Hexadezimalformat ist

<ItemID> ein Element (Achse, Kanal, ganzes System) des

Controllers ist; bei item=0 gilt die Beschreibung für alle Elemente

<ListType> bestimmt, wie die möglichen im String aufgeführten Parameterwerte zu interpretieren sind:
0 = Parameter gilt nicht für dieses Element
1 = Aufzählung
2 = min./max.

<PossibleValue> ist ein Wert aus dem zulässigen Datenbereich

<ValueDescription> ist die Bedeutung des entsprechenden Wertes

Einige Parameter sind für bestimmte Elemente schreibgeschützt (durch eine Befehlsebene > 1). Diese Parameter werden unterhalb der Zeile „#CCL levels are“ aufgeführt.

<CmdLevel> ist die Befehlsebene, die Schreibzugriff auf den Parameterwert erlaubt.

Die Zeile "#HPA_Category enabled" wird von der PC-Software für Anzeigezwecke ausgewertet.

Hinweise: Bei C-863.12 werden die Angaben

```
#Possible parameter values are:
```

und

```
#CCL levels are:
```

in der Antwort auf HPV? weggelassen, weil alle erforderlichen Informationen bereits in der Antwort auf HPA? enthalten sind.

JAS? (Query Joystick Axis Status)

Beschreibung: Fragt den aktuellen Status der angegebenen Achse des angegebenen Joystick-Gerätes ab, das an den Controller angeschlossen ist.

Format: JAS? [{<JoystickID> <JoystickAxis>}]

Argumente: <JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten.

<JoystickAxis> ist eine der Achsen des Joystick-Gerätes; nähere Angaben siehe unten.

Antwort: {<JoystickID> <JoystickAxis>"="<Amplitude>}

wobei

<Amplitude> der Faktor ist, der gerade auf die aktuell gültige Geschwindigkeitseinstellung der angesteuerten Bewegungsachse angewendet wird; er entspricht der aktuellen Auslenkung der Joystick-Achse. Nähere Angaben siehe unten.

Hinweise: An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Der C-863.12 unterstützt eine Achse des Joystick-Geräts, die Kennung der Joystick-Achse ist 1. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17).

Der Faktor <Amplitude> wird auf die mit VEL (S. 206) gesetzte Geschwindigkeit angewendet, der Bereich reicht von -1.0 bis 1.0. Beispiele: Bei einem Faktor von 0 ist die Joystick-Achse in Mittelstellung; bei Faktor -0.7 ist die Joystick-Achse um ca. 2/3 in negative Richtung ausgelenkt, vorausgesetzt, dass eine lineare Lookup-Tabelle aktuell gültig ist (ein Beispiel finden Sie in der Beschreibung des Befehls JLT (S. 168)).

JAX (Set Axis Controlled By Joystick)

Beschreibung: Bestimmt die Achse, die durch einen an den Controller angeschlossenen Joystick gesteuert wird.

Jede Achse des Controllers kann nur durch eine Joystick-Achse gesteuert werden.

Format: JAX <JoystickID> <JoystickAxis> <AxisID>

Argumente: <JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten.

<JoystickAxis> ist eine der Achsen des Joystick-Gerätes; nähere Angaben siehe unten.

<AxisID> ist eine Achse des Controllers.

Antwort: Keine

Hinweise: An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Der C-863.12 unterstützt eine Achse des Joystick-Geräts, die Kennung der Joystick-Achse ist 1. Weitere

Informationen siehe "Kommandierbare Elemente" (S. 17).

JAX? (Get Axis Controlled By Joystick)

Beschreibung:	Fragt die Achse ab, die durch einen an den Controller angeschlossenen Joystick gesteuert wird.
Format:	JAX? [{<JoystickID> <JoystickAxis>}]
Argumente:	<JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten. <JoystickAxis> ist eine der Achsen des Joystick-Gerätes; nähere Angaben siehe unten.
Antwort:	{<JoystickID> <JoystickAxis>"="{<AxisID> }LF} wobei <AxisID> eine Achse des Controllers ist.
Hinweise:	An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Der C-863.12 unterstützt eine Achse des Joystick-Geräts, die Kennung der Joystick-Achse ist 1. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17).

JBS? (Query Joystick Button Status)

Beschreibung:	Fragt den aktuellen Status der angegebenen Taste des angegebenen Joystick-Gerätes ab, das an den Controller angeschlossen ist.
Format:	JBS? [{<JoystickID> <JoystickButton>}]
Argumente:	<JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten. <JoystickButton> ist eine der Tasten des Joystick-Gerätes; nähere Angaben siehe unten.
Antwort:	{<JoystickID> <JoystickButton> "="<State>} wobei <State> angibt, ob die Joystick-Taste gedrückt ist; 0 = nicht gedrückt, 1 = gedrückt.
Hinweise:	An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Der C-863.12 unterstützt eine Taste des Joystick-Geräts, die Kennung der Joystick-Taste ist 1. Weitere

Informationen siehe "Kommandierbare Elemente" (S. 17).

JDT (Set Joystick Default Lookup Table)

Beschreibung:	Setzt den Typ der Lookup-Tabelle für die angegebene Achse des angegebenen Joystick-Gerätes, das an den Controller angeschlossen ist. Der aktuell gültige Inhalt der Lookup-Tabelle für die angegebene Joystick-Achse wird durch die mit JDT getroffene Auswahl überschrieben.
Format:	JDT {<JoystickID> <JoystickAxis> <uint>}
Argumente:	<JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten. <JoystickAxis> ist eine der Achsen des Joystick-Gerätes; nähere Angaben siehe unten. <uint> definiert den zu verwendenden Typ des Lookup-Tabellen-Profils; nähere Angaben siehe unten.
Antwort:	Keine
Hinweise:	An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Der C-863.12 unterstützt eine Achse des Joystick-Geräts, die Kennung der Joystick-Achse ist 1. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17). **Beachten Sie, dass die Anzahl von Schreibzyklen im permanenten Speicher begrenzt ist. Ändern Sie den Typ der Lookup-Tabelle nur, wenn dies notwendig ist.**
Vorhandene Lookup-Tabellen:	Der C-863.12 stellt folgende Typen der Lookup-Tabellen-Profile zur Verfügung: 1 = linear (Standard) 2 = parabolisch

JLT (Fill Joystick Lookup Table)

Beschreibung:	Füllt die Lookup-Tabelle für die angegebene Achse des angegebenen Joystick-Gerätes, das an den Controller angeschlossen ist. Die Amplituden der Joystick-Achsen (d. h. deren Auslenkungen) werden auf die aktuell gültigen Geschwindigkeitseinstellungen der Controllerachsen

abgebildet. Es gibt für jede Joystick-Achse eine Lookup-Tabelle, die diese Zuordnung festlegt. Diese Tabelle kann mit JLT geschrieben werden, oder es kann ein Standardtabellenprofil, das vom Controller zur Verfügung gestellt wird, mit dem Befehl JDT (S. 168) geladen werden.

Jede Lookup-Tabelle besteht aus 256 Punkten. Der erste Punkt entspricht standardmäßig der maximalen Auslenkung der Joystick-Achse in negativer Richtung, der 256. Punkt entspricht der maximalen Auslenkung in positiver Richtung.

Format: JLT <JoystickID> <JoystickAxis> <Addr> <floatn>

Argumente: <JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten.

<JoystickAxis> ist eine der Achsen des Joystick-Gerätes; nähere Angaben siehe unten.

<Addr> ist der Index eines Punktes in der Lookup-Tabelle, beginnt mit 1.

<floatn> ist der Wert von Punkt n.

Antwort: Keine

Hinweise: An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Der C-863.12 unterstützt eine Achse des Joystick-Geräts, die Kennung der Joystick-Achse ist 1. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17).

Die Werte <floatn> sind Faktoren, die während der Joysticksteuerung auf die mit VEL (S. 206) gesetzte Geschwindigkeit angewendet werden, der Bereich reicht von -1.0000 bis 1.0000.

Die Werte <floatn> werden automatisch im permanenten Speicher des C-863.12 abgelegt.

Beispiel: In der aktuellen Lookup-Tabelle hat Punkt 1 den Wert -1, d. h. die angesteuerte Achse bewegt sich mit voller Geschwindigkeit in negativer Richtung bei maximaler negativer Auslenkung des Joysticks. Die Punkte 124 bis 133 haben den Wert 0, d. h. in Mittelstellung des Joysticks und in einem kleinen Bereich um den Mittelpunkt herum ist die Geschwindigkeit 0 und die angesteuerte Achse bewegt sich nicht. Punkt 236 hat den Wert 0.8369, d. h. wenn die Auslenkung der Joystick-Achse ca. 2/3 in positiver Richtung beträgt, bewegt sich die angesteuerte Achse in positiver Richtung mit ca. 4/5 der vollen Geschwindigkeit. Punkt 256

hat den Wert 1, d. h. die angesteuerte Achse bewegt sich mit voller Geschwindigkeit in positiver Richtung bei maximaler positiver Auslenkung des Joysticks.

Beachten Sie, dass die Anzahl von Schreibzyklen im permanenten Speicher begrenzt ist. Schreiben Sie Werte in die Lookup-Tabelle nur, wenn dies notwendig ist.

JLT? (Get Joystick Lookup Table Values)

Beschreibung: Fragt die aktuell gültigen Werte der Lookup-Tabelle ab.

Format: JLT? [<StartPoint> <NumberOfPoints> [{<JoystickID> <JoystickAxis>}]]

Argumente: <StartPoint> ist der Startpunkt in der Lookup-Tabelle, beginnt mit 1.

<NumberOfPoints> bezeichnet die Anzahl der je Joystick-Achse zu lesenden Punkte; die maximale Anzahl ist 256.

<JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten.

<JoystickAxis> ist eine der Achsen des Joystick-Gerätes; nähere Angaben siehe unten.

Antwort: Die Inhalte der Lookup-Tabellen im GCS-Array-Format, siehe separates Handbuch für GCS Array, SM 146E, und untenstehendes Beispiel.

Hinweise: Bei C-863.12 müssen <JoystickID> und <JoystickAxis> im Befehl JLT? weggelassen werden, während <StartPoint> und <NumberOfPoints> immer erforderlich sind.

Die Werte <floatn> in der Lookup-Tabelle sind Faktoren, die bei der Joystick-Steuerung auf die mit VEL (S. 206) gesetzte Geschwindigkeit angewendet werden, der Bereich reicht von -1.0000 bis 1.0000.

Beispiel:

```
jlt? 1 20
# TYPE = 1
#
# SEPARATOR = 32
# DIM = 1
# NDATA = 20
# NAME0 = Joysticktable 1
# END HEADER
-1.0000
-0.9922
```

```
-0.9834
-0.9756
-0.9678
-0.9590
-0.9512
-0.9434
-0.9346
-0.9268
-0.9189
-0.9102
-0.9023
-0.8945
-0.8857
-0.8779
-0.8701
-0.8613
-0.8535
-0.8457
```

JON (Set Joystick Activation Status)

Beschreibung:	Aktiviert oder deaktiviert ein Joystick-Gerät, das an den Controller angeschlossen ist.
Format:	JON {<JoystickID> <uint>}
Argumente:	<JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten. <uint> 1 aktiviert das Joystick-Gerät, 0 deaktiviert das Joystick-Gerät.
Antwort:	Keine
Hinweise:	An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Weitere Informationen siehe "Analogen Joystick anschließen" (S. 49). Bevor ein Joystick-Gerät mit JON aktiviert werden kann, müssen seine Achsen den Controllerachsen mit JAX (S. 166) zugewiesen werden. Wenn ein an den C-863.12 angeschlossener Joystick mit dem Befehl JON aktiviert ist, steuert dieser Joystick die Achsengeschwindigkeit (die vom Profilgenerator ausgegebene "kommandierte Geschwindigkeit"). Während der Joystick-Steuerung wird die Zielposition auf

die Verfahrbereichsgrenze eingestellt, die durch den Parameter 0x15 bzw. 0x30 vorgegeben ist. Einzelheiten zu den Parametern finden Sie in "Stellweg und Verfahrbereichsgrenzen" (S. 31). Beim Deaktivieren des Joysticks wird die Zielposition auf die aktuelle Position der joystickgesteuerten Achse eingestellt.

Bewegungsbefehle wie MOV (S. 181) sind nicht zulässig, wenn ein Joystick für die Achse aktiv ist. Weitere Informationen siehe "Joystick-Steuerung" (S. 92).

JON? (Get Joystick Activation Status)

Beschreibung:	Fragt den Status der Aktivierung des angegebenen Joystick-Geräts ab, das an den Controller angeschlossen ist.
Format:	JON? [{<JoystickID>}]
Argumente:	<JoystickID> ist ein Joystick-Gerät, das an den Controller angeschlossen ist; nähere Angaben siehe unten.
Antwort:	{<JoystickID>"="<uint>} wobei <uint> ist der Status der Joystickaktivierung: 1 = Joystick-Gerät aktiv, 0 = Joystick-Gerät deaktiviert.
Hinweise:	An die Buchse **Joystick** (S. 271) des C-863.12 kann ein Joystick-Gerät angeschlossen werden, die Kennung ist 1. Weitere Informationen siehe "Analogen Joystick anschließen" (S. 49).

JRC (Jump Relatively Depending On Condition)

Beschreibung:	Springt relativ, abhängig von einer angegebenen Bedingung des folgenden Typs: ein angegebener Wert wird mit einem abgefragten Wert gemäß einer angegebenen Regel verglichen. Kann nur in Makros verwendet werden.
Format:	JRC <Jump> <CMD?> <OP> <Value>
Argumente:	<Jump> ist die Größe des relativen Sprungs. -1 bedeutet, dass der Makroausführungszeiger zurück zur vorherigen Zeile springt, 0 bedeutet, dass der Befehl erneut ausgeführt wird, was dem Verhalten von WAC (S. 208) entspricht. 1 springt zur nächsten Zeile, was den Befehl überflüssig macht, 2 überspringt den nächsten Befehl. Es sind nur Sprünge innerhalb des aktuellen Makros zulässig.

<CMD?> ist ein Abfragebefehl in seiner üblichen Schreibweise. Die Antwort muss ein einzelner Wert sein (und nicht mehr). Beispiel siehe unten.

<OP> ist der zu verwendende Operator. Folgende Operatoren sind möglich:
= <= < > >= !=
Wichtig: Vor und nach dem Operator muss ein Leerzeichen stehen!

<Value> ist der Wert, der mit der Antwort auf <CMD?> zu vergleichen ist.

Antwort: Keine

Fehlersuche: Korrektes Sprungziel prüfen

Beispiel: Mit dem folgenden Makro können Sie die Bewegung der Achse 1 durch einen Stoppknopf anhalten, der an einem digitalen Eingang angeschlossen ist. Die Überprüfung des Stoppknopfes erfolgt so lange, bis die Achse die Zielposition erreicht hat (Abfrage ONT?). Wenn der Stoppknopf gedrückt wird, solange die Zielposition noch nicht erreicht ist: Das Ergebnis der Abfrage POS? 1 wird in die Variable TARGET kopiert. Diese Variable wird dann als zweites Argument für den Befehl MOV verwendet. Somit bleibt der Positionierer, wo er gerade war. Zur Bereinigung wird TARGET mit dem Befehl VAR als leer definiert, wodurch die Variable gelöscht wird.

Schreiben Sie das Makro "stop":

```
MAC BEG stop
MOV 1 20
JRC 2 DIO? 1 = 1
JRC -1 ONT? 1 = 0
CPY TARGET POS? 1
MOV 1 ${TARGET}
VAR TARGET
MAC END
```

LIM? (Indicate Limit Switches)

Beschreibung: Fragt ab, ob die Achsen Endschalter haben.

Format: LIM? [{<AxisID>}]

Argumente: <AxisID>: ist eine Achse des Controllers.

Antwort: {<AxisID>"="<uint> LF}

wobei

<uint> angibt, ob die Achse Endschalter hat (=1) oder nicht (=0).

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Anhand eines Parameters (ID 0x32) ermittelt die Firmware des C-863.12 das Vorhandensein oder Fehlen von Endschaltern. Entsprechend dem Wert dieses Parameters aktiviert oder deaktiviert der C-863.12 das Stoppen der Bewegung an den Endschaltern und Referenzierungsfahrten unter Verwendung von FNL (S. 157) oder FPL (S. 158).

Passen Sie den Parameterwert Ihrer Hardware entsprechend mit SPA (S. 193) oder SEP (S. 190) an. Weitere Informationen finden Sie in "Endschaltererkennung" (S. 30).

Sie können die digitalen Eingangsleitungen anstelle der Endschalter als Quelle der negativen oder positiven Endschaltersignale verwenden. Weitere Informationen finden Sie in "Digitale Eingangssignale" (S. 86).

MAC (Call Macro Function)

Beschreibung: Ruft eine Makrofunktion auf. Erlaubt das Aufzeichnen, Löschen und Ausführen von Makros auf dem Controller.

Format: MAC <keyword> {<parameter>}

insbesondere:

MAC BEG <macroname>
MAC DEF <macroname>
MAC DEF?
MAC DEL <macroname>
MAC END
MAC ERR?
MAC NSTART <macroname> <uint> [<String1> [<String2>]]
MAC START <macroname> [<String1> [<String2>]]

Argumente: <keyword> legt fest, welche Makrofunktion aufgerufen wird. Die folgenden Schlüsselworte und Parameter werden verwendet:

MAC BEG <macroname>
Startet die Aufzeichnung eines Makros mit dem Namen *macroname* auf dem Controller; darf nicht in einem Makro verwendet werden; die Befehle, die folgen, bilden das

Makro. Die Aufzeichnung wird mit MAC END beendet. Beachten Sie, dass fehlerhafter Makroinhalt nicht durch Senden des Befehls ERR? ermittelt werden kann.

MAC END

Stoppt die Makroaufzeichnung (kann nicht Bestandteil eines Makros werden).

MAC ERR?

Meldet den letzten Fehler, der während der Ausführung eines Makros auftrat.

Antwort: <macroname> <uint1>”=”<uint2> <”<”CMD”>”>

wobei <macroname> der Name des Makros ist, <uint1> ist die Zeile im Makro, <uint2> ist der Fehlercode, und <”<”CMD”>”> ist der fehlerhafte Befehl, der an den Parser gesendet wurde.

MAC DEF <macroname>

Legt das angegebene Makro als Startup-Makro fest. Dieses Makro wird automatisch nach dem nächsten Einschalten oder Neustart des Controllers ausgeführt. Wird <macroname> weggelassen, wird die Auswahl des aktuellen Startup-Makros annulliert.

MAC DEF?

Fragt das Startup-Makro ab.
Antwort: <macroname>
Ist kein Startup-Makro festgelegt, ist die Antwort ein leerer String mit dem Abschlusszeichen.

MAC DEL <macroname>

Löscht das angegebene Makro.

MAC NSTART <macroname> <uint> [<String1> [<String2>]]

Wiederholt das angegebene Makro <uint> Mal. Eine neue Ausführung wird gestartet, wenn die letzte beendet ist.

<String1> und <String2> sind optionale Argumente, die die Werte der lokalen Variablen 1 und 2 angeben, die in dem angegebenen Makro verwendet werden. <String1> und <String2> können direkt oder über Variablenwerte angegeben werden. Die Makroausführung schlägt fehlt, wenn das Makro lokale Variablen enthält, aber <String1> und <String2> im Befehl MAC NSTART weggelassen werden. Nähere Angaben siehe "Variablen" (S. 123).

MAC START <macroname> [<String1> [<String2>]]

Startet eine Ausführung des angegebenen Makros. <String1> und <String2> haben die gleiche Funktion wie bei MAC NSTART.

Antwort: Keine

Fehlersuche: Makroaufzeichnung ist aktiv (Schlüsselworte BEG, DEL) oder inaktiv (END)

Makro enthält unzulässigerweise den Befehl MAC

Hinweise: Während einer Makroaufzeichnung ist keine Makroausführung erlaubt.

Wird ein Makro für einen Controller aufgezeichnet, dessen Adresse von 1 abweichend ist, muss die Empfängeradresse Bestandteil jeder Befehlszeile sein, sie wird jedoch nicht zum Bestandteil des Makroinhalts. PIMikroMove® sendet die Empfängeradresse bei der Makroaufzeichnung automatisch mit, so dass sie dort nicht eingegeben werden darf. Weitere Informationen siehe "Mit Makros arbeiten" (S. 102) und "Empfänger- und Senderadresse" (S. 122).

Bei der Aufzeichnung von Makros auf der Registerkarte ***Controller macros*** in PIMikroMove® müssen die Befehle `MAC BEG` und `MAC END` weggelassen werden.

Ein Makro kann von einem Makro mit demselben Namen überschrieben werden.

Makros können lokale und globale Variablen beinhalten. Weitere Informationen siehe "Variablen" (S. 123).

Ein laufendes Makro sendet keine Antworten an eine Schnittstelle.

In Abhängigkeit vom Wert des Parameters 0x72 (***Ignore Macro Error?***) bestehen die folgenden Möglichkeiten, wenn ein Fehler durch ein laufendes Makro verursacht wird:

0 = Die Makroausführung wird gestoppt (Standard).
1 = Der Fehler wird ignoriert und die Makroausführung wird fortgesetzt.

Unabhängig von der Parametereinstellung meldet MAC ERR? stets den letzten Fehler, der während einer Makroausführung aufgetreten ist.

Die folgenden vom C-863.12 zur Verfügung gestellten Befehle können nur in Makros verwendet werden:
DEL (S. 145), JRC (S. 172), MEX (S. 179) und WAC (S. 208).

Ein Makro kann ein anderes Makro starten. Die Höchstzahl der Verschachtelungsebenen beträgt 5. Ein Makro kann sich selbst aufrufen, um eine Endlosschleife zu bilden.

Von der Befehlszeile können sämtliche Befehle gesendet

werden, während ein Makro läuft. Der Makroinhalt und Bewegungsbefehle, die von der Befehlszeile empfangen werden, können sich gegenseitig überschreiben.

Die Makroausführung kann durch #24 (S. 132) und STP (S. 198) gestoppt werden.

Zeitgleiche Ausführung mehrerer Makros ist nicht möglich. Es kann jeweils nur ein Makro ausgeführt werden.

Ein laufendes Makro kann nicht gelöscht werden.

Mit #8 (S. 132) können Sie abfragen, ob ein Makro aktuell auf dem Controller ausgeführt wird.

Beachten Sie, dass die Anzahl von Schreibzyklen im permanenten Speicher begrenzt ist. Zeichnen Sie Makros nur auf, wenn dies notwendig ist.

MAC? (List Macros)

Beschreibung:	Listet Makros oder den Inhalt eines angegebenen Makros auf.
Format:	MAC? [<macroname>]
Argumente	<macroname>: Name des Makros, dessen Inhalt aufgelistet werden soll; wird diese Angabe weggelassen, werden die Namen aller gespeicherten Makros aufgelistet.
Antwort:	<string> Wenn <macroname> angegeben wurde, ist <string> der Inhalt dieses Makros. Wenn <macroname> weggelassen wurde, ist <string> eine Liste der Namen aller gespeicherten Makros.
Fehlersuche:	Makro <macroname> nicht gefunden

MAN? (Get Help String For Command)

Beschreibung:	Zeigt einen detaillierten Hilfetext zu einzelnen Befehlen an.
Format:	MAN? <CMD>
Argumente:	<CMD> ist das Befehlskürzel des Befehls, für den der Hilfetext angezeigt werden soll (siehe unten).
Antwort:	Ein String, der den Befehl beschreibt.

Hinweise: Ein detaillierter Hilfetext kann für folgende GCS-Befehle angezeigt werden:
CTO, CTO?, WPA

Beispiel: Senden: `MAN? CTO`
Empfangen:

```
CTO {<TrigOutID> <CTOPam> <Value>} Set
Configuration Of Trigger Output
#AvailableCTOparameters
<CTOPam> <Description>
<CTOPam> (configuration parameter):
1 TriggerStep
2 Axis
3 TriggerMode
7 Polarity
8 StartThreshold
9 StopThreshold
10 TriggerPosition
#AvailableTriggerModes
<Value> <Description>
0 PositionDistance
2 OnTarget
5 MotionError
6 InMotion
7 Position+Offset
8 SinglePosition
#AvailablePolarities
<Value> <Description>
0 ActiveLow
1 ActiveHigh
end of help
```

MAT (Calculate And Save To Variable)

Beschreibung: Führt eine mathematische Operation oder Bitoperation aus und speichert das Ergebnis als Variable (S. 123).

Die Variable ist nur im flüchtigen Speicher (RAM) vorhanden.

Format: MAT <Variable> "=" <FLOAT1> <OP> <FLOAT2>

Argumente: <Variable> ist der Name der Variable, in der das Ergebnis gespeichert werden soll.

<FLOAT1> und <FLOAT2> sind die Größen, aus denen das

Ergebnis berechnet werden soll. Sie können direkt angegeben werden oder über den Wert einer Variablen.

<OP> ist der zu verwendende Operator: Folgende Operatoren sind möglich:

<OP>	Operation	Typ
+	Addition	Mathematische Operation
-	Subtraktion	Mathematische Operation
*	Multiplikation	Mathematische Operation
AND	UND	Bitoperation
OR	ODER	Bitoperation
XOR	XOR	Bitoperation

Wichtig: Vor und nach dem "=" und dem Operator muss jeweils ein Leerzeichen stehen!

Antwort: Keine

Hinweise: Die Verwendung von MAT zum Setzen lokaler Variablen ist nur in Makros möglich.

Beispiel 1: Senden: `MAT TARGET = ${POS} * 2.0`

Die Variable `TARGET` erhält den 2,0-fachen Wert der Variable `POS`.

Beispiel 2: Senden: `MAT TARGET = 2 * 0x10`

Senden: `VAR? TARGET`

Empfangen: `TARGET=32`

Hinweis: Die Größen, aus denen das Ergebnis berechnet werden soll, können im Hexadezimal- oder Dezimalformat geschrieben werden. Das Ergebnis wird immer im Dezimalformat ausgegeben.

Beispiel 3: Senden: `MAT INVERT = 0x45 XOR 0xFF`

Senden: `VAR? INVERT`

Empfangen: `INVERT=186`

Hinweis: Die Bitoperation XOR mit dem Wert 0xFF entspricht einer Inversion des Werts 0x45. Das Ergebnis wird im Dezimalformat ausgegeben.

MEX (Stop Macro Execution Due To Condition)

Beschreibung: Stoppt Makroausführung aufgrund einer angegebenen Bedingung des folgenden Typs: Ein angegebener Wert wird mit einem abgefragten Wert gemäß einer angegebenen Regel verglichen.

Kann nur in Makros verwendet werden.

Wenn der Makro-Interpreter auf diesen Befehl zugreift, wird die Bedingung geprüft. Ist sie erfüllt, wird das aktuelle Makro gestoppt, andernfalls wird die Makroausführung in der nächsten Zeile fortgesetzt. Sollte die Bedingung später erfüllt sein, wird der Interpreter sie ignorieren.

Siehe auch den Befehl WAC (S. 208).

Format: MEX <CMD?> <OP> <Value>

Argumente: <CMD?> ist ein Abfragebefehl in seiner üblichen Schreibweise. Die Antwort muss ein einzelner Wert sein (und nicht mehr). Beispiel siehe unten.

<OP> ist der zu verwendende Operator. Folgende Operatoren sind möglich:
= <= < > >= !=
Wichtig: Vor und nach dem Operator muss ein Leerzeichen stehen!

<Value> ist der Wert, der mit der Antwort auf <CMD?> zu vergleichen ist.

Antwort: Keine

Beispiel: Senden: `MAC START LOOP`

Hinweis:
Makro LOOP beinhaltet Folgendes:

```
MAC START KEY1
MAC START KEY2
MEX DIO? 4 = 1
MAC START LOOP
```

Makro KEY1 beinhaltet Folgendes:

```
MEX DIO? 4 = 1
MEX DIO? 1 = 0
MVR 1 1.0
DEL 100
```

Makro KEY2 beinhaltet Folgendes:

```
MEX DIO? 4 = 1
MEX DIO? 2 = 0
MVR 1 -1.0
DEL 100
```

Makro LOOP bildet eine Endlosschleife, indem es

permanent KEY1, KEY2 und sich selbst aufruft.

KEY1 prüft den Status des digitalen Eingangskanals 1 (befindet sich auf der Buchse **I/O** (S. 270)). Wird er nicht gesetzt (0), wird das Makro gestoppt, andernfalls bewegt das Makro Achse 1 um 1.0 in positive Richtung (relative Bewegung).

KEY2 prüft den Status des digitalen Eingangskanals 2 und bewegt Achse 1 entsprechend in negative Richtung.

Durch das Verbinden der digitalen Eingangsleitungen 1, 2 und 4 mit Drucktasten, z. B. mit der Pushbutton-Box C-170.PB, ist es möglich, die interaktive Ansteuerung einer Achse ohne jegliche Softwareunterstützung zu realisieren. Die Verzögerung (DEL 100) ist erforderlich, um die Erzeugung mehrfacher Befehle MVR zu verhindern, wenn die Drucktaste kurz gedrückt wird.

Kanal 4 wird als globaler Ausstieg verwendet. Da MEX nur die Ausführung des aktuellen Makros stoppt, muss es auch in dem aufrufenden Makro enthalten sein, das andernfalls fortgesetzt würde.

MOV (Set Target Position)

Beschreibung:	Setzt eine absolute Zielposition für die angegebene Achse.
Format:	MOV {<AxisID> <Position>}
Argumente:	<AxisID> ist eine Achse des Controllers. <Position> ist die absolute Zielposition in physikalischen Einheiten.
Antwort:	Keine
Hinweise:	Der Servomodus muss bei Verwendung dieses Befehls eingeschaltet sein (geregelter Betrieb). Die Zielposition muss sich innerhalb der Verfahrbereichsgrenzen befinden. Verwenden Sie TMN? (S. 201) und TMX? (S. 202), um die aktuell gültigen Verfahrbereichsgrenzen abzufragen. Die Bewegung kann durch #24 (S. 132), STP (S. 198) und HLT (S. 162) gestoppt werden. Während einer Bewegung setzt ein neuer Bewegungsbefehl das Ziel auf einen neuen Wert; der alte

Wert wird eventuell niemals erreicht. Dies gilt auch für Makros: Bewegungsbefehle können von der Befehlszeile gesendet werden, wenn ein Makro ausgeführt wird. Der Makroinhalt und Bewegungsbefehle, die von der Befehlszeile empfangen werden, können sich gegenseitig überschreiben.

Bewegungsbefehle wie MOV sind nicht zulässig, wenn ein Joystick für die Achse aktiv ist. Weitere Informationen siehe "Joystick-Steuerung" (S. 92).

Beispiel 1: Senden: `MOV 1 10`
Hinweis: Achse 1 bewegt sich nach 10 (Zielposition in mm)

Beispiel 2: Senden: `MOV 1 243`
Senden: `ERR?`
Empfangen: `7`
Hinweis: Die Achse bewegt sich nicht. Der Fehlercode "7" in der Antwort auf den Befehl ERR? (S. 155) gibt an, dass die in den Bewegungsbefehlen angegebene Zielposition außerhalb der Grenzwerte ist.

MOV? (Get Target Position)

Beschreibung: Fragt die letzte gültige kommandierte Zielposition ab.

Format: MOV? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers.

Antwort: {<AxisID>"="<float> LF}

wobei

<float> die letzte kommandierte Zielposition in physikalischen Einheiten ist.

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Die Zielposition kann durch Befehle, die Bewegung verursachen, geändert werden (z. B. MOV (S. 181), MVR (S. 183), GOH (S. 161), STE (S. 197)) oder durch den Joystick (bei Deaktivierung eines Joysticks wird für joystickgesteuerte Achsen im geregelten Betrieb die Zielposition auf die aktuelle Position gesetzt).

MOV? fragt die kommandierten Positionen ab. Verwenden Sie POS? (S. 185), um die aktuellen Positionen abzufragen.

MVR (Set Target Relative To Current Position)

Beschreibung: Bewegt die angegebene Achse relativ zur letzten kommandierten Zielposition.

Format: MVR {<AxisID> <Distance>}

Argumente: <AxisID> ist eine Achse des Controllers.

<Distance> gibt die Strecke an, um die sich die Achse bewegen soll; die Summe der Strecke und der letzten kommandierten Zielposition wird als neue Zielposition gesetzt (in physikalischen Einheiten).

Antwort: Keine

Hinweise: Der Servomodus muss bei Verwendung dieses Befehls eingeschaltet sein (geregelter Betrieb).

Die Zielposition muss sich innerhalb der Verfahrbereichsgrenzen befinden. Verwenden Sie TMN? (S. 201) und TMX? (S. 202), um die aktuell gültigen Verfahrbereichsgrenzen abzufragen, und MOV? (S. 182) für die Abfrage des aktuellen Ziels.

Die Bewegung kann durch #24 (S. 132), STP (S. 198) und HLT (S. 162) gestoppt werden.

Während einer Bewegung setzt ein neuer Bewegungsbefehl das Ziel auf einen neuen Wert; der alte Wert wird eventuell niemals erreicht. Dies gilt auch für Makros: Von der Befehlszeile können Bewegungsbefehle gesendet werden, wenn ein Makro läuft. Der Makroinhalt und Bewegungsbefehle, die von der Befehlszeile empfangen werden, können sich gegenseitig überschreiben.

Bewegungsbefehle wie MVR sind nicht zulässig, wenn ein Joystick für die Achse aktiv ist. Weitere Informationen siehe "Joystick-Steuerung" (S. 92).

Beispiel: Senden: `MOV 1 0.5`
Hinweis: Dies ist eine absolute Bewegung.
Senden: `POS? 1`
Empfangen: `1=0.500000`
Senden: `MOV? 1`
Empfangen: `1=0.500000`
Senden: `MVR 1 2`
Hinweis: Dies ist eine relative Bewegung.
Senden: `POS? 1`
Empfangen: `1=2.500000`

Senden: `MVR 1 2000`
Hinweis: Neue Zielposition von Achse 1 würde den Bewegungsbereich überschreiten. Befehl wird ignoriert, d. h. die Zielposition bleibt unverändert und die Achse bewegt sich nicht.
Senden: `MOV? 1`
Empfangen: `1=2.500000`
Senden: `POS? 1`
Empfangen: `1=2.500000`

ONT? (Get On-Target State)

Beschreibung:	Fragt den On-Target-Status der angegebenen Achse ab. Werden alle Argumente weggelassen, wird der Status aller Achsen abgefragt.
Format:	ONT? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers.
Antwort:	{<AxisID>"="<uint> LF} wobei <uint> = "1" wenn die angegebene Achse den Zielwert erreicht hat, anderenfalls "0".
Fehlersuche:	Unzulässige Achsenkennung
Hinweise:	Die Ermittlung des On-Target-Status ist nur im geregelten Betrieb möglich (Servomodus EIN). Der On-Target-Status wird von den Einstellungen für das Einschwingfenster (Parameter 0x36) und die Verzögerungszeit (Parameter 0x3F) beeinflusst. Details siehe "On-Target-Status" (S. 29).

POS (Set Real Position)

Beschreibung:	Setzt die aktuelle Position der Achse (löst keine Bewegung aus).
Format:	POS {<AxisID> <Position>}
Argumente:	<AxisID> ist eine Achse des Controllers. <Position> ist die neue aktuelle Position in physikalischen Einheiten.
Antwort:	Keine

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Das Setzen der aktuellen Position mit POS ist nur möglich, wenn die Referenzierungsmethode "0" ausgewählt ist; siehe RON (S. 186).

Eine Achse wird als "referenziert" bezeichnet, wenn die Position mit POS gesetzt wurde (weitere Informationen siehe "Referenzierung" (S. 35)).

Die kleinsten und größten kommandierbaren Positionen (TMN? (S. 201), TMX? (S. 202)) werden nicht angepasst, wenn eine Position mit POS gesetzt wurde. Dies kann zu Zielpositionen führen, die vom C-863.12 zugelassen sind, aber von der Hardware nicht angefahren werden können. Ebenso sind Zielpositionen möglich, die von der Hardware angefahren werden können, aber vom C-863.12 verweigert werden. Darüber hinaus kann nach der Verwendung von POS die Nullposition außerhalb des physikalischen Stellwegs liegen.

POS? (Get Real Position)

Beschreibung: Fragt die aktuelle Achsenposition ab.

Werden alle Argumente weggelassen, wird die aktuelle Position aller Achsen abgefragt.

Format: POS? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers.

Antwort: {<AxisID>"="<float> LF}

wobei

<float> die aktuelle Achsenposition in physikalischen Einheiten ist.

Fehlersuche: Unzulässige Achsenkennung

RBT (Reboot System)

Beschreibung: Startet das System neu. Der Controller verhält sich wie nach dem Einschalten.

Format: RBT

Argumente: Keine

Antwort: Keine

Hinweise: RBT kann nicht in Makros verwendet werden. Dadurch werden Probleme bei der Ausführung des Startup-Makros vermieden.

RMC? (List Running Macros)

Beschreibung: Listet die aktuell laufenden Makros auf.

Format: RMC?

Argumente: Keine

Antwort: {<macroname> LF}

wobei

<macroname> der Name eines Makros ist, das auf dem Controller gespeichert ist und aktuell ausgeführt wird. Die Antwort ist eine leere Zeile, wenn kein Makro ausgeführt wird.

RON (Set Reference Mode)

Beschreibung: Wählt die Referenzierungsmethode für die angegebenen Achsen.

Format: RON {<AxisID> <ReferenceOn>}

Argumente: <AxisID> ist eine Achse des Controllers.

<ReferenceOn> ist die Referenzierungsmethode. Kann 0 oder 1 sein. 1 ist Standard. Details siehe unten.

Antwort: Keine

Fehlersuche: Unzulässige Achsenkennung

Hinweise: <ReferenceOn> = 0: Der Achse kann ein absoluter Positionswert mit POS (S. 184) zugewiesen werden, oder eine Referenzierungsfahrt kann mit FRF (S. 159), FNL (S. 157) oder FPL (S. 158) gestartet werden. Relative Bewegungen mit MVR sind möglich, auch wenn die Achse noch nicht referenziert wurde.

<ReferenceOn> = 1: Für die Achse muss eine Referenzierungsfahrt mit FRF, FNL oder FPL gestartet werden. Die Verwendung von POS ist nicht zulässig. Bewegungen im geregelten Betrieb sind erst möglich, wenn die Achse referenziert wurde.

Weitere Informationen siehe "Referenzierung" (S. 35) und "Stellweg und Verfahrbereichsgrenzen" (S. 31).

RON? (Get Reference Mode)

Beschreibung:	Fragt die Referenzierungsmethode der angegebenen Achsen ab.
Format:	RON? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers.
Antwort:	{<AxisID>"="<ReferenceOn> LF} wobei <ReferenceOn> die aktuell für die Achse ausgewählte Referenzierungsmethode ist
Fehlersuche:	Unzulässige Achsenkennung
Hinweis:	Weitere Informationen finden Sie in der Beschreibung des Befehls RON (S. 186).

RPA (Reset Volatile Memory Parameters)

Beschreibung:	Setzt den angegebenen Parameter des angegebenen Elements zurück. Der Wert aus dem permanenten Speicher wird in den flüchtigen Speicher geschrieben. Verwandte Befehle: Mit HPA? (S. 163) erhalten Sie eine Liste der verfügbaren Parameter. SPA (S. 193) beeinflusst die Parametereinstellungen im flüchtigen Speicher, WPA (S. 209) schreibt Parametereinstellungen aus dem flüchtigen in den permanenten Speicher und SEP (S. 190) schreibt Parametereinstellungen direkt in den permanenten Speicher (ohne die Einstellungen im flüchtigen Speicher zu ändern). Beispiel siehe SPA.
Format:	RPA [{<ItemID> <PamID>}]
Argumente:	<ItemID> ist das Element, für das ein Parameter zurückzusetzen ist. Nähere Angaben siehe unten. <PamID> ist die Parameterkennung, kann im Hexadezimal- oder Dezimalformat geschrieben werden. Nähere Angaben siehe unten.
Antwort:	Keine
Fehlersuche:	Unzulässige Elementkennung, falsche Parameter-ID

Hinweise:	Mit dem C-863.12 können Sie entweder alle Parameter oder einen einzelnen Parameter mit RPA zurücksetzen.
Verfügbare Element-IDs und Parameter-IDs:	Ein Element ist eine Achse (die Kennung kann mit SAI (S. 189) geändert werden) oder das gesamte System. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17). Gültige Parameter-IDs finden Sie in der Parameterübersicht (S. 244).

RTR (Set Record Table Rate)

Beschreibung:	Setzt die Aufzeichnungsrate des Datenrekorders, d. h. die Anzahl der Zyklen, die für Datenaufzeichnungsvorgänge verwendet werden. Einstellungen größer als 1 ermöglichen es, längere Zeitspannen abzudecken.
Format:	RTR <RecordTableRate>
Argumente:	<RecordTableRate> ist die Aufzeichnungsrate des Datenrekorders, die für die Aufzeichnungsvorgänge zu verwenden ist (Einheit: Anzahl der Zyklen), muss ein ganzzahliger Wert größer als Null sein.
Antwort:	Keine
Hinweise:	Die Dauer der Aufzeichnung kann wie folgt berechnet werden: Aufz.Dauer = Zykluszeit des Regelkreises * RTR Wert *Anzahl der Punkte wobei die Zykluszeit des Regelkreises für den C-863.12 50 µs beträgt die Anzahl der Punkte für den C-863.12 1024 beträgt (Länge der Datenrekordertabelle) Weitere Informationen siehe "Datenrekorder" (S. 76). Die mit RTR gesetzte Aufzeichnungsrate des Datenrekorders wird nur im flüchtigen Speicher (RAM) gespeichert.

RTR? (Get Record Table Rate)

Beschreibung:	Fragt die aktuelle Aufzeichnungsrate für die Datenrekordertabellen ab, d. h. die Anzahl der Zyklen, die

für Datenaufzeichnungsvorgänge verwendet werden.

Format: RTR?

Argumente: Keine

Antwort: <RecordTableRate> ist die Rate, die für die Aufzeichnungsvorgänge verwendet wird (Einheit: Anzahl der Zyklen).

SAI (Set Current Axis Identifiers)

Beschreibung: Setzt die Achsenkennung für die angegebenen Achsen.

Nach dem Setzen der neuen Achsenkennung mit SAI ist sie als <AxisID> in allen achsenbezogenen Befehlen zu verwenden.

Format: SAI {<AxisID> <NewIdentifier>}

Argumente: <AxisID> ist eine Achse des Controllers.

<NewIdentifier> ist die neue für die Achse zu verwendende Kennung, nähere Angaben siehe unten.

Antwort: Keine

Hinweise: Eine Achsenkennung kann aus bis zu 8 Zeichen bestehen. Verwenden Sie TVI? (S. 204), um gültige Zeichen abzufragen.

Die neue Achsenkennung wird nur in den flüchtigen Speicher des C-863.12 übernommen. Eine geänderte Achsenkennung kann mit dem Befehl WPA (S. 209) permanent im C-863.12 gespeichert werden.

SAI? (Get List Of Current Axis Identifiers)

Beschreibung: Fragt die Achsenkennung ab.

Siehe auch "Kommandierbare Elemente" (S. 17).

Format: SAI? [ALL]

Argumente: [ALL] ist optional. Bei Controllern, die Achsen-Deaktivierung zulassen, stellt [ALL] sicher, dass die Antwort auch Achsen enthält, die "deaktiviert" sind.

Antwort: {<AxisID> LF}

<AxisID> ist eine Achse des Controllers.

SEP (Set Non-Volatile Memory Parameters)

Beschreibung: Setzt einen Parameter des angegebenen Elements auf einen anderen Wert im permanenten Speicher, womit er zum neuen Standard wird.

Nach dem Setzen der Parameter mit SEP können Sie RPA (S. 187) verwenden, um sie ohne Neustart des Controllers zu aktivieren (in den flüchtigen Speicher zu schreiben).

Beachten Sie, dass dieser Befehl für die Einstellung hardwarespezifischer Parameter gilt. Falsche Werte können eventuell zu fehlerhaftem Betrieb oder zur Beschädigung Ihrer Hardware führen!

Verwandte Befehle:

HPA? (S. 163) gibt eine Liste verfügbarer Parameter zurück.

SPA (S. 193) schreibt Parametereinstellungen in den flüchtigen Speicher (ohne die Einstellungen im permanenten Speicher zu ändern).

WPA (S. 209) schreibt Parametereinstellungen vom flüchtigen in den permanenten Speicher.

Format: SEP <Pswd> {<ItemID> <PamID> <PamValue>}

Argumente <Pswd> ist das Passwort zum Schreiben in den permanenten Speicher, Standardwert ist "100".

<ItemID> ist das Element, für das ein Parameter im permanenten Speicher zu ändern ist. Nähere Angaben siehe unten.

<PamID> ist die Parameterkennung, kann im Hexadezimal- oder Dezimalformat geschrieben werden. Nähere Angaben siehe unten.

<PamValue> ist der Wert, auf den der angegebene Parameter des angegebenen Elements gesetzt wird.

Antwort: Keine

Fehlersuche: Unzulässige Elementkennung, falsche Parameter-ID, ungültiges Kennwort

Hinweise: **Beachten Sie, dass die Anzahl von Schreibzyklen im permanenten Speicher begrenzt ist. Schreiben Sie Standardeinstellungen nur, wenn dies notwendig ist.**

Beim C-863.12 können Sie nur einen Parameter pro Befehl

	SEP schreiben.
Verfügbare Element-IDs und Parameter-IDs:	Ein Element ist eine Achse (die Kennung kann mit SAI (S. 189) geändert werden) oder das gesamte System. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17). Gültige Parameter-IDs finden Sie in der Parameterübersicht (S. 244).

SEP? (Get Non-Volatile Memory Parameters)

Beschreibung:	Fragt den Wert eines Parameters für ein angegebenes Element aus dem permanenten Speicher ab. Mit HPA? (S. 163) erhalten Sie eine Liste der verfügbaren Parameter und deren Kennungen (IDs).
Format:	SEP? [{<ItemID> <PamID>}]
Argumente:	<ItemID> ist das Element, für das ein Parameterwert im permanenten Speicher abgefragt werden soll. Nähere Angaben siehe unten. <PamID> ist die Parameterkennung, kann im Hexadezimal- oder Dezimalformat geschrieben werden. Nähere Angaben siehe unten.
Antwort:	{<ItemID> <PamID>"="<PamValue> LF} wobei <PamValue> der Wert des angegebenen Parameters für das angegebene Element ist.
Fehlersuche:	Unzulässige Elementkennung, falsche Parameter-ID
Hinweise:	Mit dem C-863.12 können Sie entweder alle Parameter oder einen einzelnen Parameter pro Befehl SEP? abfragen.
Verfügbare Element-IDs und Parameter-IDs:	Ein Element ist eine Achse (die Kennung kann mit SAI (S. 189) geändert werden) oder das gesamte System. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17). Gültige Parameter-IDs finden Sie in der Parameterübersicht (S. 244).

SMO (Set Open-Loop Control Value)

Beschreibung:	Setzt den Stellwert direkt, um die Achse zu bewegen. Profilgenerator (wenn vorhanden), Sensor-Rückmeldung und Regelalgorithmus werden nicht berücksichtigt.

	Der Servomodus muss bei Verwendung dieses Befehls ausgeschaltet sein (ungeregelter Betrieb).
Format:	SMO {<AxisID> <ControlValue>}
Argumente	<AxisID> ist eine Achse des Controllers. <ControlValue> ist der neue Stellwert (dimensionslos). Nähere Angaben siehe unten.
Antwort:	Keine
Fehlersuche:	Unzulässige Achsenkennung Der Servomodus ist für eine der angegebenen Achsen eingeschaltet.
Hinweise:	**HINWEIS: Bei großen Stellwerten kann der Positionierer trotz Endschalterfunktion am mechanischen Anschlag aufprallen. Dies kann die Ausrüstung beschädigen.** Der Stellwert ohne Vorzeichen darf nicht größer als der Wert des Parameters ***Maximum Motor Output*** (0x9) sein. Wird dieser Parameter auf seinen Höchstwert gesetzt (32767), reicht <ControlValue> von -32766 bis 32766 (dimensionslos). Das Vorzeichen des Wertes bestimmt die Bewegungsrichtung. <ControlValue> steuert den PWM-Konverter für die Achse an (siehe Blockdiagramm (S. 16)). Weitere Informationen siehe "Motoransteuerung" (S. 16).
Beispiel:	Senden: `SMO 1 -16000` Hinweis: Der Stellwert beträgt ungefähr die Hälfte des maximalen Stellwerts. Die Achse bewegt sich in negativer Richtung.

SMO? (Get Control Value)

Beschreibung:	Fragt den letzten gültigen Stellwert der angegebenen Achse ab.
Format:	SMO? [{<AxisID>}]
Argumente	<AxisID> ist eine Achse des Controllers.
Antwort:	{<AxisID>"="<float> LF} wobei <float> der letzte gültige Stellwert (dimensionslos) ist.

Nähere Angaben siehe unten.

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Der Stellwert, der durch SMO? zurückgegeben wird, kann das Ergebnis des Regelalgorithmus und weiterer Korrekturen sein bzw. der Wert, der durch einen Befehl SMO (S. 191) im ungeregelten Betrieb gesetzt wird. Siehe das Blockdiagramm (S. 16) für weitere Informationen.

Der Stellwert reicht von -32766 bis 32766 (dimensionslos) und steuert den PWM-Konverter für die Achse an. Das Vorzeichen des Wertes bestimmt die Bewegungsrichtung.

SPA (Set Volatile Memory Parameters)

Beschreibung: Setzt einen Parameter des angegebenen Elements im flüchtigen Speicher (RAM) auf einen bestimmten Wert. Parameteränderungen gehen verloren, wenn der Controller abgeschaltet oder neugestartet wird.

Format: SPA {<ItemID> <PamID> <PamValue>}

Argumente: <ItemID> ist das Element, für das ein Parameter im flüchtigen Speicher geändert wird. Nähere Angaben siehe unten.

<PamID> ist die Parameterkennung, kann im Hexadezimal- oder Dezimalformat geschrieben werden. Nähere Angaben siehe unten.

<PamValue> ist der Wert, auf den der Parameter des angegebenen Elements gesetzt wird.

Antwort: Keine

Parameteränderungen gehen auch verloren, wenn die Parameter mit RPA (S. 187) auf ihre Standardwerte zurückgesetzt werden.

Beachten Sie, dass dieser Befehl für die Einstellung hardwarespezifischer Parameter gilt. Falsche Werte können eventuell zu fehlerhaftem Betrieb oder zur Beschädigung Ihrer Hardware führen!

Verwandte Befehle:

HPA? (S. 163) gibt eine Liste verfügbarer Parameter zurück.

SEP (S. 190) schreibt Parametereinstellungen direkt in den permanenten Speicher (ohne die Einstellungen im flüchtigen Speicher zu ändern).

WPA (S. 209) schreibt Parametereinstellungen vom flüchtigen in den permanenten Speicher.

RPA setzt den flüchtigen Speicher auf den Wert aus dem permanenten Speicher zurück.

Fehlersuche: Unzulässige Elementkennung, falsche Parameter-ID, Wert im unzulässigen Bereich

Hinweise: Mit dem C-863.12 können Sie nur einen Parameter je Befehl SPA schreiben.

Verfügbare Element -IDs und Parameter-IDs: Ein Element ist eine Achse (die Kennung kann mit SAI (S. 189) geändert werden) oder das gesamte System. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17).

Gültige Parameter-IDs sind in der Parameterübersicht (S. 244) angegeben.

Beispiel 1: Senden: `SPA 1 0x411 100`

Hinweis: Setzt den P-Term des Regelalgorithmus für Achse 1 auf 100, die Parameter-ID wird im Hexadezimalformat geschrieben

Senden: `SPA 1 1041 150`

Hinweis: Setzt den P-Term des Regelalgorithmus für Achse 1 auf 150, die Parameter-ID wird im Dezimalformat geschrieben

Beispiel 2: Die P-, I- und D-Parameter des Regelalgorithmus müssen an eine neue Last, die auf die angeschlossene Mechanik einwirkt, angepasst werden.

Senden: `SPA 1 0x411 150`

Hinweis: Der P-Term wird für Achse 1 auf 150 gesetzt. Die Einstellung wird nur im flüchtigen Speicher vorgenommen.

Setzen Sie mit SPA nun die I- und D-Terme im flüchtigen Speicher und prüfen anschließend die Funktion des Systems. Stellt sich die Leistung des geregelten Systems als zufriedenstellend heraus und möchten Sie die Systemkonfiguration als Standard verwenden, speichern Sie die Parametereinstellungen aus dem flüchtigen Speicher im permanenten Speicher.

Senden: `WPA 100`

Hinweis: Siehe die Befehlsbeschreibung für WPA (S. 209) für Details zum Umfang der gespeicherten Einstellungen.

Beispiel 3: Senden: `SEP 100 LEFT 0xA 20`

Hinweis: Die maximale Geschwindigkeit ist auf 20 mm/s für die Achse LEFT zu setzen (Achse wurde mit SAI umbenannt). Die Einstellung wird im permanenten Speicher vorgenommen und ist somit der neue Standard, der aber noch nicht aktiv ist. Um die neuen Einstellungen sofort zu verwenden, müssen Sie sie in den flüchtigen Speicher laden (anderenfalls würden sie aktiv werden, wenn der Controller das nächste Mal angeschaltet oder neugestartet wird).

Senden: `RPA`

Hinweis: Die neue Konfiguration ist nun aktiv.

Senden: `SPA? LEFT 0xA`

Empfangen: `LEFT 0xA=20.00000`

Hinweis: Prüfen Sie die Parametereinstellungen im flüchtigen Speicher.

SPA? (Get Volatile Memory Parameters)

Beschreibung: Fragt den Wert eines Parameters für ein angegebenes Element aus dem flüchtigen Speicher (RAM) ab.

Mit HPA? (S. 163) erhalten Sie eine Liste der verfügbaren Parameter.

Format: SPA? [{<ItemID> <PamID>}]

Argumente: <ItemID> ist das Element, für das ein Parameter im flüchtigen Speicher abgefragt werden soll. Nähere Angaben siehe unten.

<PamID> ist die Parameterkennung, kann im Hexadezimal- oder Dezimalformat geschrieben werden. Nähere Angaben siehe unten.

Antwort: {<ItemID> <PamID>"="<PamValue> LF}

wobei

<PamValue> der Wert des angegebenen Parameters für das angegebene Element ist.

Fehlersuche: Unzulässige Elementkennung, falsche Parameterkennung

Hinweise: Mit dem C-863.12 können Sie entweder alle Parameter oder einen einzelnen Parameter je Befehl SPA? abfragen.

Verfügbare Element -IDs und Parameter-IDs: Ein Element ist eine Achse (die Kennung kann mit SAI (S. 189) geändert werden) oder das gesamte System. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17).

Gültige Parameter-IDs sind in der Parameterübersicht (S. 244) angegeben.

SRG? (Query Status Register Value)

Beschreibung: Gibt Registerwerte für die abgefragten Elemente und Register zurück.

Format: SRG? [{<ItemID> <RegisterID>}]

Argumente: <ItemID> ist das Element, für das ein Register abgefragt werden soll. Nähere Angaben siehe unten.

<RegisterID> ist die ID des angegebenen Registers, verfügbare Register siehe unten.

Antwort: {<ItemID><RegisterID>"="<Value> LF}

wobei

<Value> der Wert des Registers ist, nähere Angaben siehe unten.

Hinweis: Dieser Befehl ist funktionsgleich mit #4 (S. 130), der bevorzugt werden sollte, wenn der Controller zeitaufwändige Aufgaben ausführt.

Mögliche Register-IDs und Antwortwerte: <ItemID> ist eine Achse des Controllers.

<RegisterID> kann 1 sein.

<Value> ist die bit-codierte Antwort und wird als Summe der folgenden einzelnen Codes in Hexadezimalformat zurückgegeben:

Bit	15	14	13	12	11	10	9	8
Beschreibung	On-Target-Status	Führt Referenzierung aus	In Bewegung	Servomodus Ein	-	-	-	Fehlerflag

Bit	7	6	5	4	3	2	1	0
Beschreibung	Digitale Eingangsleitung 4	Digitale Eingangsleitung 3	Digitale Eingangsleitung 2	Digitale Eingangsleitung 1	-	Pos. Endschalter	Referenzschalter	Neg. Endschalter

Beispiel: Senden: `SRG? 1 1`
Empfangen: `1 1=0x9002`
Hinweis: Die Antwort wird im Hexadezimalformat angegeben. Sie besagt: Achse 1 ist an der Zielposition (On-Target-Status = wahr), der Servomodus für diese Achse ist eingeschaltet, es ist kein Fehler aufgetreten, der Status der digitalen Eingangsleitungen 1 bis 4 ist low, und Achse 1 befindet sich auf der positiven Seite des Referenzschalters.

STE (Start Step And Response Measurement)

Beschreibung: Startet einen Sprung und die Aufzeichnung der Sprungantwort für die angegebene Achse.

Die Datenrekorderkonfiguration, d. h. die Zuweisung der Datenquellen und der Aufzeichnungsoptionen zu den Rekordertabellen, kann mit DRC (S. 149) gesetzt werden.

Die aufgezeichneten Daten können mit dem Befehl DRR? (S. 152) gelesen werden.

Format: STE <AxisID> <Amplitude>

Argumente: <AxisID> ist eine Achse des Controllers

<Amplitude> ist die Größe des Sprungs. Nähere Angaben siehe unten.

Antwort: Keine

Fehlersuche: Der Servomodus muss für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb).

Die Zielposition muss sich innerhalb der Verfahrbereichsgrenzen befinden. Verwenden Sie TMN? (S. 201) und TMX? (S. 202), um die aktuell gültigen Verfahrbereichsgrenzen abzufragen und MOV? (S. 182) für die Abfrage des aktuellen Ziels.

Bewegungsbefehle wie STE sind nicht zulässig, wenn der Joystick für die Achse aktiv ist. Weitere Informationen

siehe "Joystick-Steuerung" (S. 92).

Hinweise: Ein "Sprung" besteht aus einer relativen Bewegung mit der angegebenen Amplitude, die relativ zur aktuellen Position ausgeführt wird.

STP (Stop All Axes)

Beschreibung: Stoppt alle Achsen abrupt. Nähere Angaben siehe Hinweise unten.

Setzt den Fehlercode auf 10.

Dieser Befehl ist funktionsgleich mit dem Befehl #24 (S. 132).

Format: STP

Argumente: Keine

Antwort: Keine

Fehlersuche: Kommunikationsstörung

Hinweise: STP stoppt jede Bewegung, die durch Bewegungsbefehle (z. B. MOV (S. 181), MVR (S. 183), GOH (S. 161), STE (S. 197)), Befehle zur Referenzierung (FNL (S. 157), FPL (S. 158), FRF (S. 159)) und Makros (MAC (S. 174)) verursacht wird. Stoppt auch die Makroausführung.

Nachdem die Achsen gestoppt wurden, werden ihre Zielpositionen auf ihre aktuellen Positionen gesetzt.

HLT (S. 162) stoppt im Gegensatz zu STP die Bewegung mit vorgegebener Abbremsung im Hinblick auf die Systemträgheit.

SVO (Set Servo Mode)

Beschreibung: Setzt den Servomodus für die angegebenen Achsen (ungeregelter oder geregelter Betrieb).

Format: SVO {<AxisID> <ServoState>}

Argumente: <AxisID> ist eine Achse des Controllers

<ServoState> kann folgende Werte haben:
0 = Servomodus aus (ungeregelter Betrieb)
1 = Servomodus ein (geregelter Betrieb)

Antwort: Keine

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Beim Wechsel vom ungeregelten Betrieb in den geregelten Betrieb wird die Zielposition auf die aktuelle Position gesetzt, um Sprünge der Mechanik zu vermeiden.

Der aktuelle Zustand des Servomodus bestimmt die anwendbaren Bewegungsbefehle:
Servomodus an: Verwenden Sie die Befehle MOV (S. 181), MVR (S. 183), GOH (S. 161) oder die Joystick-Steuerung (S. 92).
Servomodus aus: Verwenden Sie SMO (S. 191).

Der Servomodus muss eingeschaltet sein, bevor Referenzierungsfahrten mit FRF (S. 159), FNL (S. 157) oder FPL (S. 158) gestartet werden können.

Wenn der Servomodus ausgeschaltet wird, während sich die Achse bewegt, stoppt die Achse.

Mit einem Startup-Makro kann der Controller so konfiguriert werden, dass der Servomodus nach dem Anschalten oder Neustart automatisch eingeschaltet ist. Nähere Angaben siehe "Startup-Makro einrichten" (S. 109).

Wenn die Achse über eine Bremse verfügt, beeinflusst das Setzen des Servomodus mit SVO den Aktivierungszustand der Bremse:

- Einschalten des Servomodus deaktiviert die Bremse.
- Ausschalten des Servomodus aktiviert die Bremse. Wenn der Servomodus ausgeschaltet ist, kann die Bremse mit BRA (S. 136) aktiviert oder deaktiviert werden. Sichern Sie den Positionierer gegen unbeabsichtigte Bewegungen, bevor Sie die Bremse mit BRA deaktivieren!

Beim Auftreten eines Bewegungsfehlers wird der Servomodus ausgeschaltet, und die Bremse wird aktiviert. Mehr Informationen siehe "Bewegungsfehler" (S. 75).

SVO? (Get Servo Mode)

Beschreibung: Fragt den Servomodus für die angegebenen Achsen ab.

Werden keine Argumente angegeben, wird der Servomodus aller Achsen abgefragt.

Format: SVO? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers.

Antwort: {<AxisID>"="<ServoState> LF}

wobei

<ServoState> der aktuelle Servomodus der Achse ist:
0 = Servomodus aus (ungeregelter Betrieb)
1 = Servomodus an (geregelter Betrieb)

Fehlersuche: Unzulässige Achsenkennung

TAC? (Tell Analog Channels)

Beschreibung: Fragt die Anzahl installierter Analogleitungen ab.

Format: TAC?

Argumente: Keine

Antwort: <uint> gibt die Gesamtanzahl der Analogleitungen (Ein- und Ausgänge) an.

Hinweise: Fragt die Anzahl der analogen Eingangsleitungen auf der Buchse **I/O** (S. 270) des C-863.12 ab (Input 1 bis Input 4). Beachten Sie, dass diese Leitungen auch für digitale Eingangssignale verwendet werden können. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17).

TAV? (Get Analog Input Voltage)

Beschreibung: Fragt die Spannung am Analogeingang ab.

Format: TAV? [{<AnalogInputID>}]

Argumente: <AnalogInputID> ist die Kennung des analogen Eingangskanals; nähere Angaben siehe unten.

Antwort: {<AnalogInputID>"="<float> LF}

wobei

<float> die aktuelle Spannung am Analogeingang ist, in Volt

Hinweise: Mit dem Befehl TAV? können die Leitungen Input 1 bis Input 4 auf der Buchse **I/O** (S. 270) des C-863.12 direkt gelesen werden. Die Kennungen der Leitungen sind 1 bis 4. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17).

Die Werte der analogen Eingangsleitungen können mit DRC-Aufzeichnungsoption 81 (S. 149) aufgezeichnet werden.

TCV? (Get Commanded Closed-Loop Velocity)

Beschreibung:	Fragt den aktuellen Wert der Geschwindigkeit ab (vom Profilgenerator berechneter Wert).
Format:	TCV? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers.
Antwort:	{<AxisID>"="<float> LF} wobei <float> der Geschwindigkeitswert in physikalischen Einheiten pro Sekunde ist.

TIO? (Tell Digital I/O Lines)

Beschreibung:	Gibt die Anzahl der installierten digitalen I/O-Leitungen an.
Format:	TIO?
Argumente:	Keine
Antwort:	I=<uint1> O=<uint2> wobei <uint1> die Anzahl der digitalen Eingangsleitungen ist. <uint2> die Anzahl der digitalen Ausgangsleitungen ist.
Hinweise:	Die durch TIO? gemeldeten digitalen Ausgangsleitungen sind Output 1 bis Output 4. Der Status der Leitungen Output 1 bis Output 4 kann durch Verwendung des Befehls DIO (S. 148) gesetzt werden. Darüber hinaus können Sie die Leitungen Output 1 bis Output 4 mit dem Befehl CTO (S. 140) (Triggerkonfiguration) und dem Befehl TRO (S. 203) (Triggeraktivierung/-deaktivierung) programmieren. Die durch TIO? gemeldeten digitalen Eingangsleitungen sind Input 1 bis Input 4. Sie können mit DIO? (S. 149), #4 (S. 130) und SRG? (S. 196) gelesen werden. Alle Leitungen befinden sich auf der Buchse **I/O** (S. 270) des C-863.12.

TMN? (Get Minimum Commandable Position)

Beschreibung:	Fragt die kleinste kommandierbare Position in physikalischen Einheiten ab.

Format:	TMN? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers.
Antwort	{<AxisID>"="<float> LF} wobei <float> die kleinste kommandierbare Position in physikalischen Einheiten ist.
Hinweis:	Die kleinste kommandierbare Position wird durch den Parameter 0x30 bestimmt. Beim Neudefinieren der Nullposition mit dem Befehl DFH (S. 146) wird die kleinste kommandierbare Position automatisch an die neue Nullposition angepasst.

TMX? (Get Maximum Commandable Position)

Beschreibung:	Fragt die größte kommandierbare Position in physikalischen Einheiten ab.
Format:	TMX? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers.
Antwort	{<AxisID>"="<float> LF} wobei <float> die größte kommandierbare Position in physikalischen Einheiten ist.
Hinweis:	Die größte kommandierbare Position wird durch den Parameter 0x15 bestimmt. Beim Neudefinieren der Nullposition mit dem Befehl DFH (S. 146) wird die größte kommandierbare Position automatisch an die neue Nullposition angepasst.

TNR? (Get Number of Record Tables)

Beschreibung:	Fragt die Anzahl der aktuell auf dem Controller verfügbaren Datenrekordertabellen ab.
Format:	TNR?
Argumente:	Keine
Antwort	<uint> ist die Anzahl der Datenrekordertabellen, die aktuell auf dem Controller verfügbar sind.
Hinweise:	Der C-863.12 hat vier Datenrekordertabellen mit 1024 Datenpunkten pro Tabelle.

Weitere Informationen siehe "Datenrekorder" (S. 76).

TRO (Set Trigger Output State)

Beschreibung:	Aktiviert oder deaktiviert die mit CTO (S. 140) gesetzten Bedingungen der Triggerausgabe für die angegebene digitale Ausgangsleitung.
Format:	TRO {<TrigOutID> <TrigMode>}
Argumente:	<TrigOutID> ist eine digitale Ausgangsleitung des Controllers; weitere Angaben siehe unten. <TrigMode> kann folgende Werte annehmen: 0 = Triggerausgabe deaktiviert 1 = Triggerausgabe aktiviert
Antwort:	Keine
Fehlersuche:	Unzulässige Kennung der digitalen Ausgangsleitung
Hinweise:	<TrigOutID> entspricht den digitalen Ausgangsleitungen Output 1 bis Output 4, IDs = 1 bis 4; weitere Informationen siehe "I/O" (S. 270). Verwenden Sie DIO (S. 148) nicht für digitale Ausgangsleitungen, bei denen die Triggerausgabe mit TRO aktiviert ist.

TRO? (Get Trigger Output State)

Beschreibung:	Fragt für die angegebene digitale Ausgangsleitung den Aktivierungsstatus der mit CTO (S. 140) vorgenommenen Triggerausgabe-Konfiguration ab. Werden alle Argumente weggelassen, wird der Status aller digitalen Ausgangsleitungen abgefragt.
Format:	TRO? [{<TrigOutID>}]
Argumente:	<TrigOutID> ist eine digitale Ausgangsleitung des Controllers; weitere Einzelheiten finden Sie in der Beschreibung des Befehls TRO (S. 203).
Antwort:	{<TrigOutID>"="<TrigMode> LF} wobei <TrigMode> der aktuelle Status der digitalen Ausgangsleitung ist: 0 = Triggerausgabe deaktiviert 1 = Triggerausgabe aktiviert

Fehlersuche: Unzulässige Kennung der digitalen Ausgangsleitung

TRS? (Indicate Reference Switch)

Beschreibung: Zeigt an, ob die Achsen einen Referenzschalter mit Richtungserkennung haben.

Format: TRS? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers

Antwort: {<AxisID>"="<uint> LF}

wobei

<uint> angibt, ob die Achse einen richtungserkennenden Referenzschalter hat (=1) oder nicht (=0).

Fehlersuche: Unzulässige Achsenkennung

Hinweise: Anhand eines Parameters (ID 0x14) ermittelt die Firmware des C-863.12 das Vorhandensein oder Fehlen eines Referenzschalters. Gemäß dem Wert dieses Parameters aktiviert oder deaktiviert der C-863.12 Referenzierungsfahrten zum Referenzschalter (Befehl FRF (S. 159)). Passen Sie den Parameterwert entsprechend Ihrer Hardware mit SPA (S. 193) oder SEP (S. 190) an. Weitere Informationen siehe "Referenzschalterkennung" (S. 29).

Sie können eine digitale Eingangsleitung anstelle des Referenzschalters als Quelle des Referenzsignals für den Befehl FRF verwenden. Weitere Informationen siehe "Digitale Eingangssignale" (S. 86).

TVI? (Tell Valid Character Set For Axis Identifiers)

Beschreibung: Gibt einen String mit den für Achsenkennungen verwendbaren Zeichen zurück.

Verwenden Sie SAI (S. 189), um die Achsenkennungen zu ändern und SAI? (S. 189), um die aktuell gültigen Achsenkennungen abzufragen.

Format: TVI?

Argumente: Keine

Antwort: <string> ist eine Liste von Zeichen.

Hinweise: Beim C-863.12 besteht der String aus

1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-_

VAR (Set Variable Value)

Beschreibung: Setzt eine Variable auf einen bestimmten Wert.

Lokale Variablen können mit VAR nur in Makros gesetzt werden. Nähere Angaben zu lokalen und globalen Variablen siehe "Variablen" (S. 123).

Die Variable ist nur im RAM vorhanden.

Format: VAR <Variable> <String>

Argumente: <Variable> ist der Name der Variablen, deren Wert gesetzt werden soll.

<String> ist der Wert, auf den die Variable zu setzen ist. Wird er weggelassen, wird die Variable gelöscht.

Der Wert kann direkt oder über den Wert einer Variablen angegeben werden.

Nähere Angaben zu Konventionen bezüglich Namen und Werten von Variablen siehe "Variablen" (S. 123).

Antwort: Keine

Beispiel: Es ist möglich, den Wert einer Variablen (z. B. TARGET) auf den einer anderen Variablen (z. B. SOURCE) zu setzen:

```
VAR TARGET ${SOURCE}
```

Verwenden Sie geschweifte Klammern, wenn der Name der Variablen länger als ein Zeichen ist:

```
VAR A ONE
VAR VARB TWO
VAR $A 1
VAR ${VARB} 2
VAR $VARB 2 // dies führt zu unerwünschtem Verhalten
VAR?
A=ONE
VARB=TWO
ONE=1
TWO=2 // ${VARB}: wird durch ihren Wert "TWO" ersetzt.
ARB=2 // $VARB: $V wird durch ihren (leeren) Wert
```

ersetzt.

Ein weiteres Beispiel finden Sie in der Beschreibung des Befehls ADD (S. 134).

VAR? (Get Variable Values)

Beschreibung: Gibt Variablenwerte zurück.

Wird VAR? mit CPY (S. 138), JRC (S. 172), MEX (S. 179) oder WAC (S. 208) kombiniert, muss die Antwort auf VAR? ein einzelner Wert sein (und nicht mehr).

Nähere Angaben zu lokalen und globalen Variablen siehe "Variablen" (S. 123).

Format: VAR? [{<Variable>}]

Argumente: <Variable> ist der Name der abzufragenden Variablen. Nähere Angaben zu Namenskonventionen siehe "Variablen" (S. 123).

Wird <Variable> weggelassen, werden alle im RAM vorhandenen globalen Variablen aufgelistet.

Antwort: {<Variable>”=”<String>LF}

wobei

<String> den Wert angibt, auf den die Variable gesetzt ist.

Hinweise: Lokale Variablen können mit VAR? nur abgefragt werden, wenn ein Makro mit lokalen Variablen läuft. Weitere Informationen zu lokalen und globalen Variablen siehe "Variablen" (S. 123).

Beispiel: Beispiel siehe ADD (S. 134).

VEL (Set Closed-Loop Velocity)

Beschreibung: Setzt die Geschwindigkeit für die angegebenen Achsen.

Format: VEL {<AxisID> <Velocity>}

Argumente: <AxisID> ist eine Achse des Controllers.

<Velocity> ist der Geschwindigkeitswert in physikalischen Einheiten pro Sekunde.

Antwort: Keine

Fehlersuche: Unzulässige Achsenkennungen

Hinweise: Die VEL-Einstellung wird nur wirksam, wenn sich die angegebene Achse im geregelten Betrieb befindet (Servomodus ein).

Der kleinstmögliche Wert für <Velocity> ist 0.

Die Geschwindigkeit kann mit VEL verändert werden, während die Achse sich bewegt.

VEL ändert den Wert des Parameters ***Closed-Loop Velocity (Phys. Unit/s)*** (ID 0x49) im flüchtigen Speicher des C-863.12. Der Parameterwert kann mit WPA (S. 209) als Standard gespeichert werden, Details siehe "Anpassen von Einstellungen" (S. 235).

Der maximale Wert, der mit dem Befehl VEL gesetzt werden kann, wird durch den Parameter ***Maximum Closed-Loop Velocity (Phys. Unit/s)***, ID 0xA, vorgegeben.

VEL? (Get Closed-Loop Velocity)

Beschreibung: Fragt die kommandierte Geschwindigkeit ab.

Werden keine Argumente angegeben, wird der Wert aller Achsen abgefragt.

Format: VEL? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers.

Antwort: {<AxisID>"="<float> LF}

wobei

<float> der aktuell gültige kommandierte Geschwindigkeitswert in physikalischen Einheiten pro Sekunde ist.

Hinweise: VEL? fragt den Wert der Geschwindigkeit für den geregelten Betrieb ab.

VER? (Get Versions Of Firmware And Drivers)

Beschreibung: Fragt die Versionen der Firmware des C-863.12 und weiterer Komponenten wie z. B. Treiber und Bibliotheken ab.

Format: VER?

Argumente: Keine

Antwort {<string1>":" <string2> [<string3>]LF}

wobei

<string1> der Name der Komponente ist;
<string2> die Versionsinformation der Komponente <string1>ist;
<string3> eine optionale Angabe ist.

WAC (Wait For Condition)

Beschreibung: Wartet, bis eine angegebene Bedingung des folgenden Typs auftritt: ein angegebener Wert wird mit einem abgefragten Wert gemäß einer angegebenen Regel verglichen.

Kann nur in Makros verwendet werden.

Siehe auch den Befehl MEX (S. 179).

Format: WAC <CMD?> <OP> <Value>

Argumente <CMD?> ist ein Abfragebefehl in seiner üblichen Schreibweise. Die Antwort muss ein einzelner Wert sein (und nicht mehr). Beispiel siehe unten.

<OP> ist der zu verwendende Operator. Folgende Operatoren sind möglich:
= <= < > >= !=
Wichtig: Vor und nach dem Operator muss ein Leerzeichen stehen!

<Value> ist der Wert, der mit der Antwort auf <CMD?> zu vergleichen ist.

Antwort: Keine

Beispiel: Senden:

```
MAC BEG LPMOTION
MVR 1 1
WAC ONT? 1 = 1
MVR 1 -1
WAC ONT? 1 = 1
MAC START LPMOTION
MAC END
MAC START LPMOTION
```

Hinweis: Makro LPMOTION wird erst aufgezeichnet, dann gestartet. WAC ONT? 1 = 1 wartet, bis die Antwort auf ONT? 1 1=1 ist. Das Makro ruft sich selbst auf, um eine Endlosschleife zu bilden.

WPA (Save Parameters To Non-Volatile Memory)

Beschreibung: Schreibt den aktuell gültigen Wert eines Parameters für ein angegebenes Element aus dem flüchtigen Speicher (RAM) in den permanenten Speicher. Die auf diese Art gespeicherten Werte werden die Standardwerte.

Hinweis: Sind die aktuellen Parameterwerte falsch, kann dies zu einer Störung des Systems führen. Vergewissern Sie sich, dass die Parametereinstellungen korrekt sind, bevor Sie den Befehl WPA ausführen.

RAM-Einstellungen, die nicht mit WPA gespeichert wurden, gehen verloren, wenn der Controller ausgeschaltet oder neugestartet wird bzw. wenn die Parameter mit RPA (S. 187) wiederhergestellt werden.

Mit HPA? (S. 163) erhalten Sie eine Liste aller verfügbaren Parameter.

Prüfen Sie die aktuellen Parametereinstellungen im flüchtigen Speicher mit SPA? (S. 193).

Ein Beispiel finden Sie in der Beschreibung des Befehls SPA (S. 193).

Format: WPA <Pswd> [{<ItemID> <PamID>}]

Argumente: <Pswd> ist das Kennwort zum Schreiben in den permanenten Speicher. Nähere Angaben siehe unten.

<ItemID> ist das Element, für das ein Parameter aus dem flüchtigen Speicher im permanenten Speicher gespeichert werden soll. Nähere Angaben siehe unten.

<PamID> ist die Parameterkennung, kann im Hexadezimal- oder Dezimalformat geschrieben werden. Nähere Angaben siehe unten.

Antwort: Keine

Fehlersuche: Unzulässige Elementkennung, falsche Parameter-ID, ungültiges Kennwort

Hinweise: Die Parameter können im flüchtigen Speicher mit SPA (S.

	193), ACC (S. 133), DEC (S. 144) und VEL (S. 206) geändert werden. Wird WPA ohne Spezifizierung eines Arguments verwendet, mit Ausnahme des Kennworts, werden die aktuell gültigen Werte aller Parameter, die von dem angegebenen Kennwort betroffen sind, gespeichert. Anderenfalls kann nur ein einzelner Parameter pro Befehl WPA gespeichert werden. **Beachten Sie, dass die Anzahl von Schreibzyklen im permanenten Speicher begrenzt ist. Schreiben Sie Standardeinstellungen nur, wenn dies notwendig ist.**
Gültige Kennwörter:	Das Kennwort zum Schreiben in den permanenten Speicher ist "100".
Verfügbare Element-IDs und Parameter-IDs:	Ein Element ist eine Achse (die Kennung kann mit SAI (S. 189) geändert werden) oder das gesamte System. Weitere Informationen siehe "Kommandierbare Elemente" (S. 17). Gültige Parameter-IDs finden Sie in der Parameterübersicht (S. 244).

8.7 Fehlercodes

Die hier aufgelisteten Fehlercodes sind Bestandteil des PI General Command Set. Einige der Fehlercodes sind für Ihren Controller möglicherweise nicht relevant und werden daher nie ausgegeben.

Controllerfehler

0	PI_CNTR_NO_ERROR	No error
1	PI_CNTR_PARAM_SYNTAX	Parameter syntax error
2	PI_CNTR_UNKNOWN_COMMAND	Unknown command
3	PI_CNTR_COMMAND_TOO_LONG	Command length out of limits or command buffer overrun
4	PI_CNTR_SCAN_ERROR	Error while scanning
5	PI_CNTR_MOVE_WITHOUT_REF_OR_NO_SERVO	Unallowable move attempted on unreferenced axis, or move attempted with servo off
6	PI_CNTR_INVALID_SGA_PARAM	Parameter for SGA not valid
7	PI_CNTR_POS_OUT_OF_LIMITS	Position out of limits

8	PI_CNTR_VEL_OUT_OF_LIMITS	Velocity out of limits
9	PI_CNTR_SET_PIVOT_NOT_POSSIBLE	Attempt to set pivot point while U,V and W not all 0
10	PI_CNTR_STOP	Controller was stopped by command
11	PI_CNTR_SST_OR_SCAN_RANGE	Parameter for SST or for one of the embedded scan algorithms out of range
12	PI_CNTR_INVALID_SCAN_AXES	Invalid axis combination for fast scan
13	PI_CNTR_INVALID_NAV_PARAM	Parameter for NAV out of range
14	PI_CNTR_INVALID_ANALOG_INPUT	Invalid analog channel
15	PI_CNTR_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
16	PI_CNTR_INVALID_STAGE_NAME	Unknown stage name
17	PI_CNTR_PARAM_OUT_OF_RANGE	Parameter out of range
18	PI_CNTR_INVALID_MACRO_NAME	Invalid macro name
19	PI_CNTR_MACRO_RECORD	Error while recording macro
20	PI_CNTR_MACRO_NOT_FOUND	Macro not found
21	PI_CNTR_AXIS_HAS_NO_BRAKE	Axis has no brake
22	PI_CNTR_DOUBLE_AXIS	Axis identifier specified more than once
23	PI_CNTR_ILLEGAL_AXIS	Illegal axis
24	PI_CNTR_PARAM_NR	Incorrect number of parameters
25	PI_CNTR_INVALID_REAL_NR	Invalid floating point number
26	PI_CNTR_MISSING_PARAM	Parameter missing
27	PI_CNTR_SOFT_LIMIT_OUT_OF_RANGE	Soft limit out of range
28	PI_CNTR_NO_MANUAL_PAD	No manual pad found
29	PI_CNTR_NO_JUMP	No more step-response values
30	PI_CNTR_INVALID_JUMP	No step-response values recorded
31	PI_CNTR_AXIS_HAS_NO_REFERENCE	Axis has no reference sensor
32	PI_CNTR_STAGE_HAS_NO_LIM_SWITCH	Axis has no limit switch
33	PI_CNTR_NO_RELAY_CARD	No relay card installed
34	PI_CNTR_CMD_NOT_ALLOWED_FOR_STAGE	Command not allowed for selected stage(s)
35	PI_CNTR_NO_DIGITAL_INPUT	No digital input installed
36	PI_CNTR_NO_DIGITAL_OUTPUT	No digital output configured

37	PI_CNTR_NO_MCM	No more MCM responses
38	PI_CNTR_INVALID_MCM	No MCM values recorded
39	PI_CNTR_INVALID_CNTR_NUMBER	Controller number invalid
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave
42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter) missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error
49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	MOV! motion still in progress
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data Record Table does not exist
58	PI_CNTR_INVALID_RECORDER_SRC_OPT	Source does not exist; number too low or too high
59	PI_CNTR_INVALID_RECORDER_SRC_CHAN	Source Record Table number too low or too high
60	PI_CNTR_PARAM_PROTECTION	Protected Param: current Command Level (CCL) too low
61	PI_CNTR_AUTOZERO_RUNNING	Command execution not

		possible while Autozero is running
62	PI_CNTR_NO_LINEAR_AXIS	Autozero requires at least one linear axis
63	PI_CNTR_INIT_RUNNING	Initialization still in progress
64	PI_CNTR_READ_ONLY_PARAMETER	Parameter is read-only
65	PI_CNTR_PAM_NOT_FOUND	Parameter not found in non-volatile memory
66	PI_CNTR_VOL_OUT_OF_LIMITS	Voltage out of limits
67	PI_CNTR_WAVE_TOO_LARGE	Not enough memory available for requested wave curve
68	PI_CNTR_NOT_ENOUGH_DDL_MEMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DELAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_ARRAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN_SINGLE_MODE_RESTART	Attempt to restart the generator while it is running in single step mode
72	PI_CNTR_ANALOG_TARGET_ACTIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERATOR_ACTIVE	Motion commands are not allowed when wave generator is active
74	PI_CNTR_AUTOZERO_DISABLED	No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)
75	PI_CNTR_NO_WAVE_SELECTED	Generator started (WGO) without having selected a wave table (WSL).
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly
77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_	Open-loop commands (SVA,

	SERVO_ON	SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM
81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command
82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
90	PI_CNTR_DIO_AND_TRACING_NOT_POSSIBLE	No tracing possible while digital IOs are used on this HW revision. Reconnect to switch operation mode.
91	PI_CNTR_COLLISION	Move not possible, would cause collision
92	PI_CNTR_SLAVE_NOT_FAST_ENOUGH	Stage is not capable of following the master. Check the gear ratio.
93	PI_CNTR_CMD_NOT_ALLOWED_WHILE_AXIS_IN_MOTION	This command is not allowed while the affected axis or its master is in motion.
94	PI_CNTR_OPEN_LOOP_JOYSTICK_ENABLED	Servo cannot be switched on when open-loop joystick control is activated.
95	PI_CNTR_INVALID_SERVO_STATE_FOR_PARAMETER	This parameter cannot be changed in current servo mode.
96	PI_CNTR_UNKNOWN_STAGE_NAME	Unknown stage name
97	PI_CNTR_INVALID_VALUE_LENGTH	Invalid length of value (too much characters)
98	PI_CNTR_AUTOZERO_FAILED	AutoZero procedure was not successful
99	PI_CNTR_SENSOR_VOLTAGE_OFF	Sensor voltage is off
100	PI_LABVIEW_ERROR	PI driver for use with NI LabVIEW reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis

201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on
206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error
210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller
212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken
216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, some controllers need CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop
218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATION	Position was calculated during MOV motion
220	PI_CNTR_INTERPOLATION_FIFO_UNDERRUN	FIFO buffer underrun during interpolation
221	PI_CNTR_INTERPOLATION_FIFO_OVERFLOW	FIFO buffer overflow during interpolation
230	PI_CNTR_INVALID_HANDLE	Invalid handle
231	PI_CNTR_NO_BIOS_FOUND	No bios found
232	PI_CNTR_SAVE_SYS_CFG_FAILED	Save system configuration

		failed
233	PI_CNTR_LOAD_SYS_CFG_FAILED	Load system configuration failed
301	PI_CNTR_SEND_BUFFER_OVERFLOW	Send buffer overflow
302	PI_CNTR_VOLTAGE_OUT_OF_LIMITS	Voltage out of limits
303	PI_CNTR_OPEN_LOOP_MOTION_SET_WHEN_SERVO_ON	Open-loop motion attempted when servo ON
304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long
305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus
307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro
310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined
402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
500	PI_CNTR_EMERGENCY_STOP_BUTTON_ACTIVATED	The \"red knob\" is still set and disables system
501	PI_CNTR_EMERGENCY_STOP_BUTTON_WAS_ACTIVATED	The \"red knob\" was activated and still disables system - reanimation required
502	PI_CNTR_REDUNDANCY_LIMIT_EXCEEDED	Position consistency check failed
503	PI_CNTR_COLLISION_SWITCH_ACTIVATED	Hardware collision sensor(s) are activated

504	PI_CNTR_FOLLOWING_ERROR	Strut following error occurred, e.g. caused by overload or encoder failure
505	PI_CNTR_SENSOR_SIGNAL_INVALID	One sensor signal is not valid
506	PI_CNTR_SERVO_LOOP_UNSTABLE	Servo loop was unstable due to wrong parameter setting and switched off to avoid damage.
507	PI_CNTR_LOST_SPI_SLAVE_CONNECTION	Digital connection to external SPI slave device is lost
508	PI_CNTR_MOVE_ATTEMPT_NOT_PERMITTED	Move attempt not permitted due to customer or limit settings
509	PI_CNTR_TRIGGER_EMERGENCY_STOP	Emergency stop caused by trigger input
530	PI_CNTR_NODE_DOES_NOT_EXIST	A command refers to a node that does not exist
531	PI_CNTR_PARENT_NODE_DOES_NOT_EXIST	A command refers to a node that has no parent node
532	PI_CNTR_NODE_IN_USE	Attempt to delete a node that is in use
533	PI_CNTR_NODE_DEFINITION_IS_CYCLIC	Definition of a node is cyclic
536	PI_CNTR_HEXAPOD_IN_MOTION	Transformation cannot be defined as long as Hexapod is in motion
537	PI_CNTR_TRANSFORMATION_TYPE_NOT_SUPPORTED	Transformation node cannot be activated
539	PI_CNTR_NODE_PARENT_IDENTICAL_TO_CHILD	A node cannot be linked to itself
540	PI_CNTR_NODE_DEFINITION_INCONSISTENT	Node definition is erroneous or not complete (replace or delete it)
542	PI_CNTR_NODES_NOT_IN_SAME_CHAIN	The nodes are not part of the same chain
543	PI_CNTR_NODE_MEMORY_FULL	Unused nodes must be deleted before new nodes can be stored
544	PI_CNTR_PIVOT_POINT_FEATURE_NOT_SUPPORTED	With some transformations pivot point usage is not supported
545	PI_CNTR_SOFTLIMITS_INVALID	Soft limits invalid due to changes in coordinate system
546	PI_CNTR_CS_WRITE_PROTECTED	Coordinate system is write protected

547	PI_CNTR_CS_CONTENT_FROM_CONFIG_FILE	Coordinate system cannot be changed because its content is loaded from a configuration file
548	PI_CNTR_CS_CANNOT_BE_LINKED	Coordinate system may not be linked
549	PI_CNTR_KSB_CS_ROTATION_ONLY	A KSB-type coordinate system can only be rotated by multiples of 90 degrees
551	PI_CNTR_CS_DATA_CANNOT_BE_QUERIED	This query is not supported for this coordinate system type
552	PI_CNTR_CS_COMBINATION_DOES_NOT_EXIST	This combination of work-and-tool coordinate systems does not exist
553	PI_CNTR_CS_COMBINATION_INVALID	The combination must consist of one work and one tool coordinate system
554	PI_CNTR_CS_TYPE_DOES_NOT_EXIST	This coordinate system type does not exist
555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
556	PI_CNTR_CS_TYPE_NOT_ACTIVATED	No coordinate system of this type is activated
557	PI_CNTR_CS_NAME_INVALID	Name of coordinate system is invalid
558	PI_CNTR_CS_GENERAL_FILE_MISSING	File with stored CS systems is missing or erroneous
559	PI_CNTR_CS_LEVELING_FILE_MISSING	File with leveling CS is missing or erroneous
601	PI_CNTR_NOT_ENOUGH_MEMORY	not enough memory
602	PI_CNTR_HW_VOLTAGE_ERROR	hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	hardware temperature out of range
604	PI_CNTR_POSITION_ERROR_TOO_HIGH	Position error of any axis in the system is too high
606	PI_CNTR_INPUT_OUT_OF_RANGE	Maximum value of input signal has been exceeded
607	PI_CNTR_NO_INTEGER	Value is not integer
608	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_RUNNING	Fast alignment process cannot be paused because it is not running

609	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_PAUSED	Fast alignment process cannot be restarted/resumed because it is not paused
650	PI_CNTR_UNABLE_TO_SET_PARAM_WITH_SPA	Parameter could not be set with SPA - SEP needed?
651	PI_CNTR_PHASE_FINDING_ERROR	Phase finding error
652	PI_CNTR_SENSOR_SETUP_ERROR	Sensor setup error
653	PI_CNTR_SENSOR_COMM_ERROR	Sensor communication error
654	PI_CNTR_MOTOR_AMPLIFIER_ERROR	Motor amplifier error
655	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_I2T	Overcurrent protection triggered by I2T-module
656	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_AMP_MODULE	Overcurrent protection triggered by amplifier module
657	PI_CNTR_SAFETY_STOP_TRIGGERED	Safety stop triggered
658	PI_SENSOR_OFF	Sensor off?
659	PI_CNTR_PARAM_CONFLICT	Parameter could not be set. Conflict with another parameter.
700	PI_CNTR_COMMAND_NOT_ALLOWED_IN_EXTERNAL_MODE	Command not allowed in external mode
710	PI_CNTR_EXTERNAL_MODE_ERROR	External mode communication error
715	PI_CNTR_INVALID_MODE_OF_OPERATION	Invalid mode of operation
716	PI_CNTR_FIRMWARE_STOPPED_BY_CMD	Firmware stopped by command (#27)
717	PI_CNTR_EXTERNAL_MODE_DRIVER_MISSING	External mode driver missing
718	PI_CNTR_CONFIGURATION_FAILURE_EXTERNAL_MODE	Missing or incorrect configuration of external mode
719	PI_CNTR_EXTERNAL_MODE_CYCLETIME_INVALID	External mode cycletime invalid
720	PI_CNTR_BRAKE_ACTIVATED	Brake is activated
725	PI_CNTR_DRIVE_STATE_TRANSITION_ERROR	Drive state transition error

731	PI_CNTR_SURFACEDETECTION_RUNNING	Command not allowed while surface detection is running
732	PI_CNTR_SURFACEDETECTION_FAILED	Last surface detection failed
733	PI_CNTR_FIELDBUS_IS_ACTIVE	Fieldbus is active and is blocking GCS control commands
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
1006	PI_CNTR_INVALID_IDENTIFIER	Invalid identifier (invalid special characters, ...)
1007	PI_CNTR_UNKNOWN_VARIABLE_OR_ARGUMENT	Variable or argument not defined
1008	PI_CNTR_RUNNING_MACRO	Controller is (already) running a macro
1009	PI_CNTR_MACRO_INVALID_OPERATOR	Invalid or missing operator for condition. Check necessary spaces around operator.
1010	PI_CNTR_MACRO_NO_ANSWER	No response was received while executing WAC/MEX/JRC/...
1011	PI_CMD_NOT_VALID_IN_MACRO_MODE	Command not valid during macro execution
1012	PI_CNTR_ERROR_IN_MACRO	Error occured during macro execution
1013	PI_CNTR_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
1015	PI_CNTR_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)

1024	PI_CNTR_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
1025	PI_CNTR_MAX_MOTOR_OUTPUT_REACHED	Maximum motor output reached
1028	PI_CNTR_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
1063	PI_CNTR_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
1064	PI_CNTR_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
1065	PI_CNTR_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
1066	PI_CNTR_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
1071	PI_CNTR_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
1072	PI_CNTR_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
1073	PI_CNTR_PROFILE_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
1090	PI_CNTR_TOO_MANY_TCP_CONNECTIONS_OPEN	There are too many open tcpip connections
2000	PI_CNTR_ALREADY_HAS_SERIAL_NUMBER	Controller already has a serial number
2100	PI_CNTR_FEATURE_LICENSE_INVALID	Entered license is invalid
4000	PI_CNTR_SECTOR_ERASE_FAILED	Sector erase failed
4001	PI_CNTR_FLASH_PROGRAM_FAILED	Flash program failed
4002	PI_CNTR_FLASH_READ_FAILED	Flash read failed
4003	PI_CNTR_HW_MATCHCODE_ERROR	HW match code missing/invalid
4004	PI_CNTR_FW_MATCHCODE_ERROR	FW match code missing/invalid

4005	PI_CNTR_HW_VERSION_ERROR	HW version missing/invalid
4006	PI_CNTR_FW_VERSION_ERROR	FW version missing/invalid
4007	PI_CNTR_FW_UPDATE_ERROR	FW update failed
4008	PI_CNTR_FW_CRC_PAR_ERROR	FW Parameter CRC wrong
4009	PI_CNTR_FW_CRC_FW_ERROR	FW CRC wrong
5000	PI_CNTR_INVALID_PCC_SCAN_DATA	PicoCompensation scan data is not valid
5001	PI_CNTR_PCC_SCAN_RUNNING	PicoCompensation is running, some actions can not be executed during scanning/recording
5002	PI_CNTR_INVALID_PCC_AXIS	Given axis cannot be defined as PPC axis
5003	PI_CNTR_PCC_SCAN_OUT_OF_RANGE	Defined scan area is larger than the travel range
5004	PI_CNTR_PCC_TYPE_NOT_EXISTING	Given PicoCompensation type is not defined
5005	PI_CNTR_PCC_PAM_ERROR	PicoCompensation parameter error
5006	PI_CNTR_PCC_TABLE_ARRAY_TOO_LARGE	PicoCompensation table is larger than maximum table length
5100	PI_CNTR_NEXLINE_ERROR	Common error in NEXLINE® firmware module
5101	PI_CNTR_CHANNEL_ALREADY_USED	Output channel for NEXLINE® can not be redefined for other usage
5102	PI_CNTR_NEXLINE_TABLE_TOO_SMALL	Memory for NEXLINE® signals is too small
5103	PI_CNTR_RNP_WITH_SERVO_ON	RNP can not be executed if axis is in closed loop
5104	PI_CNTR_RNP_NEEDED	Relax procedure (RNP) needed
5200	PI_CNTR_AXIS_NOT_CONFIGURED	Axis must be configured for this action
5300	PI_CNTR_FREQU_ANALYSIS_FAILED	Frequency analysis failed
5301	PI_CNTR_FREQU_ANALYSIS_RUNNING	Another frequency analysis is running
6000	PI_CNTR_SENSOR_ABS_INVALID_VALUE	Invalid preset value of absolute sensor

6001	PI_CNTR_SENSOR_ABS_WRITE_ERROR	Error while writing to sensor
6002	PI_CNTR_SENSOR_ABS_READ_ERROR	Error while reading from sensor
6003	PI_CNTR_SENSOR_ABS_CRC_ERROR	Checksum error of absolute sensor
6004	PI_CNTR_SENSOR_ABS_ERROR	General error of absolute sensor
6005	PI_CNTR_SENSOR_ABS_OVERFLOW	Overflow of absolute sensor position

Schnittstellenfehler

0	COM_NO_ERROR	No error occurred during function call
-1	COM_ERROR	Error during com operation (could not be specified)
-2	SEND_ERROR	Error while sending data
-3	REC_ERROR	Error while receiving data
-4	NOT_CONNECTED_ERROR	Not connected (no port with given ID open)
-5	COM_BUFFER_OVERFLOW	Buffer overflow
-6	CONNECTION_FAILED	Error while opening port
-7	COM_TIMEOUT	Timeout error
-8	COM_MULTILINE_RESPONSE	There are more lines waiting in buffer
-9	COM_INVALID_ID	There is no interface or DLL handle with the given ID
-10	COM_NOTIFY_EVENT_ERROR	Event/message for notification could not be opened
-11	COM_NOT_IMPLEMENTED	Function not supported by this interface type
-12	COM_ECHO_ERROR	Error while sending "echoed" data
-13	COM_GPIB_EDVR	IEEE488: System error
-14	COM_GPIB_ECIC	IEEE488: Function requires GPIB board to be CIC
-15	COM_GPIB_ENOL	IEEE488: Write function detected no listeners
-16	COM_GPIB_EADR	IEEE488: Interface board not addressed correctly

-17	COM_GPIB_EARG	IEEE488: Invalid argument to function call
-18	COM_GPIB_ESAC	IEEE488: Function requires GPIB board to be SAC
-19	COM_GPIB_EABO	IEEE488: I/O operation aborted
-20	COM_GPIB_ENEB	IEEE488: Interface board not found
-21	COM_GPIB_EDMA	IEEE488: Error performing DMA
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended operation
-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost
-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full
-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits
-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)
-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources

-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device
-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present
-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error
-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port: was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again
-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)
-62	COM_SOCKET_HOST_NOT_FOUND	Host not found
-63	COM_DEVICE_CONNECTED	Device already connected
-64	COM_INVALID_COM_PORT	Invalid COM port

-65	COM_USB_DEVICE_NOT_FOUND	USB device not found
-66	COM_NO_USB_DRIVER	No USB driver installed
-67	COM_USB_NOT_SUPPORTED	USB is not supported

DLL-Fehler

-1001	PI_UNKNOWN_AXIS_IDENTIFIER	Unknown axis identifier
-1002	PI_NR_NAV_OUT_OF_RANGE	Number for NAV out of range--must be in [1,10000]
-1003	PI_INVALID_SGA	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	PI_UNEXPECTED_RESPONSE	Controller sent unexpected response
-1005	PI_NO_MANUAL_PAD	No manual control pad installed, calls to SMA and related commands are not allowed
-1006	PI_INVALID_MANUAL_PAD_KNOB	Invalid number for manual control pad knob
-1007	PI_INVALID_MANUAL_PAD_AXIS	Axis not currently controlled by a manual control pad
-1008	PI_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g., reference move, fast scan algorithm)
-1009	PI_THREAD_ERROR	Internal error--could not start thread
-1010	PI_IN_MACRO_MODE	Controller is (already) in macro mode--command not valid in macro mode
-1011	PI_NOT_IN_MACRO_MODE	Controller not in macro mode--command not valid unless macro mode active
-1012	PI_MACRO_FILE_ERROR	Could not open file to write or read macro
-1013	PI_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
-1014	PI_MACRO_EDITOR_ERROR	Internal error in macro editor
-1015	PI_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	PI_AXIS_ALREADY_EXISTS	Axis identifier is already in use by a connected stage

-1017	PI_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
-1018	PI_COM_ARRAY_ERROR	Could not access array data in COM server
-1019	PI_COM_ARRAY_RANGE_ERROR	Range of array does not fit the number of parameters
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?
-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed
-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup
-1032	PI_USER_CANCELLED	The user cancelled a dialog
-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL
-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter

		file open. Close it before opening a new file
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid
-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis
-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started
-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC
-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication

		with OPM (Optical Power Meter), maybe no OPM connected
-1057	PI_WAVE_EDITOR_WRONG_PARAMNUM	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	PI_WAVE_EDITOR_FREQUENCY_OUT_OF_RANGE	WaveEditor: Frequency out of range
-1059	PI_WAVE_EDITOR_WRONG_IP_VALUE	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	PI_WAVE_EDITOR_WRONG_DP_VALUE	WaveEditor: Error during wave creation, incorrect index for floating point parameter
-1061	PI_WAVE_EDITOR_WRONG_ITEM_VALUE	WaveEditor: Error during wave creation, could not calculate value
-1062	PI_WAVE_EDITOR_MISSING_GRAPH_COMPONENT	WaveEditor: Graph display component not installed
-1063	PI_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
-1064	PI_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
-1065	PI_EXT_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
-1066	PI_EXT_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
-1067	PI_PROFILE_GENERATOR_NO_PROFILE	ProfileGenerator: No profile has been created yet
-1068	PI_PROFILE_GENERATOR_OUT_OF_LIMITS	ProfileGenerator: Generated profile exceeds limits of one or both axes
-1069	PI_PROFILE_GENERATOR_UNKNOWN_PARAMETER	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	PI_PROFILE_GENERATOR_PAR_OUT_OF_RANGE	ProfileGenerator: Parameter out of allowed range
-1071	PI_EXT_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
-1072	PI_EXT_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is

		not assigned to this axis
-1073	PI_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.
-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by another function. Please try again later.
-1077	PI_PARAM_DAT_FILE_INVALID_VERSION	Version of parameter DAT file does not match the required version. Current files are available at www.pi.ws.
-1078	PI_CANNOT_WRITE_TO_PARAM_DAT_FILE	Cannot write to parameter DAT file to store user defined stage type.
-1079	PI_CANNOT_CREATE_PARAM_DAT_FILE	Cannot create parameter DAT file to store user defined stage type.
-1080	PI_PARAM_DAT_FILE_INVALID_REVISION	Parameter DAT file does not have correct revision.
-1081	PI_USERSTAGES_DAT_FILE_INVALID_REVISION	User stages DAT file does not have correct revision.
-1082	PI_SOFTWARE_TIMEOUT	Timeout Error. Some lengthy operation did not finish within expected time.
-1083	PI_WRONG_DATA_TYPE	A function argument has an unexpected data type.
-1084	PI_DIFFERENT_ARRAY_SIZES	Length of data arrays is different.
-1085	PI_PARAM_NOT_FOUND_IN_PARAM_DAT_FILE	Parameter value not found in parameter DAT file.
-1086	PI_MACRO_RECORDING_NOT_ALLOWED_IN_THIS_MODE	Macro recording is not allowed in this mode of operation.
-1087	PI_USER_CANCELLED_COMMAND	Command cancelled by user input.

-1088	PI_TOO_FEW_GCS_DATA	Controller sent too few GCS data sets
-1089	PI_TOO_MANY_GCS_DATA	Controller sent too many GCS data sets
-1090	PI_GCS_DATA_READ_ERROR	Communication error while reading GCS data
-1091	PI_WRONG_NUMBER_OF_INPUT_ARGUMENTS	Wrong number of input arguments.
-1092	PI_FAILED_TO_CHANGE_CCL_LEVEL	Change of command level has failed.
-1093	PI_FAILED_TO_SWITCH_OFF_SERVO	Switching off the servo mode has failed.
-1094	PI_FAILED_TO_SET_SINGLE_PARAMETER_WHILE_PERFORMING_CST	A parameter could not be set while performing CST: CST was not performed (parameters remain unchanged).
-1095	PI_ERROR_CONTROLLER_REBOOT	Connection could not be reestablished after reboot.
-1096	PI_ERROR_AT_QHPA	Sending HPA? or receiving the response has failed.
-1097	PI_QHPA_NONCOMPLIANT_WITH_GCS	HPA? response does not comply with GCS2 syntax.
-1098	PI_FAILED_TO_READ_QSPA	Response to SPA? could not be received.
-1099	PI_PAM_FILE_WRONG_VERSION	Version of PAM file cannot be handled (too old or too new)
-1100	PI_PAM_FILE_INVALID_FORMAT	PAM file does not contain required data in PAM-file format
-1101	PI_INCOMPLETE_INFORMATION	Information does not contain all required data
-1102	PI_NO_VALUE_AVAILABLE	No value for parameter available
-1103	PI_NO_PAM_FILE_OPEN	No PAM file is open
-1104	PI_INVALID_VALUE	Invalid value
-1105	PI_UNKNOWN_PARAMETER	Unknown parameter
-1106	PI_RESPONSE_TO_QSEP_FAILED	Response to SEP? could not be received.
-1107	PI_RESPONSE_TO_QSPA_FAILED	Response to SPA? could not be received.

-1108	PI_ERROR_IN_CST_VALIDATION	Error while performing CST: One or more parameters were not set correctly.
-1109	PI_ERROR_PAM_FILE_HAS_DUPLICATE_ENTRY_WITH_DIFFERENT_VALUES	PAM file has duplicate entry with different values.
-1110	PI_ERROR_FILE_NO_SIGNATURE	File has no signature
-1111	PI_ERROR_FILE_INVALID_SIGNATURE	File has invalid signature
-1112	PI_ERROR_CANNOT_DETERMINE_ACTUAL_END_OF_TRAVEL_WHILE_PLATFORM_IS_MOVING	Cannot determine actual end of travel range while platform is moving.
-1113	PI_ERROR_AT_QIDN	Sending IDN? or receiving the response has failed.
-1114	PI_ERROR_AT_MAC_DEF	Sending MAC_DEF or receiving the response has failed.
-1115	PI_CONTROLLER_OR_CONTROLLER_VERSION_DOES_NOT_EXIST_IN_PISTAGES_DATABASE	Sending Controller or controller version does not exist in PIStages database.
-1116	PI_NOT_ENOUGH_MEMORY	Not enough memory
-1117	PI_ERROR_AXIS_RUNTIME_ERROR	Runtime error indicated for axis, check error log with \"LOG?\" to find more details.
-1118	PI_ERROR_SYSTEM_RUNTIME_CRITICAL_ERROR	Critical error indicated for system, check error log with \"LOG?\" to find more details.
-1119	PI_ERROR_CANNOT_START_EMULATOR	Cannot start emulation software.
-1120	COM_DEVICE_NOT_SUPPORTED	Device is not supported
-10000	PI_PARAMETER_DB_INVALID_STAGE_TYPE_FORMAT	PI stage database: String containing stage type and description has invalid format.
-10001	PI_PARAMETER_DB_SYSTEM_NOT_AVAILABLE	PI stage database: Database does not contain the selected stage type for the connected controller.
-10002	PI_PARAMETER_DB_FAILED_TO_ESTABLISH_CONNECTION	PI stage database: Establishing the connection has failed.
-10003	PI_PARAMETER_DB_COMMUNICATION_ERROR	PI stage database: Communication was interrupted (e.g. because database was deleted).

-10004	PI_PARAMETER_DB_ERROR_WHILE_QUERYING_PARAMETERS	PI stage database: Querying data failed.
-10005	PI_PARAMETER_DB_SYSTEM_ALREADY_EXISTS	PI stage database: System already exists. Rename stage and try again.
-10006	PI_PARAMETER_DB_QHPA_CONTANS_UNKNOWN_PAM_IDS	PI stage database: Response to HPA? contains unknown parameter IDs.
-10007	PI_PARAMETER_DB_AND_QHPA_ARE_INCONSISTENT	PI stage database: Inconsistency between database and response to HPA?.
-10008	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_ADDED	PI stage database: Stage has not been added.
-10009	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_REMOVED	PI stage database: Stage has not been removed.
-10010	PI_PARAMETER_DB_CONTROLLER_DB_PARAMETERS_MISMATCH	Controller does not support all stage parameters stored in PI stage database. No parameters were set.
-10011	PI_PARAMETER_DB_DATABASE_IS_OUTDATED	The version of PISTAGES3.DB stage database is out of date. Please update via PIUpdateFinder. No parameters were set.
-10012	PI_PARAMETER_DB_AND_HPA_MISMATCH_STRICT	Mismatch between number of parameters present in stage database and available in controller interface. No parameters were set.
-10013	PI_PARAMETER_DB_AND_HPA_MISMATCH_LOOSE	Mismatch between number of parameters present in stage database and available in controller interface. Some parameters were ignored.
-10014	PI_PARAMETER_DB_FAILED_TO_SET_PARAMETERS_CORRECTLY	One or more parameters could not be set correctly on the controller.
-10015	PI_PARAMETER_DB_MISSING_PARAMETER_DEFINITIONS_IN_DATABASE	One or more parameter definitions are not present in stage database. Please update PISTAGES3.DB via PIUpdateFinder. Missing parameters were ignored.

-10016	PI_PARAMETER_DB_MISSING_FIRMWARE_FEATURE_ON_CONTROLLER	Parameters could not be set on controller because the corresponding firmware feature is missing

9 Anpassen von Einstellungen

9.1 Einstellungen des C-863.12

Die Eigenschaften des C-863.12 und des angeschlossenen Positionierers sind im C-863.12 als Parameterwerte hinterlegt (z. B. Einstellungen für den Regelalgorithmus (S. 26)).

Die Parameter können in folgende Kategorien eingeteilt werden:

- Geschützte Parameter, deren Werkseinstellung nicht geändert werden kann
- Parameter, die zur Anpassung an die Anwendung vom Benutzer eingestellt werden können

Das Schreibrecht für die Parameter ist durch Befehlsebenen festgelegt.

Jeder Parameter ist sowohl im flüchtigen als auch im permanenten Speicher des C-863.12 vorhanden. Die Werte im permanenten Speicher werden als Standardwerte beim Einschalten oder Neustart des C-863.12 in den flüchtigen Speicher geladen. Die Werte im flüchtigen Speicher bestimmen das aktuelle Verhalten des Systems.

In der PC-Software von PI werden auch die Bezeichnungen "Active Values" für die Parameterwerte im flüchtigen Speicher und "Startup Values" für die Parameterwerte im permanenten Speicher verwendet.

9.2 Parameterwerte im C-863.12 ändern

HINWEIS

Unpassende Parametereinstellungen!

Die Werte im permanenten Speicher werden als Standardwerte beim Einschalten oder Neustart des C-863.12 in den flüchtigen Speicher geladen und sind sofort gültig. Unpassende Parametereinstellungen können zur Beschädigung der angeschlossenen Mechanik führen.

- ➢ Ändern Sie Parameterwerte nur nach sorgfältiger Überlegung.
- ➢ Sichern Sie die aktuellen Parameterwerte auf dem PC (S. 237), bevor Sie Änderungen im permanenten Speicher durchführen.

INFORMATION

Die Anzahl der Schreibzyklen im permanenten Speicher ist durch die begrenzte Lebensdauer des Speicherchips (EEPROM) beschränkt.

- ➢ Überschreiben Sie die Standardwerte nur, wenn es notwendig ist.
- ➢ Sichern Sie die aktuellen Parameterwerte auf dem PC (S. 237), bevor Sie Änderungen im permanenten Speicher durchführen.

➢ Wenden Sie sich an unseren Kundendienst (S. 263), wenn der C-863.12 ein unerwartetes Verhalten zeigt.

INFORMATION

Wenn Sie die PC-Software von PI verwenden, werden Informationen als Parameterwerte aus einer Positioniererdatenbank (S. 12) in den flüchtigen Speicher des C-863.12 geladen.

Parameter, die aus der Positioniererdatenbank geladen werden, sind in der Parameterübersicht (S. 244) farbig markiert.

9.2.1 Allgemeine Befehle für Parameter

Für Parameter stehen folgende allgemeine Befehle zur Verfügung:

Befehl	Funktion
`CCL`	Auf eine höhere Befehlsebene wechseln, um z. B. Schreibrecht auf bestimmte Parameter zu erhalten.
`CCL?`	Aktive Befehlsebene abfragen.
`HPA?`	Antwortet mit einem Hilfetext, der alle verfügbaren Parameter mit Kurzbeschreibungen enthält.
`RPA`	Parameterwert vom permanenten in den flüchtigen Speicher kopieren.
`SEP`	Parameter im permanenten Speicher ändern.
`SEP?`	Parameterwerte aus dem permanenten Speicher abfragen.
`SPA`	Parameter im flüchtigen Speicher ändern.
`SPA?`	Parameterwerte aus dem flüchtigen Speicher abfragen.
`WPA`	Aktuellen Parameterwert vom flüchtigen in den permanenten Speicher kopieren. Dort wird er als Standardwert verwendet.

Einzelheiten finden Sie in den Befehlsbeschreibungen (S. 130).

9.2.2 Befehle für Schnellzugriff auf einzelne Parameter

Die folgenden, speziellen Befehle ändern die zugehörigen Parameter nur im flüchtigen Speicher. Die geänderten Werte müssen bei Bedarf mit dem Befehl `WPA` (S. 209) in den permanenten Speicher geschrieben werden.

INFORMATION

Die nachfolgend aufgelisteten Parameter können auch mit den allgemeinen Befehlen geändert werden.

Befehl	Einstellbare Parameter
`ACC`	Beschleunigung im geregelten Betrieb (0xB)
`DEC`	Abbremsung im geregelten Betrieb (0xC)
`VEL`	Geschwindigkeit im geregelten Betrieb (0x49)

Einzelheiten finden Sie in den Befehlsbeschreibungen (S. 130).

9.2.3 Parameterwerte in Textdatei sichern

INFORMATION

Der C-863.12 wird über Parameter konfiguriert, z. B. zur Anpassung an die angeschlossene Mechanik. Das Ändern von Parameterwerten kann zu unerwünschten Ergebnissen führen.

- Legen Sie vor dem Ändern der Parametereinstellungen des C-863.12 eine Sicherungskopie auf dem PC an. Sie können dann jederzeit die Originaleinstellungen wiederherstellen.
- Erstellen Sie nach jeder Optimierung der Parameterwerte oder Anpassung des C-863.12 an eine bestimmte Mechanik eine weitere Sicherungskopie mit neuem Dateinamen.

INFORMATION

Parameterwerte, die in einer Textdatei auf dem PC gesichert wurden, können in PIMikroMove® oder PITerminal zurück auf den C-863.12 geladen werden. Im Fenster zum Senden von Befehlen ist dazu die Schaltfläche ***Send file...*** verfügbar. Vor dem Laden in den C-863.12 müssen die einzelnen Zeilen der Textdatei in Befehlszeilen umgewandelt werden, die entsprechende SPA- oder SEP-Befehle enthalten.

Voraussetzungen

✓ Sie haben die Kommunikation zwischen dem C-863.12 und dem PC mit PIMikroMove® oder PITerminal hergestellt (S. 57).

Parameterwerte in Textdatei sichern

1. Wenn Sie PIMikroMove® verwenden, öffnen Sie das Fenster zum Senden von Befehlen:
   – Wählen Sie im Hauptfenster den Menüeintrag ***Tools > Command entry*** oder drücken Sie die Taste F4 auf der Tastatur.

   In PITerminal ist nach dem Herstellen der Kommunikation automatisch das Hauptfenster geöffnet, aus dem Befehle gesendet werden können.
2. Fragen Sie die Parameterwerte ab, von denen Sie eine Sicherheitskopie erstellen möchten.
   – Wenn Sie die Parameterwerte aus dem flüchtigen Speicher des C-863.12 sichern möchten: Senden Sie den Befehl `SPA?`.

– Wenn Sie die Parameterwerte aus dem permanenten Speicher des C-863.12 sichern möchten: Senden Sie den Befehl `SEP?`.

3. Klicken Sie auf die Schaltfläche ***Save...***.

   Das Fenster ***Save content of terminal as textfile*** öffnet sich.

4. Speichern Sie im Fenster ***Save content of terminal as textfile*** die abgefragten Parameterwerte in einer Textdatei auf Ihrem PC.

9.2.4 Parameterwerte ändern: Generelle Vorgehensweise

Für die Arbeit mit Parametern können Sie die allgemeinen Befehle (S. 236) und die Befehle für den Schnellzugriff (S. 236) verwenden.

Für den einfacheren Zugang zu Parametern wird im Folgenden PIMikroMove® verwendet, so dass Sie sich nicht mit den entsprechenden Befehlen auseinandersetzen müssen.

HINWEIS

Unpassende Parametereinstellungen!

Die Werte im permanenten Speicher werden als Standardwerte beim Einschalten oder Neustart des C-863.12 in den flüchtigen Speicher geladen und sind sofort gültig. Unpassende Parametereinstellungen können zur Beschädigung der angeschlossenen Mechanik führen.

- ➢ Ändern Sie Parameterwerte nur nach sorgfältiger Überlegung.
- ➢ Sichern Sie die aktuellen Parameterwerte auf dem PC (S. 237), bevor Sie Änderungen im permanenten Speicher durchführen.

INFORMATION

Für das Ändern von Parameterwerten wird generell folgendes Vorgehen empfohlen:

1. Ändern Sie die Parameterwerte im flüchtigen Speicher.
2. Prüfen Sie, ob der C-863.12 mit den geänderten Parameterwerten korrekt funktioniert.

Wenn ja:

- ➢ Schreiben Sie die geänderten Parameterwerte in den permanenten Speicher.

Wenn nein:

- ➢ Ändern und prüfen Sie die Parameterwerte im flüchtigen Speicher erneut.

INFORMATION

Der Schreibzugriff auf die Parameter des C-863.12 ist durch Befehlsebenen festgelegt. Nach dem Einschalten oder Neustart des Controllers ist die aktive Befehlsebene immer 0. Auf Befehlsebenen > 1 besteht Schreibzugriff nur für PI-Servicepersonal.

- ➢ Wenn Sie Probleme mit Parametern der Befehlsebene 2 oder höher haben, wenden Sie sich an den Kundendienst (S. 263).

Voraussetzungen

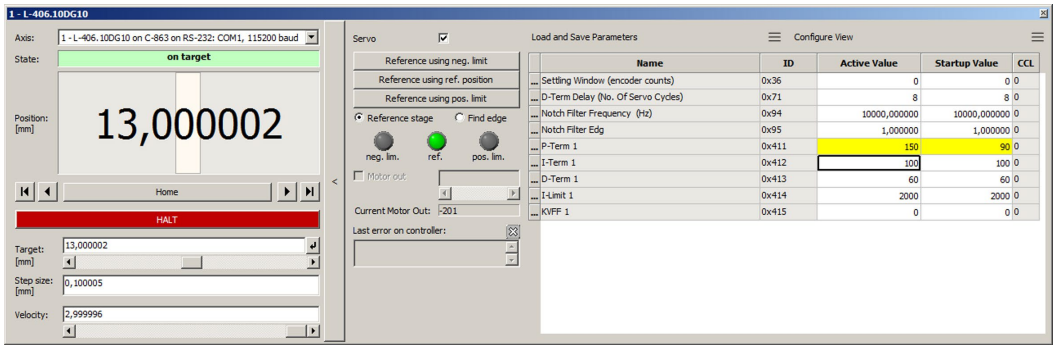
- ✓ Wenn Sie Parameterwerte im permanenten Speicher des C-863.12 ändern wollen: Sie haben die Parameterwerte des C-863.12 in einer Textdatei auf dem PC gesichert (S. 237).
- ✓ Sie haben die Kommunikation zwischen dem C-863.12 und dem PC mit PIMikroMove® hergestellt (S. 57).

Parameterwerte ändern: Generelle Vorgehensweise

1. Zeigen Sie in PIMikroMove® die Parameterliste an.

   Wenn Sie die achsbezogenen Parameter des C-863.12 ändern wollen:

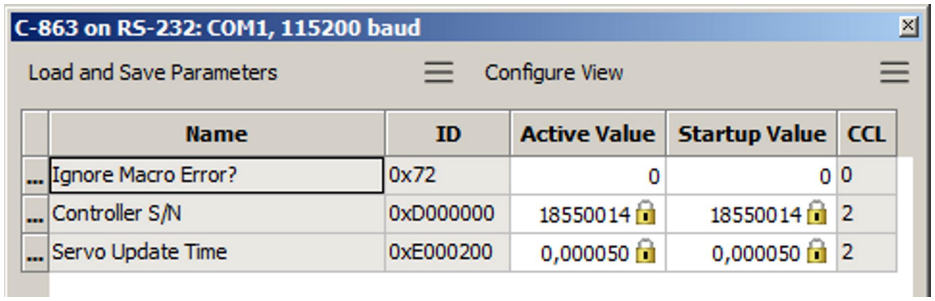
   a) Öffnen Sie im Hauptfenster von PIMikroMove® das erweiterte Einzelachsen-Fenster für den angeschlossenen Positionierer, indem Sie mit der rechten Maustaste auf die entsprechende Zeile der Registerkarte ***Axes*** klicken und im Kontextmenü ***Show Expanded Single Axis Window*** auswählen.



   b) Wenn der zu ändernde Parameter nicht in der Liste auf der rechten Seite des Fensters enthalten ist, klicken Sie auf ***Configure View > Select parameters...*** und fügen ihn zur Liste hinzu. Sie können auch bestimmte Gruppen von Parametern oder alle achsbezogenen Parameter einblenden.

   Wenn Sie die systembezogenen Parameter des C-863.12 ändern wollen:

   – Öffnen Sie im Hauptfenster von PIMikroMove® das Fenster für die systembezogenen Parameter des C-863.12, indem Sie im Menü ***C-863.12 > Show system parameters*** auswählen.



2. Ändern Sie in der entsprechenden Parameterliste die gewünschten Parameterwerte im flüchtigen oder permanenten Speicher des C-863.12.

Wenn Sie Parameterwerte im flüchtigen Speicher ändern wollen, haben Sie folgende Optionen:

- Tippen Sie den neuen Parameterwert in das entsprechende Eingabefeld in der Spalte ***Active Value*** der Liste ein. Drücken Sie auf der Tastatur des PC die Enter-Taste oder klicken Sie mit der Maus außerhalb des Eingabefeldes, um den Parameterwert in den flüchtigen Speicher des C-863.12 zu übertragen.
- Klicken Sie auf ***Load and Save Parameters -> Load all startup parameters of the axis / system from controller***, um die Werte aller achsenbezogenen / systembezogenen Parameter aus dem permanenten Speicher des C-863.12 zu laden.
- Klicken Sie im erweiterten Einzelachsen-Fenster auf ***Load and Save Parameters > Load parameters from stage database…***, um für die Achse einen ausgewählten Parametersatz aus der Positioniererdatenbank zu laden. Mit ***Load and Save Parameters > Reload parameters from stage database…*** können Sie den aktuell geladenen Parametersatz erneut laden.

Wenn Sie Parameterwerte im permanenten Speicher ändern wollen, haben Sie folgende Optionen:

- Tippen Sie den neuen Parameterwert in das entsprechende Eingabefeld in der Spalte ***Startup Value*** der Liste ein. Drücken Sie auf der Tastatur des PC die Enter-Taste oder klicken Sie mit der Maus außerhalb des Eingabefeldes, um den Parameterwert in den permanenten Speicher des C-863.12 zu übertragen.
- Klicken Sie auf ***Load and Save Parameters -> Save all currently active axis / system parameters as startup parameters to controller***, um die Werte aller achsenbezogenen / systembezogenen Parameter aus dem flüchtigen in den permanenten Speicher des C-863.12 zu schreiben. Sie können Parameter überspringen, für die auf der aktuellen Befehlsebene kein Schreibzugriff besteht.

Wenn ein Parameterwert im flüchtigen Speicher (Spalte ***Active Value***) vom Parameterwert im permanenten Speicher (Spalte ***Startup Value***) abweicht, ist die Zeile in der Liste farbig markiert.

9.3 Positioniererтyp anlegen oder ändern

Sie können in der PC-Software von PI den für Ihren Positionierer geeigneten Parametersatz aus einer Positioniererdatenbank auswählen. Die Software überträgt die Werte des ausgewählten Parametersatzes in den flüchtigen oder permanenten Speicher des Controllers. Weitere Informationen siehe "Positioniererdatenbanken" (S. 12).

In der Positioniererdatenbank PIStages3 können Sie neue Parametersätze anlegen und bearbeiten. Dies kann z. B. in folgenden Fällen erforderlich sein:

- Sie möchten einen Positionierer mit anderen Regelparameter-Einstellungen als denjenigen aus dem Standard-Parametersatz betreiben.
- Sie möchten die Verfahrbereichsgrenzen des Positionierers an Ihre Anwendung anpassen.

- Sie haben einen kundenspezifischen Positionierer.

INFORMATION

Möglichkeiten zum Anlegen und Bearbeiten von Parametersätzen in der Positioniererdatenbank PISTAGES3.DB:

- Einen neuen Positionierertyp können Sie am einfachsten anlegen, indem Sie in PIMikroMove® einen vorhandenen Positionierertyp ändern und ihn unter einer neuen Benennung abspeichern.
- Mit dem Programm PIStages3Editor, das in der PI Software Suite enthalten ist, können Sie die Positioniererdatenbank öffnen und direkt editieren.

Im Folgenden wird PIMikroMove® zum Anlegen eines neuen Positionierertyps und zum Ändern eines vorhandenen Positionierertyps verwendet.

Voraussetzungen

✓ Sie haben die neueste Version der Positioniererdatenbank PISTAGES3.DB auf dem PC installiert (S. 41).

✓ Wenn Sie für Ihren Positionierer eine kundenspezifische Positioniererdatenbank von PI erhalten haben, dann haben Sie diesen Datensatz in PIStages3 importiert (S. 43).

✓ Sie haben die Kommunikation zwischen dem C-863.12 und dem PC mit PIMikroMove® hergestellt (S. 57).

Positionierertyp in Positioniererdatenbank anlegen

1. Wählen Sie im Hauptfenster von PIMikroMove® den Menüeintrag ***C-863 > Select connected stages...***.

   Das Fenster ***Start up stages/axes for C-863*** öffnet sich, der Schritt ***Select connected stages*** ist aktiv.

2. Wählen Sie im Schritt ***Select connected stages*** einen passenden Positionierertyp aus:

   a) Markieren Sie den Positionierertyp in der Liste ***Stage database entries***.

b) Klicken Sie auf ***Assign***.

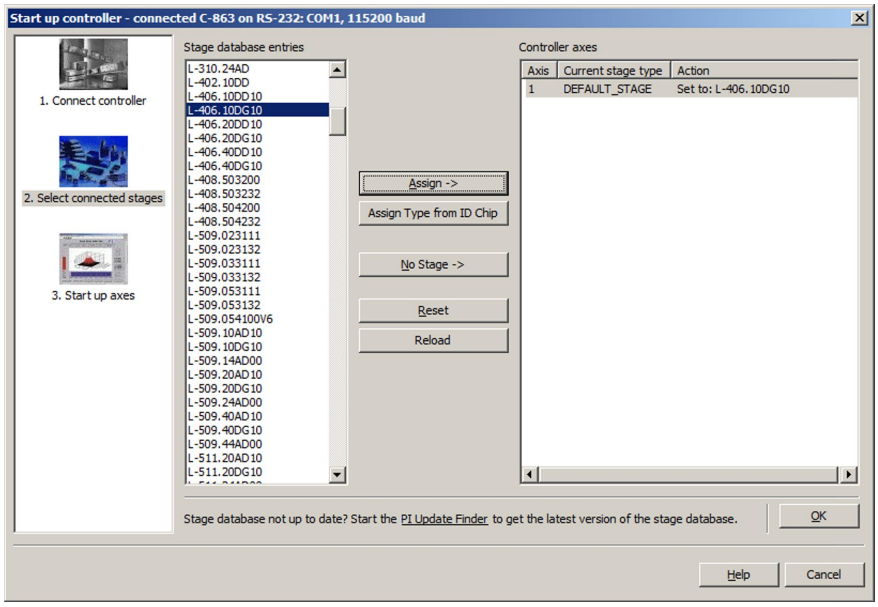


Abbildung 24: Start up controller – Select connected stages

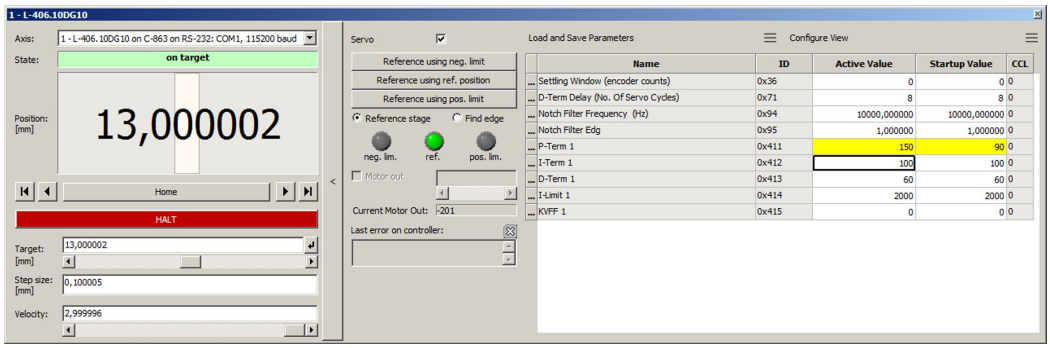
c) Bestätigen Sie die Auswahl mit ***OK***.

3. Klicken Sie im Dialog ***Save all changes permanently?*** auf ***Keep the changes temporarily***, um die Parametereinstellungen in den flüchtigen Speicher des C-863.12 zu laden.

   Das Fenster ***Start up stages/axes*** wechselt zum Schritt ***Start up axes***.

4. Klicken Sie im Schritt ***Start up axes*** auf ***Close***, um das Fenster ***Start up stages/axes*** zu schließen.

5. Öffnen Sie im Hauptfenster von PIMikroMove® das erweiterte Einzelachsen-Fenster für den ausgewählten Positionierertyp, indem Sie mit der rechten Maustaste auf die entsprechende Zeile der Registerkarte ***Axes*** klicken und im Kontextmenü ***Show Expanded Single Axis Window*** auswählen.



6. Geben Sie neue Werte für die zu ändernden Parameter ein:

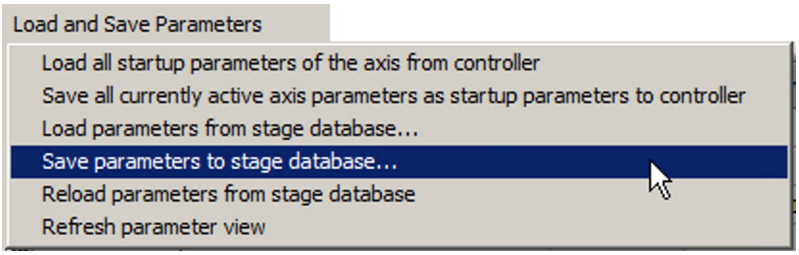
   a) Wenn der zu ändernde Parameter nicht in der Liste auf der rechten Seite des Fensters enthalten ist, klicken Sie auf ***Configure view > Select parameters...*** und

fügen ihn zur Liste hinzu. Sie können auch bestimmte Gruppen von Parametern oder alle achsenbezogenen Parameter einblenden.

b) Tippen Sie den neuen Parameterwert in das entsprechende Eingabefeld in der Spalte ***Active Value*** der Liste ein.

c) Drücken Sie auf der Tastatur des PC die Enter-Taste oder klicken Sie mit der Maus außerhalb des Eingabefeldes, um den Parameterwert in den flüchtigen Speicher des Controllers zu übertragen. Anmerkung: Wenn ein Parameterwert im flüchtigen Speicher (Spalte ***Active Value***) vom Parameterwert im permanenten Speicher (Spalte ***Startup Value***) abweicht, ist die Zeile in der Liste farbig markiert.

7. Klicken Sie auf ***Load and Save Parameters -> Save parameters to stage database...***.



Der Dialog ***Save Parameters as User Stage Type*** öffnet sich.

8. Speichern Sie im Dialog ***Save Parameters as User Stage Type*** die geänderten Parameterwerte als neuen Positionierertyp:

a) Lassen Sie den Eintrag im Feld ***Parameters of axis*** unverändert.

b) Tragen Sie im Feld ***Save as*** die Benennung für den neuen Positionierertyp ein.

c) Klicken Sie auf ***OK***.

Der neue Positionierertyp wurde in der Positioniererdatenbank PISTAGES3.DB gespeichert. Die Anzeige des angeschlossenen Positionierertyps wurde im Einzelachsen-Fenster und im Hauptfenster von PIMikroMove® aktualisiert. Der neue Positionierertyp steht ab sofort auch für die Auswahl im Schritt ***Select connected stages*** zur Verfügung.

Positionierertyp in Positioniererdatenbank ändern

1. Wählen Sie im Hauptfenster von PIMikroMove® den Menüeintrag ***C-863.12 > Select connected stages...***.

Das Fenster ***Start up stages/axes for C-863.12*** öffnet sich, der Schritt ***Select connected stages*** ist aktiv.

2. Wählen Sie im Schritt ***Select connected stages*** einen Positionierertyp aus, den Sie zuvor wie oben beschrieben (S. 241) neu angelegt haben. Gehen Sie bei der Auswahl vor wie in Schritt 2 der Anleitung **Positionierertyp in Positioniererdatenbank anlegen** beschrieben.

3. Führen Sie die Schritte 3 bis 7 aus **Positionierertyp in Positioniererdatenbank anlegen** aus.

4. Speichern Sie im Dialog ***Save Parameters as User Stage Type*** die geänderten Parameterwerte des Positionierertyps:

a) Lassen Sie den Eintrag im Feld ***Parameters of axis*** unverändert.

b) Lassen Sie den Eintrag im Feld ***Save as*** unverändert.

c) Klicken Sie auf ***OK***.

d) Klicken Sie im Dialog ***Stage type already defined*** auf ***Change settings***. Der Dialog ***Save Parameters as User Stage Type*** schließt sich nach kurzer Zeit automatisch.

Die Parameterwerte des Positionierertyps wurden in der Positioniererdatenbank PISTAGES3.DB sowie im Hauptfenster von PIMikroMove® aktualisiert.

9.4 Parameterübersicht

INFORMATION

Der Schreibzugriff auf die Parameter des C-863.12 ist durch Befehlsebenen festgelegt. Nach dem Einschalten oder Neustart des Controllers ist die aktive Befehlsebene immer 0. Auf Befehlsebenen > 1 besteht Schreibzugriff nur für PI-Servicepersonal.

➢ Wenn Sie Probleme mit Parametern der Befehlsebene 2 oder höher haben, wenden Sie sich an den Kundendienst (S. 263).

INFORMATION

Das Kennwort zum Speichern der Parameterwerte im permanenten Speicher lautet 100.

Bedeutung der farblichen Unterlegung in der Parametertabelle:

Farblos:	Der Wert des Parameters kann aus einer Positioniererdatenbank (S. 12) geladen werden.
Hellgrau:	Der Wert des Parameters stammt aus einer der folgenden Quellen: ▪ Werkseinstellung ▪ Vom Controller während der Laufzeit gesetzt und nur lesbar ▪ Vom ID-Chip des Positionierers gelesen (für künftige Verwendung)

Bezeichnungen in der Kopfzeile der nachfolgenden Tabelle:

- ID = Parameter-ID, Hexadezimal-Format
- Typ = Datentyp:
  - INT = ganzzahliger Wert, inkl. Boolesche Werte
  - FLOAT = Gleitkommazahl
  - CHAR = Stringformat
- CL = Befehlsebene (Command Level) für Schreibzugriff

- Element = Elementtyp, auf den sich der Parameter bezieht, weitere Informationen siehe "Kommandierbare Elemente" (S. 17)
- Parametername = Name des Parameters
- Beschreibung = Erläuterungen zum Parameter

ID	Typ	CL	Element	Parametername	Beschreibung
0x8	FLOAT	0	Achse	Maximum Position Error (Phys. Unit)	Maximaler Positionsfehler Wird für die Erkennung von Bewegungsfehlern verwendet. Details siehe "Bewegungsfehler" (S. 75).
0x9	INT	0	Achse	Maximum Motor Output	Maximal zulässiger Betrag des Stellwerts (dimensionslos) Details siehe "Motoransteuerung" (S. 16).
0xA	FLOAT	0	Achse	Maximum Closed-Loop Velocity (Phys. Unit/s)	Maximale Geschwindigkeit im geregelten Betrieb Gibt den Maximalwert für Parameter 0x49 an.
0xB	FLOAT	0	Achse	Closed-Loop Acceleration (Phys. Unit/s^2)	Beschleunigung im geregelten Betrieb Wird begrenzt durch Parameter 0x4A. Details siehe "Erzeugung des Dynamikprofils" (S. 23).
0xC	FLOAT	0	Achse	Closed-Loop Deceleration (Phys. Unit/s^2)	Abbremsung im geregelten Betrieb Wird begrenzt durch Parameter 0x4B. Details siehe "Erzeugung des Dynamikprofils" (S. 23).
0xE	INT	0	Achse	Numerator Of The Counts-Per-Physical-Unit Factor	Zähler des Faktors für Impulse pro physikalische Längeneinheit Details siehe "Physikalische Einheiten" (S. 21).
0xF	INT	0	Achse	Denominator Of The Counts-Per-Physical-Unit Factor	Nenner des Faktors für Impulse pro physikalische Längeneinheit Details siehe "Physikalische Einheiten" (S. 21).
0x13	INT	0	Achse	Is Rotary Stage?	Handelt es sich um einen Rotationstisch? 0 = Kein Rotationstisch 1 = Rotationstisch Keine Auswertung durch den C-863.12, sondern nur durch die PC-Software: PIMikroMove® entscheidet anhand dieses Wertes, welche

ID	Typ	CL	Element	Parametername	Beschreibung
					Bewegungen zulässig sind.
0x14	INT	0	Achse	Has Reference?	Hat der Positionierer einen Referenzschalter? Details siehe "Referenzschaltererkennung" (S. 29).
0x15	FLOAT	0	Achse	Maximum Travel In Positive Direction (Phys. Unit)	Verfahrbereichsgrenze in positiver Richtung Siehe Beispiele in "Stellweg und Verfahrbereichsgrenzen" (S. 32).
0x16	FLOAT	0	Achse	Value At Reference Position (Phys. Unit)	Positionswert am Referenzschalter Siehe Beispiele in "Stellweg und Verfahrbereichsgrenzen" (S. 32).
0x17	FLOAT	0	Achse	Distance From Negative Limit To Reference Position (Phys. Unit)	Abstand zwischen Referenzschalter und negativem Endschalter Siehe Beispiele in "Stellweg und Verfahrbereichsgrenzen" (S. 32).
0x18	INT	0	Achse	Limit Mode	Signallogik der Endschalter Details siehe "Endschaltererkennung" (S. 30).
0x1A	INT	0	Achse	Has Brake?	Hat der Positionierer eine Bremse? 0 = Keine Bremse vorhanden 1 = Bremse vorhanden. In diesem Fall sind das Ein-/Ausschalten des Servomodus und die Bremsenaktivierung/-deaktivierung miteinander verkoppelt, siehe `BRA` (S. 136) und `SVO` (S. 198). Die Bremse wird über die Buchse **Motor (S. 269)** angesteuert: ▪ Pin 15, wenn der Bremsentreiber im Positionierer integriert ist ▪ Pins 9 und 16, wenn der integrierte Bremsentreiber des C-863.12 verwendet wird. Die Konfiguration erfolgt mit den Parametern 0x3094, 0x3095, 0x3096.
0x2F	FLOAT	0	Achse	Distance From Reference Position To Positive Limit (Phys. Unit)	Abstand zwischen Referenzschalter und positivem Endschalter Siehe Beispiele in "Stellweg und Verfahrbereichsgrenzen" (S. 32).

ID	Typ	CL	Element	Parametername	Beschreibung
0x30	FLOAT	0	Achse	Maximum Travel In Negative Direction (Phys. Unit)	Verfahrbereichsgrenze in negativer Richtung Siehe Beispiele in "Stellweg und Verfahrbereichsgrenzen" (S. 32).
0x31	INT	0	Achse	Invert Reference?	Soll das Referenzsignal invertiert werden? Details siehe "Referenzschaltererkennung" (S. 29).
0x32	INT	0	Achse	Has No Limit Switches?	Hat der Positionierer Endschalter? Details siehe "Endschaltererkennung" (S. 30).
0x33	INT	0	Achse	Motor Offset Positive	Offset für die positive Bewegungsrichtung Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x34	INT	0	Achse	Motor Offset Negative	Offset für die negative Bewegungsrichtung Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x36	INT	0	Achse	Settling Window (encoder counts)	Einschwingfenster um die Zielposition Details siehe "On-Target-Status" (S. 29).
0x3C	CHAR	0	Achse	Stage Name	Positionierername Maximal 20 Zeichen; Standardwert: DEFAULT_STAGE
0x3F	FLOAT	0	Achse	Settling Time (s)	Verzögerungszeit für das Setzen des On-Target-Status. Details siehe "On-Target-Status" (S. 29).
0x47	INT	0	Achse	Reference Travel Direction	Standardrichtung für die Referenzierungsfahrt Details siehe "Referenzierung" (S. 35).
0x48	INT	0	Achse	Motor Drive Offset	Geschwindigkeitsabhängiger Offset Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x49	FLOAT	0	Achse	Closed-Loop Velocity (Phys. Unit/s)	Geschwindigkeit im geregelten Betrieb Wird begrenzt durch Parameter 0xA Details siehe "Erzeugung des Dynamikprofils" (S. 23).

ID	Typ	CL	Element	Parametername	Beschreibung
0x4A	FLOAT	0	Achse	Maximum Closed-Loop Acceleration (Phys. Unit/s^2)	Maximale Beschleunigung im geregelten Betrieb Gibt den Höchstwert für Parameter 0xB an.
0x4B	FLOAT	0	Achse	Maximum Closed-Loop Deceleration (Phys. Unit/s^2)	Maximale Abbremsung im geregelten Betrieb Gibt den Höchstwert für Parameter 0xC an.
0x50	FLOAT	0	Achse	Velocity For Reference Moves (Phys. Unit/s)	Geschwindigkeit für Referenzierungsfahrt Details siehe "Referenzierung" (S. 35).
0x5A	INT	0	Achse	Numerator Of The Servo-Loop Input Factor	Zähler des Eingangsfaktors des Regelkreises Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x5B	INT	0	Achse	Denominator Of The Servo-Loop Input Factor	Nenner des Eingangsfaktors des Regelkreises Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x5C	INT	0	Achse	Source Of Reference Signal	Referenzsignalquelle für die Befehle `FRF` oder `FED` Details siehe "Befehle und Parameter für digitale Eingänge" (S. 87) und "Digitale Eingangssignale als Schaltersignale verwenden" (S. 89).
0x5D	INT	0	Achse	Source Of Negative Limit Signal	Referenzsignalquelle für die Befehle `FNL` oder `FED` Details siehe "Befehle und Parameter für digitale Eingänge" (S. 87) und "Digitale Eingangssignale als Schaltersignale verwenden" (S. 89).
0x5E	INT	0	Achse	Source Of Positive Limit Signal	Referenzsignalquelle für die Befehle `FPL` oder `FED` Details siehe "Befehle und Parameter für digitale Eingänge" (S. 87) und "Digitale Eingangssignale als Schaltersignale verwenden" (S. 89).
0x5F	INT	0	Achse	Invert Digital Input Used For Negative Limit	Invertiert die Polarität der digitalen Eingänge, die als Quellen des negativen Endschaltersignals dienen. Details siehe "Befehle und Parameter für digitale Eingänge" (S. 87) und "Digitale Eingangssignale als

ID	Typ	CL	Element	Parametername	Beschreibung
					Schaltersignale verwenden" (S. 89).
0x60	INT	0	Achse	Invert Digital Input Used For Positive Limit	Invertiert die Polarität der digitalen Eingänge, die als Quellen des positiven Endschaltersignals dienen. Details siehe "Befehle und Parameter für digitale Eingänge" (S. 87) und "Digitale Eingangssignale als Schaltersignale verwenden" (S. 89).
0x61	INT	0	Achse	Invert Direction Of Motion For Joystick-Controlled Axis?	Soll die Bewegungsrichtung für joystickgesteuerte Achsen invertiert werden? Details siehe "Befehle und Parameter für Joystick-Steuerung" (S. 93).
0x63	FLOAT	0	Achse	Distance Between Limit And Hard Stop (Phys. Unit)	Abstand zwischen eingebautem Endschalter und mechanischem Anschlag Details siehe "Referenzierung" (S. 35).
0x70	INT	0	Achse	Reference Signal Type	Art des Referenzsignals Details siehe "Referenzschaltererkennung" (S. 29).
0x71	INT	0	Achse	D-Term Delay (No. Of Servo Cycles)	D-Term-Verzögerung Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x72	INT	0	System	Ignore Macro Error?	Makrofehler ignorieren? Details siehe "Befehle und Parameter für Makros" (S. 100).
0x77	INT	0	Achse	Use Limit Switches Only For Reference Moves?	Sollen die Endschalter nur für Referenzierungsfahrten verwendet werden? Details siehe "Endschaltererkennung" (S. 30).
0x78	FLOAT	0	Achse	Distance From Limit To Start Of Ref. Search (Phys. Unit)	Abstand zwischen Endschalter oder mechanischem Anschlag und der Startposition für die Referenzierungsfahrt zum Indexpuls Details siehe "Referenzierung" (S. 35).
0x79	FLOAT	0	Achse	Distance For Reference Search (Phys. Unit)	Maximale Strecke für die Referenzierungsfahrt zum Indexpuls Details siehe "Referenzierung" (S. 35).
0x7C	FLOAT	0	Achse	Maximum Motor Output (V)	Maximal zulässige Betriebsspannung des Motors. Details siehe "Motoransteuerung" (S.

ID	Typ	CL	Element	Parametername	Beschreibung
					16).
0x94	FLOAT	0	Achse	Notch Filter Frequency 1 (Hz)	Frequenz des ersten Kerbfilters Details siehe "Regelalgorithmus und itere Stellwertkorrekturen" (S. 26).
0x95	FLOAT	0	Achse	Notch Filter Edge 1	Anstieg der Flanke des ersten Kerbfilters Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x3094	INT	0	Achse	Internal Brake	Integrierten Bremsentreiber des C-863.12 verwenden? 0 = Bremsentreiber nicht verwenden 1 = Bremsentreiber verwenden. Die Bremse wird aktiviert (geschlossen), wenn die Versorgungsspannung den Wert des Parameters 0x3096 unterschreitet. Außerdem sind das Ein-/Ausschalten des Servomodus und die Bremsenaktivierung/-deaktivierung miteinander verkoppelt, siehe `BRA` (S. 136) und `SVO` (S. 198). Die Einstellung wird nur wirksam, wenn Parameter 0x1A den Wert 1 hat ("Bremse vorhanden"). Die Signale des Bremsentreibers werden an den Pins 9 und 16 der Buchse **Motor (S. 269)** ausgegeben.
0x3095	FLOAT	0	Achse	Brake Activation Voltage (V)	Versorgungsspannung für das Lösen der Bremse 0 bis 48 V Wird nur verwendet, wenn Parameter 0x1A und 0x3094 jeweils den Wert 1 haben.
0x3096	FLOAT	0	Achse	Brake Continuous Voltage (V)	Versorgungsspannung, bei der die Bremse dauerhaft gelöst bleibt 0 bis 48 V Sollte kleiner sein als der Wert von Parameter 0x3095. Für möglichst geringe Wärmeentwicklung sollte die dauerhafte Versorgungsspannung der Bremse so niedrig wie möglich sein. Wird nur verwendet, wenn Parameter 0x1A und 0x3094 jeweils den Wert 1 haben.

ID	Typ	CL	Element	Parametername	Beschreibung
0x411	INT	0	Achse	P-Term 1	0 bis 32767; Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x412	INT	0	Achse	I-Term 1	0 bis 32767; Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x413	INT	0	Achse	D-Term 1	0 bis 32767; Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x414	INT	0	Achse	I-Limit 1	0 bis 32767; Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x415	INT	0	Achse	Kvff 1	0 bis 32767; Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x03003900	INT	0	Achse	Quadrature Encoder Filter	Filterung des Encodersignals zur Reduktion von Störungen? 0 - aus (Standardeinstellung) 1 - für schnelle Bewegungen 2 - für mittelschnelle Bewegungen 3 - für langsame Bewegungen Details siehe "Regelalgorithmus und weitere Stellwertkorrekturen" (S. 26).
0x07000601	CHAR	0	Achse	Axis Unit	Einheitenzeichen Details siehe "Physikalische Einheiten" (S. 21).
0x0D000000	CHAR	2	System	Controller S/N	Seriennummer des C-863.12 Details siehe "Typenschild" (S. 9).
0x0E000200	FLOAT	2	System	Servo Update Time	Servo-Zykluszeit in Sekunden
0x0F000100	CHAR	2	Achse	Stage Type	Positionierertyp Form für Standardpositionierer: x-xxx Form für kundenspezifische Positionierer: x-xxxKxxx
0x0F000200	CHAR	2	Achse	Stage Serial Number	Seriennummer des Positionierers 9-stellige Nummer
0x0F000300	CHAR	2	Achse	Stage Assembly Date	Herstellungsdatum des Positionierers Datumsformat: TTMMJJ
0x0F000400	INT	2	Achse	Stage HW Version	Versionsnummer der Positioniererhardware

10 Wartung

10.1 C-863.12 reinigen

HINWEIS

Kurzschlüsse oder Überschläge!
Der C-863.12 enthält elektrostatisch gefährdete Bauteile, die beim Eindringen von Reinigungsflüssigkeiten in das Gehäuse durch Kurzschlüsse oder Überschläge beschädigt werden können.

- Trennen Sie vor dem Reinigen den C-863.12 von der Stromversorgung, indem Sie den Netzstecker ziehen.
- Vermeiden Sie das Eindringen von Reinigungsflüssigkeit in das Gehäuse.

- Wenn notwendig, reinigen Sie die Gehäuseoberflächen des C-863.12 mit einem Tuch, das leicht mit einem milden Reinigungs- oder Desinfektionsmittel angefeuchtet wurde.

10.2 Firmware aktualisieren

HINWEIS

Funktionsstörung durch fehlerhafte Firmwareaktualisierung!
Eine falsch oder unvollständig ausgeführte Aktualisierung der Firmware des C-863.12 kann dazu führen, dass der C-863.12 nur durch den PI Kundendienst wieder in Betriebsbereitschaft versetzt werden kann.

- Aktualisieren Sie die Firmware des C-863.12 nur mit Zustimmung des PI Kundendienstes. Wenn möglich, lassen Sie die Firmwareaktualisierung durch den PI Kundendienst vornehmen.
- Stellen Sie vor dem Start der Firmwareaktualisierung sicher, dass Sie vom PI Kundendienst eine passende Firmware erhalten und an einem Ort abgelegt haben, der für das Update-Programm zugänglich ist.
- Schalten Sie den C-863.12 während der Firmwareaktualisierung **nicht** aus.

INFORMATION

Wenn sich der C-863.12 im Firmware-Update-Modus befindet, blinkt die LED **STA**. Der C-863.12 verlässt den Firmware-Update-Modus erst, wenn er nach einer **erfolgreichen** Aktualisierung der Firmware **neu gestartet** wird. Wenn die Aktualisierung der Firmware nicht erfolgreich war oder abgebrochen wurde, bleibt der C-863.12 nach einem Neustart im Firmware-Update-

Modus.

Wenn die LED **STA** noch blinkt, obwohl der C-863.12 nach der Firmware-Aktualisierung neu gestartet wurde:

- ➢ Wiederholen Sie die Aktualisierung der Firmware.
- ➢ Wenn die erneute Aktualisierung der Firmware fehlschlägt, kontaktieren Sie unseren Kundendienst (S. 263).

INFORMATION

Wenn mit der Aktualisierung der Firmware neue Parameter eingeführt werden oder das Speichermanagement des C-863.12 verändert wird, ist nach der Aktualisierung der Firmware eine Initialisierung des C-863.12 erforderlich.

Voraussetzungen

- ✓ Sie haben den C-863.12 über die USB- oder die RS-232-Schnittstelle an den PC angeschlossen (S. 46).
- ✓ Sie haben sichergestellt, dass der C-863.12 **nicht** Bestandteil eines Daisy-Chain-Netzwerks ist.
- ✓ Sie haben sichergestellt, dass **kein** Kabel an der Buchse **RS-232 Out** angeschlossen ist.
- ✓ Das Programm ***PIFirmwareManager*** ist auf dem PC installiert (S. 41).
- ✓ Sie haben die neue Firmwaredatei, die Sie von unserem Kundendienst erhalten haben, in ein Verzeichnis auf dem PC kopiert.
- ✓ Sie haben die Dokumentation gelesen und verstanden, die Sie mit der neuen Firmware von unserem Kundendienst erhalten haben. Der Dokumentation haben Sie entnommen, ob mit der Aktualisierung der Firmware neue Parameter eingeführt werden oder das Speichermanagement des C-863.12 verändert wird.
- ✓ Sie haben die Parameterwerte des C-863.12 in einer Textdatei auf dem PC gesichert (S. 237).
- ✓ Sie haben die Controllermakros des C-863.12 in Dateien auf dem PC gesichert (S. 110).
- ✓ Sie haben die Kommunikation zwischen dem C-863.12 und dem PC mit PIMikroMove® oder PITerminal hergestellt (S. 57).

Firmware des C-863.12 aktualisieren

- ➢ Starten Sie am PC das Programm ***PIFirmwareManager*** und führen Sie die Aktualisierung der Firmware des Controllers durch.

  Gehen Sie dazu vor wie im Benutzerhandbuch SM164E (S. 3) beschrieben.

Neustart des C-863.12 durchführen

1. Schalten Sie den C-863.12 aus.
2. Schalten Sie den C-863.12 wieder ein.

Wenn die Aktualisierung der Firmware erfolgreich war, hat der C-863.12 den Firmware-Update-Modus verlassen, und die LED **STA** leuchtet dauerhaft.

Wurden mit der Aktualisierung der Firmware neue Parameter eingeführt, oder wurde das Speichermanagement des C-863.12 verändert?

- Wenn nein: Die Aktualisierung der Firmware ist beendet.
- Wenn ja: Eine Initialisierung des C-863.12 ist erforderlich, siehe unten.

C-863.12 nach Firmware-Aktualisierung initialisieren

Die Initialisierung des C-863.12 setzt **alle** Parameter auf ihre Werkseinstellung zurück und löscht alle Controllermakros. Ungesicherte Parameterwerte und Controllermakros gehen somit bei der Initialisierung verloren.

1. Stellen Sie sicher, dass die aktuellen Parameterwerte und Controllermakros des C-863.12 auf dem PC gesichert wurden.
2. Starten Sie am PC PITerminal oder PIMikroMove®, stellen Sie die Verbindung zum C-863.12 her, und öffnen Sie gegebenenfalls das Fenster zum Senden von Befehlen.
3. Initialisieren Sie den C-863.12, indem Sie nacheinander folgende Befehle senden:

   `ZZZ 100 parameter`

   `ZZZ 100 macros`

   Nach erfolgreicher Initialisierung gibt der Controller jeweils eine entsprechende Meldung aus.
4. Passen Sie die Parameterwerte des C-863.12 an.

   Eine Anleitung zum allgemeinen Vorgehen finden Sie in "Parameterwerte ändern: Generelle Vorgehensweise" (S. 238).

   – Setzen Sie die Parameter, die bereits vor der Firmware-Aktualisierung vorhanden waren, auf die gesicherten Werte aus der Textdatei zurück.

   – Setzen Sie die Parameter, die mit der Aktualisierung der Firmware eingeführt wurden, auf geeignete Werte.
5. Wenn Sie Controllermakros auf dem PC gesichert haben: Laden Sie die Controllermakros zurück in den C-863.12, siehe "Controllermakros sichern und laden" (S. 110).

11 Störungsbehebung

Störung: Positionierer bewegt sich nicht	
Mögliche Ursachen	**Maßnahmen zur Behebung**
Kabel nicht korrekt angeschlossen	➢ Prüfen Sie die Kabelanschlüsse.
Positionierer oder Positioniererkabel defekt	➢ Wenn vorhanden, tauschen Sie den defekten Positionierer gegen einen anderen Positionierer aus und testen Sie die neue Kombination.
Ungeeignetes Positioniererkabel verwendet	Bei der Verwendung ungeeigneter Kabel können Störungen in der Signalübertragung zwischen Positionierer und C-863.12 auftreten. ➢ Wenn Positionierer, Kabel und C-863.12 als zusammengehörendes System gekennzeichnet sind, tauschen Sie die Systemkomponenten nur nach Rücksprache mit PI gegen andere Komponenten aus.
Positionierer nicht an Netzteil angeschlossen	Positionierer mit integriertem PWM-Verstärker werden über ein separates Netzteil versorgt. ➢ Wenn der Positionierer über einen integrierten PWM-Verstärker verfügt, schließen Sie ihn an ein geeignetes Netzteil an. ➢ Um die optimale Motorleistung zu erreichen, verwenden Sie für den C-863.12 ein Netzteil, das die gleiche Ausgangsspannung liefert wie das Netzteil des PWM-Verstärkers. ➢ Stellen Sie sicher, dass das Netzteil ordnungsgemäß arbeitet.
Signallogik der Endschalter falsch eingestellt	Damit sich der Positionierer bewegen kann, müssen die Einstellungen des C-863.12 mit dem Endschalter-Logikpegel des Positionierer übereinstimmen; siehe "Endschaltererkennung" (S. 30). ➢ Stellen Sie den Parameter ***Limit Mode*** (0x18) passend ein.
Endschaltersignale nicht kompatibel mit dem C-863.12	Positionierer von Drittanbietern verwenden möglicherweise ungeeignete Endschaltersignale. ➢ Wenden Sie sich an den Kundendienst (S. 263) und an den Hersteller des Positionierers.
Falsche Konfiguration	➢ Prüfen Sie die Parametereinstellungen des C-863.12 mit den Befehlen `SPA?` (flüchtiger Speicher) und `SEP?` (permanenter Speicher); Details siehe "Anpassen von Einstellungen" (S. 235).

Störung: Positionierer bewegt sich nicht	
Mögliche Ursachen	**Maßnahmen zur Behebung**
Falscher Befehl oder falsche Syntax	➢ Senden Sie den Befehl `ERR?` und prüfen Sie den zurückgemeldeten Fehlercode.
Falsche Achse kommandiert	➢ Stellen Sie sicher, dass die richtige Achsenkennung verwendet wird und dass die kommandierte Achse zum richtigen Positionierer gehört.
Joystick-Steuerung ist aktiv	Bewegungsbefehle sind nicht zulässig, wenn ein Joystick für die Achse aktiv ist. ➢ Deaktivieren Sie den Joystick mit dem Befehl `JON` (S. 171).

Störung: Positionierer führt unbeabsichtigte Bewegung aus	
Mögliche Ursachen	**Maßnahmen zur Behebung**
Joystick nicht angeschlossen, aber im C-863.12 aktiviert	➢ Aktivieren Sie den Joystick in der Software nur, wenn tatsächlich ein Joystick an den C-863.12 angeschlossen ist (S. 94).
Joystick nicht kalibriert	➢ Kalibrieren Sie den Joystick (S. 95).
Startup-Makro wird ausgeführt	➢ Prüfen Sie, ob ein Makro als Startup-Makro festgelegt ist, und heben Sie die Auswahl des Startup-Makros gegebenenfalls auf (S. 102).
Bremse des Positionierers bei ausgeschaltetem Servomodus mit dem Befehl `BRA` (S. 136) deaktiviert	➢ Sichern Sie den Positionierer gegen unbeabsichtigte Bewegungen, bevor Sie die Bremse per Befehl deaktivieren!

Störung: Positionierer schwingt oder wird ungenau positioniert	
Mögliche Ursachen	**Maßnahmen zur Behebung**
Die Last wurde geändert.	➢ Stellen Sie das System der Laständerung entsprechend neu ein (S. 69).
Störung des Encodersignals	Bei deaktivierter Filterung des Encodersignals (S. 28): Wenn große Störungen auf das Signal wirken, kann es vorkommen, dass der Encoder zu viele Takte zählt oder die Flanken nicht mehr genau erkennt. ➢ Setzen Sie den Parameter ***Quadrature Encoder Filter*** (0x03003900) auf einen Wert > 0, um den Filter für das Encodersignal zu aktivieren. ➢ Führen Sie eine erneute Referenzierung (S. 35) aus. Bei aktivierter Filterung des Encodersignals: Wenn zu hohe Geschwindigkeiten gefahren werden, kann es vorkommen,

Störung: Positionierer schwingt oder wird ungenau positioniert	
Mögliche Ursachen	**Maßnahmen zur Behebung**
	dass der Encoder nicht alle Flanken erkennt. ➢ Setzen Sie ggf. den Parameter ***Quadrature Encoder Filter*** (0x03003900) auf einen kleineren Wert. ➢ Führen Sie eine erneute Referenzierung (S. 35) aus.

Störung: Positionierer schwingt bereits während der Referenzierungsfahrt	
Mögliche Ursachen	**Maßnahmen zur Behebung**
Sehr hohe Last auf dem Positionierer	Gehen Sie bei einer sehr hohen Last während der Referenzierungsfahrt mit PIMikroMove® wie folgt vor: 1. Starten Sie die Referenzierungsfahrt **nicht** im Schritt ***Start up axes***, sondern klicken Sie auf ***Close***, um das Fenster ***Start up controller*** zu schließen. 2. Öffnen Sie im Hauptfenster das Einzelachsen-Fenster für den angeschlossenen Positionierer, indem Sie den Positionierer im Menü ***View > Single Axis Window*** auswählen. 3. Erweitern Sie die Ansicht des Einzelachsen-Fensters durch Anklicken der Schaltfläche **>** am rechten Fensterrand. 4. Stellen Sie mit dem Kontrollkästchen ***Servo*** sicher, dass der Servomodus eingeschaltet ist. 5. Starten Sie die Referenzierungsfahrt durch Klicken auf eine der Schaltflächen ***Reference…***. 6. Wenn der Positionierer schwingt: Halten Sie die Referenzierungsfahrt im Dialog ***Reference Axes*** sofort an, schließen Sie den Dialog und schalten Sie den Servomodus aus, indem Sie den Haken aus dem entsprechenden Kontrollkästchen im Einzelachsen-Fenster entfernen. 7. Geben Sie neue Werte für die Regelparameter ein, siehe "Regelparameter optimieren" (S. 69). 8. Starten Sie die Referenzierungsfahrt erneut. 9. Wenn der Positionierer immer noch schwingt, wiederholen Sie die Schritte 6 bis 8 so lange, bis die Referenzierungsfahrt ohne Schwingungen erfolgreich

Störung: Positionierer schwingt bereits während der Referenzierungsfahrt	
Mögliche Ursachen	**Maßnahmen zur Behebung**
	beendet wird.

Störung: Kommunikation zwischen Controller und PC funktioniert nicht	
Mögliche Ursachen	**Maßnahmen zur Behebung**
Falsches Kommunikationskabel wird verwendet oder es ist defekt	➢ Verwenden Sie für die RS-232-Verbindung ein Nullmodemkabel. ➢ Prüfen Sie gegebenenfalls, ob das Kabel an einem fehlerfreien System funktioniert.
Baudrate nicht richtig konfiguriert	➢ Überprüfen Sie die Einstellungen der DIP-Schalter 5 und 6 für die Baudrate (S. 55). ➢ Stellen Sie in einem Daisy-Chain-Netzwerk sicher, dass für alle Controller die gleiche Baudrate eingestellt ist.
Controlleradresse nicht richtig konfiguriert	➢ Überprüfen Sie die Einstellungen der DIP-Schalter 1 bis 4 für die Controlleradresse (S. 54).
Ein anderes Programm greift auf die Schnittstelle zu.	➢ Schließen Sie das andere Programm.
Probleme mit spezieller Software	➢ Prüfen Sie, ob das System mit einer anderen Software, wie z. B. einem Terminal-Programm oder einer Entwicklungsumgebung, funktioniert. Sie können die Kommunikation testen, indem Sie ein Terminal-Programm (z. B. PITerminal) starten und `*IDN?` oder `HLP?` eingeben. ➢ Achten Sie darauf, dass Sie Befehle mit einem LF (line feed) abschließen. Ein Befehl wird erst ausgeführt, wenn der LF empfangen wurde.

Störung: Kunden-Software funktioniert nicht mit den PI-Treibern	
Mögliche Ursachen	**Maßnahmen zur Behebung**
Falsche Kombination der Treiberroutinen/VIs	➢ Prüfen Sie, ob das System mit einem Terminal-Programm (z. B. PITerminal) funktioniert. Wenn ja: ➢ Lesen Sie die Angaben im zugehörigen Softwarehandbuch und vergleichen Sie Ihren Programmcode mit dem Beispielcode auf dem Datenträger mit der PI Software Suite.

Wenn die Störung Ihres Systems nicht in der Liste angeführt ist oder wenn sie nicht wie beschrieben behoben werden kann, kontaktieren Sie unseren Kundendienst (S. 263).

12 Kundendienst

Wenden Sie sich bei Fragen und Bestellungen an Ihre PI Vertretung oder schreiben Sie uns eine E-Mail (mailto:service@pi.de).

- Geben Sie bei Fragen zu Ihrem System folgende Systeminformationen an:
  - Produkt- und Seriennummern von allen Produkten im System
  - Firmwareversion des Controllers (sofern vorhanden)
  - Version des Treibers oder der Software (sofern vorhanden)
  - PC-Betriebssystem (sofern vorhanden)
- Wenn möglich: Fertigen Sie Fotografien oder Videoaufnahmen Ihres Systems an, die Sie unserem Kundendienst auf Anfrage senden können.

13 Technische Daten

Änderungen vorbehalten. Die aktuellen Produktspezifikationen finden Sie auf der Seite des Produkts unter www.pi.de (https://www.physikinstrumente.de/de/).

13.1 Spezifikationen

13.1.1 Datentabelle

	C-863.12
Funktion	DC-Motorsteuerung
Antriebsarten	DC-Motor, servogeregelt Motoren mit PWM-Ansteuerung, z. B. ActiveDrive-Verstärker oder bürstenlose Motoren mit integrierter Blockkommutierung
Achsen	1
Unterstützte Funktionen	Punkt-zu-Punkt-Bewegung. Startup-Makro. Datenrekorder zur Aufnahme von Betriebsgrößen wie Motorspannung, Geschwindigkeit, Position oder Positionsfehler. ID-Chip-Erkennung. Interne Sicherheitsschaltung: Watchdog Timer.

Bewegung und Regler	C-863.12
Reglertyp	PID-Regler, Parameteränderung im Betrieb
Servozykluszeit	50 µs
Profilgenerator	Trapezförmiges Geschwindigkeitsprofil
Encodereingang	A/B-Quadratur TTL-Pegel, einseitig geerdet oder differenziell gem. RS-422; 60 MHz
Blockiererkennung	Automatischer Motorstopp bei Überschreitung eines programmierbaren Positionsfehlers
Endschalter	2 × TTL (Polarität programmierbar)
Referenzschalter	1 × TTL
Motorbremse	1 × TTL, per Software schaltbar

Elektrische Eigenschaften	C-863.12
Max. Ausgangsspannung*	0 V bis Betriebsspannung, für direkte Ansteuerung von DC-Motoren
Max. Ausgangsleistung	60 W
Dauerausgangsleistung	48 W

Elektrische Eigenschaften	C-863.12
Leistungsaufnahme Volllast	48 W
Leistungsaufnahme Leerlauf	3 W
Strombegrenzung	2,5 A

Schnittstellen und Bedienung	C-863.12
Kommunikations-Schnittstellen	USB; RS-232, D-Sub 9 (m)
Motoranschluss	HD D-Sub 26 (f)
Controllernetzwerk	Bis zu 16 Einheiten** an einer Schnittstelle
I/O-Leitungen	4 analoge / digitale Eingänge (0 bis 5 V / TTL), 4 digitale Ausgänge (TTL)
Befehlssatz	PI General Command Set (GCS)
Bedienersoftware	PIMikroMove®
Schnittstellen zur Anwendungsprogrammierung	C, C++, C#, MATLAB, NI LabVIEW, Python
Manuelle Bedienhilfe	Joystick, Y-Kabel für 2D-Bewegungen, Pushbutton-Box

Umgebung	C-863.12
Betriebsspannung	12 bis 48 V DC *** von externem Netzteil (Netzteil 24 V DC im Lieferumfang)
Max. Stromaufnahme	40 mA im Leerlauf (bei Versorgung mit 48 V) 80 mA im Leerlauf (bei Versorgung mit 24 V)
Betriebstemperaturbereich	5 bis 50 °C (Temperaturschutz schaltet bei zu hohen Temperaturen ab)
Masse	0,48 kg
Abmessungen	130 mm × 76 mm × 40 mm (inkl. Montageschienen)

* Die Ausgangsspannung ist abhängig vom verwendeten Netzteil.

** 16 Einheiten über USB; 6 Einheiten über RS-232.

*** Empfohlene Betriebsspannung: 24 bis 48 V DC

13.1.2 Bemessungsdaten

Der C-863.12 ist für folgende Betriebsgrößen ausgelegt:

Eingang an:	**Maximale Betriebsspannung** ⚠	**Betriebsfrequenz** ⚠	**Maximale Stromaufnahme** ⚠
Mini-DIN 4-pol. (f)	48 V	⎓	3 A

Ausgang an:	**Maximale Ausgangsspannung** ⚠	**Maximaler Ausgangsstrom** ⚠	**Maximale Ausgabefrequenz** ⚠
HD D-Sub 26 (f)	48 V	2,5 A	20 kHz (PWM)

13.1.3 Umgebungsbedingungen und Klassifizierungen

Folgende Umgebungsbedingungen und Klassifizierungen sind für den C-863.12 zu beachten:

Einsatzbereich	Nur zur Verwendung in Innenräumen
Maximale Höhe	2000 m
Luftdruck	1100 hPa bis 0,1 hPa
Relative Luftfeuchte	Höchste relative Luftfeuchte 80 % für Temperaturen bis 31 °C Linear abnehmend bis 50 % relativer Luftfeuchte bei 40 °C
Lagertemperatur	0 °C bis 70 °C
Transporttemperatur	–25 °C bis +85 °C
Überspannungskategorie	II
Schutzklasse	I
Verschmutzungsgrad	2
Schutzart gemäß IEC 60529	IP20

13.2 Abmessungen

Abmessungen in mm.

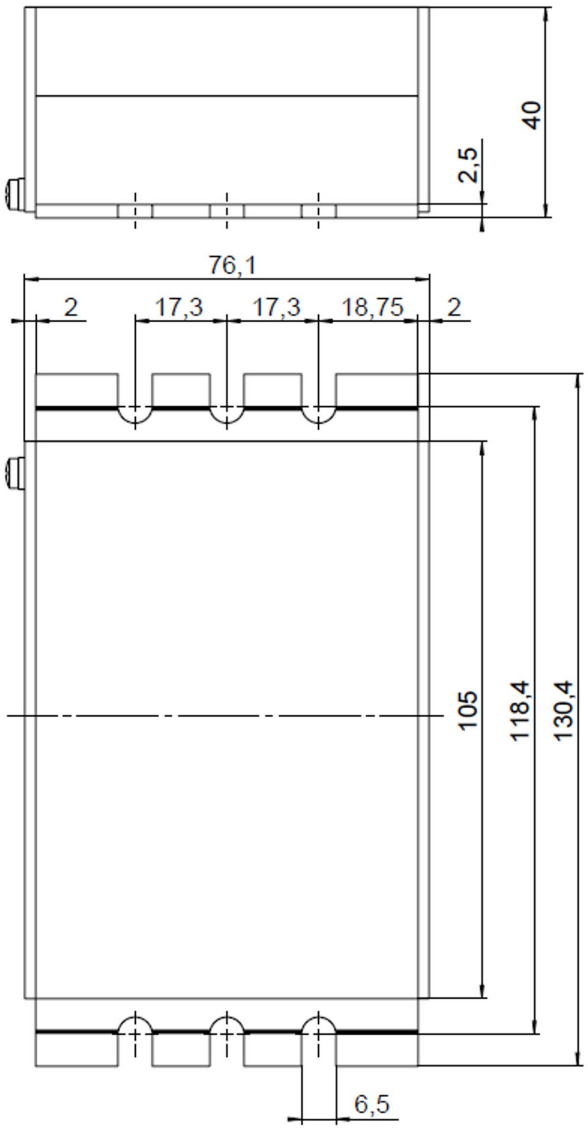
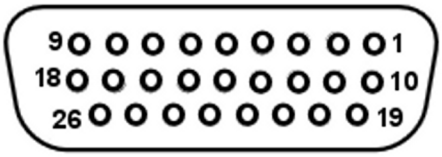


Abbildung 25: C-863.12, Abmessungen in mm

13.3 Pinbelegung

13.3.1 Motor

HD D-Sub 26 (w)



Pin	Signal	Richtung	Funktion
1	OUT0	Ausgang	Motor + (differentiell; Power PWM); für Positionierer ohne PWM-Verstärker
2	OUT0	Ausgang	Motor + (differentiell; Power PWM); für Positionierer ohne PWM-Verstärker
3	OUT1	Ausgang	Motor – (differentiell; Power PWM); für Positionierer ohne PWM-Verstärker
4	OUT1	Ausgang	Motor – (differentiell; Power PWM); für Positionierer ohne PWM-Verstärker
5	-	-	reserviert
6	-	-	reserviert
7	-	-	reserviert
8	-	-	reserviert
9	BRAKE_OUT	Ausgang	Treiber Motorbremse
10	REF	Eingang	Referenzschalter (5 V TTL Eingang, single-ended)
11	NLIM	Eingang	Negativer Endschalter (5 V TTL Eingang)
12	PLIM	Eingang	Positiver Endschalter (5 V TTL Eingang)
13	PWM-SIGNE	Ausgang	PWM Sign (TTL); für Positionierer mit PWM-Verstärker
14	PWM-MAGE	Ausgang	PWM Magnitude (TTL); für Positionierer mit PWM-Verstärker
15	BRAKEE	Ausgang	Motorbremse 5 V TTL, für Positionierer mit integriertem Bremsentreiber
16	VB_HC	Ausgang	Treiber Motorbremse (Versorgung 0 bis 48 V)
17	ID-Chip	Bidirektional	ID-Chip (vorgesehen für künftige Verwendung)
18	VCC_ENC	Ausgang	Versorgung Positionssensor (5 V, 200 mA)
19	ENCA+	Eingang	Encoder-Eingang A+ (RS-422)
20	ENCA-	Eingang	Encoder-Eingang A- (RS-422)

Pin	Signal	Richtung	Funktion
21	ENCB+	Eingang	Encoder-Eingang B+ (RS-422)
22	ENCB-	Eingang	Encoder-Eingang B- (RS-422)
23	INDEX+	Eingang	Referenzschalter, differenziell
24	INDEX-	Eingang	Referenzschalter, differenziell
25	GND		GND
26	VCC_ENC	Ausgang	Versorgung Positionssensor (5 V, 200 mA)

An reservierte Pins darf nichts angeschlossen werden.

13.3.2 I/O

Mini-DIN-Buchse, 9-polig, weiblich

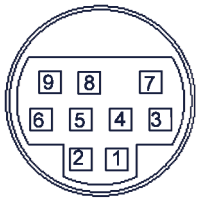


Abbildung 26: Vorderansicht der Mini-DIN-Buchse

Pin	Funktion
1	Eingang 1 (analog: 0 bis +5V/ digital: TTL)
2	Eingang 2 (analog: 0 bis +5V/ digital: TTL)
3	Eingang 3 (analog: 0 bis +5V/ digital: TTL)
4	Eingang 4 (analog: 0 bis +5V/ digital: TTL)
5	Ausgang 1 (digital: TTL)
6	Ausgang 2 (digital: TTL)
7	Ausgang 3 (digital: TTL)
8	Ausgang 4 (digital: TTL)
9	Vcc (+5 V)
Schirm	GND

13.3.3 Kabel C-170.IO für den Anschluss an die I/O-Buchse

Mini-DIN-Stecker, 9-polig, männlich, offenes Ende

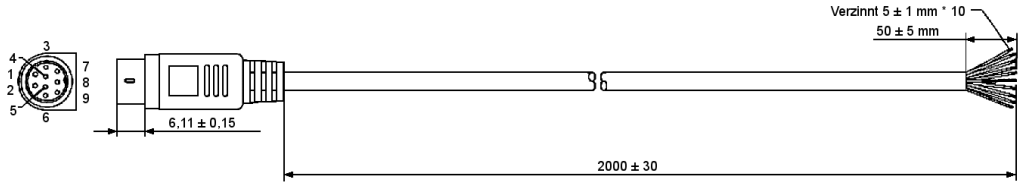


Abbildung 27: C-170.IO-Kabel

Pin	Adernfarbe	Funktion an der I/O-Buchse des C-863.12
1	schwarz	Eingang 1 (analog: 0 bis +5V / digital: TTL)
2	weiß	Eingang 2 (analog: 0 bis +5V / digital: TTL)
3	rot	Eingang 3 (analog: 0 bis +5V / digital: TTL)
4	gelb	Eingang 4 (analog: 0 bis +5V / digital: TTL)
5	violett	Ausgang 1 (digital, TTL)
6	blau	Ausgang 2 (digital, TTL)
7	grün	Ausgang 3 (digital, TTL)
8	braun	Ausgang 4 (digital, TTL)
9	grau	Vcc (+5V)
Mantel	Schirm, schwarz beschichtet (dicker als der an Pin 1 angeschlossene Draht)	GND

13.3.4 Joystick

Mini-DIN-Buchse, 6-polig, weiblich (PS/2)

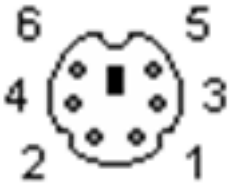


Abbildung 28: Vorderansicht der Mini-DIN-Buchse

Pin	Funktion
1	GND
2	Nicht verbunden
3	Ausgang: Vcc (3,3 V)
4	Eingang: Achse 1 von Joystick-Gerät 1 (0 bis 3,3 V)
5	Nicht verbunden
6	Eingang: Taste 1 von Joystick-Gerät 1 (0 oder 3,3 V)

13.3.5 Kabel C-819.20Y für Joystick C-819.20

Das Kabel C-819.20Y ermöglicht den Anschluss von 2 Controllern an den Joystick C-819.20.

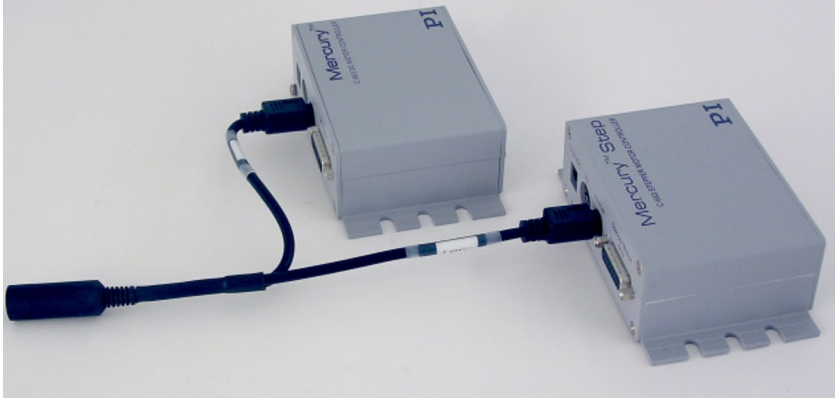


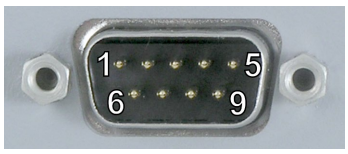
Abbildung 29: Y-Kabel C-819.20Y für Joystick mit 2 Controllern

Mini-DIN-Stecker, 6-polig, weiblich auf 2 Mini-DIN-Stecker, 6-polig, männlich

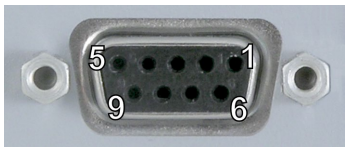
Mini-DIN 6-polig, weiblich (zum Joystick)	Signal	Mini-DIN 6-polig, männlich, X-Zweig (zu Controller 1)	Mini-DIN 6-polig, männlich, Y-Zweig (zu Controller 2)
Pin 1	GND	Pin 1	Pin 1
Pin 2	Taste für Y-Achse des Joysticks	Nicht verbunden	Pin 6
Pin 3	Spannungsversorgung des Joysticks	Pin 3	Nicht verbunden
Pin 4	X-Achse des Joysticks	Pin 4	Nicht verbunden
Pin 5	Y-Achse des Joysticks	Nicht verbunden	Pin 4
Pin 6	Taste für X-Achse des Joysticks	Pin 6	Nicht verbunden

13.3.6 RS-232 In und RS-232 Out

RS-232 In: D-Sub-Einbaustecker, 9-polig, männlich



RS-232 Out: D-Sub-Buchse, 9-polig, weiblich



Pin	Funktion
1	Nicht verbunden
2	RxD (PC zum Controller)
3	TxD (Controller zum PC)
4	Nicht verbunden
5	GND
6	Nicht verbunden
7	Nicht verbunden
8	Nicht verbunden
9	Nicht verbunden

INFORMATION

Die Pins der Buchsen **RS-232 In** und **RS-232 Out** sind im C-863.12 1:1 miteinander verbunden.

INFORMATION

In einem Daisy-Chain-Netzwerk, das über die RS-232-Schnittstelle des ersten Controllers an den PC angeschlossen ist, speist nur der PC die RxD-Leitung. In Abhängigkeit davon, wie leistungsfähig der RS-232-Treiber des PC ist, kann daher der Umfang des Netzwerks auf 6 Geräte beschränkt sein.

INFORMATION

Der C-863.12 kopiert alle Signale, die er vom PC über USB erhält, auf die RxD-Leitung der Buchsen **RS-232 In** und **RS-232 Out**. Das Signal der TxD-Leitung kopiert der C-863.12 über USB zum PC.

13.3.7 Netzteilanschluss

DC-Power-Buchse (Kycon), 4-polig (w), verriegelbar

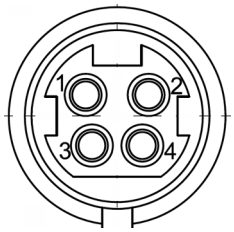


Abbildung 30: DC-Power-Buchse (Kycon), 4-polig (w)

Pin	Signal	Richtung
1	GND	GND
2	Versorgungsspannung 48 V DC	Eingang
3	GND	GND
4	Versorgungsspannung 48 V DC	Eingang
Schirm	Über das Gehäuse verbunden mit GND	GND

14 Altgerät entsorgen

Nach geltendem EU-Recht dürfen Elektrogeräte in den Mitgliedsstaaten der EU nicht über den kommunalen Restmüll entsorgt werden.

Entsorgen Sie das Altgerät unter Beachtung der internationalen, nationalen und regionalen Richtlinien.

Um der Produktverantwortung als Hersteller gerecht zu werden, übernimmt die Physik Instrumente (PI) SE & Co. KG kostenfrei die umweltgerechte Entsorgung eines Altgeräts von PI, sofern es nach dem 13. August 2005 in Verkehr gebracht wurde.

Falls Sie ein solches Altgerät von PI besitzen, können Sie es versandkostenfrei an folgende Adresse senden:

Physik Instrumente (PI) SE & Co. KG

Auf der Römerstr. 1

D-76228 Karlsruhe

15 Europäische Konformitätserklärungen

Für den C-863.12 wurden Konformitätserklärungen gemäß den folgenden europäischen gesetzlichen Anforderungen ausgestellt:

EMV-Richtlinie

RoHS-Richtlinie

Die zum Nachweis der Konformität zugrunde gelegten Normen sind nachfolgend aufgelistet.

EMV: EN 61326-1

Sicherheit: EN 61010-1

RoHS: EN IEC 63000