MS245D

C-886 Parallelkinematik-Controller

Benutzerhandbuch

Version: 1.0.0 Datum: 22.11.2017

Dieses Dokument beschreibt die folgenden Produkte:

- **C-886.1**
  Controller für parallelkinematische Positionierer, 6 Achsen, für DC-Motoren, TCP/IP, USB, 2 optionale Einzelachsen
- **C-886.2**
  Controller für parallelkinematische Positionierer, 6 Achsen, für 2-Phasen-Schrittmotoren, TCP/IP, USB, 2 optionale Einzelachsen
- **C-886.31**
  Controller für parallelkinematische Positionierer, 6 Achsen, für Q-Motion® Piezoträgheitsantriebe, TCP/IP, USB, 2 optionale Einzelachsen

Physik Instrumente (PI) GmbH & 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 folgenden aufgeführten Firmennamen oder Marken sind eingetragene Warenzeichen der Physik Instrumente (PI) GmbH & Co. KG:

PI®, NanoCube®, PICMA®, PIFOC®, PILine®, NEXLINE®, PiezoWalk®, PicoCube®, PiezoMove®, PIMikroMove®, NEXACT®, Picoactuator®, PInano®, NEXSHIFT®, PITOUCH®, PIMag®, PIHera, Q-Motion®

Hinweise zu Markennamen und Warenzeichen Dritter:
Microsoft® und Windows® sind eingetragene Warenzeichen oder Warenzeichen der Microsoft Corporation in den USA und/oder anderen Ländern.
Bei den nachfolgend aufgeführten Bezeichnungen handelt es sich um geschützte Firmennamen, Warenzeichen oder eingetragene Warenzeichen fremder Inhaber:
Linux, LabVIEW, MATLAB, National Instruments, MathWorks, EtherCAT

Die von PI gehaltenen Patente finden Sie in unserer Patentliste (http://www.physikinstrumente.de/de/ueber-pi/patente).

Von PI zur Verfügung gestellte Softwareprodukte unterliegen den Allgemeinen Softwarelizenzbestimmungen der Physik Instrumente (PI) GmbH & Co. KG und können Drittanbieter-Softwarekomponenten beinhalten und/oder verwenden. Weitere Informationen finden Sie in den Allgemeinen Softwarelizenzbestimmungen http://www.physikinstrumente.com/download/EULA_PhysikInstrumenteGmbH_Co_KG.pdf und in den Drittanbieter-Softwarehinweisen http://www.physikinstrumente.com/download/TPSWNote_PhysikInstrumenteGmbH_Co_KG.pdf auf unserer Webseite.

© 2017 Physik Instrumente (PI) GmbH & Co. KG, Karlsruhe, Deutschland. Die Texte, Bilder und Zeichnungen dieses Handbuchs sind urheberrechtlich geschützt. Physik Instrumente (PI) GmbH & 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: 22.11.2017
Dokumentnummer: MS245D, BRo, Version 1.0.0

Änderungen vorbehalten. Dieses Handbuch verliert seine Gültigkeit mit Erscheinen einer neuen Revision. Die jeweils aktuelle Revision ist auf unserer Webseite (http://www.pi.ws) 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 ... 3
1.4 Mitgeltende Dokumente ... 6
1.5 Handbücher herunterladen ... 7

2 Sicherheit 9

2.1 Bestimmungsgemäße Verwendung ... 9
2.2 Allgemeine Sicherheitshinweise ... 9
2.2.1 Organisatorische Maßnahmen ... 10
2.2.2 Maßnahmen bei der Installation ... 10
2.2.3 Maßnahmen bei Inbetriebnahme und Betrieb ... 11
2.2.4 Maßnahmen bei der Wartung ... 13

3 Produktbeschreibung 15

3.1 Merkmale und Anwendungsbereich ... 15
3.1.1 Funktionen im Überblick ... 15
3.1.2 Master- und Slave-Module ... 16
3.2 Modellübersicht ... 17
3.3 Produktansicht ... 18
3.3.1 Vorderwand ... 18
3.3.2 Rückwand ... 22
3.4 Typenschild ... 23
3.5 Lieferumfang ... 23
3.6 Optionales Zubehör ... 24
3.7 Kommandierbare Elemente ... 25
3.7.1 Kennungen für den Normalbetrieb des C-886 ... 25
3.7.2 Kennungen für direkten Zugriff auf Slave-Module ... 26
3.8 Wichtige Komponenten der Firmware ... 28
3.9 Bewegungen des Positionierers ... 29
3.9.1 Unterstützte Bewegungsarten ... 29
3.9.2 Profilgenerator für Punkt-zu-Punkt-Bewegungen ... 31
3.9.3 Zyklische Übertragung von Zielpositionen ... 33
3.9.4 Koordinatensysteme ... 35
3.9.5 Rotationen ... 37
3.10 Kommunikationsschnittstellen ... 37
3.11 PC-Softwareübersicht ... 38

4 Auspacken 41

5 Installation 43

5.1 PC-Software installieren ... 43
5.1.1 Erstinstallation ausführen ... 43
5.1.2 Updates installieren ... 44
5.2 Belüftung sicherstellen ... 46
5.3 C-886 an Schutzleiter anschließen ... 46
5.4 Slave-Module im C-886 installieren ... 47
5.5 Netzteil an C-886 anschließen ... 48
5.6 Antriebe anschließen ... 49
5.7 PC anschließen ... 50
5.7.1 C-886 über die TCP/IP-Schnittstelle anschließen ... 50
5.7.2 C-886 über die USB-Schnittstelle anschließen ... 51

6 Inbetriebnahme 53

6.1 Allgemeine Hinweise zur Inbetriebnahme ... 53
6.2 C-886 einschalten ... 56
6.3 Kommunikation über TCP/IP-Schnittstelle herstellen ... 57
6.3.1 PC und C-886 für Verwendung statischer IP-Adressen vorbereiten ... 58
6.3.2 Kommunikation über TCP/IP in der PC-Software herstellen ... 61
6.4 Kommunikation über USB-Schnittstelle herstellen ... 64
6.5 Slave-Module für Einzelachsen konfigurieren ... 66
6.6 Bewegungen starten ... 70

7 Betrieb 75

7.1 Allgemeine Hinweise zum Betrieb ... 75
7.2 Datenrekorder ... 76
7.2.1 Eigenschaften des Datenrekorders ... 76
7.2.2 Datenrekorder einrichten ... 77
7.2.3 Aufzeichnung starten ... 78
7.2.4 Aufgezeichnete Daten auslesen ... 78
7.3 Funktionsgenerator ... 79
7.3.1 Funktionsweise des Funktionsgenerators ... 79
7.3.2 Befehle und Parameter für den Funktionsgenerator ... 80
7.3.3 Kurvenform definieren ... 82
7.3.4 Funktionsgenerator konfigurieren ... 90
7.3.5 Ausgabe starten und stoppen ... 92
7.3.6 Anwendungstipps: Kundenspezifische Daten laden ... 95
7.3.7 Anwendungstipps: Makros für Funktionsgenerator verwenden ... 100
7.4 Controllermakros ... 103
7.4.1 Übersicht: Makrofunktionalitäten und Beispielmakros ... 103
7.4.2 Befehle und Parameter für Makros ... 103
7.4.3 Mit Makros arbeiten ... 105

7.4.4 Variablen ... 112

8 GCS-Befehle 115

8.1 Schreibweise ... 115
8.2 GCS-Syntax für Syntaxversion 2.0 ... 115
8.3 Befehlsübersicht ... 118
8.4 Befehlsbeschreibungen für GCS 2.0 ... 123
8.5 Fehlercodes ... 230

9 Anpassen von Einstellungen 247

9.1 Überblick über die Einstellungen des C-886 ... 247
9.2 Parameterwerte des C-886-Master-Moduls ändern ... 248
9.3 Parameterwerte in Textdatei sichern ... 249
9.4 Parameterübersicht ... 250

10 Wartung 257

10.1 C-886 reinigen ... 257
10.2 Firmware aktualisieren ... 258

11 Störungsbehebung 263

12 Kundendienst 267

13 Technische Daten 269

13.1 Spezifikationen ... 269
13.1.1 Datentabelle ... 269
13.1.2 Bemessungsdaten ... 273
13.1.3 Umgebungsbedingungen und Klassifizierungen ... 273
13.2 Systemanforderungen ... 273
13.3 Abmessungen ... 274
13.4 Pinbelegung ... 275
13.4.1 Axis 1, Axis 2 (nur C-886.1) ... 275
13.4.2 Motor (nur C-886.2) ... 276
13.4.3 Motor & Sensor (nur C-886.31) ... 277

14 Altgerät entsorgen 279

15 EU-Konformitätserklärung 281

1 Über dieses Dokument

In diesem Kapitel

Ziel und Zielgruppe dieses Benutzerhandbuchs 1
Symbole und Kennzeichnungen 1
Begriffserklärung 3
Mitgeltende Dokumente 6
Handbücher herunterladen 7

1.1 Ziel und Zielgruppe dieses Benutzerhandbuchs

Dieses Benutzerhandbuch enthält die erforderlichen Informationen für die bestimmungsgemäße Verwendung des C-886.

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

Die aktuellen Versionen der Benutzerhandbücher stehen auf unserer Website zum Herunterladen (S. 7) bereit.

1.2 Symbole und Kennzeichnungen

In diesem Benutzerhandbuch werden folgende Symbole und Kennzeichnungen verwendet:

VORSICHT

Gefährliche Situation
Bei Nichtbeachtung drohen leichte Verletzungen.

➢ Maßnahmen, um die Gefahr zu vermeiden.

HINWEIS

Gefährliche Situation
Bei Nichtbeachtung drohen Sachschäden.

➢ Maßnahmen, um die Gefahr zu vermeiden.

INFORMATION

Informationen zur leichteren Handhabung, Tricks, Tipps, etc.

Symbol/ Kennzeichnung	Bedeutung
1. 2.	Handlung mit mehreren Schritten, deren Reihenfolge eingehalten werden muss
➢	Handlung mit einem Schritt oder mehreren Schritten, deren Reihenfolge nicht relevant ist
▪	Aufzählung
S. 5	Querverweis auf Seite 5
RS-232	Bedienelement-Beschriftung auf dem Produkt (Beispiel: Buchse der RS-232 Schnittstelle)
⚠ ⚡	Auf dem Produkt angebrachte Warnzeichen, die auf ausführliche Informationen in diesem Handbuch verweisen.
Start > Einstellungen	Menüpfad in der PC-Software (Beispiel: Zum Aufrufen des Menüs muss nacheinander auf die Menüeinträge ***Start*** und ***Einstellungen*** geklickt 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)
5	Wert, der über die PC-Software eingegeben bzw. ausgewählt werden muss

1.3 Begriffserklärung

Absolut messender Positionssensor

Sensor (Encoder) zur Erfassung von Lageänderungen oder Winkeländerungen. Die Signale des absolut messenden Positionssensors werden für die Rückmeldung der Achsenposition verwendet. Nach dem Einschalten des Controllers können sofort absolute Zielpositionen kommandiert und erreicht werden. Eine Referenzwertbestimmung ist nicht erforderlich.

Achse

Auch als „logische Achse" bezeichnet. Logische Achsen bilden die Translationen und Rotationen der bewegten Plattform des Positionierers und die Bewegungen der optional verwendbaren Versteller in der Firmware des C-886 ab. Jede Bewegungsrichtung entspricht einer logischen Achse. Weitere Informationen zu Translationen und Rotationen finden Sie im Handbuch des Positionierers.

Alle Bewegungsbefehle des C-886 beziehen sich auf logische Achsen.

Achsorientierungs-Koordinatensystem

Mit dem Achsorientierungs-Koordinatensystem kann die Richtung der Translationsachsen X und/oder Y und/oder Z dauerhaft geändert werden (z. B. wenn Z immer in die Richtung der werkseitigen X-Achse zeigen soll). In der Werkseinstellung ist das Achsorientierungs-Koordinatensystem PI_Base aktiv.

Arbeitsraum

Die Gesamtheit aller Kombinationen von Translationen und Rotationen, die der Positionierer von der aktuellen Position aus anfahren kann, wird als Arbeitsraum bezeichnet.

Der Arbeitsraum kann durch folgende externe Faktoren eingeschränkt werden:

- Vorhandener Einbauraum
- Abmessungen und Position der Last

Betriebs-Koordinatensystem

Mit dem Betriebs-Koordinatensystem werden die Positionsanzeige, die Bewegungsrichtung und der Drehpunkt für die Bewegungsplattform des Positionierers an die Anwendung angepasst. Es ist auch möglich, --> Work-und Tool-Koordinatensysteme zu verwenden. In der Werkseinstellung ist das Betriebs-Koordinatensystem ZERO aktiv.

Drehpunkt

Der Drehpunkt beschreibt das Rotationszentrum (Schnittpunkt der Rotationsachsen U, V und W).

Der Drehpunkt bewegt sich immer zusammen mit der Plattform.

In Abhängigkeit vom aktiven --> Betriebs-Koordinatensystem kann der Drehpunkt mit dem Befehl `SPI` aus dem Ursprung des Koordinatensystems heraus in X- und/oder Y- und/oder Z-Richtung verschoben werden. Der mit dem Befehl `SPI` verschiebbare Drehpunkt wird auch als „Pivotpunkt“ bezeichnet.

Dynamikprofil	Umfasst die für jeden Zeitpunkt der Bewegung berechnete Zielposition, Geschwindigkeit und Beschleunigung der Achse. Die errechneten Werte werden "kommandierte Werte" genannt. Das Dynamikprofil kann vom Profilgenerator des C-886 oder von den Funktionsgeneratoren erzeugt werden, oder es kann extern erzeugt und durch zyklische Übertragung von Zielpositionen an den C-886 übergeben werden.
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. Piezosteuerungen und Servocontroller können dank GCS mit minimalem Programmieraufwand gemeinsam betrieben werden.
Positionierer-System	Die Kombination aus Positionierer, Controller, Kabelsatz und Netzteil(en) wird in diesem Handbuch als „Positionierer-System“ bezeichnet.
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 Referenzwertbestimmung durchgeführt werden, bevor absolute Zielpositionen kommandiert und erreicht werden können.
Koordinatensystem	Positionsanzeige, Bewegungsrichtung und Drehpunkt für die Bewegungsplattform des Positionierers werden durch miteinander verkettete Koordinatensysteme bestimmt. Die Kette ist grundsätzlich wie folgt aufgebaut (Ausgangspunkt > Endpunkt): --> Koordinatensystem HEXAPOD, --> Korrektur-Koordinatensystem, --> Achsorientierungs-Koordinatensystem, --> Betriebs-Koordinatensystem. Die Koordinatensysteme sind immer rechtshändige Systeme. Das Koordinatensystem HEXAPOD bestimmt die grundsätzlichen Eigenschaften aller anderen Koordinatensysteme. HEXAPOD basiert auf der Konfigurationsdatei mit den Geometriedaten des Positionierers. Die Maßzeichnung im Handbuch des Positionierers zeigt jeweils die Lage des Koordinatensystems HEXAPOD. Mit dem Controller können eigene Koordinatensysteme definiert und anstelle der werkseitig voreingestellten Koordinatensysteme verwendet werden. Achsorientierungs- und Korrektur-Koordinatensystem passen, basierend auf HEXAPOD, grundlegende Eigenschaften des aktiven Betriebs-Koordinatensystems an und müssen in den meisten Anwendungen gar nicht oder nur einmalig anwenderspezifisch definiert und aktiviert werden.

Korrektur-Koordinatensystem	Mit dem Korrektur-Koordinatensystem können Fehler in der Ausrichtung des Positionierers dauerhaft korrigiert werden (z. B. Einbaufehler). In der Werkseinstellung ist das Korrektur-Koordinatensystem PI_Levelling aktiv.
Master-Modul	Das Master-Modul des C-886 führt zentrale Aufgaben aus: ▪ Parallelkinematische Ansteuerung: Umrechnung zwischen den kartesischen Koordinaten der Bewegungsplattform und den Positionen der einzelnen Antriebe ▪ Schnittstelle zur Kommunikation mit dem PC ▪ Interne Kommunikation mit den Slave-Modulen zur Ansteuerung der Antriebe des parallelkinematischen Positionierers und der optionalen Einzelachsen ▪ Bereitstellung von Datenrekorder, Funktionsgeneratoren und Makrofunktionalität ▪ Bereitstellung anwenderdefinierter Koordinatensysteme für den parallelkinematischen Positionierer
Parallelkinematischer Positionierer	Der C-886 ist für den Betrieb eines parallelkinematischen Positionierers vorgesehen, der in diesem Handbuch auch kurz als "Positionierer" bezeichnet wird. Ein parallelkinematischer Positionierer kann ein Hexapod sein oder ein Aufbau, der mit passiven Beinen ausgestattet ist, die jeweils durch eine Kombination aus zwei Linearantrieben bewegt werden. Für die Bewegung einer logischen Achse der Plattform des Positionierers ist die Bewegung mehrerer Linearantriebe notwendig. Der C-886 berechnet aus den vorgegebenen Zielpositionen für die Translations- und Rotationsachsen des Positionierers die Zielpositionen für die einzelnen Linearantriebe. Die Geschwindigkeiten und Beschleunigungen der Linearantriebe werden so berechnet, dass alle Antriebe zur selben Zeit starten und stoppen.
PC-Software	Software, die auf dem PC installiert wird.
Permanenter Speicher	EEPROM-Speicherchip (Festspeicher), 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.
Pivotpunkt	In Abhängigkeit vom aktiven --> Betriebs-Koordinatensystem kann der Drehpunkt mit dem Befehl `SPI` aus dem Ursprung des Koordinatensystems heraus in X- und/oder Y- und/oder Z-Richtung verschoben werden. Der mit dem Befehl `SPI` verschiebbare Drehpunkt wird auch als „Pivotpunkt“ bezeichnet.

Slave-Modul

Die Slave-Module des C-886 (auch "Antriebsmodule") sind die Schnittstellen zu den Antrieben des parallelkinematischen Positionierers und der optionalen Einzelachsen. Die Slave-Module erhalten vom Master-Modul die Ansteuerbefehle für die einzelnen Antriebe.

- Slave-Module für den parallelkinematischen Positionierer:
  Bei Auslieferung des C-886 sind die Module für den verwendeten Positionierer konfiguriert. Der Austausch der Module und direkte Zugriff auf die Einstellungen der Module darf nur nach Rücksprache mit PI erfolgen.
- Slave-Module für die zwei optionalen Einzelachsen:
  Die Module sind als optionales Zubehör erhältlich. Der Einbau und die Konfiguration der Module können vom Anwender vorgenommen werden.

Work- und Tool-Koordinaten-systeme

Für die Arbeit mit anwenderdefinierten Koordinatensystemen kann das Work-und-Tool-Konzept eingesetzt werden.

Das Work-und-Tool-Konzept verwendet eine Kombination aus zwei aktiven --> Betriebs-Koordinatensystemen ("Work-Koordinatensystem" und "Tool-Koordinatensystem"). Die Achsen X, Y, Z des Tool-Koordinatensystems sind immer fest mit der bewegten Plattform des Positionierers verbunden, d.h. das Tool-Koordinatensystem bewegt sich zusammen mit der Plattform. Die Achsen X, Y, Z des Work-Koordinatensystems sind immer raumfest, d.h. das Work-Koordinatensystem bewegt sich **nicht** mit, wenn sich die Plattform des Positionierers bewegt.

Die aktuelle Position der bewegten Plattform des Positionierers ist als Position des Tool-Koordinatensystems im Work-Koordinatensystem zu verstehen.

Der Drehpunkt für Rotationen liegt immer im Ursprung des Tool-Koordinatensystems und bewegt sich deshalb genau wie das Tool-Koordinatensystem zusammen mit der Plattform.

1.4 Mitgeltende Dokumente

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

Dokumentation für den C-886 und die mitgelieferte PC-Software:

Beschreibung	Dokument
Koordinatensysteme für Hexapod-Mikroroboter	Technical Note C887T0007
Bewegungen des Hexapods Position und Orientierung im Raum, Drehpunkt	Technical Note C887T0021
GCS LabVIEW-Treiberbibliothek für C-886	MS246E Software Manual

Beschreibung	Dokument
LabVIEW Merge Tool	SM154E Software Manual
PI GCS 2.0 DLL	SM151E Software Manual
PI MATLAB-Treiber GCS 2.0	SM155D Software Manual
PIPython	SM157D Benutzerhandbuch
GCS Array Datenformatbeschreibung	SM146E Software Manual
PIMikroMove®	SM148E Software Manual
PI Frequency Generator Tool In PIMikroMove® verfügbar für Controller mit Funktionsgenerator	Technical Note A000T0057
PIStages3Editor	SM156D Benutzerhandbuch
PI Update Finder: Updates suchen und herunterladen	A000T0028 Technical Note
PI Update Finder: PC ohne Internetverbindung aktualisieren	A000T0032 Technical Note

Benutzerhandbuch des parallelkinematischen Positionierers

Benutzerhandbücher der optionalen Einzelachsen

Wenn Slave-Module für optionale Einzelachsen verwendet werden:

Beschreibung	Dokument
C-663.12C885 Controllermodul	C663T0004 User Manual, MS241D Benutzerhandbuch
C-863.20C885 Controllermodul	C863T0005 User Manual, MS205D Benutzerhandbuch
E-873.10C885 Controllermodul	E873T0002 User Manual, PZ273D Benutzerhandbuch

1.5 Handbücher herunterladen

INFORMATION

Wenn ein Handbuch fehlt oder Probleme beim Herunterladen auftreten:

➢ Wenden Sie sich an unseren Kundendienst (S. 267).

INFORMATION

Für Produkte, die mit Software ausgeliefert werden (CD im Lieferumfang), ist der Zugang zu den Handbüchern durch ein Kennwort geschützt. Geschützte Handbücher werden auf der Website erst nach Eingabe des Kennworts angezeigt.
Das Kennwort ist auf der CD des Produkts enthalten.

Für Produkte mit CD: Kennwort identifizieren

1. Legen Sie die CD des Produkts in das PC-Laufwerk ein.
2. Wechseln Sie auf der CD in das Verzeichnis Manuals.
3. Öffnen Sie im Verzeichnis Manuals die Release News (Datei mit dem Namensbestandteil ***Releasenews***).
4. Entnehmen Sie dem Abschnitt "User login for software download" in den Release News den Benutzernamen (user name) und das Kennwort (password).

Handbücher herunterladen

1. Öffnen Sie die Website **www.pi.de**.
2. Wenn der Zugang zu den Handbüchern durch ein Kennwort geschützt ist:
   a) Klicken Sie auf ***Login***.
   b) Melden Sie sich mit dem Benutzernamen und dem Kennwort an.
3. Klicken Sie auf ***Suche***.
4. Geben Sie die Produktnummer bis zum Punkt (z. B. P-882) oder die Produktfamilie (z. B. PICMA® Bender) in das Suchfeld ein.
5. Klicken Sie auf ***Suche starten*** oder drücken Sie die Enter-Taste.
6. Öffnen Sie über die Liste der Suchergebnisse die entsprechende Produktdetailseite:
   a) Wenn notwendig: Scrollen Sie in der Liste nach unten.
   b) Wenn notwendig: Klicken Sie am Ende der Liste auf ***Weitere Suchergebnisse laden***.
   c) Klicken Sie in der Liste auf das entsprechende Produkt.
7. Scrollen Sie auf der Produktdetailseite nach unten zum Bereich ***Downloads***.

   Die Handbücher werden unter ***Dokumentation*** angezeigt.
8. Klicken Sie auf das gewünschte Handbuch und speichern Sie es auf der Festplatte Ihres PC oder auf einem Datenträger.

2 Sicherheit

In diesem Kapitel

Bestimmungsgemäße Verwendung 9
Allgemeine Sicherheitshinweise 9

2.1 Bestimmungsgemäße Verwendung

Der C-886 ist ein Laborgerät im Sinne der DIN EN 61010-1. Er ist für die Verwendung in Innenräumen und in einer Umgebung vorgesehen, die frei von Schmutz, Öl und Schmiermitteln ist.

Entsprechend seiner Bauform ist der C-886 vorgesehen für den geregelten Betrieb eines parallelkinematischen Positionierers von PI. An den C-886 dürfen nur Positionierer mit passendem Antriebstyp gemäß der Spezifikationen (S. 269) angeschlossen werden.

Der C-886 darf nicht für andere als die in diesem Benutzerhandbuch genannten Zwecke verwendet werden.

Der C-886 darf nur unter Einhaltung der technischen Spezifikationen und Anweisungen in diesem Benutzerhandbuch verwendet werden. Für die Prozessvalidierung ist der Betreiber verantwortlich.

2.2 Allgemeine Sicherheitshinweise

Der C-886 ist nach dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln gebaut. Bei unsachgemäßer Verwendung des C-886 können Benutzer gefährdet werden und/oder Schäden am C-886 entstehen.

- Benutzen Sie den C-886 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-886 verantwortlich.

- Installieren Sie den C-886 in der Nähe der Stromversorgung, damit der Netzstecker schnell und einfach vom Netz getrennt werden kann.
- Verwenden Sie zum Anschließen des C-886 an die Stromversorgung die mitgelieferten Komponenten (Netzteil, Adapter, Netzkabel (S. 24)).

- Wenn eine der mitgelieferten Komponenten für das Anschließen an die Stromversorgung ersetzt werden muss, verwenden Sie eine ausreichend bemessene Komponente.

2.2.1 Organisatorische Maßnahmen

Benutzerhandbuch

- Halten Sie dieses Benutzerhandbuch ständig am C-886 verfügbar.
  Die aktuellen Versionen der Benutzerhandbücher stehen auf unserer Website zum Herunterladen (S. 7) bereit.
- Fügen Sie alle vom Hersteller bereitgestellten Informationen, z. B. Ergänzungen und Technical Notes, zum Benutzerhandbuch hinzu.
- Wenn Sie den C-886 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 leichten Verletzungen und zu Sachschäden führen.
- Installieren und bedienen Sie den C-886 nur, nachdem Sie dieses Benutzerhandbuch gelesen und verstanden haben.

Personalqualifikation

Nur autorisiertes und entsprechend qualifiziertes Personal darf den C-886 installieren, in Betrieb nehmen, bedienen, warten und reinigen.

2.2.2 Maßnahmen bei der Installation

Unzulässige mechanische Belastung und Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung können den Positionierer beschädigen.

- Beachten Sie die Sicherheitshinweise und Anweisungen im Handbuch des Positionierers.
- Überschreiten Sie **nicht** die Kräfte und Drehmomente, die für den Positionierer spezifiziert sind, siehe Handbuch des Positionierers.

Das Anschließen eines ungeeigneten Antriebstyps kann zu Schäden am Antrieb oder C-886 führen.

- Schließen Sie an die Slave-Module des C-886 nur einen geeigneten Antriebstyp an.

2.2.3 Maßnahmen bei Inbetriebnahme und Betrieb

Bei fehlendem oder nicht ordnungsgemäß angeschlossenem Schutzleiter können im Falle eines Fehlers oder Defekts gefährliche Berührungsspannungen am C-886 entstehen. Wenn Berührungsspannungen vorhanden sind, kann das Berühren des C-886 zu leichten Verletzungen durch Stromschlag führen.

- Schließen Sie den C-886 vor Inbetriebnahme an einen Schutzleiter an (S. 46).
- Entfernen Sie den Schutzleiter **nicht** während des Betriebs.
- Wenn der Schutzleiter vorübergehend entfernt werden muss (z. B. bei Umbauten), schließen Sie den C-886 vor erneuter Inbetriebnahme wieder an den Schutzleiter an.

Zwischen den bewegten Teilen des Positionierers und einem feststehenden Teil oder Hindernis besteht die Gefahr von leichten Verletzungen durch Quetschung.

- Halten Sie Ihre Finger von Bereichen fern, in denen sie von bewegten Teilen erfasst werden können.

Wenn die Kommunikation zwischen C-886 und PC über TCP/IP hergestellt wird, bietet die PC-Software alle im selben Netzwerk vorhandenen Controller zur Auswahl an. Nach Auswahl eines C-886 für die Verbindung werden alle Befehle an diesen Controller geschickt. Bei Auswahl eines falschen Controllers besteht für das Bedien- und Wartungspersonal des angeschlossenen Positionierers die Gefahr von leichten Verletzungen durch Quetschung aufgrund von unerwartet kommandierten Bewegungen.

- Wenn in der PC-Software mehrere C-886 angezeigt werden, vergewissern Sie sich, dass Sie den richtigen C-886 auswählen.

Die Konfiguration des C-886 muss auf den Positionierer abgestimmt sein. Bei Verwendung einer falschen Konfiguration kann der Positionierer durch unkontrollierte Bewegungen oder Kollisionen beschädigt werden. Bei Auslieferung ist der C-886 für den verwendeten Positionierer konfiguriert.

- Prüfen Sie die Konfiguration: Nachdem Sie die Kommunikation über TCP/IP (S. 57) oder USB (S. 64) hergestellt haben, senden Sie den Befehl `CST?`. Die Antwort zeigt an, auf welchen Positionierer der C-886 abgestimmt ist.
- Betreiben Sie den Positionierer nur mit einem C-886, dessen Konfigurationsdaten auf den Positionierer abgestimmt sind.
- Ändern Sie die Einstellungen der Slave-Module für die Antriebe des Positionierers nur nach Rücksprache mit PI.

Kollisionen können den Positionierer, die zu bewegende Last und die Umgebung beschädigen.

Generelle Maßnahmen zur Vermeidung von Kollisionen:

- Stellen Sie sicher, dass im Arbeitsraum des Positionierers keine Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich sind.

- Platzieren Sie keine Gegenstände in Bereichen, in denen sie von bewegten Teilen erfasst werden können.
- Halten Sie bei einer Fehlfunktion des Controllers die Bewegung sofort an.
- Beachten Sie, dass sich der Positionierer während einer Referenzfahrt auf unvorhersehbare Weise bewegt.

In Abhängigkeit von der Quelle des Bewegungsprofils kann sich die Plattform des Positionierers unter bestimmten Bedingungen auf einer undefinierten Bahn bewegen. Dadurch sind Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich.

Wenn das Bewegungsprofil durch den Profilgenerator des C-886 festgelegt wird (Standard):

- Vermeiden Sie das Senden neuer Zielpositionen, wenn sich der Positionierer (Achsen X, Y, Z, U, V, W) noch bewegt.
- Wenn neue Zielpositionen gesendet werden müssen, während sich der Positionierer noch bewegt (Achsen X, Y, Z, U, V, W): Setzen Sie mit Bewegungsbefehlen nur Zielpositionen, die von der aktuellen Position maximal um den Wert des Parameters ***Path Control Step Size*** (ID 0x19001504) abweichen.

Wenn das Bewegungsprofil durch aufeinander folgende `MOV`-Befehle festgelegt wird:

- Setzen Sie mit den aufeinanderfolgenden `MOV`-Befehlen nur Zielpositionen, deren Abstand zueinander maximal so groß wie der Wert des Parameters ***Path Control Step Size*** (ID 0x19001504) ist.

Wenn die tatsächliche Belastung der Bewegungsplattform des Positionierers die auf der Selbsthemmung der Linearantriebe basierende maximale Haltekraft überschreitet, kann das Ausschalten des Servomodus ungewollte Positionsänderungen des Positionierers verursachen. Dadurch sind Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich.

- Stellen Sie sicher, dass die tatsächliche Belastung der Bewegungsplattform des Positionierers die auf der Selbsthemmung der Linearantriebe basierende maximale Haltekraft nicht überschreitet, bevor Sie den Servomodus ausschalten, den C-886 neu starten oder ausschalten.
- Sorgen Sie für eine unterbrechungsfreie Stromversorgung, um eine ungewollte Deaktivierung des Positionierer-Systems und daraus resultierende ungewollte Positionsänderungen des Positionierers zu vermeiden.

Unpassende Parametereinstellungen können zu unsachgemäßem Betrieb oder zur Beschädigung der angeschlossenen Mechanik führen.

- Ändern Sie Parameter nur nach sorgfältiger Überlegung.

2.2.4 Maßnahmen bei der Wartung

Der C-886 enthält elektrostatisch gefährdete Bauteile, die bei Kurzsschlüssen oder Überschlägen beschädigt werden können.

- Trennen Sie vor dem Reinigen des Gehäuses den C-886 von der Stromversorgung, indem Sie den Netzstecker ziehen.

3 Produktbeschreibung

In diesem Kapitel

Merkmale und Anwendungsbereich 15
Modellübersicht 17
Produktansicht 18
Typenschild 23
Lieferumfang 23
Optionales Zubehör 24
Kommandierbare Elemente 25
Wichtige Komponenten der Firmware 28
Bewegungen des Positionierers 29
Kommunikationsschnittstellen 37
PC-Softwareübersicht 38

3.1 Merkmale und Anwendungsbereich

3.1.1 Funktionen im Überblick

Der C-886 Controller ist für die Ansteuerung eines parallelkinematischen Positionierers in sechs Freiheitsgraden vorgesehen. Die Bewegungsbefehle verwenden kartesische Koordinaten. Der C-886 rechnet diese um in die jeweiligen Positionen und Geschwindigkeiten der einzelnen Antriebe, bevor sich die Plattform in die gewünschte Position bewegt.

- TCP/IP- und USB-Schnittstelle für die Kommunikation
- Wenn die entsprechenden Slave-Module im C-886 installiert sind: Anschlüsse für bis zu zwei optionale Einzelachsen mit frei wählbarem Antriebstyp
- Anwenderdefinierte Koordinatensysteme möglich, inklusive Work- und Tool-Koordinatensysteme
- Nichtflüchtiger Makrospeicher
- Datenrekorder
- Funktionsgenerator
- Umfangreiches Softwarepaket verfügbar

3.1.2 Master- und Slave-Module

Um die breite Palette der von PI angebotenen Antriebskonzepte zu unterstützen, ist der C-886 modular aufgebaut. Jeder C-886 besteht aus einem Master-Modul und mehreren Slave-Modulen.

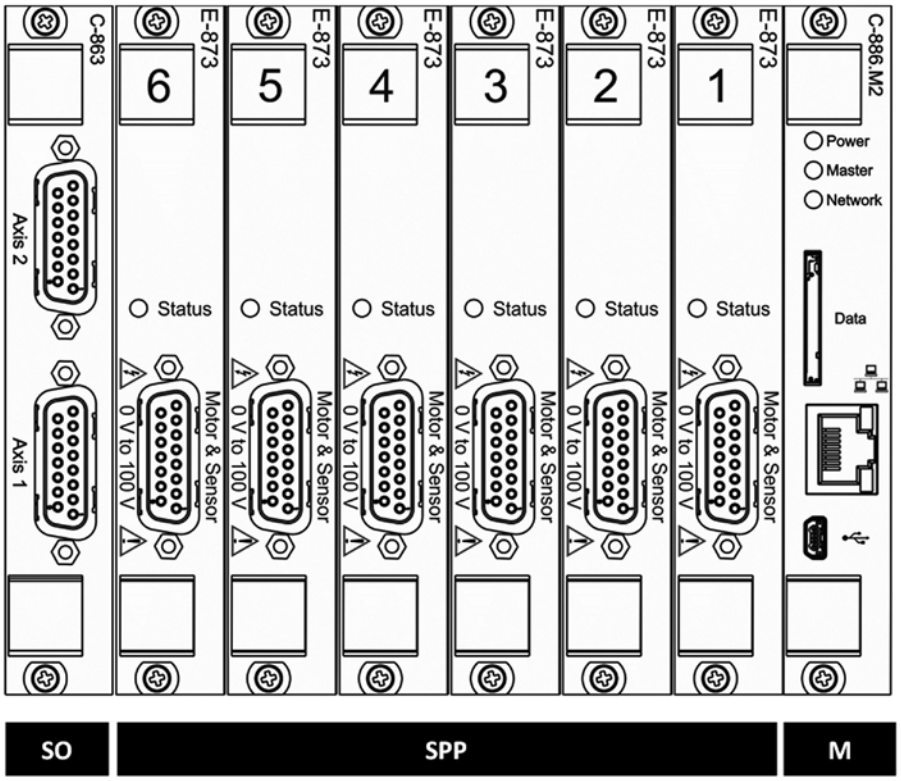


Abbildung 1: Beispiel: C-886.31 mit Modul C-863.20C885 für optionale Einzelachsen

SO Slave-Modul für optionale Einzelachsen
SPP Slave-Module für den parallelkinematischen Positionierer
M Master-Modul

Das Master-Modul des C-886 führt zentrale Aufgaben aus:

- Parallelkinematik-Berechnung: Umrechnung zwischen den kartesischen Koordinaten der Bewegungsplattform und den Positionen der einzelnen Antriebe
- Schnittstelle zur Kommunikation mit dem PC
- Interne Kommunikation mit den Slave-Modulen zur Ansteuerung der Antriebe des parallelkinematischen Positionierers und der optionalen Einzelachsen
- Bereitstellung von Datenrekorder, Funktionsgeneratoren und Makrofunktionalität
- Bereitstellung anwenderdefinierter Koordinatensysteme für den parallelkinematischen Positionierer

Die Slave-Module des C-886 (auch "Antriebsmodule") sind die Schnittstellen zu den Antrieben des parallelkinematischen Positionierers und der optionalen Einzelachsen. Die Slave-Module erhalten vom Master-Modul die Befehle für die Ansteuerung der einzelnen Antriebe.

- Slave-Module für den parallelkinematischen Positionierer:

  Bei Auslieferung des C-886 sind die Module für den verwendeten Positionierer konfiguriert. Der Austausch der Module und Zugriff auf die Einstellungen der Module darf nur nach Rücksprache mit PI erfolgen.

- Slave-Module für die optionalen Einzelachsen:

  Die Module sind als optionales Zubehör (S. 24) erhältlich. Der Einbau und die Konfiguration der Module können vom Anwender vorgenommen werden. Für die Konfiguration der Module können die Einträge der Verstellerdatenbank PIStages3 verwendet werden.

3.2 Modellübersicht

Der C-886 Controller ist in folgenden Ausführungen erhältlich:

Modell	Bezeichnung
C-886.1	Controller für parallelkinematische Positionierer, 6 Achsen, für DC-Motoren, TCP/IP, USB, 2 optionale Einzelachsen
C-886.2	Controller für parallelkinematische Positionierer, 6 Achsen, für 2-Phasen-Schrittmotoren, TCP/IP, USB, 2 optionale Einzelachsen
C-886.31	Controller für parallelkinematische Positionierer, 6 Achsen, für Q-Motion® Piezoträgheitsantriebe, TCP/IP, USB, 2 optionale Einzelachsen

3.3 Produktansicht

3.3.1 Vorderwand

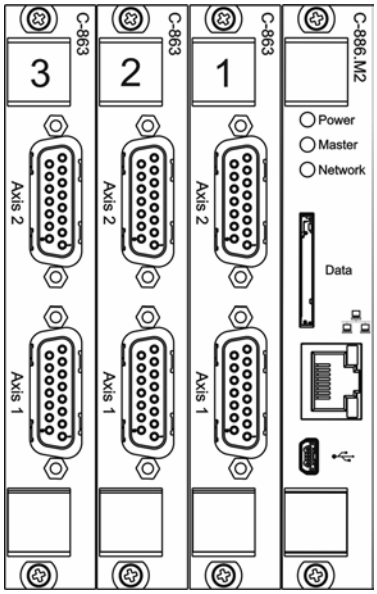


Abbildung 2: C-886.1 Vorderwand, ohne Slave-Module für optionale Einzelachsen

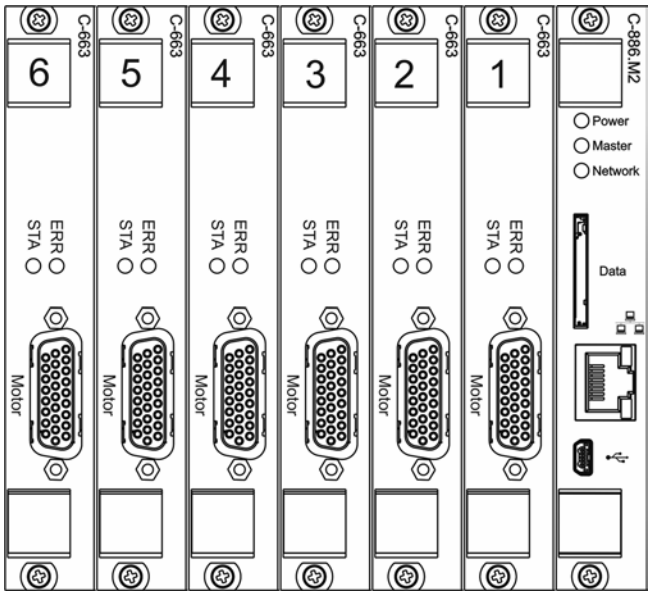


Abbildung 3: C-886.2 Vorderwand, ohne Slave-Module für optionale Einzelachsen

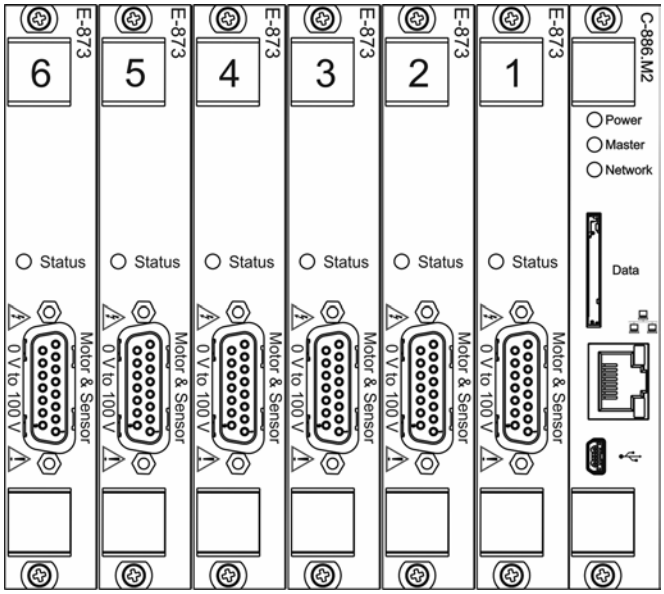


Abbildung 4: C-886.31 Vorderwand, ohne Slave-Module für optionale Einzelachsen

Master-Modul:

Beschriftung	Typ	Funktion
Power	LED grün/aus	Power: ▪ An: Der C-886 ist bereit für den Normalbetrieb. ▪ Aus: Der C-886 bootet. / Der C-886 ist nicht an der Stromversorgung angeschlossen.
Master	LED rot/aus	Fehleranzeige für das Master-Modul: ▪ An: Fehler (Fehlercode ≠ 0) ▪ Aus: Kein Fehler (Fehlercode = 0). / Der C-886 ist nicht an der Stromversorgung angeschlossen. 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
Network	LED rot/grün/aus	Statusanzeige für Slave-Module: ▪ Aus: Es liegt kein Fehler vor (Fehlercode = 0). / Der C-886 bootet. / Der C-886 ist nicht an der Stromversorgung angeschlossen. ▪ Grün: Alle Slave-Module sind bereit für den Normalbetrieb. ▪ Rot: Fehler in der internen Kommunikation oder in mindestens einem der Slave-Module (Fehlercode ≠ 0). Wenn das betroffene Slave-Modul mit einer LED für die Fehleranzeige ausgestattet ist, zeigt auch diese LED den Fehler an.
Data	Slot für SD-Karte	Für künftige Verwendung; derzeit keine Funktion.
	Buchse RJ45	Netzwerkverbindung über TCP/IP
	Mini-USB Typ B	Universal Serial Bus für Verbindung zum PC

Nur C-886.1: Slave-Module für den parallelkinematischen Positionierer:

Beschriftung	Typ	Funktion
Axis 1 **Axis 2**	D-Sub 15 (f) 	Anschlüsse für die Antriebe des parallelkinematischen Positionierers. Nur für DC-Motoren!

Nur C-886.2: Slave-Module für den parallelkinematischen Positionierer:

Beschriftung	Typ	Funktion
STA	LED grün/aus	Status des Slave-Moduls: ▪ Dauerhaftes Leuchten: Modul ist bereit für den Normalbetrieb ▪ Blinken: Modul befindet sich im Firmware-Update-Modus ▪ Aus: Modul ist nicht an der Versorgungsspannung angeschlossen

Beschriftung	Typ	Funktion
ERR	LED rot/aus	Fehleranzeige für Slave-Modul: ▪ An: Fehler. Der Fehler wird auch durch die LED **Network** des Master-Moduls angezeigt. ▪ Aus: Kein Fehler
Motor	D-Sub 26 (f) 	Anschluss für einen Antrieb des parallelkinematischen Positionierers. Nur für 2-Phasen-Schrittmotoren!

Nur C-886.31: Slave-Module für den parallelkinematischen Positionierer:

Beschriftung	Typ	Funktion
Status	LED rot/grün/aus	Status des Slave-Moduls: ▪ Rot: Fehler. Der Fehler wird auch durch die LED **Network** des Master-Moduls angezeigt. ▪ Dauerhaft grün: Modul ist bereit für den Normalbetrieb ▪ Blinken grün: Modul befindet sich im Firmware-Update- Modus ▪ Aus: Modul ist nicht an der Versorgungsspannung angeschlossen
Motor & Sensor ⚠ **0 V to 100 V** ⚠	D-Sub 15 (f) 	Anschlüsse für die Antriebe des parallelkinematischen Positionierers. Nur für Q-Motion® Piezoträgheitsantriebe!

3.3.2 Rückwand

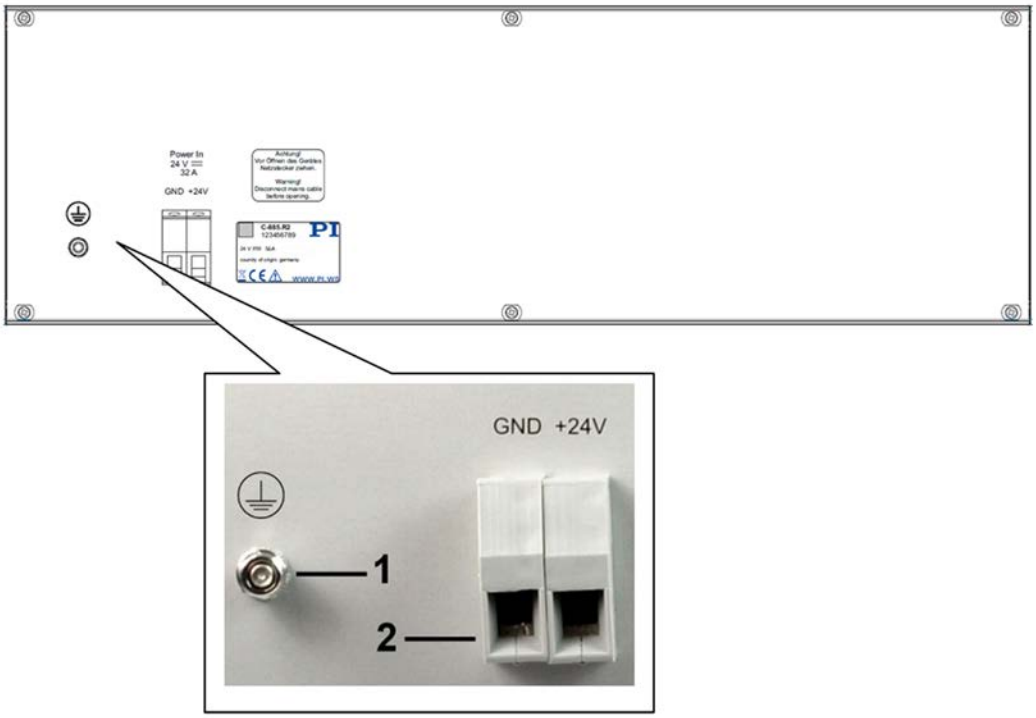


Abbildung 5: C-886 Rückwand

1 Schutzleiteranschluss
2 Anschluss für die Versorgungsspannung

Beschriftung	Typ	Funktion
⏚	Gewindebolzen mit Befestigungs-material für Schutzleiter	Schutzleiteranschluss (S. 46) Der Gewindebolzen muss an einen Schutzleiter angeschlossen werden, da der C-886 nicht über den Netzteil-Anschluss geerdet ist.
GND +24V	Durchführungs-klemmen mit Schraubanschluss, Querschnitt 0,5 mm^2 bis 16 mm^2	Anschluss für die Versorgungsspannung Anschlussbelegung gemäß Beschriftung Maximale Stromaufnahme: 32 A

3.4 Typenschild

Beschriftung	Funktion
	DataMatrix-Code (Beispiel; enthält die Seriennummer)
C-886.31	Produktbezeichnung (Beispiel), die Stellen nach dem Punkt kennzeichnen das Modell
PI	Herstellerlogo
117567891	Seriennummer (Beispiel), individuell für jeden C-886 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. 279)
CE	Konformitätszeichen CE
WWW.PI.WS	Herstelleradresse (Website)

3.5 Lieferumfang

Artikel-nummer	Komponenten
C-886	Controller gemäß Ihrer Bestellung
000036360	USB-Kabel (Typ A auf Mini-B) zur Verbindung mit dem PC
C-885.PS	Separates 24-V-Weitbereichsnetzteil (85 W / 3,54 A) zur Verwendung bei Netzspannungen von 100 bis 240 V AC und Spannungsfrequenzen von 50 oder 60 Hz Anschluss zum C-886: Molex Mini Fit 6-pol. (f)
3763	Netzkabel
000058055	Adapter für den Netzteil-Anschluss; Molex Mini Fit 6-pol. (m) auf Aderendhülsen 4 mm^2
C-886.CD	CD mit PC-Software und Dokumentation
MS245D	Benutzerhandbuch für den C-886

3.6 Optionales Zubehör

Folgende Module können als Slave-Module für optionale Einzelachsen im C-886 verwendet werden:

Bestell-nummer	Beschreibung
C-863.20C885	Motion-Controller-Modul für DC-Motoren, 2 Achsen, für PIMotionMaster, PID-Regler
C-663.12C885	Mercury Step Schrittmotor-Controllermodul, 1 Achse, HD D-Sub 26, für PIMotionMaster, 1 Achse, geregelter und ungeregelter Betrieb, Unterstützung externer Sensoren
E-873.10C885	Q-Motion® Controllermodul für PIMotionMaster, 1 Achse, für Systeme mit piezoelektrischem Trägheitsantrieb

Nach dem Einbau von Slave-Modulen für optionale Einzelachsen kann das Abdecken freier Steckplätze erforderlich sein. Verfügbare Abdeckplatten:

Bestell-nummer	Beschreibung
C-885.AP1	Abdeckplatte für PIMotionMaster, 3 HE, 4 TE
C-885.AP2	Abdeckplatte für PIMotionMaster, 3 HE, 8 TE
C-885.AP4	Abdeckplatte für PIMotionMaster, 3 HE, 16 TE
C-885.AP8	Abdeckplatte für PIMotionMaster, 3 HE, 32 TE

➢ Wenden Sie sich bei Bestellungen an den Kundendienst (S. 267).

3.7 Kommandierbare Elemente

3.7.1 Kennungen für den Normalbetrieb des C-886

Die folgende Tabelle enthält die mit den Befehlen des GCS (S. 115) kommandier- oder auslesbaren Elemente.

<table>
<tr><th>Element</th><th>Anzahl</th><th>Kennung</th><th>Beschreibung</th></tr>
<tr><td>Logische Achse</td><td>6</td><td>X, Y, Z, U, V, W</td><td>Die logischen Achsen X bis W bilden die Translationen und Rotationen der Bewegungsplattform des parallelkinematischen Positionierers in der Firmware des C-886 ab.
Translationsachsen: X, Y und Z
Rotationsachsen: U, V und W</td></tr>
<tr><td>Logische Achse</td><td>2</td><td>S1, S2</td><td>Die logischen Achsen S1 und S2 bilden die Bewegung der optionalen Einzelachsen in der Firmware des C-886 ab.</td></tr>
<tr><td>Funktionsgenerator</td><td>8</td><td>1 bis 8</td><td>Jeder Funktionsgenerator (S. 79) ist einer logischen Achse fest zugeordnet:
<table>
<tr><th>Mechanik</th><th colspan="6">Parallelkinematischer Positionierer</th></tr>
<tr><td>Achse</td><td>X</td><td>Y</td><td>Z</td><td>U</td><td>V</td><td>W</td></tr>
<tr><td>Funktions-generator</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr>
</table>
<table>
<tr><th>Mechanik</th><th colspan="2">Einzelachsen</th></tr>
<tr><td>Achse</td><td>S1</td><td>S2</td></tr>
<tr><td>Funktionsgenerator</td><td>7</td><td>8</td></tr>
</table></td></tr>
<tr><td>Kurventabelle</td><td>100</td><td>1 bis 100</td><td>Die Kurventabellen enthalten die gespeicherten Daten (insgesamt 10.000 Punkte) für die Kurvenformen, die durch die Funktionsgeneratoren ausgegeben werden.
Der Wert des Parameters Number Of Waves (ID 0x1300010A) gibt die Anzahl der Kurventabellen (S. 79) an.</td></tr>
</table>

Element	Anzahl	Kennung	Beschreibung
Koordinatensystem	Unbegrenzt	Name wird beim Definieren des Koordinatensystems vergeben	Mit dem Controller können eigene Koordinatensysteme definiert und anstelle der werkseitig voreingestellten Koordinatensysteme verwendet werden. Weitere Informationen siehe "Bewegungen des Positionierers" (S. 29). **Konventionen für Namen von Koordinatensystemen:** Zulässige Zeichen: 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ_ Die Zeichenanzahl ist unbegrenzt. Der Name muss mit einem Buchstaben beginnen. Reservierte Namen, die nicht zum Definieren, Kopieren oder Löschen verwendet werden dürfen: HEXAPOD, PI_LEVELLING, PI_BASE, ZERO, 0, NULL, XML, KLF, KLF(USER), KLF(PI), KLD, KLD(USER), KLD(PI), KSB, KSB(USER), KSB(PI), KSD, KSF, KST, KSW Jeder Name darf nur einmal vorhanden sein. Ein vorhandenes Koordinatensystem, das nicht verwendet wird, wird beim Anlegen (Definieren, Erzeugen einer Kopie) eines Koordinatensystems mit demselben Namen überschrieben.
Datenrekordertabelle	36	1 bis 36	Die Datenrekordertabellen enthalten die aufgezeichneten Daten. Der C-886 hat 36 Datenrekordertabellen (Abfrage mit `TNR?` (S. 206)) mit maximal 8192 Datenpunkten pro Tabelle. Die Anzahl der Punkte pro Datenrekordertabelle kann mit dem Parameter ***Data Recorder Points Per Table*** (ID 0x16000201) eingestellt werden.
System	1	1	C-886 als Gesamtsystem.

3.7.2 Kennungen für direkten Zugriff auf Slave-Module

Die Kennungen in der folgenden Tabelle sind nur relevant, wenn der direkte Zugriff auf Slave-Module oder einzelne Antriebe erforderlich ist. Typischer Anwendungsfall: Die Slave-Module für die optionalen Einzelachsen müssen konfiguriert werden, Details siehe "Slave-Module für Einzelachsen konfigurieren" (S. 66).

Die Slave-Module für den parallelkinematischen Positionierer sind bei Auslieferung des C-886 verwendungsfertig konfiguriert. Das Ändern von Einstellungen oder die Kommandierung von Bewegungen über den direkten Zugriff auf diese Slave-Module kann zu Schäden am Positionierer führen.

➢ Greifen Sie nur nach Rücksprache mit PI direkt auf die Slave-Module für die Antriebe des parallelkinematischen Positionierers zu.

Grundsätzlich sind beim direkten Zugriff auf Slave-Module in jeder Befehlszeile die Kennungen des anzusprechenden Slave-Moduls und des anzusprechenden Antriebs erforderlich. Weitere Informationen finden Sie in der Dokumentation der Slave-Module (S. 6), insbesondere in "Empfänger- und Senderadresse". Anmerkung: Die Bezeichnungen in diesem Handbuch weichen von den Bezeichnungen in der Dokumentation der Slave-Module ab:

- Slave-Modul = anzusprechender Controller (Empfänger)
- Unterstützter Antrieb = Achse

INFORMATION

Bei direktem Zugriff auf die Slave-Module werden Befehle ohne Modul-Kennung oder mit der Modul-Kennung 1 an das Master-Modul des C-886 gesendet.

Wenn Sie mit der PC-Software PIMikroMove® direkt auf die Slave-Module zugreifen: PIMikroMove® übernimmt die Adressierung der Slave-Module, so dass im Fenster ***Command entry*** die Modul-Kennungen weggelassen werden müssen.

Die Zählung der Slave-Module beginnt mit dem Modul, das sich im Gehäuse des C-886 direkt neben dem Master-Modul befindet.

Element	**Anzahl**	**Kennung**	**Beschreibung**
Slave-Module für den parallelkinematischen Positionierer	3 / 6	1 bis 3 / 6 oder 2 bis 4 / 7	Die Anzahl der Slave-Module für den parallelkinematischen Positionierer hängt davon ab, wie viele Antriebe pro Modul unterstützt werden: ▪ Drei Slave-Module, wenn pro Modul zwei Antriebe unterstützt werden ▪ Sechs Slave-Module, wenn pro Modul ein Antrieb unterstützt wird Die Zählweise der Kennungen ist unterschiedlich: ▪ 1 bis 3 / 6 in der Antwort auf den Befehl SPA? ▪ 2 bis 4 / 7 in der Antwort auf den Befehl VER?

Element	Anzahl	Kennung	Beschreibung
Slave-Module für optionale Einzelachsen	maximal 2	5, 6 oder 8, 9	Die Kennungen der Slave-Module für optionale Einzelachsen hängen von der Anzahl der Slave-Module für den parallelkinematischen Positionierer ab: ▪ Wenn drei Module für den Positionierer vorhanden sind: 5, 6 ▪ Wenn sechs Module für den Positionierer vorhanden sind: 8, 9 Die maximale Anzahl der Slave-Module für optionale Einzelachsen hängt davon ab, wie viele Antriebe pro Modul unterstützt werden. Optionen: ▪ Ein Slave-Modul, das zwei Antriebe unterstützt ▪ Zwei Slave-Module, die jeweils einen Antrieb unterstützen
Antriebe pro Slave-Module	maximal 2	1, 2	Die Anzahl der unterstützten Antriebe pro Slave-Modul hängt vom Modul-Typ ab: 1 oder 2. In der Dokumentation der Slave-Module werden die Antriebe als "Achsen" bezeichnet.

3.8 Wichtige Komponenten der Firmware

Die Firmware des C-886 stellt die folgenden funktionalen Einheiten bereit:

Firmware-Komponente	Beschreibung
ASCII-Befehle	Die Kommunikation mit dem C-886 kann mit den Befehlen des PI General Command Set (GCS; Version 2.0) geführt werden. Der GCS ist von der Hardware (Controller, angeschlossene Versteller) unabhängig. Beispiele für die Verwendung des GCS: ▪ Bewegungen des Positionierers starten ▪ System- und Bewegungswerte abfragen ▪ Koordinatensystem definieren Eine Liste der verfügbaren Befehle finden Sie im Abschnitt "Befehlsübersicht" (S. 118).
Konfigurations-Parameter	▪ Schnittstellenparameter siehe "Kommunikation über TCP/IP-Schnittstelle herstellen" (S. 57) ▪ Weitere Parameter siehe "Anpassen von Einstellungen" (S. 247)

Firmware-Komponente	Beschreibung
Befehlsebenen	Die Verfügbarkeit von Befehlen und das Schreibrecht auf die Parameter werden durch Befehlsebenen festgelegt. Die aktuelle Befehlsebene kann mit dem Befehl CCL geändert werden. Dazu kann die Eingabe eines Kennworts erforderlich sein. Weitere Informationen finden Sie im Abschnitt "Anpassen von Einstellungen" (S. 247).
Datenrekorder	Der C-886 besitzt einen Echtzeit-Datenrekorder (S. 76) für Positionswerte und Statusregister-Bits.
Funktionsgenerator	Jede Achse kann von einem Funktionsgenerator gesteuert werden, der Kurvenformen ausgibt. Der Funktionsgenerator eignet sich besonders für dynamische Anwendungen, bei denen periodische Bewegungen der Achse ausgeführt werden (S. 79).
Makros	Der C-886 kann Makros speichern. Über die Makrofunktion können Befehlssequenzen festgelegt und dauerhaft gespeichert werden. Ein Startup-Makro kann festgelegt werden, das bei jedem Einschalten oder Neustart des C-886 ausgeführt wird. Dies vereinfacht den Betrieb ohne Verbindung zum PC. Weitere Informationen finden Sie im Abschnitt "Controllermakros" (S. 103).

Die Firmware kann mit einem Hilfsprogramm aktualisiert werden (S. 258).

3.9 Bewegungen des Positionierers

3.9.1 Unterstützte Bewegungsarten

Der C-886 unterstützt die folgenden Bewegungsarten:

Bewegungsart	Befehle zum Auslösen der Bewegung
Referenzfahrt (S. 140)	FRF
Punkt-zu-Punkt-Bewegung; Profilgenerator erzeugt das Dynamikprofil (S. 31)	MOV: Bewegung zu absoluter Zielposition Der Parameter ***Trajectory Source*** (ID 0x19001900) muss den Wert 0 haben (Standard).
	STE, IMP: Sprung oder Impuls starten, mit Datenaufzeichnung MVR: Bewegung zu relativer Zielposition MRT, MRW: Bewegt die angegebene Achse relativ im Tool- bzw. Work-Koordinatensystem. Siehe "Begriffserklärung" (S. 3) für weiterführende Erläuterungen zu Koordinatensystemen

Bewegungsart	Befehle zum Auslösen der Bewegung
Zyklische Übertragung von Zielpositionen (S. 33)	Aufeinander folgende MOV-Befehle Der Parameter ***Trajectory Source*** (ID 0x19001900) muss den Wert 1 = "Dynamikprofil wird durch aufeinander folgende `MOV`-Befehle festgelegt" haben.
Funktionsgenerator (S. 79)	WGO: Startet/stoppt die Funktionsgeneratorausgabe

INFORMATION

Der C-886 kann Befehlsfolgen als Controllermakros (S. 103) speichern und abarbeiten.

Über die Kommunikationsschnittstellen des C-886 können sämtliche Befehle gesendet werden, während auf dem C-886 ein Makro läuft. Der Makroinhalt und Befehle, die über die Kommunikationsschnittstellen empfangen werden, können sich gegenseitig überschreiben.

INFORMATION

Für Achsen mit inkrementellen Sensoren können Bewegungen erst nach einer erfolgreichen Referenzfahrt kommandiert werden (S. 140) (auch als "Initialisierung" bezeichnet).

Das Verhalten des Positionierers nach der Referenzfahrt wird durch die Parameter ***Behaviour After Reference Move*** (ID 0x07030401) und ***Target For Motion After Reference Move*** (ID 0x07030402) festgelegt. Entsprechend der Parameterwerte kann die Plattform z. B. nach der Referenzfahrt automatisch zur einer vorgegebenen Position bewegt werden.

- Wert des Parameters 0x07030401 = 0: Bewegungsplattform bleibt nach der Referenzfahrt in der Referenzposition.
- Wert des Parameters 0x07030401 = 1: Bewegungsplattform fährt nach der Referenzfahrt zur absoluten Zielposition, die durch Parameter 0x07030402 vorgegeben ist.

Für Achsen mit absolut messenden Sensoren ist keine Referenzfahrt erforderlich. Die Verwendung des Befehls FRF wird für diese Achsen trotzdem empfohlen. FRF startet für Achsen mit absolut messenden Sensoren **keine** Referenzfahrt, sondern setzt die Zielpositionen auf die aktuellen Positionswerte. Zusätzlich werden die oben beschriebenen Parameterwerte wirksam, so dass die Achsen z. B. zu einer definierten "Ausgangsposition" bewegt werden können.

Die folgenden Einstellungen gelten für alle Bewegungsarten:

Parameter	Beschreibung und mögliche Werte
Maximum System Velocity (Phys. Unit/s) 0x19001500	Maximale Systemgeschwindigkeit Dieser Parameter ist schreibgeschützt und wird vor Auslieferung auf den zum System gehörenden Positionierer abgeglichen. Obergrenze der mit `VLS` (S. 211) einstellbaren Geschwindigkeit für die Bewegungsplattform des Positionierers.

Parameter	Beschreibung und mögliche Werte
Minimum System Velocity (Phys. Unit/s) 0x19001501	Minimale Systemgeschwindigkeit Dieser Parameter ist schreibgeschützt und wird vor Auslieferung auf den zum System gehörenden Positionierer abgeglichen. Hängt ab von der kleinsten Schrittweite des Positionierers. Untergrenze der mit VLS einstellbaren Geschwindigkeit für die Bewegungsplattform des Positionierers.
Maximum System Acceleration (Phys.Unit/s2) 0x19001502	Maximale Systembeschleunigung Dieser Parameter ist schreibgeschützt und wird vor Auslieferung auf den zum System gehörenden Positionierer abgeglichen.

3.9.2 Profilgenerator für Punkt-zu-Punkt-Bewegungen

Für Punkt-zu-Punkt-Bewegungen legt der Profilgenerator des C-886 das Dynamikprofil fest.

INFORMATION

Während einer Punkt-zu-Punkt-Bewegung setzt ein beliebiger neuer Bewegungsbefehl die Zielposition auf einen neuen Wert, und die Bewegungsplattform fährt auf einer undefinierten Bahn sofort die neue Zielposition an.

Parameter für den Profilgenerator

Parameter	Beschreibung und mögliche Werte
Path Control Step Size (mm) 0x19001504	Schrittweite für die Berechnung des Dynamikprofils der Plattform Dieser Parameter ist schreibgeschützt und wird vor Auslieferung auf den zum System gehörenden Positionierer abgeglichen.
Trajectory Velocity (Phys. Unit/s) 0x19001510	Geschwindigkeit für die Bewegungsplattform des Positionierers Das Ändern der Geschwindigkeit mit dem Befehl VLS überschreibt den Wert des Parameters im flüchtigen Speicher.
Trajectory Acceleration (Phys. Unit/s2) 0x19001511	Beschleunigung für die Bewegungsplattform des Positionierers
Trajectory Jerk (Phys. Unit/s3) 0x19001512	Ruck für die Bewegungsplattform des Positionierers
Trajectory Source 0x19001900	Quelle des Dynamikprofils für MOV-Befehle Für Punkt-zu-Punkt-Bewegungen, die mit dem Befehl MOV ausgelöst werden, muss der Parameter den Wert 0 haben (Standard).

Befehle für den Profilgenerator

Befehl	Syntax	Funktion
VLS	VLS <SystemVelocity>	Setzt die Geschwindigkeit für die Bewegungsplattform des Positionierers. Wird begrenzt durch die Parameter ***Maximum System Velocity (Phys. Unit/s)*** (ID 0x19001500) und ***Minimum System Velocity (Phys. Unit/s)*** (ID 0x19001501).

Prüfung des vom Profilgenerator erzeugten Dynamikprofils

Wenn das Dynamikprofil für den Positionierer durch den Profilgenerator festgelegt wird, erfolgt vor dem Start jeder Bewegung eine Prüfung, ob die Bewegungsplattform die Stützstellen des berechneten Profils und die kommandierte Zielposition tatsächlich erreichen kann. Wenn eine Stützstelle oder die Zielposition nicht erreicht werden kann, wird die Bewegung nicht ausgeführt. Geprüft wird dabei Folgendes:

- Liegen die Stützstellen und die Zielposition außerhalb der Stellwegsgrenzen, die mit TMN? (S. 204) und TMX? (S. 205) oder TRA? (S. 206) abgefragt werden können?
- Sind die mit NLM (S. 188) und PLM (S. 190) gesetzten Verfahrbereichsgrenzen mit SSL (S. 199) aktiviert, und wenn ja, liegen die Stützstellen und die Zielposition außerhalb dieser Verfahrbereichsgrenzen?
- Sind die einzelnen Antriebe in der Lage, die Plattform zu den notwendigen Stützstellen und zur vorgegebenen Zielposition zu bewegen?

Der Befehl VMO? (S. 212) fragt ab, ob eine vorgegebene Zielposition erreicht werden kann.

3.9.3 Zyklische Übertragung von Zielpositionen

Bei der zyklischen Übertragung von Zielpositionen wird das Dynamikprofil durch aufeinander folgende MOV-Befehle vorgegeben.

HINWEIS

Zyklische Übertragung von Zielpositionen!

Beschleunigung / Abbremsung, Geschwindigkeit und Stetigkeit der Bewegung hängen während der zyklischen Übertragung von Zielpositionen von folgenden Faktoren ab:

- Zielpositionswerte
- Einhalten der Zykluszeit

Die Ausführung eines ungeeigneten Dynamikprofils kann den Positionierer verkippen. Verkippen kann den Positionierer und/oder die auf ihm angebrachte Last beschädigen.

➢ Beachten Sie deshalb bei der zyklischen Übertragung von Zielpositionen durch aufeinander folgende MOV-Befehle:
  - Die Bahn, die durch die Zielpositionen vorgegeben wird, muss mindestens zweimal stetig differenzierbar sein.
  - Bei der Ausführung des Dynamikprofils dürfen die maximal zulässige Geschwindigkeit und Beschleunigung des Positionierers **nicht** überschritten werden.
  - Für die Erzeugung der Zielpositionen und deren stetige Übergabe an den C-886 während der Bewegung wird die Verwendung eines geeigneten Programms empfohlen.

➢ Verwenden Sie den Zwischenspeicher des C-886, um sicherzustellen, dass die Zykluszeit eingehalten wird:
  - Legen Sie das Dynamikprofil vor Ausführung im Zwischenspeicher des C-886 ab. Setzen Sie dazu mit `SPA` den Parameter ***Trajectory Execution*** (ID 0x19001901) auf den Wert 1.
  - Für eine sinnvolle Verwendung des Zwischenspeichers erhöhen Sie den Wert des Parameters ***Threshold for Trajectory Execution*** (0x19001903) mit `SPA` (Standard = 1).

Parameter für zyklische Übertragung von Zielpositionen

Parameter	Beschreibung und mögliche Werte
Path Control Step Size (mm) 0x19001504	Schrittweite für die Berechnung des Dynamikprofils der Plattform Dieser Parameter ist schreibgeschützt und wird vor Auslieferung auf den zum System gehörenden Positionierer abgeglichen. Der Abstand der mit aufeinander folgenden `MOV`-Befehlen gesetzten Zielpositionen zueinander darf maximal so groß sein wie der Wert des Parameters 0x19001504, um ein Verkippen des Positionierers zu vermeiden.

Parameter	Beschreibung und mögliche Werte
Trajectory Source 0x19001900	Quelle des Dynamikprofils für MOV-Befehle Für die zyklische Übertragung von Zielpositionen durch aufeinander folgende MOV-Befehle muss der Parameter den Wert 1 haben.
Trajectory Execution 0x19001901	Ausführung des Dynamikprofils Bestimmt, wie das durch aufeinander folgende `MOV`-Befehle festgelegte Dynamikprofil ausgeführt wird: 0 = Dynamikprofil wird sofort ausgeführt (Standard) 1 = Dynamikprofil wird vor Ausführung in einem Zwischenspeicher abgelegt Dieser Parameter wird nur ausgewertet, wenn Parameter 0x19001900 den Wert 1 hat.
Maximum Number of Trajectory Points 0x19001902	Maximale Anzahl der Dynamikprofilpunkte Gibt die maximale Größe des Zwischenspeichers an. Dieser Parameter ist schreibgeschützt und wird nur ausgewertet, wenn die Parameter 0x19001900 und 0x19001901 jeweils den Wert 1 haben.
Threshold for Trajectory Execution 0x19001903	Schwellenwert für Ausführung des Dynamikprofils Bestimmt, wie viele Dynamikprofilpunkte im Zwischenspeicher abgelegt sein müssen (durch aufeinander folgende `MOV`-Befehle), bis die Ausführung des Dynamikprofils beginnt. Dieser Parameter wird nur ausgewertet, wenn die Parameter 0x19001900 und 0x19001901 jeweils den Wert 1 haben.
Current Number Threshold for Trajectory Execution 0x19001904	Zeigt die aktuelle Anzahl der Dynamikprofilpunkte im Zwischenspeicher an. Der Parameterwert ist immer 0, wenn das Dynamikprofil durch den Profilgenerator festgelegt wird oder das durch aufeinander folgende `MOV`-Befehle festgelegte Dynamikprofil sofort ausgeführt wird. Dieser Parameter ist schreibgeschützt.

Befehle für zyklische Übertragung von Zielpositionen

Befehl	Syntax	Funktion
`MOV`	MOV {<AxisID> <Position>}	Aufeinander folgende MOV-Befehle geben die einzelnen Zielpositionen vor. Andere Bewegungsbefehle und das Starten der Funktionsgeneratorausgabe sind nicht zulässig.

Befehl	Syntax	Funktion
SCT	SCT "T" <CycleTime>	Legt die Zykluszeit für das Ausführen des Dynamikprofils fest. Die Zykluszeit wird verwendet, um während der Bewegung die Geschwindigkeit so zu berechnen, dass die vorgegebenen Punkte des Dynamikprofils jeweils genau am Ende des Zeitintervalls erreicht werden (sofern unter Einhaltung der Grenzwerte für Geschwindigkeit und Beschleunigung möglich).
VLS	VLS <SystemVelocity>	Begrenzt die Geschwindigkeiten der einzelnen Antriebe.
#11	#11	Fragt den freien Speicherplatz des Zwischenspeichers ab, dessen Inhalt das Dynamikprofil des Positionierers festlegt, wenn die Parameter 0x19001900 und 0x19001901 jeweils den Wert 1 haben.

3.9.4 Koordinatensysteme

Positionsanzeige, Bewegungsrichtung und Drehpunkt für die Bewegungsplattform des Positionierers werden durch miteinander verkettete Koordinatensysteme bestimmt. Die Kette ist grundsätzlich wie folgt aufgebaut (Ausgangspunkt > Endpunkt): --> Koordinatensystem HEXAPOD, --> Korrektur-Koordinatensystem, --> Achsorientierungs-Koordinatensystem, --> Betriebs-Koordinatensystem.

Die Koordinatensysteme sind immer rechtshändige Systeme.

Das Koordinatensystem HEXAPOD bestimmt die grundsätzlichen Eigenschaften aller anderen Koordinatensysteme. HEXAPOD basiert auf der Konfigurationsdatei mit den Geometriedaten des Positionierers. Die Maßzeichnung im Handbuch des Positionierers zeigt jeweils die Lage des Koordinatensystems HEXAPOD.

Mit dem Controller können eigene Koordinatensysteme definiert und anstelle der werkseitig voreingestellten Koordinatensysteme verwendet werden.

Achsorientierungs- und Korrektur-Koordinatensystem passen, basierend auf HEXAPOD, grundlegende Eigenschaften des aktiven Betriebs-Koordinatensystems an und müssen in den meisten Anwendungen gar nicht oder nur einmalig anwenderspezifisch definiert und aktiviert werden.

Mit dem Betriebs-Koordinatensystem werden die Positionsanzeige, die Bewegungsrichtung und der Drehpunkt für die Bewegungsplattform des Positionierers an die Anwendung angepasst. Es ist auch möglich, --> Work-und Tool-Koordinatensysteme zu verwenden. In der Werkseinstellung ist das Betriebs-Koordinatensystem ZERO aktiv.

Die wichtigsten Befehle für die Arbeit mit anwenderdefinierten Koordinatensystemen:

- KSD (S. 169): Betriebskoordinatensystem durch Angabe von Offsetwerten für die Achsen X, Y, Z, U, V und W definieren
- KSF (S. 171): Betriebskoordinatensystem an der aktuellen Position der Bewegungsplattform des Positionierers definieren ("Home-Koordinatensystem")
- KLN (S. 162): Koordinatensysteme miteinander verketten
- KEN (S. 152): Koordinatensysteme aktivieren
- WPA (S. 224): Sichern der aktuell gültigen Einstellungen für Koordinatensysteme
- DPA (S. 133): Einstellungen für Parameter und Koordinatensysteme auf Werkseinstellungen zurücksetzen

INFORMATION

In PIMikroMove® können Koordinatensysteme komfortabel definiert, verkettet, aktiviert und gesichert werden.

Im Fenster ***Positioner Platform*** sind dafür die Schaltflächen ***Define Home Coordinate System (KSF)*** und ***Manage Coordinate Systems ...*** verfügbar. Auf der Karte ***Positioner 3D View*** werden der Positionierer und das aktive Koordinatensystem grafisch abgebildet.

Wenn im Hauptfenster **nicht** das Fenster ***Positioner Platform*** (standardmäßig angedockt) und die Karte ***Positioner 3D View*** angezeigt werden:

- ➢ Blenden Sie das Fenster ***Positioner Platform*** mit dem Menüeintrag ***C-886 > Show Positioner Platform Settings*** ein.
- ➢ Blenden Sie die Karte ***Positioner 3D View*** mit dem Menüeintrag ***C-886 > Positioner 3D View > Show*** ein.

Weitere Informationen zu Koordinatensystemen und zum Work-und-Tool-Konzept finden Sie in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007) und in "Begriffserklärung" (S. 3).

3.9.5 Rotationen

Rotationen erfolgen um den Drehpunkt. Eine beliebige Rotation im Raum wird aus den Einzelrotationen in der Reihenfolge U-> V -> W berechnet. Dies geschieht unabhängig davon, ob die Werte für U, V und W mit dem aktuellen Befehl explizit gegeben wurden oder aus den vorangegangenen Befehlen resultieren.

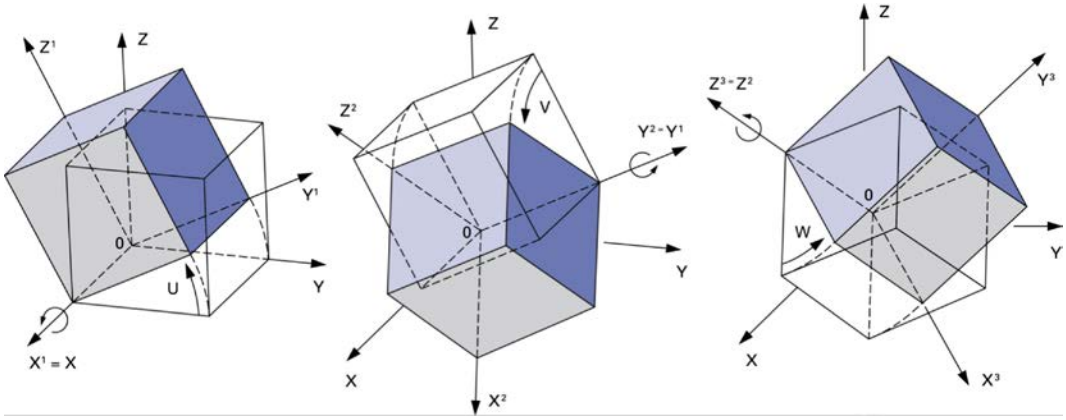


Abbildung 6: Reihenfolge der Elementardrehungen der Hexapod-Plattform beim Erreichen einer Position (von links nach rechts)

Der Drehpunkt beschreibt das Rotationszentrum (Schnittpunkt der Rotationsachsen U, V und W).

Der Drehpunkt bewegt sich immer zusammen mit der Plattform.

In Abhängigkeit vom aktiven --> Betriebs-Koordinatensystem kann der Drehpunkt mit dem Befehl `SPI` aus dem Ursprung des Koordinatensystems heraus in X- und/oder Y- und/oder Z-Richtung verschoben werden. Der mit dem Befehl `SPI` verschiebbare Drehpunkt wird auch als „Pivotpunkt“ bezeichnet.

Der mit SPI definierte Pivotpunkt wird für Rotationen nur verwendet und kann nur geändert werden, wenn als Betriebs-Koordinatensystem das Koordinatensystem ZERO oder ein Koordinatensystem vom Typ KSF aktiv ist.

Weitere Informationen zur Berechnung von Translationen und Rotationen durch den C-886 finden Sie in der Technical Note "Bewegungen des Hexapods - Position und Orientierung im Raum, Drehpunkt" (C887T0021).

3.10 Kommunikationsschnittstellen

Verfügbare Kommunikationsschnittstellen

Der C-886 kann über folgende Schnittstellen mit ASCII-Befehlen gesteuert werden:

- TCP/IP
- USB

3.11 PC-Softwareübersicht

Die folgende Tabelle zeigt die PC-Software, die sich auf der C-886.CD befindet. Die angegebenen Betriebssysteme stehen für folgende Versionen:

- Windows: Vista Service Pack 1, Windows 7, 8 und 10 (32 Bit, 64 Bit)
- Linux: Kernel 2.6, GTK 2.0, glibc 2.4 (für die Entwicklung der PC-Software verwendete Konfiguration)

PC-Software	Unterstützendes Betriebssystem	Kurzbeschreibung	Empfohlene Verwendung
PIMikroMove®	Windows	Grafische Benutzerschnittstelle für Windows, mit der der C-886 und andere Controller von PI bedient werden können: ▪ Das System kann ohne Programmieraufwand gestartet werden ▪ Grafische Darstellung der Bewegungen ▪ Makrofunktionalität zum Abspeichern von Befehlsfolgen auf dem PC (Hostmakros) ▪ Komplette Umgebung für die Befehlseingabe, zum Ausprobieren von verschiedenen Befehlen Für die Bedienung von PIMikroMove® sind keine Befehlskenntnisse erforderlich.	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.
Dynamische Programm-bibliothek für GCS	Windows, Linux	Ermöglicht die Software-Programmierung für den C-886 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 LabVIEW-Treiber benötigt.
PI Hexapod 3D Library	Windows	Programmbibliothek, die folgende Funktionen für PIMikroMove® bereitstellt: ▪ 3D-Darstellung der Bewegungen des parallelkinematischen Positionierers und des aktiven Koordinatensystems ▪ Konfiguration und Verwaltung von anwenderdefinierten Koordinatensystemen	Wird für PIMikroMove® benötigt.
LabVIEW-Treiber	Windows, Linux	LabVIEW ist eine Software für die Datenerfassung und Prozesssteuerung (von National Instruments separat zu beziehen). Die C-886-LabVIEW-Software ist eine Sammlung von Virtual-Instrument-Treibern (VI-Treiber) für den C-886-Controller. Die Treiber unterstützen das GCS.	Für Anwender, die LabVIEW zur Programmierung ihrer Anwendung verwenden möchten.

PC-Software	Unterstützendes Betriebssystem	Kurzbeschreibung	Empfohlene Verwendung
LabVIEW Merge Tool	Windows	Mit dem LabVIEW Merge Tool können produktspezifische LabVIEW Treiber von PI miteinander kombiniert werden.	Für Anwender, die mehrere Produkte von PI unter Verwendung von LabVIEW gemeinsam betreiben wollen.
MATLAB-Treiber	Windows	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.
PIPython	Windows, Linux, OS X	Python ist eine Programmiersprache, die auch als Skriptsprache genutzt wird (als Open-Source-Software separat zu beziehen). PIPython ist eine Sammlung von Python-Modulen, die das GCS unterstützen. Die PIPython-Module können mit Python 2.7+ und 3.4+ eingesetzt werden. Über Sockets ist auch der Einsatz auf weiteren Betriebssystemen möglich.	Für Anwender, die Python zur Programmierung von Skripten für ihre Anwendung verwenden möchten.
PITerminal	Windows	Terminalprogramm, das für nahezu alle PI-Controller verwendet werden kann (siehe die Beschreibung des Fensters ***Command Entry*** im PIMikroMove®-Benutzerhandbuch).	Für Anwender, die die Befehle des GCS direkt an den Controller senden möchten.
PI Update Finder	Windows	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.
PI Hexapod DataFiles	Windows	Konfigurationsdateien für die grafische Darstellung der parallelkinematischen Positionierer in der PC-Software	Werden für die PI Hexapod 3D Library benötigt.
PIStages3Editor	Windows	Programm zum Öffnen und Editieren von Verstellerdatenbanken des Formats .db.	Für Anwender, die optionale Einzelachsen verwenden.
PIFirmware-Manager	Windows	Programm zur Aktualisierung der Firmware des C-886.	Für Anwender, die die Firmware aktualisieren möchten.
USB-Treiber	Windows	Treiber für die USB-Schnittstelle	Für Anwender, die den Controller über die USB-Schnittstelle an den PC anschließen möchten.

4 Auspacken

1. Packen Sie den C-886 vorsichtig aus.
2. Vergleichen Sie die erhaltene Lieferung mit dem Inhalt laut Vertrag und mit der Packliste.
3. Überprüfen Sie den Inhalt auf Anzeichen von Schäden. Bei Anzeichen von Beschädigungen oder fehlenden Teilen wenden Sie sich sofort an PI.
4. Bewahren Sie das komplette Verpackungsmaterial auf für den Fall, dass das Produkt zurückgeschickt werden muss.

5 Installation

In diesem Kapitel

PC-Software installieren 43
Belüftung sicherstellen 46
C-886 an Schutzleiter anschließen 46
Slave-Module im C-886 installieren 47
Netzteil an C-886 anschließen 48
Antriebe anschließen 49
PC anschließen 50

5.1 PC-Software installieren

Die Kommunikation zwischen dem C-886 und einem PC ist zur Konfiguration des C-886 und zur Bewegungskommandierung mit den Befehlen des GCS notwendig. Dafür stehen verschiedene PC-Software-Anwendungen zur Verfügung.

5.1.1 Erstinstallation ausführen

Zubehör

- PC mit Betriebssystem Windows (7, 8, 10; 32 Bit, 64 Bit) oder Linux und mindestens 30 MB freiem Speicherplatz
- Produkt-CD (im Lieferumfang)

PC-Software auf Windows installieren

1. Starten Sie den Installationsassistenten, indem Sie im Installationsverzeichnis (Hauptverzeichnis der CD) auf die Datei ***PI_C-886.CD_Setup.exe*** doppelklicken.

   Das Fenster ***InstallShield Wizard*** für die Installation von Programmen und Handbüchern für den C-886 öffnet sich.

2. Folgen Sie den Anweisungen am Bildschirm.

   Sie können zwischen der Standardinstallation (*Complete*) und der benutzerdefinierten Installation (*Custom*) wählen.

   Bei der Standardinstallation (empfohlen) werden alle Komponenten installiert. Dazu gehören unter anderem:

- LabVIEW-Treiber
  Ausnahme: Für einige Controller von PI wird die Komponente *Analog LabVIEW drivers* bereitgestellt. Diese Komponente ist nur über die benutzerdefinierte Installation verfügbar.
- Dynamische Programmbibliothek für GCS
- PIMikroMove®
- PC-Software zum Aktualisieren der Firmware des C-886
- PI Update Finder zum Aktualisieren der PC-Software
- Für Controller, die eine USB-Schnittstelle zur Kommunikation mit dem PC haben: USB-Treiber

Bei der benutzerdefinierten Installation haben Sie die Möglichkeit, einzelne Komponenten von der Installation auszuschließen.

PC-Software auf Linux installieren

1. Entpacken Sie das tar-Archiv aus dem Verzeichnis /linux der Produkt-CD 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. Geben Sie ./INSTALL ein, um die Installation zu starten.
   Achten Sie bei der Befehlseingabe auf Groß-/Kleinschreibung.
5. Folgen Sie den Anweisungen am Bildschirm.

Sie können einzelne Komponenten zur Installation auswählen.

5.1.2 Updates installieren

Die PC-Software wird von PI ständig verbessert.

➢ Installieren Sie immer die neueste Version der PC-Software und der Verstellerdatenbank.

Voraussetzungen

✓ Aktive Verbindung zum Internet.

✓ Wenn Ihr PC ein Windows-Betriebssystem verwendet:

- Sie haben den PI Update Finder von der Produkt-CD installiert (S. 43).
- Sie haben die Technical Note A000T0028 für den PI Update Finder parat. Sie finden das Dokument auf der Produkt-CD.

- Wenn der zu aktualisierende PC **nicht** direkt mit dem Internet verbunden ist: Sie haben die Technical Note A000T0032 für den PI Update Finder parat. Sie finden das Dokument auf der Produkt-CD.

✓ Wenn Ihr PC ein Linux-Betriebssystem verwendet:

- Sie haben Benutzername und Kennwort für den C-886 parat. Beide Angaben finden Sie in der Datei "xxx_Releasenews.pdf" (x_x_x: Versionsnummer der CD) im Ordner \Manuals auf der Produkt-CD.

PC-Software auf Windows aktualisieren

➢ Verwenden Sie den PI Update Finder:

- Wenn der zu aktualisierende PC direkt mit dem Internet verbunden ist: Folgen Sie den Anweisungen in der Technical Note A000T0028 (TECHNICAL_NOTE_PI_UPDATE_FINDER_xx.pdf).
- Wenn der zu aktualisierende PC **nicht** direkt mit dem Internet verbunden ist: Folgen Sie den Anweisungen in der Technical Note A000T0032.

PC-Software auf Linux aktualisieren

1. Öffnen Sie die Webseite www.pi.de.
2. Klicken Sie auf ***Login***.
3. Melden Sie sich mit dem Benutzernamen (username) und dem Kennwort (password) aus der Datei "xxx_Releasenews.pdf" von der Produkt-CD an.
4. Klicken Sie auf ***Suche***.
5. Geben Sie die Produktnummer bis zum Punkt (z. B. C-886) in das Suchfeld ein.
6. Klicken Sie auf ***Suche starten*** oder drücken Sie die Enter-Taste.
7. Öffnen Sie über die Liste der Suchergebnisse die entsprechende Produktdetailseite:
   a) Wenn notwendig: Scrollen Sie in der Liste nach unten.
   b) Wenn notwendig: Klicken Sie am Ende der Liste auf ***Weitere Suchergebnisse laden***.
   c) Klicken Sie in der Liste auf das entsprechende Produkt.
8. Scrollen Sie auf der Produktdetailseite nach unten zum Bereich ***Downloads***.

   Die Softwaredateien werden unter ***Software Downloads*** angezeigt.
9. Klicken Sie auf die Archivdatei "CD Mirror" oder auf den zugehörigen Download-Link.
10. Wählen Sie in der folgenden Abfrage die Option zum Speichern der Datei auf Ihrem PC.

    Wenn Sie nichts anderes angeben, wird die Archivdatei "CD Mirror" im Standard-Downloadverzeichnis Ihres PC gespeichert.
11. Entpacken Sie die Archivdatei in ein separates Installationsverzeichnis.
12. Wechseln Sie im Verzeichnis mit den entpackten Dateien in das Unterverzeichnis ***linux***.

13. Entpacken Sie die Archivdatei im Verzeichnis ***linux***, indem Sie in der Konsole den Befehl tar -xvpf <Name der Archivdatei> eingeben.
14. Lesen Sie die Begleitinformationen zum Software-Update durch (Readme-Datei und/oder Datei "xxx_Releasenews.pdf" und entscheiden Sie, ob das Update für Ihre Anwendung sinnvoll ist.
    - Wenn nein: Beenden Sie den Update-Vorgang.
    - Wenn ja: Führen Sie die nachfolgenden Schritte aus.
15. Melden Sie sich am PC als Superuser (Root-Rechte) an.
16. Installieren Sie das Update.

INFORMATION

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

➢ Wenden Sie sich an unseren Kundendienst (S. 267).

5.2 Belüftung sicherstellen

Hohe Temperaturen können den C-886 überhitzen.

➢ Installieren Sie den C-886 mit einem Abstand von mindestens 10 cm zur Ober- und Rückseite und mindestens 5 cm zu dessen Seiten. Wenn dies nicht möglich ist, kühlen Sie die Umgebung ausreichend.

➢ Sorgen Sie für ausreichende Belüftung am Aufstellungsort.

➢ Halten Sie die Umgebungstemperatur auf einem unkritischen Wert (<50 °C).

5.3 C-886 an Schutzleiter anschließen

INFORMATION

➢ Beachten Sie die jeweils geltenden Normen für die Schutzleiterbefestigung.

Voraussetzungen

✓ Der C-886 ist ausgeschaltet, d. h. das Netzteil ist **nicht** am C-886 angeschlossen.

Werkzeug und Zubehör

- Geeigneter Schutzleiter:
  - Kabelquerschnitt ≥0,75 mm^2
  - Übergangswiderstand <0,1 Ohm bei 25 A an allen für die Schutzleitermontage relevanten Stellen

- Befestigungsmaterial für den Schutzleiter, sitzt bei Auslieferung des C-886 in folgender Reihenfolge auf dem Schutzleiteranschluss (Gewindebolzen), ausgehend vom Gehäuse:
  - Zahnscheibe
  - Mutter
  - Unterlegscheibe
  - Sicherungsscheibe
  - Mutter
- Geeigneter Schraubenschlüssel

C-886 an Schutzleiter anschließen

1. Wenn nötig, befestigen Sie einen geeigneten Kabelschuh am Schutzleiter.
2. Entfernen Sie die äußere Mutter vom Schutzleiteranschluss auf der Rückwand des C-886 (mit ⏚ gekennzeichneter Gewindebolzen (S. 18)).
3. Schließen Sie den Schutzleiter an:
   a) Schieben Sie den Kabelschuh des Schutzleiters auf den Gewindebolzen.
   b) Schrauben Sie die Mutter auf den Gewindebolzen. Der Kabelschuh des Schutzleiters wird auf diese Weise zwischen der Zahnscheibe und der Mutter eingeklemmt.
   c) Ziehen Sie die Mutter mit mindestens drei Umdrehungen und einem Drehmoment von 1,2 Nm bis 1,5 Nm fest.

5.4 Slave-Module im C-886 installieren

INFORMATION

Im C-886 können Slave-Module für bis zu zwei optionale Einzelachsen installiert werden.

Wenn Sie die optionalen Einzelachsen mit dem Positionierer-System bestellt haben, konfiguriert PI den C-886 entsprechend Ihrer Bestellung, so dass die entsprechenden Slave-Module bereits installiert sind.

Voraussetzung

✓ Der C-886 ist **nicht** über das Netzkabel an der Steckdose angeschlossen.

Werkzeug und Zubehör

- Slave-Modul, das als optionales Zubehör (S. 24) bestellt wurde
- Wenn erforderlich: Abdeckplatten für freie Steckplätze, erhältlich als optionales Zubehör (S. 24)
- Kreuzschlitzschraubendreher

Slave-Module im C-886 installieren

1. Entfernen Sie die Abdeckung des Steckplatzes, der sich im Gehäuse des C-886 direkt neben dem letzten belegten Steckplatz befindet.
2. Setzen Sie das Slave-Modul in den C-886 ein:
   a) Schieben Sie das Slave-Modul **in korrekter Orientierung** vorsichtig in den Steckplatz.
   b) Wenn Sie einen Widerstand spüren, schieben Sie das Slave-Modul weiter, bis es in der Anschlussleiste des Gehäuses einrastet.
   c) Prüfen Sie den korrekten Sitz des Slave-Moduls.
   d) Befestigen Sie das Slave-Modul mit den Kreuzschlitzschrauben am Gehäuse des C-886.
3. Wenn im C-886 freie Steckplätze ohne Abdeckung vorhanden sind, schließen Sie diese Steckplätze mit einer passenden Abdeckplatte.

5.5 Netzteil an C-886 anschließen

Voraussetzungen

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

✓ Der C-886 ist in der Nähe der Stromversorgung installiert, damit der Netzstecker schnell und einfach vom Netz getrennt werden kann.

Werkzeug und Zubehör

- Mitgeliefertes 24-V-Weitbereichsnetzteil (für Netzspannungen zwischen 100 und 240 Volt Wechselspannung bei 50 oder 60 Hz)
- Alternativ: ausreichend bemessenes Netzteil
- Mitgelieferter Adapter für den Netzteil-Anschluss (S. 23); Molex Mini Fit 6-pol. (m) auf Aderendhülsen 4 mm^2
- Alternativ: ausreichend bemessener Adapter
- Mitgeliefertes Netzkabel
- Alternativ: ausreichend bemessenes Netzkabel

Mitgeliefertes Netzteil an den C-886 anschließen

➢ Verbinden Sie den Adapter mit dem 24-V-Anschluss des C-886:
   a) Führen Sie die Aderendhülse des schwarzen Kabels in die Durchführungsklemme **GND** ein.
   b) Führen Sie die Aderendhülse des roten Kabels in die Durchführungsklemme **+24V** ein.
   c) Sichern Sie die Steckverbindungen mit den integrierten Schrauben gegen unbeabsichtigtes Abziehen.

- ➢ Verbinden Sie den Molex-Stecker des Adapters mit der Molex-Buchse des Netzteils.
  - – Stellen Sie sicher, dass die Verriegelung der Steckverbindung eingerastet ist.
- ➢ Verbinden Sie das Netzkabel mit dem Netzteil.

5.6 Antriebe anschließen

HINWEIS

Schäden bei Anschluss eines ungeeigneten Antriebstyps!
Das Anschließen eines ungeeigneten Antriebstyps kann zu Schäden am Antrieb oder C-886 führen.

- ➢ Schließen Sie an die Slave-Module des C-886 nur einen geeigneten Antriebstyp an.

INFORMATION

C-886 und Positionierer werden als vorkonfiguriertes System ausgeliefert.

- ➢ Wenn durch die Beschriftung von C-886 und/oder Positionierer eine Zuordnung der Anschlüsse vorgegeben ist, halten Sie diese Zuordnung beim Anschließen des Positionierers ein.

Voraussetzung

- ✓ Der C-886 ist ausgeschaltet, d. h. das Netzteil ist **nicht** über das Netzkabel an der Steckdose angeschlossen.
- ✓ Wenn Sie optionale Einzelachsen anschließen wollen: Die entsprechenden Slave-Module sind im C-886 installiert (S. 47).
- ✓ Sie haben die Benutzerhandbücher aller anzuschließenden Antriebe (parallelkinematischer Positionierer, optionale Einzelachsen) gelesen und verstanden.
- ✓ Sie haben die anzuschließenden Antriebe gemäß den Anweisungen in den entsprechenden Benutzerhandbüchern installiert.

Werkzeug und Zubehör

- Parallelkinematischer Positionierer mit geeignetem Antriebstyp
- Wenn Sie optionale Einzelachsen verwenden wollen: Versteller mit geeignetem Antriebstyp

Antriebe anschließen

1. Schließen Sie die Antriebe an den Buchsen der Slave-Module an.
2. Sichern Sie die Steckverbindungen mit den integrierten Schrauben gegen unbeabsichtigtes Abziehen.

5.7 PC anschließen

Die Kommunikation zwischen dem C-886 und einem PC ist zur Konfiguration des C-886 und zur Bewegungskommandierung mit den Befehlen des GCS notwendig. Der C-886 verfügt dazu über folgende Schnittstellen:

- TCP/IP-Schnittstelle
- USB-Schnittstelle

In diesem Abschnitt erfahren Sie, wie Sie die entsprechenden Kabelverbindungen zwischen C-886 und PC sowie in einem TCP/IP-Netzwerk herstellen. Alle weiteren Schritte, die für die Herstellung der Kommunikation zwischen C-886 und PC erforderlich sind, sind in den folgenden Abschnitten beschrieben:

- "Kommunikation über TCP/IP-Schnittstelle herstellen" (S. 57)
- "Kommunikation über USB herstellen" (S. 64)

5.7.1 C-886 über die TCP/IP-Schnittstelle anschließen

Voraussetzungen

✓ Wenn der C-886 direkt an den PC angeschlossen werden soll:
Der PC verfügt über eine freie RJ45-EthernetAnschlussbuchse.

✓ Wenn C-886 und PC gemeinsam in einem Netzwerk betrieben werden sollen:
Für den C-886 ist ein freier Zugangspunkt zum Netzwerk vorhanden, gegebenenfalls ist dazu ein geeigneter Hub oder Switch an das Netzwerk angeschlossen.

Werkzeug und Zubehör

- Wenn der C-886 direkt an den PC angeschlossen werden soll: Crossover-Netzwerkkabel
- Wenn der C-886 an einen Netzwerk-Zugangspunkt angeschlossen werden soll: Straight-Through-Netzwerkkabel

C-886 direkt an den PC anschließen

➢ Verbinden Sie die RJ45-Buchse des C-886 über das Crossover-Netzwerkkabel mit der RJ45-Ethernet-Anschlussbuchse des PC.

C-886 an das Netzwerk anschließen, in dem sich auch der PC befindet

➢ Verbinden Sie die RJ45-Buchse des C-886 über das Straight-Through-Netzwerkkabel mit dem Netzwerk-Zugangspunkt.

5.7.2 C-886 über die USB-Schnittstelle anschließen

Voraussetzungen

✓ Der PC verfügt über eine freie USB-Schnittstelle.

Werkzeug und Zubehör

- USB-Kabel (Typ A auf Mini-B) zur Verbindung mit dem PC (000036360 im Lieferumfang)

C-886 an den PC anschließen

➢ Verbinden Sie die USB-Buchse des C-886 und die USB-Schnittstelle des PC mit dem USB-Kabel.

6 Inbetriebnahme

In diesem Kapitel

Allgemeine Hinweise zur Inbetriebnahme 53
C-886 einschalten 56
Kommunikation über TCP/IP-Schnittstelle herstellen 57
Kommunikation über USB-Schnittstelle herstellen 64
Slave-Module für Einzelachsen konfigurieren 66
Bewegungen starten 70

6.1 Allgemeine Hinweise zur Inbetriebnahme

VORSICHT

Stromschlaggefahr bei fehlendem Schutzleiter!

Bei fehlendem oder nicht ordnungsgemäß angeschlossenem Schutzleiter können im Falle eines Fehlers oder Defekts gefährliche Berührungsspannungen am C-886 entstehen. Wenn Berührungsspannungen vorhanden sind, kann das Berühren des C-886 zu leichten Verletzungen durch Stromschlag führen.

- Schließen Sie den C-886 vor Inbetriebnahme an einen Schutzleiter an (S. 46).
- Entfernen Sie den Schutzleiter **nicht** während des Betriebs.
- Wenn der Schutzleiter vorübergehend entfernt werden muss (z. B. bei Umbauten), schließen Sie den C-886 vor erneuter Inbetriebnahme wieder an den Schutzleiter an.

VORSICHT

Quetschgefahr durch bewegte Teile!

Zwischen den bewegten Teilen des Positionierers und einem feststehenden Teil oder Hindernis besteht die Gefahr von leichten Verletzungen durch Quetschung.

- Halten Sie Ihre Finger von Bereichen fern, in denen sie von bewegten Teilen erfasst werden können.

HINWEIS

Falsche Konfiguration des C-886!

Die Konfiguration des C-886 muss auf den Positionierer abgestimmt sein. Bei Verwendung einer falschen Konfiguration kann der Positionierer durch unkontrollierte Bewegungen oder Kollisionen beschädigt werden. Bei Auslieferung ist der C-886 für den verwendeten Positionierer konfiguriert.

- Prüfen Sie die Konfiguration: Nachdem Sie die Kommunikation über TCP/IP (S. 57) oder USB (S. 64) hergestellt haben, senden Sie den Befehl `CST?`. Die Antwort zeigt an, auf welchen Positionierer der C-886 abgestimmt ist.
- Betreiben Sie den Positionierer nur mit einem C-886, dessen Konfigurationsdaten auf den Positionierer abgestimmt sind.
- Ändern Sie die Einstellungen der Slave-Module für die Antriebe des Positionierers nur nach Rücksprache mit PI.

HINWEIS

Schäden durch Kollisionen!

Kollisionen können den Positionierer, die zu bewegende Last und die Umgebung beschädigen.

- Stellen Sie sicher, dass im Arbeitsraum des Positionierers keine Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich sind.
- Platzieren Sie keine Gegenstände in Bereichen, in denen sie von bewegten Teilen erfasst werden können.
- Halten Sie bei einer Fehlfunktion des Controllers die Bewegung sofort an.

HINWEIS

Schäden durch ungewollte Positionsänderungen!

Die maximale Haltekraft bei ausgeschaltetem Servomodus basiert auf der Selbsthemmung der Antriebe und kann geringer ausfallen als die maximale Belastbarkeit bei eingeschaltetem Servomodus (siehe Handbuch des Positionierers).

Wenn die tatsächliche Belastung des Positionierers die auf der Selbsthemmung der Antriebe basierende maximale Haltekraft überschreitet, können in folgenden Fällen ungewollte Positionsänderungen des Positionierers auftreten:

- Ausschalten des C-886
- Neustart des C-886
- Ausschalten des Servomodus für die Achsen derBewegungsplattform des Positionierers

Dadurch sind Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich. Kollisionen können den Positionierer, die zu bewegende Last oder die Umgebung beschädigen.

- Stellen Sie sicher, dass die tatsächliche Belastung der Bewegungsplattform des Positionierers die auf der Selbsthemmung der Antriebe basierende maximale Haltekraft nicht überschreitet, bevor Sie den Servomodus ausschalten, den C-886 neu starten oder ausschalten.

HINWEIS

Schäden durch Kollisionen während der Referenzfahrt!

Während einer Referenzfahrt bewegt sich der Positionierer auf unvorhersehbare Weise.

Dadurch sind Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich. Kollisionen können den Positionierer, die zu bewegende Last und die Umgebung beschädigen.

- Stellen Sie sicher, dass während der Referenzfahrt des Positionierers keine Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich sind.
- Platzieren Sie keine Gegenstände in Bereichen, in denen sie während der Referenzfahrt von bewegten Teilen erfasst werden können.
- Kommandieren Sie nach erfolgreicher Referenzfahrt die entsprechende Zielposition, um von einer beliebigen Position zur Referenzposition (Standard: Nullposition) zurückzukehren. Starten Sie **nicht** eine erneute Referenzfahrt.

HINWEIS

Schäden durch unkontrollierte Bewegung des Positionierers!

Geschwindigkeit und Beschleunigung der Bewegungsplattform des Positionierers werden in folgenden Fällen **nicht** vom C-886 vorgegeben:

- Zyklische Übertragung von Zielpositionen (der Parameter ***Trajectory Source*** (ID 0x19001900) hat den Wert 1)
- Der Positionierer (Achsen X, Y, Z, U, V, W) bewegt sich noch, während ein neuer Bewegungsbefehl gesendet wird. Die bisherige Zielposition wird dabei überschrieben, ohne dass Geschwindigkeit und Beschleunigung der Bewegungsplattform des Positionierers neu berechnet werden.

Die Plattform des Positionierers bewegt sich dann auf einer undefinierten Bahn. Auf dieser undefinierten Bahn sind Kollisionen mit der Umgebung des Positionierers möglich. Kollisionen können den Positionierer, die zu bewegende Last und die Umgebung beschädigen.

Wenn Sie Bewegungen durch zyklische Übertragung von Zielpositionen auslösen (S. 33):

- Setzen Sie mit den aufeinanderfolgenden `MOV`-Befehlen nur Zielpositionen, deren Abstand zueinander maximal so groß wie der Wert des Parameters ***Path Control Step Size*** (ID 0x19001504) ist.

Wenn Sie Punkt-zu-Punkt-Bewegungen mit Bewegungsbefehlen kommandieren (S. 29):

- Vermeiden Sie das Senden neuer Zielpositionen, wenn sich der Positionierer (Achsen X, Y, Z, U, V, W) noch bewegt.
- Wenn neue Zielpositionen gesendet werden müssen, während sich der Positionierer noch bewegt (Achsen X, Y, Z, U, V, W): Setzen Sie mit Bewegungsbefehlen nur Zielpositionen, die von der aktuellen Position maximal um den Wert des Parameters ***Path Control Step Size*** (ID 0x19001504) abweichen.

INFORMATION

Die Kommunikation zwischen dem C-886 und einem PC kann zur Konfiguration des C-886 und zur Bewegungskommandierung mit den Befehlen des GCS genutzt werden.

- Über die USB-Schnittstelle ist die Kommunikation ohne weitere Einstellungen möglich.
- Für die Kommunikation über TCP/IP kann einmalig die Anpassung der Schnittstellenparameter erforderlich sein.

INFORMATION

Die Kommunikationsschnittstellen des C-886 (TCP/IP, USB) sind gleichzeitig aktiv. Befehle werden in der Reihenfolge abgearbeitet, in der die kompletten Befehlszeilen eintreffen. Die gleichzeitige Verwendung mehrerer Kommunikationsschnittstellen kann jedoch Probleme mit der PC-Software verursachen.

➢ Verwenden Sie immer nur eine Schnittstelle des C-886.

INFORMATION

Für Achsen mit inkrementellen Sensoren können Bewegungen erst nach einer erfolgreichen Referenzfahrt kommandiert werden (S. 140) (auch als "Initialisierung" bezeichnet).

Das Verhalten des Positionierers nach der Referenzfahrt wird durch die Parameter ***Behaviour After Reference Move*** (ID 0x07030401) und ***Target For Motion After Reference Move*** (ID 0x07030402) festgelegt. Entsprechend der Parameterwerte kann die Plattform z. B. nach der Referenzfahrt automatisch zur einer vorgegebenen Position bewegt werden.

- Wert des Parameters 0x07030401 = 0: Bewegungsplattform bleibt nach der Referenzfahrt in der Referenzposition.
- Wert des Parameters 0x07030401 = 1: Bewegungsplattform fährt nach der Referenzfahrt zur absoluten Zielposition, die durch Parameter 0x07030402 vorgegeben ist.

Für Achsen mit absolut messenden Sensoren ist keine Referenzfahrt erforderlich. Die Verwendung des Befehls FRF wird für diese Achsen trotzdem empfohlen. FRF startet für Achsen mit absolut messenden Sensoren **keine** Referenzfahrt, sondern setzt die Zielpositionen auf die aktuellen Positionswerte. Zusätzlich werden die oben beschriebenen Parameterwerte wirksam, so dass die Achsen z. B. zu einer definierten "Ausgangsposition" bewegt werden können.

6.2 C-886 einschalten

Voraussetzungen

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

✓ Der C-886 wurde ordnungsgemäß installiert (S. 43).

C-886 einschalten

➢ Verbinden Sie das Netzkabel des Netzteils mit der Steckdose.

Der C-886 startet die Firmware. Der Startvorgang dauert circa 10 Sekunden. Nach dem Ende des Startvorgangs leuchtet die LED **Power** grün.

6.3 Kommunikation über TCP/IP-Schnittstelle herstellen

Anpassung der Schnittstellenparameter

Vor der Herstellung der Kommunikation kann einmalig die Anpassung der werkseitigen Einstellungen der Schnittstellenparameter erforderlich sein. Mit dem Befehl `IFS` (S. 147) können im permanenten Speicher des C-886 folgende Schnittstellenparameter für die TCP/IP-Kommunikation angepasst werden:

Schnittstellen-parameter	Werkseitige Einstellung	Bemerkung
Standard-IP-Adresse (IPADR)	192.168.1.28:50000	Ermöglicht die Definition einer statischen (d.h. festen) Adresse. Diese statische Adresse wird **nicht** verwendet, wenn der C-886 für die Zuweisung einer IP-Adresse durch einen DHCP-Server konfiguriert ist (werkseitige Einstellung des Startup-Verhaltens zur Konfiguration der IP-Adresse). Wenn die statische Adresse verwendet werden soll: ▪ Das Startup-Verhalten muss umgestellt werden, damit der C-886 die mit "IPADR" definierte IP-Adresse verwendet. ▪ Die IP-Adressen und Subnetzmasken von C-886 und PC bzw. allen restlichen Netzwerkteilnehmern müssen aufeinander abgestimmt werden. Details siehe "PC und C-886 für Verwendung statischer IP-Adressen vorbereiten".
Startup-Verhalten zur Konfiguration der IP-Adresse für die TCP/IP-Kommu-nikation (IPSTART)	DHCP wird verwendet, um die IP-Adresse zu erhalten	Die IP-Adresse des C-886 wird mit der werkseitigen Einstellung des Startup-Verhaltens über DHCP zugewiesen. Die werkseitige Einstellung des Startup-Verhaltens muss nur geändert werden, wenn die Netzwerkteilnehmer stattdessen statische Adressen verwenden sollen.
Subnetzmaske (IPMASK)	255.255.255.0	Die werkseitige Einstellung der Subnetzmaske muss eventuell geändert werden, wenn die Netzwerkteilnehmer statische Adressen verwenden sollen. Details siehe "PC und C-886 für Verwendung statischer IP-Adressen vorbereiten".

Nach dem Einschalten oder dem Neustart des C-886

Der Startvorgang des C-886 muss beendet sein, bevor die Kommunikation zwischen C-886 und PC hergestellt werden kann. Der Startvorgang dauert circa 10 Sekunden.

Wenn die IP-Adresse des C-886 über DHCP zugewiesen wird, ist nach dem Ende des Startvorgangs des C-886 eine zusätzliche Wartezeit erforderlich, bevor die Kommunikation über TCP/IP möglich ist.

Anschluss des Netzwerkkabels bei eingeschaltetem Controller

Die Herstellung der Kommunikation über TCP/IP kann fehlschlagen, wenn das Netzwerkkabel bei eingeschaltetem C-886 an die RJ45-Buchse des C-886 angeschlossen wurde.

➢ Wenn die Herstellung der Kommunikation fehlschlägt, schalten Sie den C-886 aus und bei gestecktem Netzwerkkabel wieder ein.

Port-Einstellung

Für die Kommunikation über TCP/IP steht beim C-886 ein unveränderlicher Port (50000) zur Verfügung.

6.3.1 PC und C-886 für Verwendung statischer IP-Adressen vorbereiten

Wenn ein Netzwerk ohne DHCP-Server vorliegt oder wenn der C-886 direkt an die Ethernet-Anschlussbuchse des PC angeschlossen ist, sind folgende Anpassungen der Schnittstellenparameter notwendig:

- Startup-Verhalten zur Konfiguration der IP-Adresse des C-886 so einstellen, dass eine statische Adresse verwendet wird
- IP-Adressen und Subnetzmasken von C-886 und PC bzw. allen restlichen Netzwerkteilnehmern aufeinander abstimmen

Zur Anpassung von IP-Adressen und Subnetzmasken können Sie sich für eine der beiden folgenden Optionen entscheiden:

- Die PC-Einstellungen und gegebenenfalls die Einstellungen weiterer Netzwerkteilnehmer anpassen. Die Einstellungen des C-886 bleiben unverändert.
- Die Einstellungen des C-886 anpassen. Die PC-Einstellungen und gegebenenfalls die Einstellungen weiterer Netzwerkteilnehmer bleiben unverändert.

Voraussetzung

✓ Sie haben die Kommunikation zwischen dem C-886 und dem PC über USB hergestellt, um die Einstellungen des C-886 zu ermitteln und gegebenenfalls ändern zu können.

IP-Adresse und Subnetzmaske des PC ermitteln

1. Öffnen Sie an Ihrem PC auf geeignete Weise das Fenster, in dem die Eigenschaften des Internetprotokolls TCP/IP angezeigt und eingestellt werden. Die erforderlichen Schritte hängen vom verwendeten Betriebssystem ab.

Wenn Ihr Betriebssystem zwischen Internetprotokoll Version 4 (TCP/IPv4) und Version 6 (TCP/IPv6) unterscheidet (z. B. Windows 7), öffnen Sie das Fenster für Version 4.

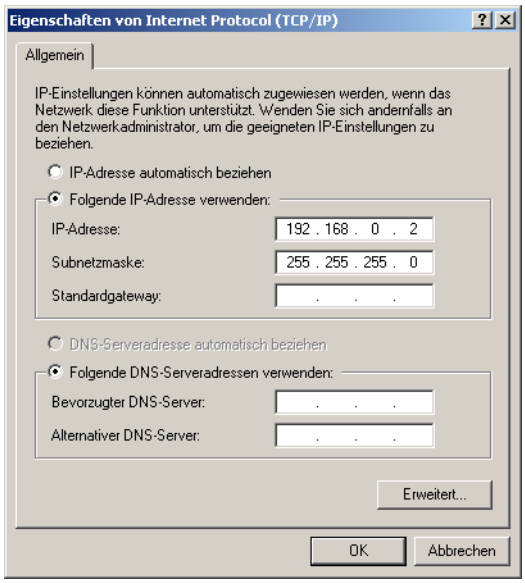


Abbildung 7: Fenster „Eigenschaften von Internet Protocol (TCP/IP)“ mit beispielhaften Einstellungen (nicht unbedingt für Ihr System geeignet)

Die Abbildung zeigt beispielhafte Einstellungen, die nicht unbedingt für Ihr System geeignet sind.

2. Notieren Sie die Einstellungen.

IP-Adresse und Subnetzmaske des C-886 ermitteln, Startverhalten des C-886 anpassen

1. Wenn Sie die Kommunikation zwischen C-886 und PC über die USB-Schnittstelle hergestellt haben, öffnen Sie im verwendeten Programm das Fenster zur Befehlseingabe.
2. Geben Sie den Befehl `IFS?` ein.

   Dieser Befehl fragt die Werte der Schnittstellenparameter im permanenten Speicher ab.
3. Notieren Sie die Einstellungen für ***IPMASK*** und ***IPADR***.
4. Stellen Sie sicher, dass der Parameter ***IPSTART*** korrekt eingestellt ist:
   - Wenn ***IPSTART*** = 0 (die mit ***IPADR*** definierte statische IP-Adresse wird verwendet), ist die Einstellung korrekt.
   - Wenn ***IPSTART*** ≠ 0: Senden Sie den Befehl `IFS 100 IPSTART 0`.

INFORMATION

In PIMikroMove® können Sie im Fenster ***Configure Interface*** die IP-Adresse und die Subnetzmaske des C-886 ermitteln und das Startverhalten des C-886 anpassen, ohne Befehle senden zu müssen.

IP-Einstellungen des PC anpassen

➢ Wenn Sie die PC-Einstellungen unverändert lassen wollen, fahren Sie mit dem Abschnitt "C-886-Einstellungen anpassen" fort.

1. Aktivieren Sie ***Folgende IP-Adresse verwenden*** in dem Fenster, in dem die Eigenschaften des Internetprotokolls TCP/IP (TCP/IPv4) angezeigt und eingestellt werden.
2. Passen Sie die IP-Adresse und die Subnetzmaske an die Einstellungen des C-886 an:
   a) Übernehmen Sie für die IP-Adresse am PC die ersten drei Abschnitte der IP-Adresse des C-886.
   b) Stellen Sie sicher, dass sich der letzte Abschnitt der IP-Adresse am PC vom letzten Abschnitt der IP-Adresse des C-886 unterscheidet und nicht "255" oder "0" ist.
   c) Übernehmen Sie für die Subnetzmaske am PC die Subnetzmaske des C-886.

   Beispiel:
   Neue IP-Adresse des PC: 192.168.1.29 (wenn der C-886 die IP-Adresse 192.168.1.28 hat)
   Neue Subnetzmaske des PC: 255.255.255.0 (wenn der C-886 die Subnetzmaske 255.255.255.0 hat)
3. Bestätigen Sie die Einstellungen mit der Schaltfläche ***OK***.
4. Wenn noch weitere Netzwerkteilnehmer angepasst werden müssen:
   Passen Sie die IP-Adressen und Subnetzmasken wie in den vorhergehenden Schritten an.
   Weisen Sie jedem Netzwerkteilnehmer eine eigene, eindeutige IP-Adresse zu.
   IP-Adressen dürfen im selben Netzwerk nicht doppelt vorkommen.
5. Schließen Sie die Verbindung über die USB-Schnittstelle, z. B. in PIMikroMove® im Hauptfenster über den Menüeintrag ***Connections > Close > C-886***.
6. Schalten Sie den C-886 aus.
7. Fahren Sie mit dem Abschnitt "Kommunikation über TCP/IP in der PC-Software herstellen" fort (S. 61).

C-886-Einstellungen anpassen

1. Passen Sie mit dem Befehl `IFS` die Einstellungen des C-886 an die des PC an:

   a) Ändern Sie die Subnetzmaske mit dem Befehl `IFS 100 IPMASK xxx.xxx.xxx.xxx`, wobei *xxx.xxx.xxx.xxx* die Subnetzmaske des PC ist.

   b) Ändern Sie die IP-Adresse mit dem Befehl `IFS 100 IPADR xxx.xxx.xxx.yyy:50000`, wobei Folgendes gilt:

   – *xxx.xxx.xxx.* stimmt mit den ersten drei Abschnitten der IP-Adresse des PC überein

   – *yyy* unterscheidet sich vom letzten Abschnitt der IP-Adresse des PC und jedes anderen Geräts im gleichen Netzwerk

   – *yyy* ist nicht "255" und nicht "0" und liegt im Adressbereich, der durch den letzten Abschnitt der Subnetzmaske vorgegeben ist

   – die Port-Adresse "50000" darf nicht geändert werden

   Beispiel:
   Wenn die IP-Adresse des PC 192.168.0.1 ist und kein anderes Gerät die IP-Adresse 192.168.0.2 hat, senden Sie den Befehl `IFS 100 IPADR 192.168.0.2:50000`.

2. Schließen Sie die Verbindung über die USB-Schnittstelle, z. B. in PIMikroMove® im Hauptfenster über den Menüeintrag ***Connections > Close > C-886***.

3. Schalten Sie den C-886 aus.

4. Fahren Sie mit dem Abschnitt "Kommunikation über TCP/IP in der PC-Software herstellen" fort (S. 61).

6.3.2 Kommunikation über TCP/IP in der PC-Software herstellen

VORSICHT

Quetschgefahr durch unerwartete Bewegung

Wenn die Kommunikation zwischen C-886 und PC über TCP/IP hergestellt wird, bietet die PC-Software alle im selben Netzwerk vorhandenen Controller zur Auswahl an. Nach Auswahl eines C-886 für die Verbindung werden alle Befehle an diesen Controller geschickt. Bei Auswahl eines falschen Controllers besteht für das Bedien- und Wartungspersonal des angeschlossenen Positionierers die Gefahr von leichten Verletzungen durch Quetschung aufgrund von unerwartet kommandierten Bewegungen.

- ➢ Wenn in der PC-Software mehrere C-886 angezeigt werden, vergewissern Sie sich, dass Sie den richtigen C-886 auswählen.
- ➢ Vergeben Sie einen eindeutigen Namenszusatz für jeden C-886 im selben Netzwerk: Tragen Sie den Namenszusatz als Wert des Parameters ***Customer Device Name*** (ID 0x0D001000) ein.

INFORMATION

Für den <Produktname kann ein frei wählbarer Namenszusatz vergeben werden, um mehrere C-886 im selben Netzwerk oder am selben PC unterscheiden zu können. Gehen Sie wie folgt vor, um den Namenszusatz zu vergeben:

1. Stellen Sie die Kommunikation über eine der vorhandenen Schnittstellen her.
2. Wechseln Sie auf Befehlsebene 1, indem Sie senden:
   `CCL 1 advanced`
3. Tragen Sie einen eindeutigen Namenszusatz als Wert des Parameters ***Customer Device Name*** (ID 0x0D001000) ein, indem Sie senden:
   `SPA 1 0x0D001000` *Namenszusatz*
4. Speichern Sie den neuen Parameterwert im permanenten Speicher, indem Sie senden:
   `WPA 101 1 0x0D001000`.
5. Schließen Sie die Verbindung.

Alternativ:

➢ Identifizieren Sie den zu verbindenden C-886 in der Auflistung der gefundenen Controller anhand seiner Seriennummer (*SN*). Die Seriennummer des Controllers können Sie dem Typenschild auf der Rückwand des C-886 entnehmen.

Voraussetzungen

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

✓ Der C-886 ist über die RJ45-Buchse an das Netzwerk oder direkt an den PC angeschlossen.

✓ Wenn der C-886 an ein Netzwerk angeschlossen ist:
Der zur Kommunikation mit dem C-886 zu verwendende PC ist auf geeignete Weise an dasselbe Netzwerk wie der C-886 angeschlossen.

✓ Wenn das verwendete Netzwerk keinen DHCP-Server besitzt oder wenn der C-886 direkt an die Ethernet-Anschlussbuchse des PC angeschlossen ist, so dass statische IP-Adressen verwendet werden müssen:
Sie haben durch Anpassen der Schnittstellenparameter das passende Startup-Verhalten zur Konfiguration der IP-Adresse des C-886 eingestellt und die IP-Adressen und Subnetzmasken von C-886 und PC bzw. allen restlichen Netzwerkteilnehmern aufeinander abgestimmt.

✓ Wenn mehrere C-886 über ihre TCP/IP-Schnittstellen mit dem gleichen Netzwerk verbunden sind: Sie haben über den Parameter ***Customer Device Name*** (ID 0x0D001000) einen eindeutigen Namenszusatz für den C-886 vergeben, mit dem die Kommunikation hergestellt werden soll. Alternativ haben Sie die Seriennummer dieses C-886 parat. Die Seriennummer können Sie dem Typenschild auf der Rückwand des C-886 entnehmen.

✓ Der PC ist eingeschaltet.

✓ Die benötigte Software ist auf dem PC installiert (S. 43).

✓ Sie haben das Handbuch der verwendeten PC-Software gelesen und verstanden. Die Software-Handbücher finden Sie auf der Produkt-CD.

✓ Der C-886 ist ausgeschaltet.

Kommunikation über TCP/IP herstellen

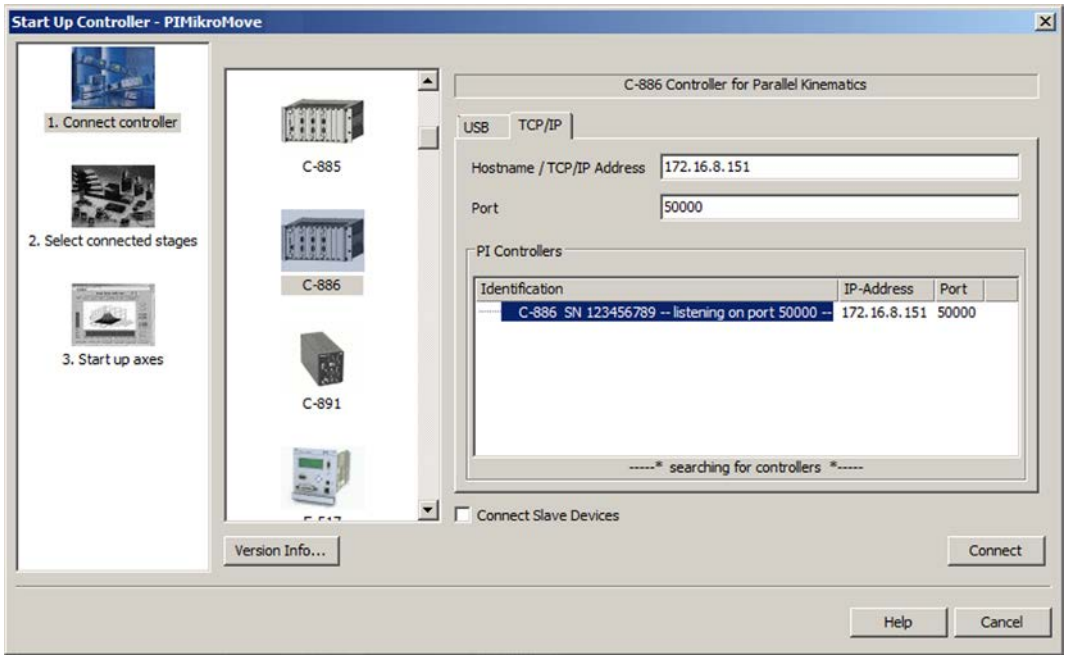
Im Folgenden ist das Vorgehen für PIMikroMove® beschrieben. Das Vorgehen bei anderen PC-Software-Programmen (z. B. PITerminal, LabVIEW-Treiber) ist ähnlich.

1. Schalten Sie den C-886 ein.
2. 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...***.
3. Wählen Sie im Feld für die Controllerauswahl ***C-886*** aus.
4. Wählen Sie auf der rechten Seite des Fensters die Registerkarte ***TCP/IP*** aus.

   Alle Controller im selben Netzwerk werden angezeigt.



5. Klicken Sie in der Liste der gefundenen Controller auf den Eintrag ***C-886 ...*** .

   – Wenn mehrere Einträge ***C-886 ...*** angezeigt werden, identifizieren Sie Ihren C-886 anhand des Namenszusatzes, den Sie zuvor vergeben haben, oder anhand seiner neunstelligen Seriennummer (*SN* ...).

   – Wenn der C-886 nicht in der Liste der gefundenen Controller angezeigt wird, prüfen Sie die Netzwerkeinstellungen (S. 263). Wenden Sie sich gegebenenfalls an Ihren Netzwerkadministrator.

6. Prüfen Sie die IP-Adresse im Feld ***Hostname / TCP/IP Address*** und die Port-Nummer im Feld ***Port***.
7. Stellen Sie sicher, dass das Kontrollkästchen ***Connect Slave Devices*** deaktiviert ist.

   Das Kontrollkästchen ***Connect Slave Devices*** darf nur aktiviert werden, wenn die direkte Kommunikation mit den Slave-Modulen für die optionalen Einzelachsen erforderlich ist, weitere Informationen siehe "Slave-Module für Einzelachsen konfigurieren" (S. 66) und "Kennungen für direkten Zugriff auf Slave-Module" (S. 26).
8. Klicken Sie auf die Schaltfläche ***Connect***, um die Kommunikation herzustellen.

Wenn die Kommunikation erfolgreich hergestellt wurde, wechselt das Fenster ***Start up controller*** zum Schritt ***Start up axes***.

6.4 Kommunikation über USB-Schnittstelle herstellen

INFORMATION

Für den <Produktname kann ein frei wählbarer Namenszusatz vergeben werden, um mehrere C-886 im selben Netzwerk oder am selben PC unterscheiden zu können. Gehen Sie wie folgt vor, um den Namenszusatz zu vergeben:

1. Stellen Sie die Kommunikation über eine der vorhandenen Schnittstellen her.
2. Wechseln Sie auf Befehlsebene 1, indem Sie senden:
   `CCL 1 advanced`
3. Tragen Sie einen eindeutigen Namenszusatz als Wert des Parameters ***Customer Device Name*** (ID 0x0D001000) ein, indem Sie senden:
   `SPA 1 0x0D001000` *Namenszusatz*
4. Speichern Sie den neuen Parameterwert im permanenten Speicher, indem Sie senden:
   `WPA 101 1 0x0D001000`.
5. Schließen Sie die Verbindung.

Alternativ:

➢ Identifizieren Sie den zu verbindenden C-886 in der Auflistung der gefundenen Controller anhand seiner Seriennummer (*SN*). Die Seriennummer des Controllers können Sie dem Typenschild auf der Rückwand des C-886 entnehmen.

Voraussetzungen

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

✓ Der C-886 ist an die USB-Schnittstelle des PC angeschlossen

✓ Der C-886 ist eingeschaltet, und der Startvorgang des C-886 ist beendet (S. 56).

✓ 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. Die Software-Handbücher finden Sie auf der Produkt-CD.

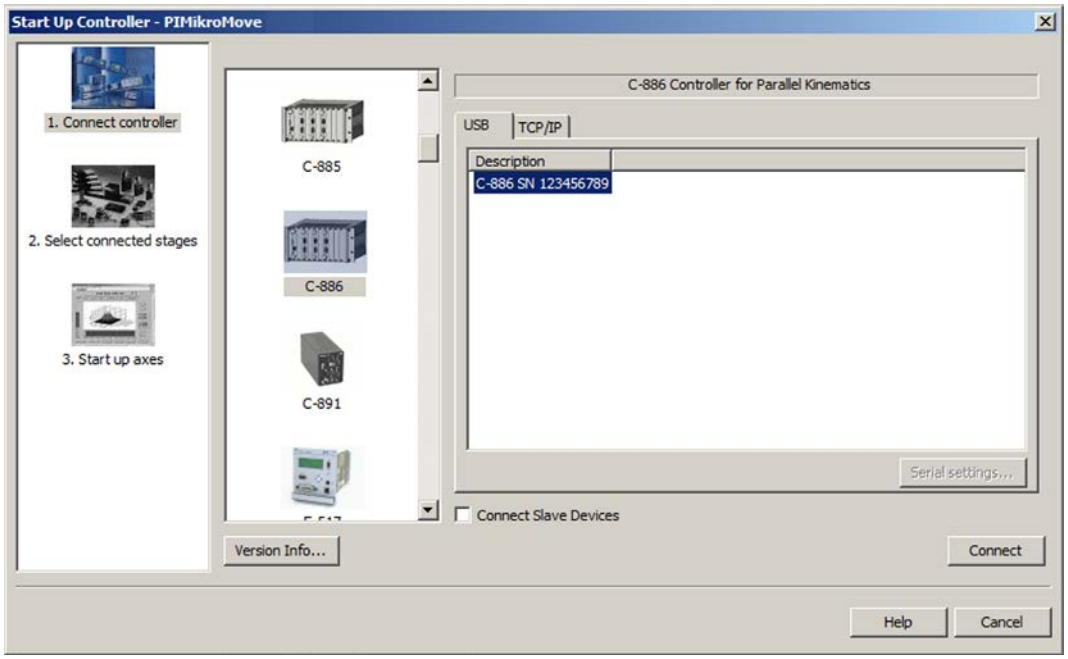
Kommunikation über USB herstellen

Im Folgenden ist das Vorgehen für PIMikroMove® beschrieben. Das Vorgehen bei anderen PC-Software-Programmen (z. B. PITerminal, LabVIEW-Treiber) ist ähnlich.

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 ***C-886*** aus.
3. Wählen Sie auf der rechten Seite des Fensters die Registerkarte ***USB*** aus.
4. Wählen Sie auf der Registerkarte ***USB*** den angeschlossenen ***C-886*** aus.
5. Stellen Sie sicher, dass das Kontrollkästchen ***Connect Slave Devices*** deaktiviert ist.

   Das Kontrollkästchen ***Connect Slave Devices*** darf nur aktiviert werden, wenn die direkte Kommunikation mit den Slave-Modulen für die optionalen Einzelachsen erforderlich ist, weitere Informationen siehe "Slave-Module für Einzelachsen konfigurieren" (S. 66) und "Kennungen für direkten Zugriff auf Slave-Module" (S. 26).
6. Klicken Sie auf ***Connect***, um die Kommunikation herzustellen.

Wenn die Kommunikation erfolgreich hergestellt wurde, wechselt das Fenster ***Start up controller*** zum Schritt ***Start up axes***.

6.5 Slave-Module für Einzelachsen konfigurieren

HINWEIS

Schäden durch direkten Zugriff auf Slave-Module für den Positionierer!

Bei Auslieferung des C-886 sind die Slave-Module für den Positionierer verwendungsfertig konfiguriert. Das Ändern von Einstellungen oder die Kommandierung von Bewegungen über den direkten Zugriff auf diese Slave-Module kann zu Schäden am Positionierer führen.

- Greifen Sie nur nach Rücksprache mit PI direkt auf die Slave-Module für die Antriebe des Positionierer zu.

HINWEIS

Schäden durch direkten Zugriff auf Slave-Module für Einzelachsen!

Die Slave-Module für die optionalen Einzelachsen können durch den Anwender im C-886 eingebaut (S. 47) und konfiguriert werden. Unpassende Parametereinstellungen der Slave-Module können zu unsachgemäßem Betrieb oder zur Beschädigung der angeschlossenen Mechanik führen.

- Laden Sie die Parametereinstellungen für die optionalen Einzelachsen aus einer Verstellerdatenbank von PI, siehe "Slave-Module für Einzelachsen in PIMikroMove® konfigurieren".
- Ändern Sie die Parameterwerte für die optionalen Einzelachsen nur nach sorgfältiger Überlegung.
- Informieren Sie sich im Handbuch des entsprechenden Moduls (S. 6) über das Anpassen von Einstellungen.

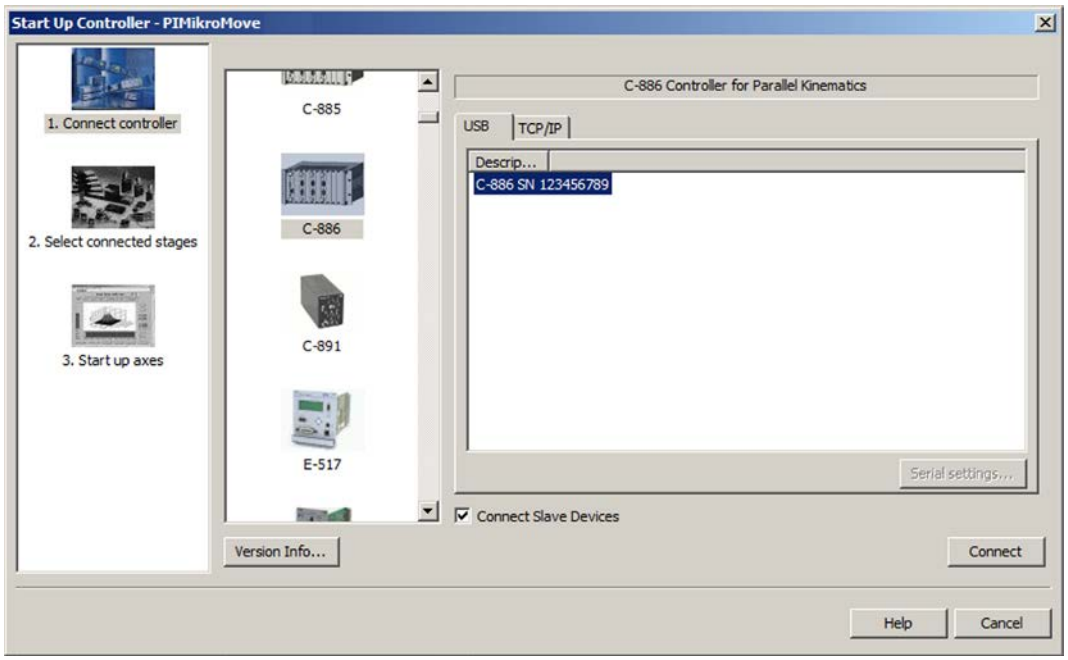
Voraussetzungen

- ✓ Der C-886 wurde ordnungsgemäß installiert (S. 43).
- ✓ Wenn der C-886 an die USB-Schnittstelle des PC angeschlossen ist: Sie haben die Schritte 1 bis 4 aus "Kommunikation über USB-Schnittstelle herstellen" (S. 64) in PIMikroMove® ausgeführt.
- ✓ Wenn der C-886 über die RJ45-Buchse an das Netzwerk oder direkt an den PC angeschlossen ist: Sie haben die Schritte 1 bis 6 aus "Kommunikation über TCP/IP in der PC-Software herstellen" (S. 61) in PIMikroMove® ausgeführt.

Slave-Module für Einzelachsen in PIMikroMove® konfigurieren

Im Beispiel in der folgenden Anleitung wird ein C-886 mit sechs Slave-Modulen für den parallelkinematischen Positionierer verwendet, der um ein Slave-Modul für eine optionale Einzelachse erweitert wurde.

1. Aktivieren Sie im Fenster ***Start up controller*** das Kontrollkästchen ***Connect Slave Devices***.

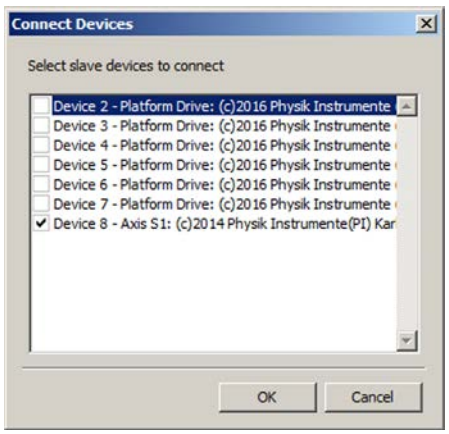


2. Klicken Sie auf ***Connect***.

   Das Fenster ***Connect Devices*** öffnet sich.

3. Wählen Sie im Fenster ***Connect Devices*** die Slave-Module für die optionalen Einzelachsen aus, indem Sie die entsprechenden Kontrollkästchen aktivieren (***Device 8*** und/oder ***Device 9***).

   – Aktivieren Sie **nicht** die Kontrollkästchen der Slave-Module für die Antriebe des Positionierers (***Device 2*** bis ***Device 7***). Die Slave-Module für die Antriebe des Positionierers wurden vor Auslieferung des C-886 von PI verwendungsfertig konfiguriert.

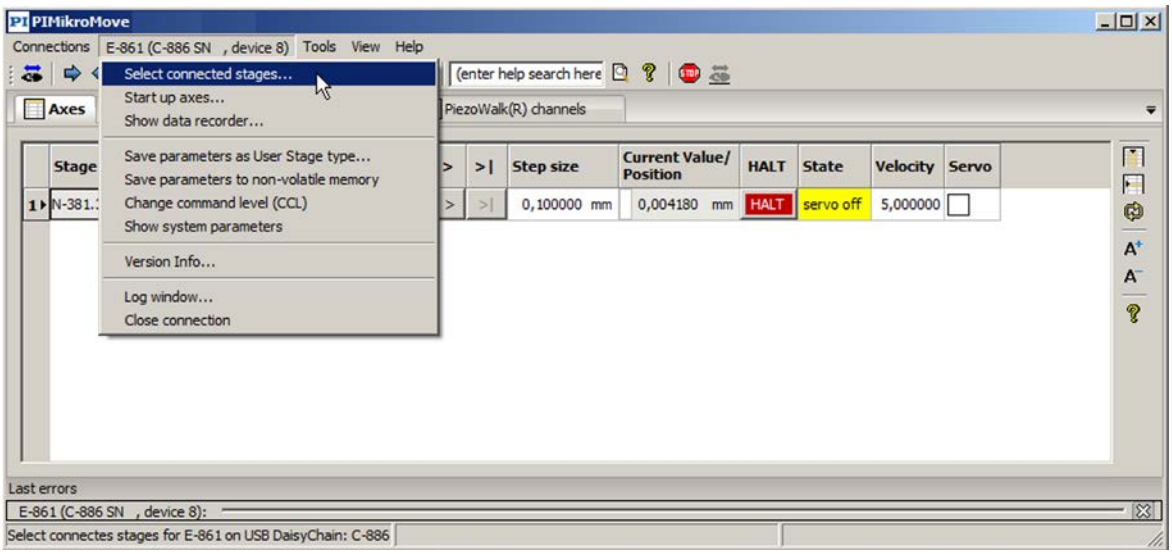


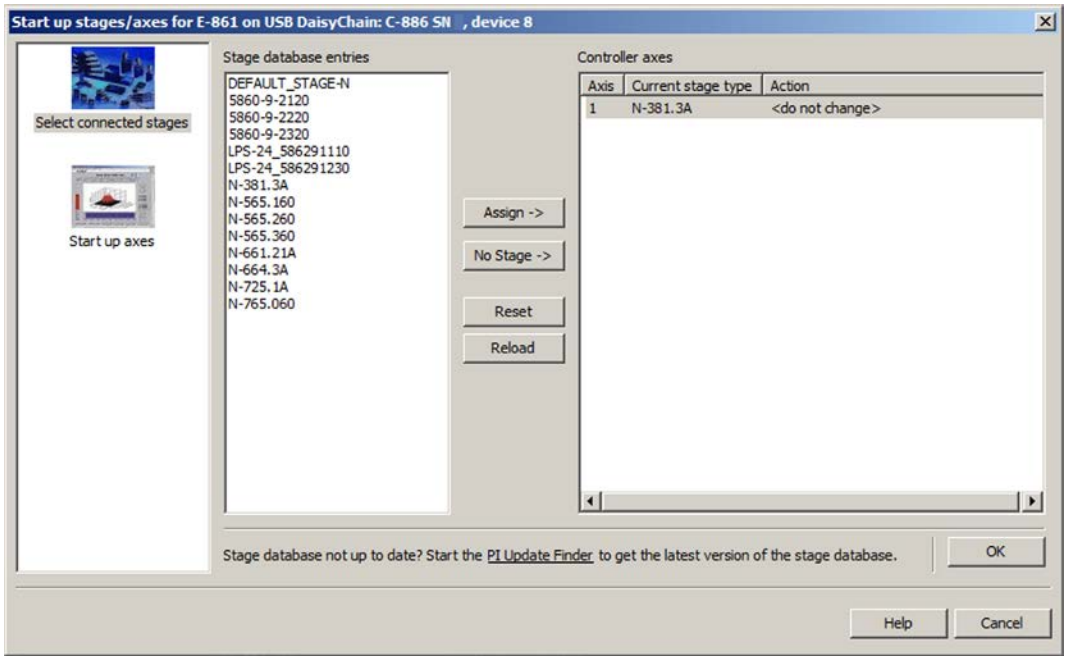
4. Klicken Sie im Fenster ***Connect Devices*** auf ***OK***, um die direkte Kommunikation mit den Slave-Modulen für die optionalen Einzelachsen herzustellen.

Das Herstellen der Kommunikation kann einige Sekunden dauern. Wenn die Kommunikation erfolgreich hergestellt wurde, schließt sich das Fenster ***Start up controller***, und das Hauptfenster von PIMikroMove® öffnet sich.

5. Konfigurieren Sie die Parametereinstellungen der Slave-Module für die optionalen Einzelachsen:

   a) Öffnen Sie das Fenster ***Start up controller*** mit dem Schritt ***Select connected stages***, indem Sie im Hauptfenster im Menü des Slave-Moduls den Eintrag ***Select connected stages ...*** wählen.



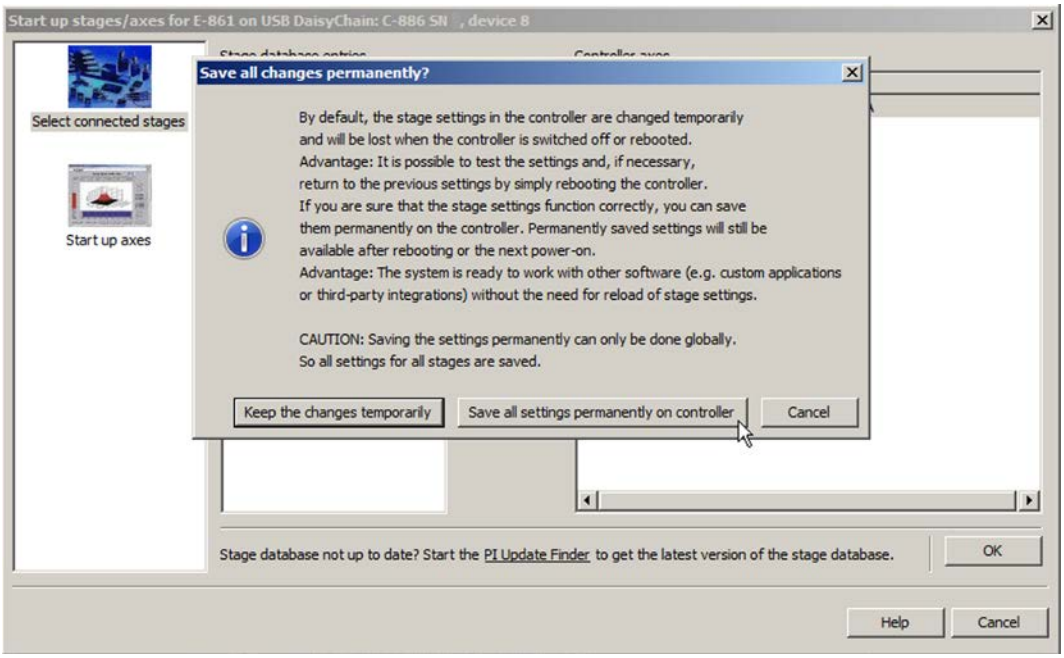


   b) Wählen Sie den passenden Verstellertyp aus. Sie haben zwei Möglichkeiten:

      - Klicken Sie auf ***Assign from ID Chip*** (nur vorhanden, wenn das Slave-Modul ID-Chips auslesen kann).

- Markieren Sie den passenden Verstellertyp in der Liste ***Stage database entries***, und klicken Sie auf ***Assign***.

c) Bestätigen Sie die Auswahl mit ***OK***, um die Parametereinstellungen für den ausgewählten Verstellertyp aus der Verstellerdatenbank in das Slave-Modul zu laden. Der Dialog ***Save all changes permanently?*** öffnet sich.



6. Geben Sie im Dialog ***Save all changes permanently?*** an, wie Sie die Parametereinstellungen in das Slave-Modul laden wollen:

   – Als Standardwerte laden (empfohlen): Klicken Sie auf ***Save all settings permanently on controller***, um die Parametereinstellungen in den permanenten Speicher des Slave-Moduls zu laden. Die Einstellungen sind nach dem Einschalten oder Neustart des C-886 sofort vorhanden und müssen nicht erneut geladen werden.

   – Temporär laden: Klicken Sie auf ***Keep the changes temporarily***, um die Parametereinstellungen in den flüchtigen Speicher des Slave-Moduls zu laden. Die Einstellungen gehen beim Ausschalten oder Neustart des C-886 verloren.

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

7. Schließen Sie das Fenster ***Start up controller***, indem Sie auf ***Close*** klicken.

8. Wiederholen Sie die Schritte 5 bis 7, wenn Sie ein weiteres Slave-Modul für optionale Einzelachsen konfigurieren wollen.

9. Beenden Sie die direkte Kommunikation mit den Slave-Modulen, indem Sie im Hauptfenster den Menüeintrag ***Connections > Close all*** wählen.

6.6 Bewegungen starten

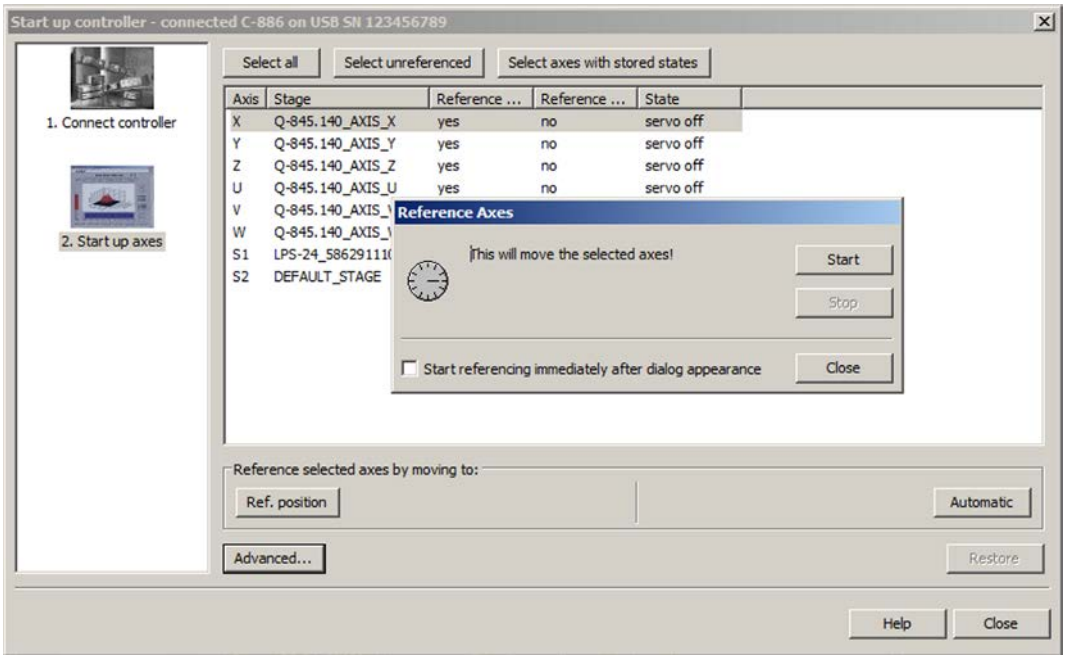
Im Folgenden wird PIMikroMove® verwendet, um die Achsen zu bewegen.

Voraussetzungen

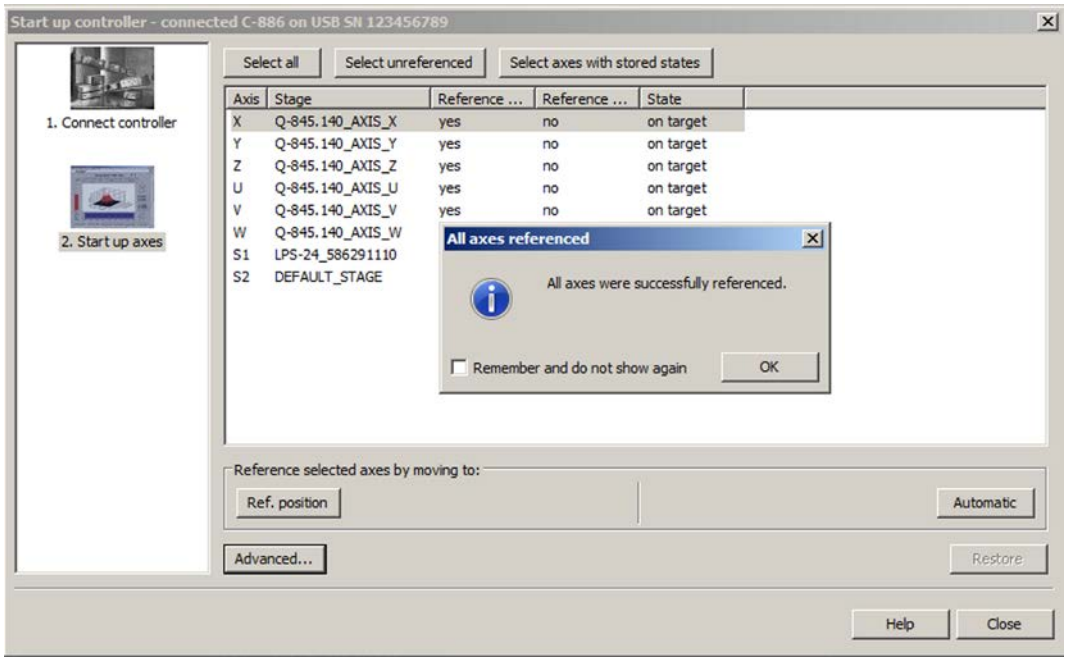
- ✓ Sie haben die allgemeinen Hinweise zur Inbetriebnahme gelesen und verstanden (S. 53).
- ✓ PIMikroMove® ist auf dem PC installiert (S. 43).
- ✓ Sie haben das PIMikroMove®-Handbuch gelesen und verstanden. Das Handbuch finden Sie auf der Produkt-CD.
- ✓ Wenn erforderlich: Sie haben die Slave-Module für die optionalen Einzelachsen im C-886 installiert (S. 47) und konfiguriert (S. 66).
- ✓ Sie haben die Benutzerhandbücher aller angeschlossenen Antriebe (parallelkinematischer Positionierer, optionale Einzelachsen) gelesen und verstanden.
- ✓ Sie haben die Antriebe gemäß den Anweisungen in den entsprechenden Benutzerhandbüchern installiert und am C-886 angeschlossen (S. 49).
- ✓ Sie haben die Kommunikation zwischen dem C-886 und dem PC mit PIMikroMove® über die TCP/IP-Schnittstelle (S. 61) oder die USB-Schnittstelle (S. 64) hergestellt.

Bewegungen starten mit PIMikroMove®

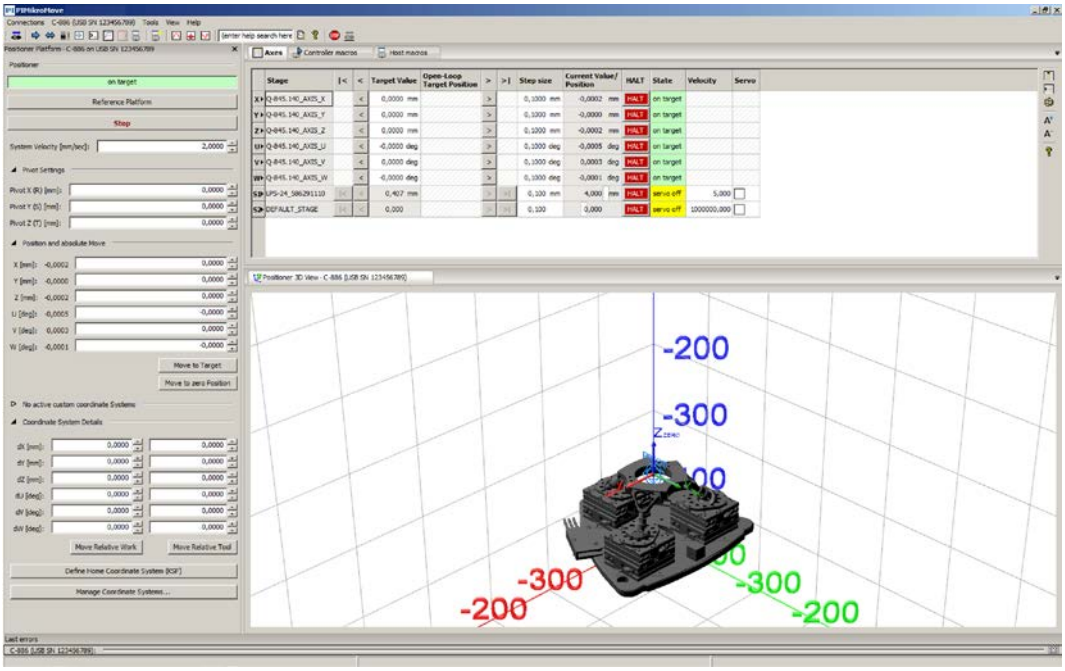
1. Führen Sie im Schritt ***Start up axes*** die Referenzfahrt für die Achsen aus (erforderlich für Achsen mit inkrementellen Sensoren). Klicken Sie dazu auf ***Ref.position*** oder auf ***Automatic***.



2. Nach erfolgreicher Referenzfahrt klicken Sie auf ***OK > Close***.



3. Wenn im Hauptfenster von PIMikroMove® **nicht** das Fenster ***Positioner Platform*** (links, angedockt) und die Karte ***Positioner 3D View*** angezeigt werden:
   - Blenden Sie das Fenster ***Positioner Platform*** mit dem Menüeintrag ***C-886 > Show Positioner platform settings*** ein.
   - Blenden Sie die Karte ***Positioner 3D View*** mit dem Menüeintrag ***C-886 > Positioner 3D View > Show*** ein.
   - Wenn die Karte ***Positioner 3D View*** nicht im Vordergrund angezeigt wird, klicken Sie auf den entsprechenden Kartenreiter.



4. Starten Sie einige Testbewegungen für die Achsen der Bewegungsplattform des Positionierers (X bis W):

   a) Geben Sie im Fenster ***Positioner Platform*** im Bereich ***Position and absolute Move*** eine Zielposition für mindestens eine Achse der Bewegungsplattform des Positionierers ein.

   b) Klicken Sie auf ***Move to Target***, um die Bewegung zur angegebenen Zielposition zu starten.

   c) Wenn die Bewegung beendet ist, wiederholen Sie die Schritte a und b für eine neue Zielposition.

   Wenn eine Stützstelle des berechneten Dynamikprofils oder die Zielposition nicht erreicht werden kann, wird die Bewegung nicht ausgeführt, und ein Fenster mit einer Fehlermeldung öffnet sich.

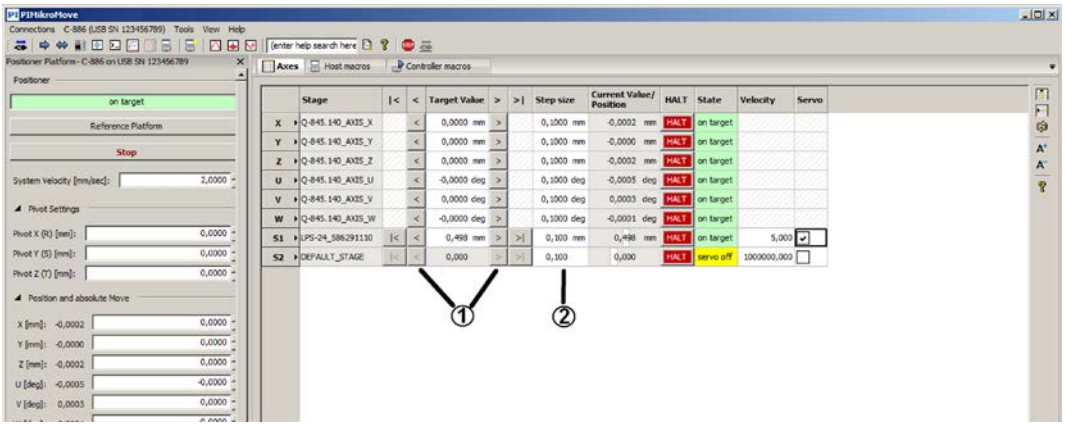
   Während einer Bewegung kann keine neue Zielposition eingegeben werden.

   Auf der Karte ***Positioner 3D View*** wird die Bewegung grafisch abgebildet.

5. Wenn optionale Einzelachsen vorhanden sind: Starten Sie im Hauptfenster von PIMikroMove® auf der Karte ***Axes*** einige Testbewegungen der optionalen Einzelachsen (S1 und S2).

   a) Wenn erforderlich, schalten Sie den Servomodus für die optionalen Einzelachsen ein, indem Sie das entsprechende Kontrollkästchen in der Spalte ***Servo*** aktivieren.

b) Starten Sie die Testbewegungen: Sie können z. B. Schritte mit einer bestimmten Schrittweite (2) ausführen, indem Sie auf die entsprechenden Pfeiltasten (1) für eine Achse klicken.



7 Betrieb

In diesem Kapitel

Allgemeine Hinweise zum Betrieb 75
Datenrekorder 76
Funktionsgenerator 79
Controllermakros 103

7.1 Allgemeine Hinweise zum Betrieb

VORSICHT

Stromschlaggefahr bei fehlendem Schutzleiter!

Bei fehlendem oder nicht ordnungsgemäß angeschlossenem Schutzleiter können im Falle eines Fehlers oder Defekts gefährliche Berührungsspannungen am C-886 entstehen. Wenn Berührungsspannungen vorhanden sind, kann das Berühren des C-886 zu leichten Verletzungen durch Stromschlag führen.

- Schließen Sie den C-886 vor Inbetriebnahme an einen Schutzleiter an (S. 46).
- Entfernen Sie den Schutzleiter **nicht** während des Betriebs.
- Wenn der Schutzleiter vorübergehend entfernt werden muss (z. B. bei Umbauten), schließen Sie den C-886 vor erneuter Inbetriebnahme wieder an den Schutzleiter an.

VORSICHT

Quetschgefahr durch bewegte Teile!

Zwischen den bewegten Teilen des Positionierers und einem feststehenden Teil oder Hindernis besteht die Gefahr von leichten Verletzungen durch Quetschung.

- Halten Sie Ihre Finger von Bereichen fern, in denen sie von bewegten Teilen erfasst werden können.

HINWEIS

Schäden durch Kollisionen!

Kollisionen können den Positionierer, die zu bewegende Last und die Umgebung beschädigen.

- Stellen Sie sicher, dass im Arbeitsraum des Positionierers keine Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich sind.
- Platzieren Sie keine Gegenstände in Bereichen, in denen sie von bewegten Teilen erfasst werden können.
- Halten Sie bei einer Fehlfunktion des Controllers die Bewegung sofort an.

HINWEIS

Schäden durch ungewollte Positionsänderungen!

Die maximale Haltekraft bei ausgeschaltetem Servomodus basiert auf der Selbsthemmung der Antriebe und kann geringer ausfallen als die maximale Belastbarkeit bei eingeschaltetem Servomodus (siehe Handbuch des Positionierers).

Wenn die tatsächliche Belastung des Positionierers die auf der Selbsthemmung der Antriebe basierende maximale Haltekraft überschreitet, können in folgenden Fällen ungewollte Positionsänderungen des Positionierers auftreten:

- Ausschalten des C-886
- Neustart des C-886
- Ausschalten des Servomodus für die Achsen derBewegungsplattform des Positionierers

Dadurch sind Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich. Kollisionen können den Positionierer, die zu bewegende Last oder die Umgebung beschädigen.

- Stellen Sie sicher, dass die tatsächliche Belastung der Bewegungsplattform des Positionierers die auf der Selbsthemmung der Antriebe basierende maximale Haltekraft nicht überschreitet, bevor Sie den Servomodus ausschalten, den C-886 neu starten oder ausschalten.

7.2 Datenrekorder

7.2.1 Eigenschaften des Datenrekorders

Der C-886 besitzt einen Echtzeit-Datenrekorder für Positionswerte und Statusregister-Bits.

Die aufgezeichneten Daten werden temporär in 36 Datenrekordertabellen mit jeweils maximal 8192 Punkten gespeichert. Jede Datenrekordertabelle enthält die Daten von einer Datenquelle.

Sie können festlegen, wie die Aufzeichnung gestartet werden soll.

7.2.2 Datenrekorder einrichten

Informationen über den Datenrekorder auslesen

- Senden Sie den Befehl HDR? (S. 143).

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

- Senden Sie den Befehl DRC? (S. 134), um die Konfiguration des Datenrekorders auszulesen.Die Datenrekordertabellen des C-886 zeichnen Folgendes auf:
  - Datenrekordertabelle 1, 3, 5, 7, 9, 11: kommandierte Position der Achsen des parallelkinematischen Positionierers (X, Y, Z, U, V und W)
  - Datenrekordertabelle 2, 4, 6, 8, 10, 12: aktuelle Position der Achsen des parallelkinematischen Positionierers (X, Y, Z, U, V und W)
  - Datenrekordertabelle 13, 16, 19, 22, 25, 28: kommandierte Position der Antriebe des parallelkinematischen Positionierers (1, 2, 3, 4, 5 und 6)
  - Datenrekordertabelle 14, 17, 20, 23, 26, 29: aktuelle Position der Antriebe des parallelkinematischen Positionierers (1, 2, 3, 4, 5 und 6)
  - Datenrekordertabelle 15, 18, 21, 24, 27, 30: Statusregister für die Antriebe des parallelkinematischen Positionierers (1, 2, 3, 4, 5 und 6)
  - Datenrekordertabelle 31, 34: kommandierte Position der optionalen Einzelachsen (S1, S2)
  - Datenrekordertabelle 32, 35: aktuelle Position der optionalen Einzelachsen (S1, S2)
  - Datenrekordertabelle 33, 36: Statusregister für die optionalen Einzelachsen (S1, S2)

INFORMATION

Mit der Aufzeichnungsoption 80 können die Bits des Statusregisters aufgezeichnet werden.

- Verwenden Sie zur Aufzeichnung des Statusregisters das Fenster ***Data Recorder*** in PIMikroMove®, das die gezielte Auswahl einzelner Bits für die grafische Darstellung der aufgezeichneten Daten erlaubt. Details siehe PIMikroMove®-Handbuch.

Datenrekorder konfigurieren

Sie können festlegen, wie die Aufzeichnung ausgelöst werden soll.

- Fragen Sie mit DRT? (S. 138) die aktuelle Triggeroption ab.
- Ändern Sie die Triggeroption mit dem Befehl DRT (S. 137). Die Triggeroption gilt für alle Datenrekordertabellen.

Sie können festlegen, wieviele Punkte pro Datenrekordertabelle maximal aufgezeichnet werden sollen.

- Ändern Sie mit dem Befehl SPA (S. 196) den Parameter ***Data Recorder Points Per Table***, ID 0x16000201. Maximaler Wert: 8192 Punkte.

Aufzeichnungsrate abfragen

- Senden Sie den Befehl `RTR?` (S. 193), um die Aufzeichnungsrate des Datenrekorders auszulesen.
  Die Aufzeichnungsrate gibt über einen Faktor an, mit welcher Frequenz Datenpunkte aufgezeichnet werden. Der Faktor 1 entspricht einer Frequenz von 100 Hz.

7.2.3 Aufzeichnung starten

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

Unabhängig von der eingestellten Triggeroption wird die Datenaufzeichnung immer in folgenden Fällen ausgelöst:

- Starten einer Sprungantwortmessung mit `STE` (S. 201)
- Starten einer Impulsantwortmessung mit `IMP` (S. 149)
- Starten der Funktionsgeneratorausgabe mit `WGO` (S. 220)
- Während der Funktionsgeneratorausgabe: `WGR` (S. 222) startet die Aufzeichnung mit dem nächsten Ausgabezyklus

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

7.2.4 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. 135) aus.

  Die Daten werden im GCS-Array-Format ausgegeben (siehe Benutzerhandbuch SM146E auf der Produkt-CD).
- Fragen Sie die Anzahl der in der letzten Aufzeichnung enthaltenen Punkte mit dem Befehl `DRL?` (S. 135) ab.

7.3 Funktionsgenerator

7.3.1 Funktionsweise des Funktionsgenerators

Die Funktionsgeneratoren sind den Achsen des C-886 fest zugeordnet, siehe "Kommandierbare Elemente" (S. 25).

Ein Funktionsgenerator gibt auf Basis von definierten Kurvenformen absolute Zielpositionen für die Achsenbewegung aus und bestimmt damit das Dynamikprofil. Die Funktionsgeneratorausgabe eignet sich besonders für dynamische Anwendungen mit periodischen Achsenbewegungen.

Das folgende Blockdiagramm zeigt die Einbindung eines Funktionsgenerators im C-886.

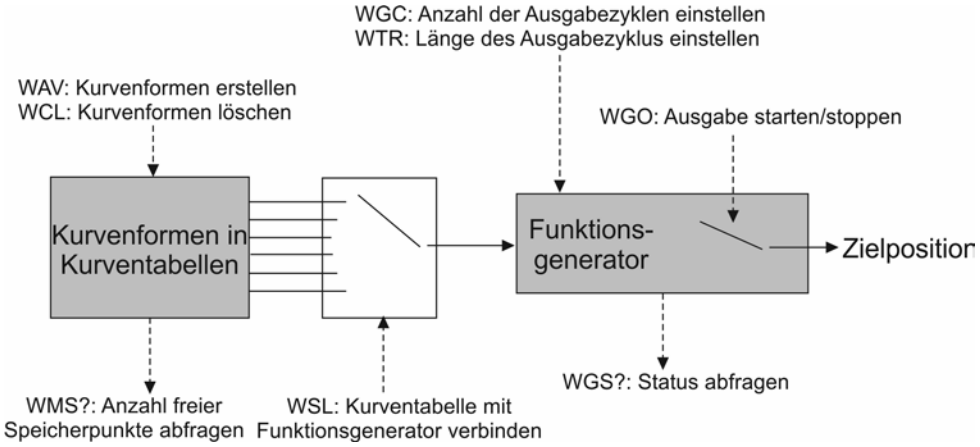


Abbildung 8: Blockdiagramm eines Funktionsgenerators

Kurvenformen können definiert und im flüchtigen Speicher des C-886 temporär in bis zu 100 Kurventabellen gespeichert werden (S. 82). Jede Kurventabelle enthält die Daten einer Kurvenform. Insgesamt sind 10.000 Speicherpunkte für Kurventabellen verfügbar. Die Speicherpunkte werden bei der Definition der Kurvenformen auf die einzelnen Kurventabellen verteilt.

Die Kurventabellen können den Funktionsgeneratoren und damit den Achsen beliebig zugeordnet werden (S. 90). Eine Kurventabelle kann gleichzeitig von mehreren Funktionsgeneratoren verwendet werden.

Die Anzahl der Ausgabezyklen (S. 90) und die Ausgaberate (S. 90) des Funktionsgenerators kann eingestellt werden.

INFORMATION

Die Einstellungen zur Verwendung des Funktionsgenerators lassen sich nur im flüchtigen Speicher des C-886 ändern und gehen beim Ausschalten oder Neustart des C-886 verloren.

INFORMATION

Für die Arbeit mit dem Funktionsgenerator wird die Verwendung des PI Frequency Generator Tool oder des PI Wave Generator Tool empfohlen (beide verfügbar in PIMikroMove®).

Für die Arbeit mit PIMikroMove® sind keine Befehlskenntnisse erforderlich. Trotzdem wird empfohlen, dass Sie sich in diesem Kapitel mit dem Funktionsgenerator vertraut machen.

Alle Beispiele in diesem Kapitel können in PIMikroMove® oder PITerminal als Befehlsfolgen eingegeben werden.

Die Verwendung des PI Frequency Generator Tool ist in der Technical Note A000T0057 beschrieben. Die Verwendung des PI Wave Generator Tool ist im PIMikroMove®-Handbuch beschrieben (SM148E).

Sie können Makros verwenden, um Kurvenformen zu definieren und die Funktionsgeneratoren zu konfigurieren. Für die Arbeit mit Makros siehe "Controllermakros" (S. 103).

7.3.2 Befehle und Parameter für den Funktionsgenerator

Befehle

Folgende Befehle stehen für die Verwendung des Funktionsgenerators zur Verfügung:

Befehl	Syntax	Funktion
GWD?	GWD? [<StartPoint> <NumberOfPoints> [{<WaveTableID>}]]	Fragt den Inhalt der Kurventabellen (d. h. die Kurvenformen) ab.
TWG?	TWG?	Fragt die Anzahl der Funktionsgeneratoren (= Anzahl der Achsen) ab.
WAV	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>	Definiert die Kurvenform.
WAV?	WAV? [{<WaveTableID> <WaveParameterID>}]	Fragt die aktuelle Länge der Kurventabellen ab (Punkteanzahl).
WCL	WCL {<WaveTableID>}	Löscht den Inhalt der Kurventabellen.
WGC	WGC {<WaveGenID> <Cycles>}	Stellt die Anzahl der Ausgabezyklen ein.
WGC?	WGC? [{<WaveGenID>}]	Fragt die Anzahl der Ausgabezyklen ab.
WGO	WGO {<WaveGenID> <StartMode>}	Startet und stoppt die Ausgabe des Funktionsgenerators.
WGO?	WGO? [{<WaveGenID>}]	Fragt den Aktivierungszustand ab, der zuletzt für den Funktionsgenerator kommandiert wurde.
WGR	WGR	Startet die Datenaufzeichnung bei laufendem Funktionsgenerator neu.

Befehl	Syntax	Funktion
WGS?	[<WaveGenID> [<InfoType>]]	Fragt den Status des Funktionsgenerators ab.
WMS?	[{<WaveTableID>}]	Fragt die Anzahl freier Speicherpunkte für die Kurventabelle ab.
WSL	WSL {<WaveGenID> <WaveTableID>}	Stellt die Verbindung zwischen Kurventabelle und Funktionsgenerator her.
WSL?	WSL? [{<WaveGenID>}]	Fragt die Verbindung zwischen Kurventabelle und Funktionsgenerator ab.
WTR	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Stellt die Ausgaberate des Funktionsgenerators ein (beeinflusst damit die Dauer eines Ausgabezyklus).
WTR?	WTR? [{<WaveGenID>}]	Fragt die Ausgaberate des Funktionsgenerators ab.
#9	#9	Fragt den aktuellen Aktivierungszustand des Funktionsgenerators ab.

Parameter

Die folgenden Parameter definieren die Eigenschaften des Funktionsgenerators:

Parameter	Beschreibung und mögliche Werte
Maximum Number Of Wave Points (ID 0x13000004)	Gesamtanzahl der für Kurvenformen verfügbaren Punkte Die Kurventabellen des C-886 haben insgesamt 10.000 Punkte. Die verfügbaren Punkte werden während der Definition von Kurvenformen mit dem Befehl WAV (S. 214) auf die Kurventabellen verteilt. Dieser Parameter ist schreibgeschützt.
Number of Wave Tables (ID 0x1300010A)	Anzahl der Kurventabellen zum Speichern von Kurvenformen Der C-886 hat 100 Kurventabellen. Dieser Parameter ist schreibgeschützt.
Start All Hexapod Wave Generators (ID 0x19002001)	Startverhalten der Funktionsgeneratoren für die Achsen der Bewegungsplattform des Positionierers (X, Y, Z, U, V, W): 0 = Jeder Funktionsgenerator, der gestartet werden soll, muss im WGO-Befehl addressiert werden (Standardeinstellung). 1 = Das Starten eines Funktionsgenerators startet alle Funktionsgeneratoren, die mit einer Kurventabelle verbunden sind. Für die Achsen der Bewegungsplattform des Positionierers, deren Funktionsgenerator **nicht** gestartet wurde, wird immer die zuletzt gültige Zielposition kommandiert.

7.3.3 Kurvenform definieren

Das Definieren von Kurvenformen umfasst folgende Schritte:

- Optional: Informationen zu Kurventabellen abfragen (S. 82)
- Kurvenform in Kurventabelle erstellen (S. 82)
- Optional: Inhalt der Kurventabelle löschen (S. 83)

In diesem Handbuch finden Sie Beispiele für das Erstellen von Kurvenformen (S. 83).

INFORMATION

Der Inhalt der Kurventabellen (= definierte Kurvenformen) ist nur im flüchtigen Speicher des C-886 vorhanden und geht beim Ausschalten oder Neustart des C-886 verloren.

Optional: Informationen zu Kurventabellen abfragen

- Senden Sie den Befehl `SPA? 1 0x13000004`, um die Gesamtanzahl der Punkte abzufragen, die der C-886 für das Definieren von Kurvenformen in Kurventabellen bereitstellt.
- Senden Sie den Befehl `SPA? 1 0x1300010A`, um die Anzahl der im C-886 verfügbaren Kurventabellen abzufragen.
- Fragen Sie mit dem Befehl `WAV?` (S. 219) für die Kurventabellen die aktuelle Anzahl bereits definierter Kurvenformpunkte ab.
- Fragen Sie mit dem Befehl `GWD?` (S. 142) den aktuellen Inhalt der Kurventabellen ab (= bereits definierte Kurvenformen).

  Die Antwort enthält den Inhalt der Kurventabellen im GCS-Array-Format (siehe separates Handbuch für GCS Array, SM 146E).
- Fragen Sie mit dem Befehl `WMS?` (S. 224) für die Kurventabellen die Anzahl der verfügbaren freien Speicherpunkte ab.

Kurvenform in Kurventabelle erstellen

1. Stellen Sie sicher, dass die gewählte Kurventabelle **nicht** mit einem Funktionsgenerator verbunden ist, für den die Ausgabe gestartet ist. Details siehe "Funktionsgenerator konfigurieren" (S. 90) und "Funktionsgeneratorausgabe stoppen" (S. 93).
2. Erstellen Sie mit dem Befehl `WAV` (S. 214) (WAV + max. 12 Argumente) in der gewählten Kurventabelle die Kurvenform aus einzelnen Segmenten. Unterstützte Kurventypen:

   "PNT" (benutzerdefinierte Kurve)

   "SIN_P" (invertierte Kosinuskurve)

   "RAMP" (Rampenkurve)

   "LIN" (Kurve in Form einer einzelnen Abtastzeile)

   Die Kurvenform wird in die gewählte Kurventabelle im flüchtigen Speicher geschrieben. Details siehe "Beispiele für das Erstellen von Kurvenformen" (S. 83).

INFORMATION

Bei der Definition einer Kurvenform mit `WAV` (S. 214) werden die Zielpositionen und die resultierenden Geschwindigkeiten **nicht** geprüft. Die Prüfung erfolgt erst während der Funktionsgeneratorausgabe (S. 92).

INFORMATION

Die Kurvenform beeinflusst die Geschwindigkeit während der Bewegungen.
Die Geschwindigkeit wird unter anderem durch folgende Faktoren begrenzt:

- Typ der Mechanik
- Kombination der zu bewegenden Achsen
- Aktuelle Einstellungen für Koordinatensystem und Drehpunkt
- Amplitude der Bewegung

➢ Definieren Sie die Kurvenform so, dass die Spezifikationen der angeschlossenen Mechanik während der Funktionsgeneratorausgabe eingehalten werden.

Optional: Inhalt der Kurventabellen löschen

1. Stellen Sie sicher, dass die gewählte Kurventabelle **nicht** mit einem Funktionsgenerator verbunden ist, für den die Ausgabe gestartet ist. Details siehe "Funktionsgenerator konfigurieren" (S. 90) und "Funktionsgeneratorausgabe stoppen" (S. 93).
2. Löschen Sie mit dem Befehl `WCL` (S. 219) den Inhalt der Kurventabellen.

   Der komplette Inhalt der gewählten Kurventabelle wird gelöscht. Das segmentweise Löschen des Kurventabelleninhaltes ist **nicht** möglich.

INFORMATION

Beim Ausschalten oder Neustart des C-886 wird der Inhalt der Kurventabellen automatisch gelöscht.

Beispiele für das Erstellen von Kurvenformen

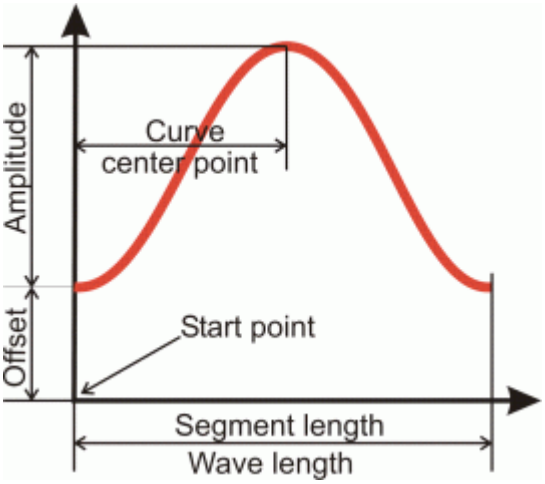
Die folgenden Beispiele helfen Ihnen beim Erstellen der Kurvenform.

Sinuskurve 1

- Symmetrische Sinuskurve mit Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 2 X SIN_P 2000 0.2 0.1 2000 0 1000`

<WaveTableID> = 2
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 2000
<Amp> = 0.2
<Offset> = 0.1
<WaveLength> = 2000
<StartPoint> = 0
<CurveCenterPoint> = 1000

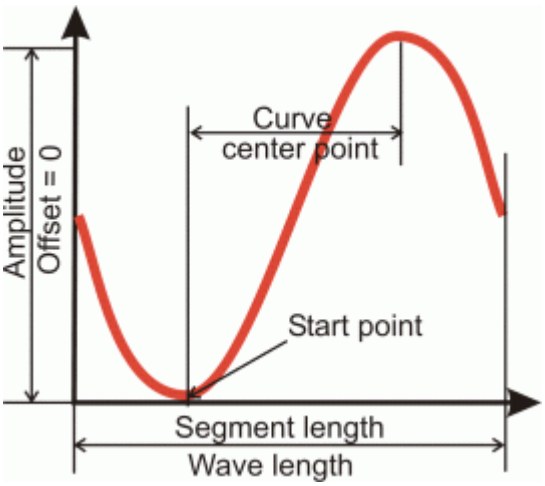


Sinuskurve 2

- Symmetrische Sinuskurve ohne Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 2 X SIN_P 2000 0.3 0 2000 499 1000`

<WaveTableID> = 2
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 2000
<Amp> = 0.3
<Offset> = 0
<WaveLength> = 2000
<StartPoint> = 499
<CurveCenterPoint> = 1000

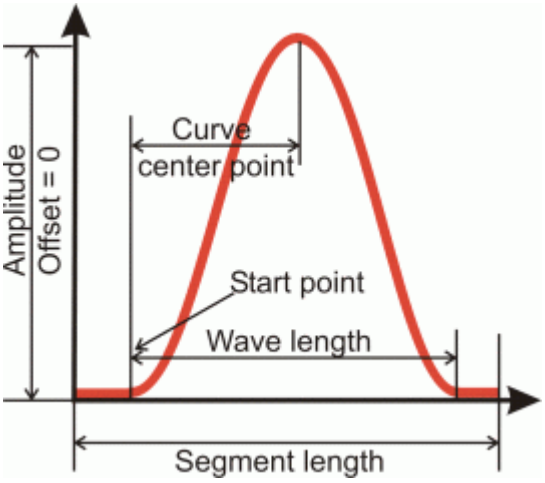


Sinuskurve 3

- Symmetrische Sinuskurve ohne Offset
- Segment wird an den Inhalt der Kurventabelle angehängt

Befehl: `WAV 2 & SIN_P 2000 0.25 0 1800 100 900`

<WaveTableID> = 2
<AppendWave> = &
<WaveType> = SIN_P
<SegLength> = 2000
<Amp> = 0.25
<Offset> = 0
<WaveLength> = 1800
<StartPoint> = 100
<CurveCenterPoint> = 900

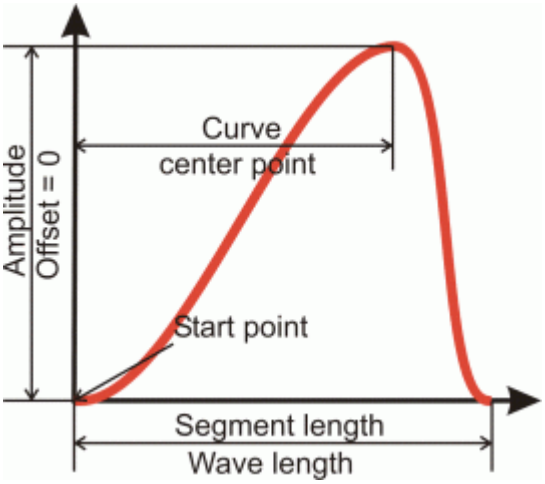


Sinuskurve 4

- Asymmetrische Kurve ohne Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 3 X SIN_P 4000 0.2 0 4000 0 3100`

<WaveTableID> = 3
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 4000
<Amp> = 0.2
<Offset> = 0
<WaveLength> = 4000
<StartPoint> = 0
<CurveCenterPoint> = 3100

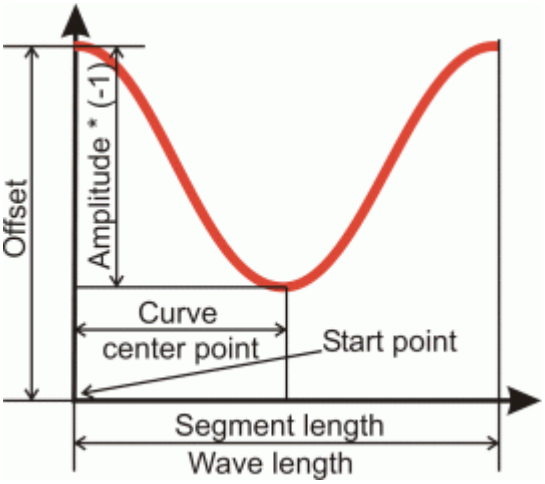


Sinuskurve 5

- Symmetrische Kurve mit negativer Amplitude
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 1 X SIN_P 1000 -0.3 0.45 1000 0 500`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = SIN_P
<SegLength> = 1000
<Amp> = –0.3
<Offset> = 0.45
<WaveLength> = 1000
<StartPoint> = 0
<CurveCenterPoint> = 500

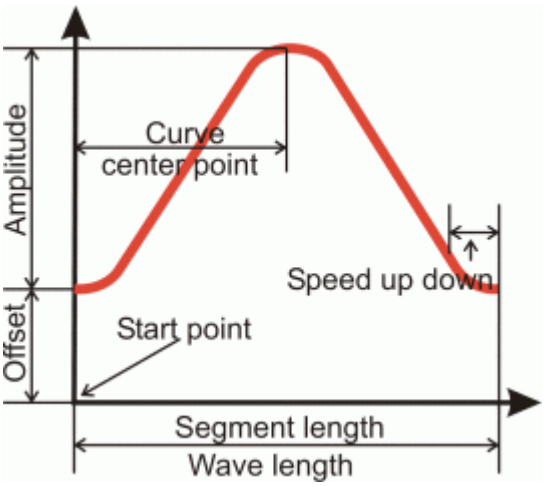


Rampenkurve 1

- Symmetrische Rampenkurve mit Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 4 X RAMP 2000 0.2 0.1 2000 0 300 1000`

<WaveTableID> = 4
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 2000
<Amp> = 0.2
<Offset> = 0.1
<WaveLength> = 2000
<StartPoint> = 0
<SpeedUpDown> = 300
<CurveCenterPoint> = 1000

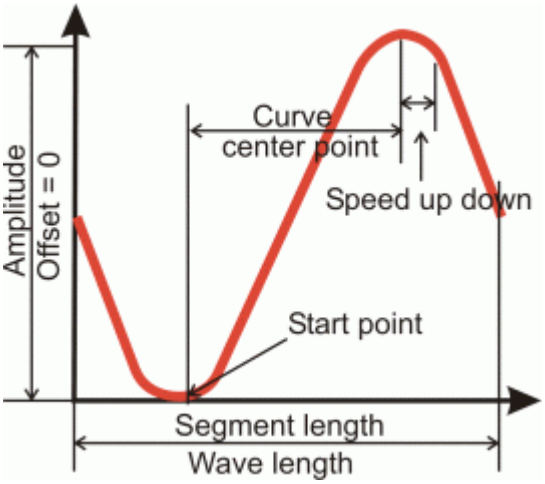


Rampenkurve 2

- Symmetrische Rampenkurve ohne Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 4 X RAMP 2000 0.35 0 2000 499 300 1000`

<WaveTableID> = 4
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 2000
<Amp> = 0.35
<Offset> = 0
<WaveLength> = 2000
<StartPoint> = 499
<SpeedUpDown> = 300
<CurveCenterPoint> = 1000

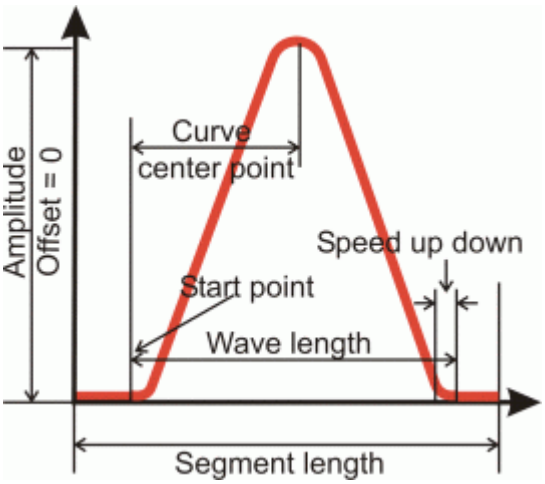


Rampenkurve 3

- Symmetrische Rampenkurve ohne Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 5 X RAMP 2000 0.15 0 1800 120 150 900`

<WaveTableID> = 5
<AppendWave> = X
<WaveType> = RAMP
<SegLength> = 2000
<Amp> = 0.15
<Offset> = 0
<WaveLength> = 1800
<StartPoint> = 120
<SpeedUpDown> = 150
<CurveCenterPoint> = 900

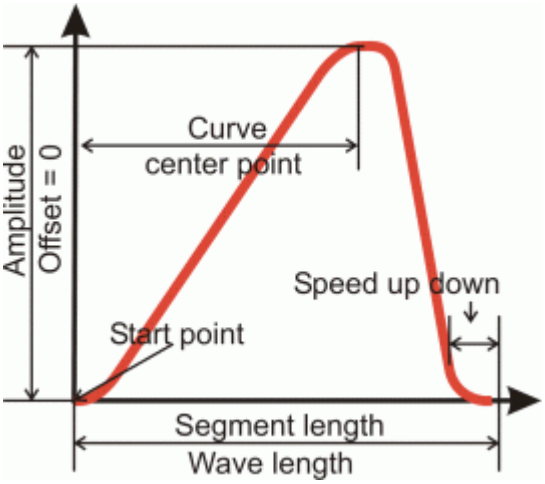


Rampenkurve 4

- Asymmetrische Rampenkurve ohne Offset
- Segment wird an den Inhalt der Kurventabelle angehängt

Befehl: `WAV 5 & RAMP 3000 0.35 0 3000 0 200 2250`

<WaveTableID> = 5
<AppendWave> = &
<WaveType> = RAMP
<SegLength> = 3000
<Amp> = 0.35
<Offset> = 0
<WaveLength> = 3000
<StartPoint> = 0
<SpeedUpDown> = 200
<CurveCenterPoint> = 2250

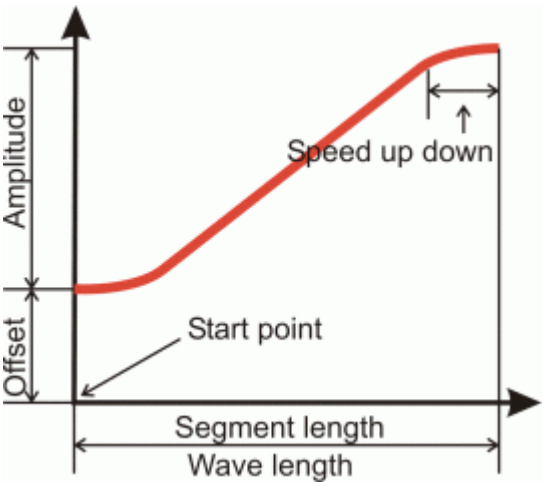


Einzelne Abtastzeile 1

- Abtastzeile mit Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 1 X LIN 1500 0.3 0.15 1500 0 370`

<WaveTableID> = 1
<AppendWave> = X
<WaveType> = LIN
<SegLength> = 1500
<Amp> = 0.3
<Offset> = 0.15
<WaveLength> = 1500
<StartPoint> = 0
<SpeedUpDown> = 370

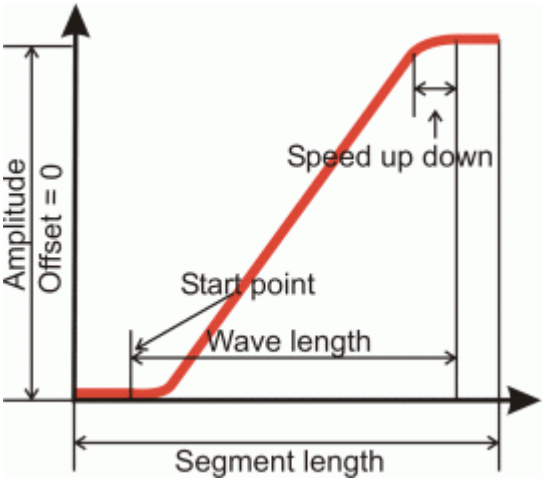


Einzelne Abtastzeile 2

- Abtastzeile ohne Offset
- Segment überschreibt den Inhalt der Kurventabelle

Befehl: `WAV 2 X LIN 1500 0.4 0 1100 210 180`

<WaveTableID> = 2
<AppendWave> = X
<WaveType> = LIN
<SegLength> = 1500
<Amp> = 0.4
<Offset> = 0
<WaveLength> = 1100
<StartPoint> = 210
<SpeedUpDown> = 180

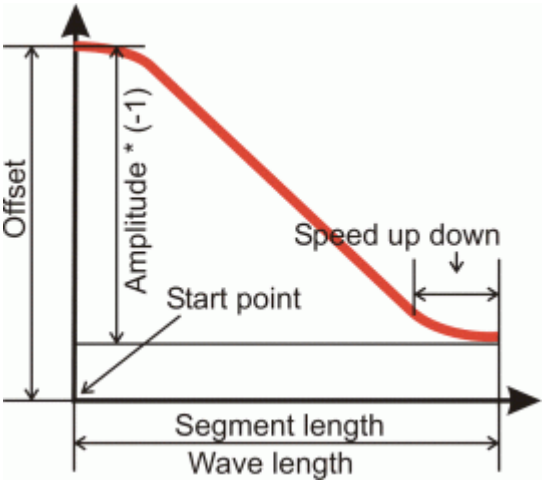


Einzelne Abtastzeile 3

- Abtastzeile mit negativer Amplitude
- Segment wird an den Inhalt der Kurventabelle angehängt

Befehl: `WAV 2 & LIN 3000 -0.4 0.5 3000 0 650`

<WaveTableID> = 2
<AppendWave> = &
<WaveType> = LIN
<SegLength> = 3000
<Amp> = –0.4
<Offset> = 0.5
<WaveLength> = 3000
<StartPoint> = 0
<SpeedUpDown> = 650



7.3.4 Funktionsgenerator konfigurieren

Das Konfigurieren des Funktionsgenerators umfasst folgende Schritte:

- Funktionsgenerator und Kurventabelle verbinden oder trennen (S. 90)
- Optional: Anzahl der Ausgabezyklen einstellen (S. 90)
- Optional: Ausgaberate einstellen (S. 90)

In diesem Handbuch finden Sie ein Beispiel für das Einstellen der Ausgaberate (S. 91).

Funktionsgenerator und Kurventabelle verbinden oder trennen

➢ Fragen Sie mit dem Befehl `WSL?` (S. 227) die aktuelle Verbindung von Funktionsgenerator und Kurventabelle ab.

➢ Verbinden oder trennen Sie Funktionsgenerator und Kurventabelle:

a) Stellen Sie sicher, dass für den gewählten Funktionsgenerator **nicht** die Ausgabe gestartet ist. Details siehe "Funktionsgeneratorausgabe stoppen" (S. 93).

b) Verbinden Sie mit dem Befehl `WSL` (S. 227) die gewählte Kurventabelle mit dem gewählten Funktionsgenerator, oder trennen Sie die Verbindung des ausgewählten Generators zu einer Kurventabelle.

Zwei oder mehr Generatoren können mit derselben Kurventabelle verbunden sein, ein Generator kann jedoch nicht mit mehreren Kurventabellen verbunden sein.

Optional: Anzahl der Ausgabezyklen einstellen

Die werkseitige Standardeinstellung für die Anzahl der Ausgabezyklen ist 0. Mit der werkseitigen Einstellung erfolgt die Ausgabe der Kurvenform ohne zeitliche Begrenzung, bis sie mit den Befehlen `WGO` (S. 220) oder `HLT` (S. 145) oder `#24` (S. 127) oder `STP` (S. 202) gestoppt wird.

➢ Senden Sie den Befehl `WGC?` (S. 220), um die aktuelle Einstellung für die Anzahl der Ausgabezyklen des Funktionsgenerators abzufragen.

➢ Stellen Sie mit dem Befehl `WGC` (S. 220) die Anzahl der Ausgabezyklen des Funktionsgenerators ein.

Optional: Ausgaberate einstellen

Durch das Anpassen der Ausgaberate können die einzelnen Ausgabezyklen der Kurvenform verlängert werden. Die Dauer eines Ausgabezyklus für die Kurvenform kann wie folgt berechnet werden:

Ausgabedauer = Zykluszeit des C-886 * Ausgaberate * Anzahl der Punkte

wobei

- die Zykluszeit des C-886 durch den Parameter 0x0E000200 angegeben wird (in Sekunden)

- die Ausgaberate der Anzahl der C-886-Zyklen entspricht, über die sich die Ausgabe eines Kurvenpunkts zeitlich erstreckt
- die Anzahl der Punkte der Länge der Kurvenform (d. h. der Länge der Kurventabelle) entspricht

Die Ausgabe der Kurvenpunkte erfolgt standardmäßig mit Interpolation, um Positionssprünge der Achsen zu vermeiden.

➢ Senden Sie den Befehl `WTR?` (S. 230), um die aktuellen Einstellungen für Ausgaberate und Interpolation abzufragen.

➢ Stellen Sie mit dem Befehl `WTR` (S. 228) die Ausgaberate und die Interpolation ein.

Für die einzelnen Funktionsgeneratoren des C-886 können unterschiedliche Ausgaberaten eingestellt werden.

INFORMATION

Wenn beim Einstellen der Ausgaberate mit dem Befehl `WTR` die Funktionsgeneratorkennung 0 verwendet wird, wird die Ausgaberate für alle Funktionsgeneratoren auf denselben Wert gesetzt.

Beispiel für das Einstellen der Ausgaberate

Aktion	Befehl	Ergebnis
Sinuskurve für Kurventabelle 2 definieren.	`WAV 2 X SIN_P 2000 0.2 0.1 2000 0 1000`	Die Länge der Kurvenform und damit die Anzahl der Punkte in der Kurventabelle ist 2000.
Zykluszeit des C-886 auslesen.	`SPA? 1 0x0E000200`	Die Zykluszeit des C-886 ist 10 ms
Aktuelle Ausgaberate auslesen.	`WTR?`	Standardwert für die Ausgaberate = 1 (jeder Punkt in der Kurventabelle wird während eines C-886-Zyklus ausgegeben) Dauer eines Ausgabezyklus (siehe Berechnungsformel oben): 0,01 s • 1 • 2000 = 20 s

Aktion	Befehl	Ergebnis
Anzahl der C-886-Zyklen pro Punkt der Kurventabelle für alle Funktionsgeneratoren verdreißigfachen.	`WTR 0 30 1`	Dauer eines Ausgabezyklus (siehe Berechnungsformel oben): 0,01 s • 30 • 2000 = 10 min Zwischen der Ausgabe der Punkte interpoliert der C-886 linear, um Positionssprünge zu vermeiden.

7.3.5 Ausgabe starten und stoppen

Der Funktionsgenerator gibt absolute Zielpositionen aus.

- Funktionsgeneratorausgabe starten (S. 93)
- Funktionsgeneratorausgabe stoppen (S. 93)
- Optional: Aktivierungszustand und Status des Funktionsgenerators abfragen (S. 93)
- Optional: Datenaufzeichnung während Funktionsgeneratorausgabe starten (S. 94)

In diesem Handbuch finden Sie Beispiele für das Starten/Stoppen der Funktionsgeneratorausgabe (S. 94).

INFORMATION

Wenn die Funktionsgeneratorausgabe aktiv ist, sind Befehle zum Starten oder Konfigurieren von Bewegungen sowie das Ausführen entsprechender Makros **nicht** zulässig.

INFORMATION

Während der Funktionsgeneratorausgabe prüft der C-886 ständig, ob die Bewegung noch möglich ist. In folgenden Fällen stoppt der C-886 die Bewegung abrupt und setzt einen Fehlercode:

- Die auszugebenden Zielpositionen können nicht erreicht werden.
- Die erforderliche Geschwindigkeit kann nicht erreicht werden.
- Die Bewegung würde eine Kollision verursachen.

➢ Fragen Sie mit dem Befehl `WGS?` den aktuellen Status der Funktionsgeneratoren ab, insbesondere den Index der Kurvenformpunkte, an denen ein Fehler aufgetreten ist.

➢ Fragen Sie mit dem Befehl `ERR?` (S. 139) den Fehlercode des zuletzt aufgetretenen Fehlers ab.

Voraussetzungen

✓ Sie haben die gewünschte Kurvenform erstellt (S. 82).

✓ Sie haben den Funktionsgenerator mit der entsprechenden Kurventabelle verbunden (S. 90).

Funktionsgeneratorausgabe starten

- Starten Sie die Funktionsgeneratorausgabe mit dem Befehl `WGO` (S. 220).

  Alle Funktionsgeneratoren, deren Ausgabe gleichzeitig aktiv sein soll, müssen im selben Befehl gestartet werden. Für die Achsen der Bewegungsplattform des Positionierers, deren Funktionsgenerator nicht gestartet wurde, wird immer die zuletzt gültige Zielposition kommandiert.

  Die Ausgabe erfolgt synchron mit den Servozyklen des C-886.

  Beim Start der Funktionsgeneratorausgabe wird automatisch ein Datenaufzeichnungszyklus gestartet.

Funktionsgeneratorausgabe stoppen

- Stoppen Sie die Funktionsgeneratorausgabe durch Senden eines der folgenden Befehle:
  - `WGO` (S. 220) stoppt die angegebenen Funktionsgeneratoren
  - `STP` (S. 202) stoppt alle Funktionsgeneratoren
  - `#24` (S. 127) stoppt alle Funktionsgeneratoren
  - `HLT` (S. 145) stoppt die Funktionsgeneratoren der angegebenen Achsen

  Wenn die Funktionsgeneratorausgabe durch Senden von `STP`, `#24` oder `HLT` gestoppt wird, setzt der C-886 den Fehlercode 10 (Abfrage mit dem Befehl `ERR?` (S. 139)).

Wenn die Anzahl der Ausgabezyklen begrenzt wurde (S. 90), wird die Funktionsgeneratorausgabe nach Erreichen der vorgegebenen Zyklenzahl automatisch gestoppt.

INFORMATION

Das Beenden der PC-Software stoppt **nicht** die Funktionsgeneratorausgabe.

Optional: Aktivierungszustand des Funktionsgenerators abfragen

- Fragen Sie mit dem Befehl `#9` (S. 126) ab, ob die Funktionsgeneratorausgabe läuft.
- Fragen Sie mit dem Befehl `WGO?` (S. 222) die zuletzt kommandierten Start-/Stopp-Einstellungen des Funktionsgenerators ab.

  Die Antwort auf `WGO?` wird auch durch das Stoppen der Funktionsgeneratorausgabe mit `#24` (S. 127), `STP` (S. 202) oder `HLT` (S. 145) und Ende der Ausgabe nach einer festgelegten Anzahl von Ausgabezyklen beeinflusst.

- Fragen Sie mit dem Befehl `WGS?` den Status des Funktionsgenerators ab:
  - Läuft der Funktionsgenerator?
  - Wie viele Zyklen hat der Funktionsgenerator seit dem letzten Start ausgegeben?
  - Fehlercode des zuletzt während der Ausgabe aufgetretenen Fehlers
  - Index des Kurvenformpunktes, an dem der Fehler aufgetreten ist

Optional: Datenaufzeichnung während Funktionsgeneratorausgabe starten

- Starten Sie die Datenaufzeichnung während der Funktionsgeneratorausgabe, indem Sie den Befehl `WGR` (S. 222) senden.

Beim Starten der Funktionsgeneratorausgabe (S. 93) startet automatisch ein erster Datenaufzeichnungszyklus.

Die aufgezeichneten Daten können mit dem Befehl `DRR?` (S. 135) ausgelesen werden. Weitere Informationen siehe "Datenrekorder" (S. 76).

Beispiele für das Starten/Stoppen der Funktionsgeneratorausgabe

Aktion	Befehl	Ergebnis
Sinuskurve für Kurventabelle 4 definieren.	`WAV 4 X SIN_P 2000 1 0 2000 0 1000`	Die Länge der Kurvenform und damit die Anzahl der Punkte in der Kurventabelle ist 2000.
Sinuskurve für Kurventabelle 5 definieren.	`WAV 5 X SIN_P 2000 0.2 0 2000 0 1000`	Die Länge der Kurvenform und damit die Anzahl der Punkte in der Kurventabelle ist 2000.
Funktionsgenerator 1 (Achse X) mit der Kurventabelle 4 verbinden. Funktionsgenerator 3 (Achse Z) mit der Kurventabelle 5 verbinden.	`WSL 1 4 3 5`	Voraussetzung für Ausgabe des Funktionsgenerators erfüllt: ohne Zuweisung einer Kurventabelle ist keine Ausgabe des Funktionsgenerators möglich.
Funktionsgeneratoren 1 und 3 starten.	`WGO 1 1 3 1`	Die in den Kurventabellen 4 und 5 definierten Kurvenformen werden ausgegeben. Für die Achsen Y, U, V und W der Bewegungsplattform des Positionierers, deren Funktionsgeneratoren nicht gestartet wurden, werden die zuletzt gültigen Zielpositionen kommandiert.
Funktionsgeneratoren 1 und 3 stoppen.	`WGO 1 0 3 0`	Die Ausgabe der Kurvenpunkte (und damit die Bewegung des Positionierers) wird gestoppt.

Alle Befehle bis auf die letzten beiden Zeilen wie im Beispiel oben; die letzten beiden Zeilen werden durch die folgenden Befehle ersetzt:

Aktion	**Befehl**	**Ergebnis**
Anzahl der Ausgabezyklen für Funktionsgeneratoren 1 und 3 auf 100 begrenzen.	`WGC 1 100 3 100`	Ein Befehl zum Stoppen der Funktionsgeneratoren ist nicht erforderlich.
Funktionsgeneratoren 1 und 3 starten.	`WGO 1 1 3 1`	Die Ausgabe der Funktionsgeneratoren endet automatisch nach 100 Ausgabezyklen.

7.3.6 Anwendungstipps: Kundenspezifische Daten laden

Sie können kundenspezifische Daten (Zeitreihen) aus Dateien vom PC in den C-886 laden und als benutzerdefinierte Kurven verwenden. Vor dem Laden müssen Sie die Daten in das GCS-Array-Format umwandeln. Für das Laden der umgewandelten Daten in den C-886 verwenden Sie das PI Wave Generator Tool oder das PI Frequency Generator Tool (beide verfügbar in PIMikroMove®). Die Verwendung des PI Wave Generator Tool ist nachfolgend beschrieben; für die Verwendung des PI Frequency Generator Tool siehe die Technical Note A000T0057.

Daten in das GCS-Array-Format umwandeln

Das GCS-Array-Format basiert auf ASCII-Zeichen. Eine GCS-Array-Datei besteht aus einem Header-Abschnitt und einem Datentabellen-Abschnitt. Als Dezimaltrennzeichen kann ein Komma oder ein Punkt verwendet werden. Eine ausführliche Beschreibung des GCS-Array-Formats siehe separates Handbuch (SM146E).

Wandeln Sie die Daten Ihrer Zeitreihe wie folgt in das GCS-Array-Format um:

1. Erzeugen Sie eine Textdatei mit der Dateiendung ".dat" auf dem PC, z. B. "New_GCS_Array.dat".
2. Fügen Sie den Header zur Datei hinzu. Struktur und Inhalt der Headers müssen wie folgt aufgebaut sein:

   Template für den GCS-Array-Header:

   `[GCS_ARRAY` **Target Positions XY**`]`

   `# VERSION = 1`

   `# TYPE = 1`

   `# SEPARATOR =` **32**

   `# DIM = 2`

   `# SAMPLE_TIME =` **0,001**

   `# NDATA =` **256**

```
#
# NAME0 = TARGET POSITION X
# NAME1 = TARGET POSITION Y
#
# DISP_UNIT0 = mm
# DISP_UNIT1 = mm
# END_HEADER
```

Passen Sie die folgenden Angaben im Header an Ihre Daten an (**markiert** in der Liste oben):

`DIM`
Anzahl der Spalten in der Datentabelle (d.h. Anzahl der Zeitreihen), Minimum ist 1

`NDATA`
Anzahl der Zeilen in der Datentabelle (d.h. Anzahl der Datenpunkte pro Zeitreihe)

`NAME%`
Optional: Beschreibung der Spalte % (wobei % ein positiver ganzzahliger Wert < DIM ist; die Zählung beginnt mit 0)

`DISP_UNIT%`
Optional: Beschreibung der physikalischen Einheit für Spalte % (wobei % ein positiver ganzzahliger Wert < DIM ist; die Zählung beginnt mit 0)

`SAMPLE_TIME`
Samplezeit der Daten (in s), d.h. der Zeitabstand zwischen zwei Datenpunkten der Zeitreihe

`SEPARATOR`
Separator für die Spalten in der Datentabelle. Muss als dezimales ASCII-Zeichen angegeben werden (z. B. 9 für TAB oder 32 für SPACE)

`[GCS_ARRAY %]`
Optional: % ist der Name der Datentabelle

3. Fügen Sie Ihre Daten zur Datei hinzu gemäß den Header-Angaben für `DIM`, `NDATA` und `SEPARATOR`:

   Datentabellen-Abschnitt der GCS-Array-Datei, Leerzeichen werden hier als SP dargestellt:

   `Value1_TimeSeries0`SP`Value1_TimeSeries1`SP…

   `Value2_TimeSeries0`SP`Value2_TimeSeries1`SP…

   `Value3_TimeSeries0`SP`Value3_TimeSeries1`SP…

   …

   Das folgende Beispiel zeigt ein komplettes GCS-Array mit Daten für zwei Zeitreihen. Die erste Zeitreihe "TARGET POSITION X" enthält die Werte 1/2/3, und die zweite Zeitreihe "TARGET POSITION Y" enthält die Werte 0.1/0.2/0.3.

```
# REM data recorded with C-886 controller
#
# VERSION = 1
# TYPE = 1
# SEPARATOR = 32
# DIM = 2
# SAMPLE_TIME = 0.009009
# NDATA = 3
#
# NAME0 = TARGET POSITION X
# NAME1 = TARGET POSITION Y
#
# DISP_UNIT0 = mm
# DISP_UNIT1 = mm
# END_HEADER
1 0.1
2 0.2
3 0.3
```

Das Ergebnis der Umwandlung ist eine GCS-Array-Datei mit der Dateiendung ".dat", die die Zeitreihen-Daten enthält. Die Datei kann mit dem PI Wave Generator Tool (siehe unten) oder dem PI Frequency Generator Tool in den C-886 importiert werden.

GSC-Array-Daten laden

Das PI Wave Generator Tool ist in PIMikroMove® verfügbar. Weitere Informationen zum PI Wave Generator Tool siehe PIMikroMove®-Handbuch (SM148E).

1. Öffnen Sie das PI Wave Generator Tool aus dem Hauptfenster von PIMikroMove® über den Menüeintrag ***C-886 > Show wave generator...***.

2. Klicken Sie im Hauptfenster des PI Wave Generator Tools auf ***Load Data Set*** , um das Fenster ***Load Data Set for Wave Tables*** zu öffnen.

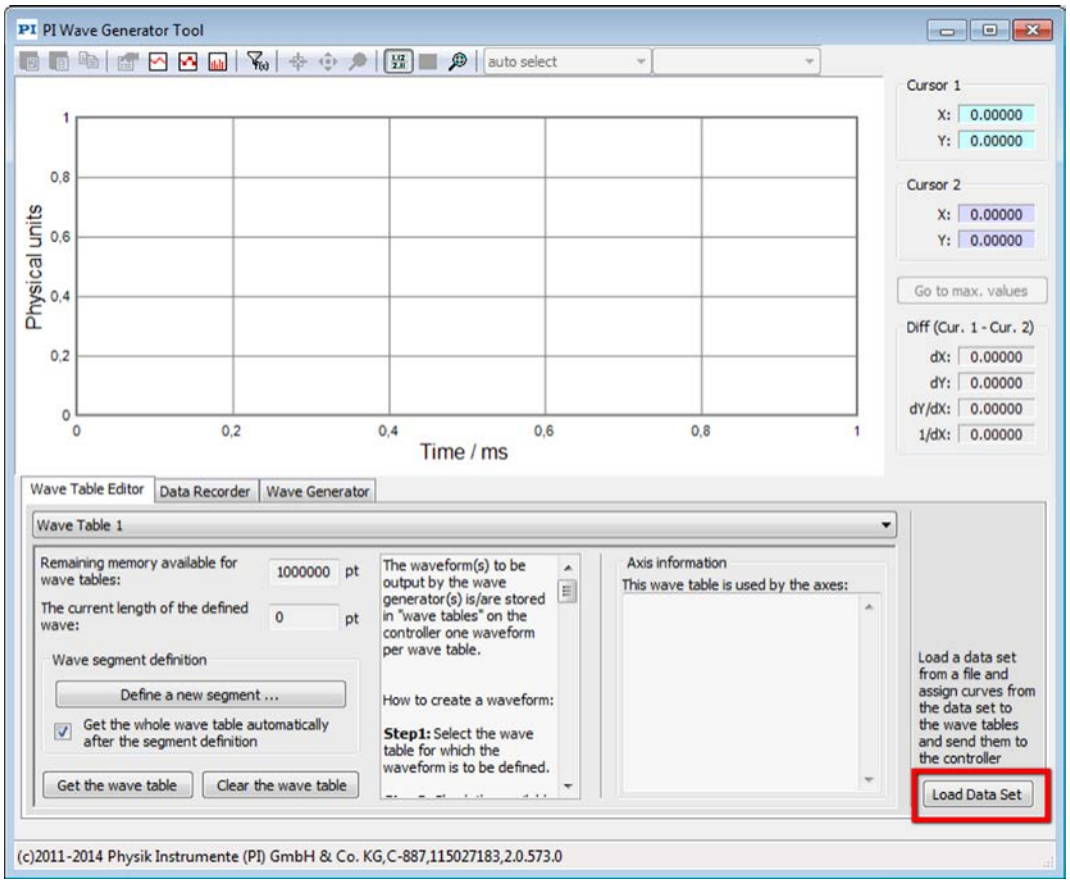


Abbildung 9: Hauptfenster des PI Wave Generator Tools

3. Importieren Sie die GCS-Array-Datei im Fenster ***Load Data Set for Wave Tables***.

Die Zeitreihen-Daten, die aus der Datei geladen wurden, werden im Grafikfeld angezeigt. Die Abbildung unten zeigt die Daten aus dem Beispiel in "Daten in das GCS-Array-Format umwandeln".

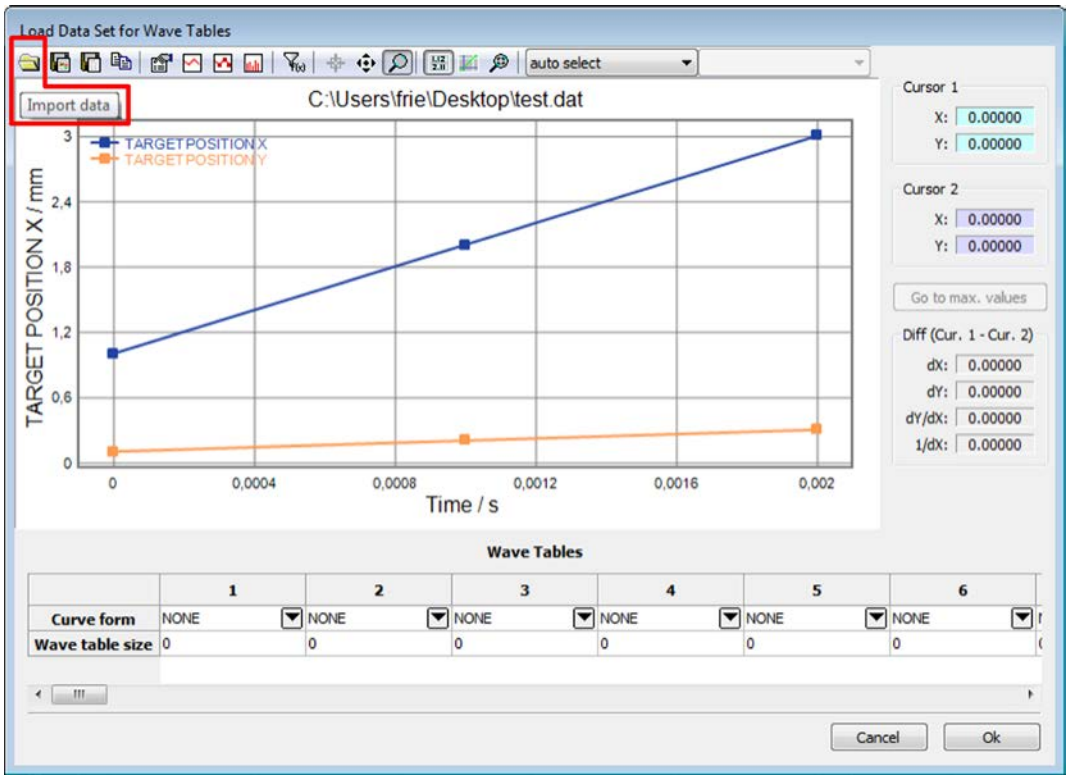


Abbildung 10: Fenster "Load Data Set for Wave Tables" mit grafischer Anzeige der geladenen Daten

4. Weisen Sie die einzelnen Zeitreihen den Kurventabellen des C-886 zu. Für die Zuweisung öffnen Sie das Auswahlmenü einer Kurventabelle und wählen die gewünschten Daten (siehe Abbildung unten).

Beachten Sie, dass die Daten erst dann an den C-886 gesendet werden, wenn Sie in der unteren rechten Ecke des Fensters auf ***Ok*** klicken.

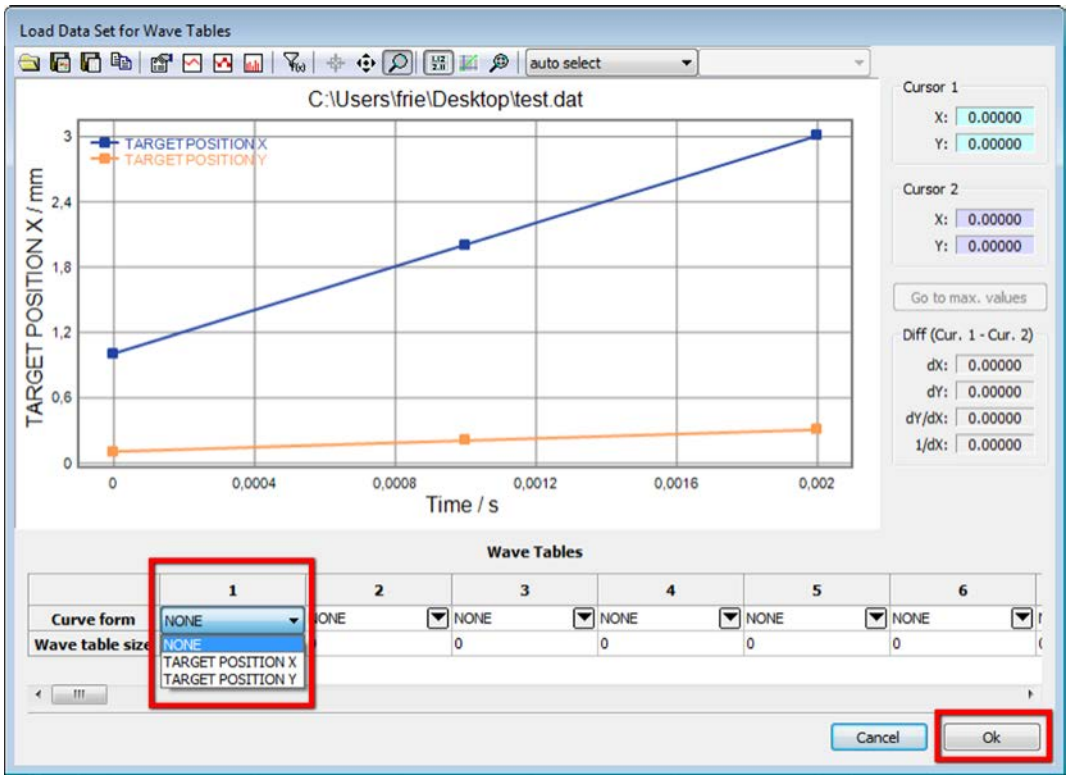


Abbildung 11: Zuweisung der geladenen Daten zu Kurventabellen, hier für Kurventabelle 1

7.3.7 Anwendungstipps: Makros für Funktionsgenerator verwenden

Die folgenden Einstellungen für den Funktionsgenerator lassen sich nur im flüchtigen Speicher des C-886 ändern und gehen beim Ausschalten oder Neustart des C-886 verloren:

- Inhalt der Kurventabellen (definiert mit WAV)
- Verbindung von Kurventabellen mit Funktionsgeneratoren (eingestellt mit WSL)
- Ausgaberate (eingestellt mit WTR)
- Anzahl der Ausgabezyklen (eingestellt mit WGC)

Mit der Makrofunktionalität des C-886 können Sie die Einstellungen des Funktionsgenerators dauerhaft im C-886 speichern. Zusätzlich können Sie ein Startup-Makro verwenden, um bei jedem Einschalten oder Neustart des C-886 den Funktionsgenerator zu konfigurieren und die Ausgabe zu starten.

Weitere Informationen zur Makrofunktionalität siehe "Controllermakros" (S. 103).

Im folgenden Beispiel werden zwei Makros verwendet, um für die Achsen U und V (Funktionsgeneratoren 4 und 5) des Positionierers die Funktionsgeneratorausgabe zu konfigurieren und zu starten:

- Das Makro "UVdata" enthält die Definition der Kurvenformen für die Kurventabellen 1 und 2 (WAV-Befehle).
- Das Makro "Start" führt die folgenden Aktionen aus:
  a) Durch Aufruf des Makros "UVdata" die Kurvenformen in die Kurventabellen 1 und 2 schreiben.
  b) Kurventabelle 1 mit Funktionsgenerator 4 verbinden und Kurventabelle 2 mit Funktionsgenerator 5 (Befehl WSL).
  c) Ausgaberate für Funktionsgeneratoren 4 und 5 auf 10 setzen, mit linearer Interpolation (Befehl WTR).
  d) Anzahl der Ausgabezyklen für Funktionsgeneratoren 4 und 5 auf 100 setzen (Befehl WGC).
  e) Optional, aber empfohlen, um Sprünge der Achsen zu vermeiden: Achsen U und V zur Startposition der Funktionsgeneratorausgabe kommandieren.
  f) Ausgabe für Funktionsgeneratoren 4 und 5 - und damit die Bewegung der Achsen U und V - starten (Befehl WGO).

Inhalt des Makro "UVdata":

```
WAV 1 X PNT 1 1 0
WAV 1 & PNT 1 1 5.654625319357E-06
WAV 1 & PNT 1 1 3.091004951757E-05
[…]
WAV 1 & PNT 1 1 0.000148233661

WAV 2 X PNT 1 1 0
WAV 2 & PNT 1 1 -4.742444189387E-06
WAV 2 & PNT 1 1 -3.027352414704E-05
[…]
WAV 2 & PNT 1 1 -0.000257643502
```

Inhalt des Makros "Start", wobei `startposU` und `startposV` die Startposition der Funktionsgeneratorausgabe bezeichnen:

```
MAC START UVdata
WSL 4 1 5 2
WTR 4 10 1 5 10 1
```

```
WGC 4 100 5 100
MOV U startposU V startposV
WGO 4 1 5 1
```

Anmerkungen:

Im Beispielmakro "UVdata" wird zur besseren Übersicht pro WAV-Befehl nur ein Kurvenpunkt (= Zielposition) angegeben. Für schnellere Verarbeitung wird empfohlen, mehr als einen Punkt pro WAV-Befehl zu definieren (pro Befehlszeile sind maximal 256 Zeichen zulässig).

Kurvenform-Definitionen können leicht erstellt werden, indem der Inhalt des Logfensters von PIMikroMove® in ein Makro kopiert wird:

1. Öffnen Sie das Logfenster über den Menüeintrag ***C-886 > Log window...*** im Hauptfenster von PIMikroMove®.
2. Öffnen Sie aus dem Menü ***C-886*** im Hauptfenster von PIMikroMove® das PI Frequency Generator Tool oder das PI Wave Generator Tool.
3. Definieren Sie mit dem Tool eine Kurvenform und senden Sie die Kurvenform an den C-886.

   Die gesendeten Befehle werden im Logfenster aufgelistet.

   Beispielsweise können Sie mit dem PI Frequency Generator Tool eine Sinuskurve mit einer bestimmten Frequenz definieren. Die entsprechenden Befehle WAV … PNT können Sie dann aus dem Logfenster in ein Makro kopieren. Die Befehle WAV … PNT im Logfenster werden durch die Funktion PI_WAV_PNT() der von PIMikroMove® verwendeten PI GCS2 DLL erzeugt. Die Funktion PI_WAV_PNT() verteilt die Kurvenpunkte automatisch auf die einzelnen Befehle WAV … PNT.

Wenn das Beispielmakro "Start" als Startup-Makro verwendet wird, kann vor dem Start der Funktionsgeneratorausgabe eine Referenzfahrt erforderlich sein. In diesem Fall müssen die folgenden Zeilen vor der Zeile `MOV U startposU V startposV` in das Makro eingefügt werden:

```
FRF X
WAC FRF? X = 1
```

Die Zeilen starten eine Referenzfahrt und warten, bis die Referenzfahrt erfolgreich beendet wurde.

7.4 Controllermakros

7.4.1 Übersicht: Makrofunktionalitäten und Beispielmakros

Der C-886 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-886 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 in mehreren Verschachtelungsebenen aufrufen.
- Variablen (S. 112) können für das Makro und im Makro selbst gesetzt und in verschiedenen Operationen verwendet werden.

In diesem Handbuch finden Sie Beispielmakros für folgende Aufgaben:

- Achse hin und her bewegen (S. 107)
- Achse mit variablem Verfahrweg hin und her bewegen (S. 109)
- Referenzfahrt für Positionierer durch Startup-Makro auslösen (S. 111)

7.4.2 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. 128)	ADD <Variable> <FLOAT1> <FLOAT2>	Kann nur in Makros verwendet werden. Addiert zwei Werte und speichert das Ergebnis als Variable (S. 112).
CPY (S. 131)	CPY <Variable> <CMD?>	Kann nur in Makros verwendet werden. Kopiert eine Antwort auf einen Befehl in eine Variable (S. 112).
DEL (S. 132)	DEL <uint>	Kann nur in Makros verwendet werden. Bewirkt eine Verzögerung um <uint> Millisekunden.
JRC (S. 150)	JRC <Jump> <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Löst einen relativen Sprung des Makroausführungszeigers in Abhängigkeit von einer Bedingung aus.

Befehl	Syntax	Funktion
MAC (S. 179)	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 FREE?	Fragt nach dem freien Speicherplatz für die Makroaufzeichnung.
	MAC NSTART <macroname> <uint> [{<String>}]	Startet das angegebene Makro n-mal hintereinander (n = Anzahl der Ausführungen). Mit <String> können die Werte lokaler Variablen für das Makro gesetzt werden.
	MAC START <macroname> [{<String>}]	Startet eine Ausführung des angegebenen Makros. Mit <String> können die Werte lokaler Variablen für das Makro gesetzt werden.
MAC? (S. 183)	MAC? [<macroname>]	Listet alle Makros oder den Inhalt des angegebenen Makros auf.
MEX (S. 184)	MEX <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Stoppt die Makroausführung in Abhängigkeit von einer Bedingung.
RMC? (S. 192)	RMC?	Listet die aktuell laufenden Makros auf.
VAR (S. 208)	VAR <Variable> <String>	Setzt eine Variable (S. 112) auf einen bestimmten Wert oder löscht sie.
VAR? (S. 209)	VAR? [{<Variable>}]	Gibt Variablenwerte zurück.
WAC (S. 213)	WAC <CMD?> <OP> <Value>	Kann nur in Makros verwendet werden. Wartet, bis eine Bedingung erfüllt ist.
#8 (S. 126)	-	Prüft, ob ein Makro auf dem Controller ausgeführt wird.

INFORMATION

Pro Befehlszeile sind maximal 256 Zeichen zulässig.

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

7.4.3 Mit Makros arbeiten

Die Arbeit mit Makros umfasst Folgendes:

- Aufzeichnen von Makros (S. 105)
- Starten der Makroausführung (S. 108)
- Stoppen der Makroausführung (S. 110)
- Einrichten eines Startup-Makros (S. 111)
- Löschen von Makros (S. 112)

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-886 kann eine unbegrenzte Anzahl von Makros gleichzeitig speichern. Maximal 10 Verschachtelungsebenen sind in Makros möglich.

INFORMATION

Grundsätzlich können alle GCS-Befehle (S. 115) Bestandteil eines Makros werden. Ausnahmen:

- `RBT` für den Neustart des C-886
- `MAC BEG` und `MAC END` für die Makroaufzeichnung
- `MAC DEL` zum Löschen eines Makros
- `#27` zum Anhalten des C-886

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 weiterleiten.

INFORMATION

Um die Anwendung von Makros flexibler zu gestalten, können Sie in Makros lokale und globale Variablen verwenden. Weitere Informationen siehe "Variablen" (S. 112).

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. 112) in Makros, und geben Sie beim Starten der Makroausführung die entsprechenden Variablenwerte an.
- ➢ Wenden Sie sich an unseren Kundendienst (S. 267), wenn der C-886 unerwartetes Verhalten zeigt.

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-886 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-886 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 X 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 X 12.5
WAC ONT? X = 1
MAC END
MAC BEG macro2
MVR X -12.5
```

```
WAC ONT? X = 1
MAC END
MAC BEG macro3
MAC START macro1
MAC START macro2
MAC END
```

Makroausführung starten

INFORMATION

Während auf dem C-886 ein Makro ausgeführt wird, können über alle Kommunikationsschnittstellen Befehle an den C-886 gesendet werden. Befehle werden in der Reihenfolge ihres Eintreffens abgearbeitet. Die Zeitdauer für das Abarbeiten einzelner Befehle hängt vom jeweiligen Befehl ab. Die Ausführung eines zeitaufwändigen Befehls, der über eine Kommunikationsschnittstelle gesendet wurde, kann daher zu einer Pause in der Makroausführung führen. Makros sind deshalb z.B. nicht für das zeitkritische Abfahren festgelegter Dynamikprofile geeignet.

- Verwenden Sie zum Abfahren eines festgelegten zeitkritischen Dynamikprofils kein Makro, sondern aufeinander folgende MOV-Befehle, die in einem Zwischenspeicher abgelegt werden, siehe "Zyklische Übertragung von Zielpositionen" (S. 33).
- Wenn möglich, starten oder verwenden Sie während der Makroausführung keine PC-Software wie PIMikroMove® oder LabVIEW-Treiber.
- Wenn möglich, vermeiden Sie während der Makroausführung das Senden von Befehlen.

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 das 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. 247).

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 X 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-886 erneut an, da sie nur in den flüchtigen Speicher des C-886 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 X $1
   WAC ONT? X = 1
   MAC END
   ```

   MOVWAI bewegt die Achse X 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 X bewegt sich fünfmal im Wechsel zu den Positionen 5 und 15. Für die Anzahl der Ausführungen können Sie auch einen beliebigen anderen Wert wählen.

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 das PITerminal oder das Fenster ***Command entry*** von PIMikroMove® verwendet, um Befehle einzugeben. Details zur Arbeit mit der Registerkarte ***Controller macros*** in PIMikroMove® finden Sie imPIMikroMove® -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-886 ausgeführt.

INFORMATION

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

Im Folgenden wird das 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: Referenzfahrt für Positionierer durch Startup-Makro auslösen

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 startet eine Referenzfahrt. STARTCL wird als Startup-Makro festgelegt, so dass der Positionierer die Referenzfahrt sofort nach dem Einschalten ausführt.

- Senden Sie:

```
MAC BEG startcl
DEL 1000
FRF X
MAC END
MAC DEF startcl
```

Makro löschen

INFORMATION

Ein laufendes Makro kann nicht gelöscht werden.

Im Folgenden wird das 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.

7.4.4 Variablen

Für eine flexiblere Programmierung unterstützt der C-886 die Verwendung von Variablen in Makros. Während globale Variablen für alle Makros verwendet werden können, gelten lokale Variablen immer nur für ein bestimmtes Makro. Die Anzahl der globalen und lokalen Variablen ist nicht beschränkt.

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

Die Arbeit mit Variablen umfasst Folgendes:

- Erzeugen von Variablen:
  - Lokale Variablen: siehe "Besonderheiten lokaler Variablen"
  - Globale Variablen: innerhalb eines Makros mit den Befehlen `ADD` (S. 128), `CPY` (S. 131) und `VAR` (S. 208), außerhalb eines Makros mit `VAR`
- Ändern von Variablenwerten: innerhalb eines Makros mit den Befehlen `ADD`, `CPY` und `VAR`
- Abfragen von Variablenwerten: mit `VAR?` innerhalb oder außerhalb eines Makros
- Löschen von Variablen: mit `VAR` innerhalb eines Makros; globale Variablen auch außerhalb eines Makros

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.

Besonderheiten lokaler Variablen

- Die Namen der in einem Makro verwendeten lokalen Variablen müssen eine fortlaufende Reihe bilden. Beispiel für erlaubte Benennung: 1, 2, 3, 4. Nicht erlaubt ist z.B. die Benennung mit 1, 2, 5, 6.
- Die Werte lokaler Variablen werden als Argumente <String> der Befehle `MAC START` oder `MAC NSTART` beim Start des Makros angegeben.

  Befehlsformate:

  `MAC START <macroname> [{<String>}]`

  `MAC NSTART <macroname> <uint> [{<String>}]`

  <String> steht für den Wert einer im Makro enthaltenen lokalen Variablen. 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. Pro Befehlszeile sind maximal 256 Zeichen zulässig. <String> kann direkt oder über den Wert einer anderen Variable angegeben werden.

  <uint> bestimmt, wie oft das Makro ausgeführt werden soll. Weitere Informationen finden Sie in der Beschreibung des Befehls `MAC` (S. 179).
- 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.
- Die Werte lokaler Variablen können mit `VAR?` abgefragt werden, solange das Makro ausgeführt wird.

8 GCS-Befehle

In diesem Kapitel

Schreibweise 115
GCS-Syntax für Syntaxversion 2.0 115
Befehlsübersicht 118
Befehlsbeschreibungen für GCS 2.0 123
Fehlercodes 230

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 (ASCII-Zeichen 10) ist das Standard-Abschlusszeichen (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 endet mit dem Abschlusszeichen (LF).

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-886 unterstützten Kennungen sind im Abschnitt "Kommandierbare Elemente" (S. 25) beschrieben.

Beispiel 1:

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

Senden: `MOVSP1SP10.0LF`

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: `MOVSP1SP17.3SP2SP2.05LF`

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: `RPALF`

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

Pro Befehlszeile sind maximal 256 Zeichen zulässig.

8.3 Befehlsübersicht

Befehl	Format	Beschreibung
#3 (S. 123)	#3	Get Real Position
#4 (S. 123)	#4	Request Status Register
#5 (S. 124)	#5	Request Motion Status
#6 (S. 125)	#6	Query for Position Change
#7 (S. 125)	#7	Request Controller Ready Status
#8 (S. 126)	#8	Query if Macro is Running
#9 (S. 126)	#9	Get Wave Generator Status
#11 (S. 126)	#11	Get Memory Space for Trajectory Points
#24 (S. 127)	#24	Stop All Axes
#27 (S. 127)	#27	System Abort
*IDN? (S. 128)	*IDN?	Get Device Identification
ADD (S. 128)	ADD <Variable> <FLOAT1> <FLOAT2>	Add and Save to Variable
CCL (S. 129)	CCL <Level> [<PSWD>]	Set Command Level
CCL? (S. 130)	CCL?	Get Command Level
CPY (S. 131)	CPY <Variable> <CMD?>	Copy into Variable
CST? (S. 131)	CST? [{<AxisID>}]	Get Assignment Of Stages To Axes
CSV? (S. 132)	CSV?	Get Current Syntax Version
DEL (S. 132)	DEL <uint>	Delay the Command Interpreter
DIA? (S. 132)	DIA? [{<MeasureID>}]	Get Diagnosis Information
DPA (S. 133)	DPA <Pswd> [{<ItemID> <PamID>}]	Reset Volatile Memory Settings To Default
DRC? (S. 134)	DRC? [{<RecTableID>}]	Get Data Recorder Configuration
DRL? (S. 135)	DRL? [{<RecTableID>}]	Get Number Of Recorded Points
DRR? (S. 135)	DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]	Get Recorded Data Values
DRT (S. 137)	DRT {<RecTableID> <TriggerSource> <Value>}	Set Data Recorder Trigger Source
DRT? (S. 138)	DRT? [{<RecTableID>}]	Get Data Recorder Trigger Source
ECO? (S. 139)	ECO? <String>	Echo a String

Befehl	Format	Beschreibung
ERR? (S. 139)	ERR?	Get Error Number
FRF (S. 140)	FRF [{<AxisID>}]	Fast Reference Move To Reference Switch
FRF? (S. 141)	FRF? [{<AxisID>}]	Get Referencing Result
GWD? (S. 142)	GWD? [<StartPoint> <NumberOfPoints> [{<WaveTableID>}]]	Get Wave Table Data
HDI? (S. 143)	HDI?	Get Help For Interpretation Of DIA? Response
HDR? (S. 143)	HDR?	Get All Data Recorder Options
HLP? (S. 144)	HLP?	Get List of Available Commands
HLT (S. 145)	HLT [{<AxisID>}]	Halt Motion Smoothly
HPA? (S. 145)	HPA?	Get List Of Available Parameters
IFC? (S. 146)	IFC? [{<InterfacePam>}]	Get Current Interface Parameters
IFS (S. 147)	IFS <Pswd> {<InterfacePam> <PamValue>}	Set Interface Parameters as Default Values
IFS? (S. 148)	IFS? [{<InterfacePam>}]	Get Interface Parameters as Default Values
IMP (S. 149)	IMP <AxisID> <Amplitude>	Start Impulse and Response Measurement
JRC (S. 150)	JRC <Jump> <CMD?> <OP> <Value>	Jump Relatively Depending on Condition
KCP (S. 151)	KCP <CSNameSource> <CSNameCopy>	Copy Coordinate System
KEN (S. 152)	KEN <CSName>	Enable Coordinate System
KEN? (S. 154)	KEN? [{<CSName>}]	Get Enabled Coordinate Systems
KET? (S. 154)	KET? [{<CSType>}]	Get Enabled Coordinate System Types
KLC? (S. 155)	KLC? [<CSName1> [<CSName2>[<Item1>[<Item2>]]]]	Get Properties Of Work-And-Tool Combinations
KLD (S. 158)	KLD <CSName> [{<AxisID> <Offset>}]	Define Leveling Coordinate System By Specifying Values Vor dem Definieren eines Korrektur-Koordinatensystems ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).

Befehl	Format	Beschreibung
KLF (S. 160)	KLF <CSName>	Define Leveling Coordinate System At Current Position Vor dem Definieren eines Korrektur-Koordinatensystems ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).
KLN (S. 162)	KLN <CSName1> <CSName2>	Link Coordinate Systems
KLN? (S. 163)	KLN? [{<CSName>}]	Get Coordinate System Chains
KLS? (S. 164)	KLS? [<CSName>[<Item1>[<Item2>]]]	Get Coordinate System Properties
KLT? (S. 166)	KLT? [<StartCS> [<EndCS>]]	Get Offsets Resulting From A Chain
KRM (S. 167)	KRM <CSName>	Remove Coordinate System
KSB (S. 168)	KSB <CSName> [{<AxisID> <Angle>}]	Define Orientational Coordinate System Vor dem Definieren eines Achsorientierungs-Koordinatensystems ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).
KSD (S. 169)	KSD <CSName> [{<AxisID> <Offset>}]	Define Operating Coordinate System By Specifying Values
KSF (S. 171)	KSF <CSName>	Define Operating Coordinate System At Current Position
KST (S. 173)	KST <CSName> [{<AxisID> <Offset>}]	Define “Tool” Operating Coordinate System
KSW (S. 175)	KSW <CSName> [{<AxisID> <Offset>}]	Define “Work” Operating Coordinate System
LIM? (S. 179)	LIM? [{<AxisID>}]	Indicate Limit Switches
MAC (S. 179)	MAC <keyword> {<parameter>}	Call Macro Function
MAC? (S. 183)	MAC? [<macroname>]	List Macros
MAN? (S. 183)	MAN? {<CMD>}	Get Help String For Command
MEX (S. 184)	MEX <CMD?> <OP> <value>	Stop Macro Execution due to Condition
MOV (S. 184)	MOV {<AxisID> <Position>}	Set Target Position
MOV? (S. 186)	MOV? [{<AxisID>}]	Get Target Position
MRT (S. 177)	MRT {<AxisID> <Distance>}	Set Target Relative In Tool Coordinate System
MRW (S. 178)	MRW {<AxisID> <Distance>}	Set Target Relative In Work Coordinate System

Befehl	Format	Beschreibung
MVR (S. 186)	MVR {<AxisID> <Distance>}	Set Target Relative To Current Position
NLM (S. 188)	NLM {<AxisID> <LowLimit>}	Set Low Position Soft Limit
NLM? (S. 189)	NLM? [{<AxisID>}]	Get Low Position Soft Limit
ONT? (S. 189)	ONT? [{<AxisID>}]	Get On-Target State
PLM (S. 190)	PLM {<AxisID> <HighLimit>}	Set High Position Soft Limit
PLM? (S. 191)	PLM? [{<AxisID>}]	Get High Position Soft Limit
POS? (S. 191)	POS? [{<AxisID>}]	Get Real Position
PUN? (S. 192)	PUN? [{<AxisID>}]	Get Axis Unit
RBT (S. 192)	RBT	Reboot System
RMC? (S. 192)	RMC?	List Running Macros
RON? (S. 193)	RON? [{<AxisID>}]	Get Reference Mode
RTR? (S. 193)	RTR?	Get Record Table Rate
SAI? (S. 194)	SAI? [ALL]	Get List Of Current Axis Identifiers
SCT (S. 194)	SCT "T" <CycleTime>	Set Cycle Time
SCT? (S. 195)	SCT? [<T>]	Get Cycle Time
SPA (S. 196)	SPA {<ItemID> <PamID> <PamValue>}	Set Volatile Memory Parameters
SPA? (S. 197)	SPA? [{<ItemID> <PamID>}]	Get Volatile Memory Parameters
SPI (S. 197)	SPI {<PPCoordinate> <Position>}	Set Pivot Point
SPI? (S. 198)	SPI? [{<PPCoordinate>}]	Get Pivot Point
SSL (S. 199)	SSL {<AxisID> <SoftLimitsOn>}	Set Soft Limit
SSL? (S. 200)	SSL? [{<AxisID>}]	Get Soft Limit Status
SSN? (S. 200)	SSN?	Get Device Serial Number
STA? (S. 200)	STA?	Query Status Register Value
STE (S. 201)	STE <AxisID> <Amplitude>	Start Step And Response Measurement
STP (S. 202)	STP	Stop All Axes
SVO (S. 203)	SVO {<AxisID> <ServoState>}	Set Servo Mode
SVO? (S. 203)	SVO? [{<AxisID>}]	Get Servo Mode
TMN? (S. 204)	TMN? [{<AxisID>}]	Get Minimum Commandable Position
TMX? (S. 205)	TMX? [{<AxisID>}]	Get Maximum Commandable Position
TNR? (S. 206)	TNR?	Get Number Of Record Tables

Befehl	Format	Beschreibung
TRA? (S. 206)	TRA? {<AxisID> <Component>}	Get Maximum Commandable Position For Direction Vector
TRS? (S. 207)	TRS? [{<AxisID>}]	Indicate Reference Switch
TWG? (S. 208)	TWG?	Get Number of Wave Generators
VAR (S. 208)	VAR <Variable> <String>	Set Variable Value
VAR? (S. 209)	VAR? [{<Variable>}]	Get Variable Value
VEL (S. 209)	VEL {<AxisID> <Velocity>}	Set Closed-Loop Velocity
VEL? (S. 210)	VEL? [{<AxisID>}]	Get Closed-Loop Velocity
VER? (S. 210)	VER?	Get Version
VLS (S. 211)	VLS <SystemVelocity>	Set System Velocity
VLS? (S. 212)	VLS?	Get System Velocity
VMO? (S. 212)	VMO? {<AxisID> <Position>}	Virtual Move
WAC (S. 213)	WAC <CMD?> <OP> <value>	Wait for Condition
WAV (S. 214)	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>	Set Waveform Definition
WAV? (S. 219)	WAV? [{<WaveTableID> <WaveParameterID>}]	Get Waveform Definition
WCL (S. 219)	WCL {<WaveTableID>}	Clear Wave Table Data
WGC (S. 220)	WGC {<WaveGenID> <Cycles>}	Set Number Of Wave Generator Cycles
WGC? (S. 220)	WGC? [{<WaveGenID>}]	Get Number Of Wave Generator Cycles
WGO (S. 220)	WGO {<WaveGenID> <StartMode>}	Set Wave Generator Start/Stop Mode
WGO? (S. 222)	WGO? [{<WaveGenID>}]	Get Wave Generator Start/Stop Mode
WGR (S. 222)	WGR	Starts Recording In Sync With Wave Generator
WGS? (S. 223)	WGS? [<WaveGenID> [<ItemID>]]	Get Status Information Of Wave Generator
WMS? (S. 224)	WMS? [{<WaveTableID>}]	Get Maximum Number of Values for the Waveform
WPA (S. 224)	WPA <Pswd> [{<ItemID> <PamID>}]	Save Parameters To Non-Volatile Memory
WSL (S. 227)	WSL {<WaveGenID> <WaveTableID>}	Set Connection Of Wave Table To Wave Generator
WSL? (S. 227)	WSL? [{<WaveGenID>}]	Get Connection Of Wave Table To Wave Generator

Befehl	Format	Beschreibung
WTR (S. 228)	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Set Wave Generator Table Rate
WTR? (S. 230)	WTR? [{<WaveGenID>}]	Get Wave Generator Table Rate

8.4 Befehlsbeschreibungen für GCS 2.0

#3 (Get Real Position)

Beschreibung: Fragt die aktuelle Achsenposition ab.

Format: #3 (einzelnes ASCII Zeichen Nummer 3)

Argumente: Keine

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

wobei

<float> die aktuelle Achsenposition in physikalischen Einheiten ist.

Hinweise: Dieser Befehl ist funktionsgleich mit POS? (S. 191), aber es muss nur ein Zeichen über die Schnittstelle gesendet werden. Deshalb kann #3 auch verwendet werden, wenn der Controller zeitaufwändige Aufgaben ausführt.

Die aktuelle Position der Achsen X, Y, Z, U, V und W wird aus den gemessenen Positionen der einzelnen Antriebe berechnet.

Die physikalische Einheit, in der die Achsenposition angegeben wird, kann mit PUN? (S. 192) abgefragt werden.

#4 (Request Status Register)

Beschreibung: Fragt die Systemstatus-Information ab.

Format: #4

Argumente: Keine

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

Hinweise: Dieser Befehl ist funktionsgleich mit STA? (S. 200), aber es muss nur ein Zeichen über die Schnittstelle gesendet werden.

Die Antwort ist die Summe der untenstehenden Codes in Hexadezimalformat. Bei der Auswertung der Antwort ist Folgendes zu beachten:

- Die Bits 14 und 15 für den Bewegungsstatus der optionalen Einzelachsen S1 und S2 werden nur dann gesetzt, wenn die Bewegung durch einen Befehl ausgelöst wurde.
- Unbelegte Bits haben den Wert 0.

Bit:	23	22	21	20	19	18	17	16
	-	-	-	-	Referenz-fahrt wird ausgeführt	Referenz-fahrt Achse S2 erfolg-reich	Referenz-fahrt Achse S1 erfolg-reich	Referenz-fahrt Posi-tionierer erfolgreich
Bit:	15	14	13	12	11	10	9	8
	Achse S2 in Bewegung	Achse S1 in Bewegung	Antrieb 6 in Bewegung	Antrieb 5 in Bewegung	Antrieb 4 in Bewegung	Antrieb 3 in Bewegung	Antrieb 2 in Bewegung	Antrieb 1 in Bewegung
Bit:	7	6	5	4	3	2	1	0
	Bewegungs-fehler Achse S2	Bewegungs-fehler Achse S1	Bewe-gungsfehler Antrieb 6	Bewe-gungsfehler Antrieb 5	Bewe-gungsfehler Antrieb 4	Bewe-gungsfehler Antrieb 3	Bewe-gungsfehler Antrieb 2	Bewe-gungsfehler Antrieb 1

Beispiel: Senden: #4
Empfangen: 0x71804
Hinweis: Die Antwort wird im Hexadezimalformat angegeben. Sie besagt: Für Antrieb 3 wurde ein Bewegungsfehler gemeldet, die Antriebe 4 und 5 sind in Bewegung. Die Referenzfahrt des Positionierers und der optionalen Einzelachsen S1 und S2 wurde erfolgreich abgeschlossen.

#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
...

Beispiele: 0 gibt an, dass die Bewegung aller Achsen abgeschlossen ist
3 gibt an, dass sich die erste und die zweite Achse bewegen

49 gibt an, dass sich die Achsen X, U und S1 bewegen.

Hinweis: Die Achsen 1 bis 8 entsprechen den Achsen X, Y, Z, U, V, W, S1 und S2 in dieser Reihenfolge.

#6 (Query for Position Change)

Beschreibung: Fragt, ob sich die Achsenpositionen seit der zuletzt gesendeten Positionsabfrage geändert haben.

Format: #6 (einzelnes ASCII Zeichen Nummer 6)

Argumente: Keine

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

1 = Position der ersten Achse hat sich geändert
2 = Position der zweiten Achse hat sich geändert
4 = Position der dritten Achse hat sich geändert
...

Beispiele: 0 gibt an, dass sich keine Achsenposition geändert hat.
3 gibt an, dass sich die Positionen der ersten und der zweiten Achse geändert haben.

49 gibt an, dass sich die Positionen der Achsen X, U und S1 geändert haben.

Hinweise: Als Positionsänderung wird gewertet, wenn seit dem zuletzt gesendeten POS? (S. 191) oder #3 (S. 123) eine neue Zielposition per Befehl - auch innerhalb eines Makros - vorgegeben wurde.

#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. 124) 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 Referenzfahrt 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

#9 (Get Wave Generator Status)

Beschreibung:	Fragt den Status der Funktionsgeneratoren ab.
Format:	#9
Argumente:	Keine
Antwort:	Die Antwort <uint> ist bit-codiert und wird als hexadezimale Summe der folgenden Codes ausgegeben: 1 = Funktionsgenerator 1 ist aktiv, 2 = Funktionsgenerator 2 ist aktiv, 4 = Funktionsgenerator 3 ist aktiv usw. "Aktiv" = Funktionsgeneratorausgabe läuft
Beispiele:	0 gibt an, dass kein Funktionsgenerator aktiv ist 5 gibt an, dass die Funktionsgeneratoren 1 und 3 aktiv sind

#11 (Get Memory Space for Trajectory Points)

Beschreibung:	Fragt den freien Speicherplatz für die Punkte des Dynamikprofils ab.
Format:	#11 (einzelnes ASCII Zeichen Nummer 11)
Argumente:	Keine

Antwort: <uint> ist der freie Speicherplatz, Angabe als Anzahl der Dynamikprofilpunkte.

Hinweise: #11 fragt den freien Speicherplatz eines Zwischenspeichers ab, der Dynamikprofilpunkte für den Positionierer enthält. Ein Dynamikprofilpunkt entspricht einem Satz von Zielpositionen für die Achsen des Positionierers (X, Y, Z, U, V, W). Der Inhalt des Zwischenspeichers wird nur dann zur Festlegung des Dynamikprofils verwendet, wenn die Parameter 0x19001900 und 0x19001901 jeweils den Wert 1 haben.

Weitere Informationen siehe "Zyklische Übertragung von Zielpositionen" (S. 33).

#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. 202), aber es wird nur ein Zeichen über die Schnittstelle gesendet.

Format: #24

Argumente: Keine

Antwort: Keine

Hinweise: #24 stoppt alle Achsenbewegungen, die durch Bewegungsbefehle oder Funktionsgeneratorausgabe verursacht werden, und die Referenzfahrt.

#24 stoppt Makros.

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

#27 (System Abort)

Beschreibung: Hält den Controller an.

Format: #27 (einzelnes ASCII Zeichen Nummer 27)

Argumente: Keine

Antwort: Keine

Hinweise: #27 löst Folgendes aus:

- Antriebe werden abgeschaltet
- Servomodus wird ausgeschaltet
- Befehle werden nicht mehr abgearbeitet
- Einstellungen im flüchtigen Speicher werden auf Standardwerte gesetzt

Für einen Neustart muss der C-886 aus- und wieder eingeschaltet werden.

***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

Hinweis: Für C-886 antwortet *IDN? etwa Folgendes:

```
(c)2011-2017 Physik Instrumente (PI) GmbH & Co. KG,C-886,117567891,2.3.3.30
```

ADD (Add And Save To Variable)

Beschreibung: Addiert zwei Werte und speichert das Ergebnis als Variable (S. 112).

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

Hinweis: ADD kann nur in Makros verwendet werden.

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

Senden: `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
```

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 Werkseinstellung, 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 PI-Servicepersonal 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. 267).

Antwort: Keine

Fehlersuche: Ungültiges Kennwort

Hinweise: HLP? (S. 144) führt alle in der aktuellen Befehlsebene verfügbaren Befehle auf.

HPA? (S. 145) listet die Parameter einschließlich der Information darüber, welche Befehlsebene Schreibzugriff auf sie erlaubt. Weitere Informationen zur Parameterverwendung siehe "Anpassen von Einstellungen" (S. 247).

Nach dem Anschalten 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, alle Befehle, die dem „normalen“ Benutzer zur Verfügung gestellt werden, sowie der Lesezugriff auf alle Parameter sind zugänglich

<Level> = 1 lässt zusätzliche Befehle und Schreibzugriff für Parameter der Ebene 1 zu (Befehle und Parameter von Ebene 0 sind inbegriffen)

CPY (Copy Into Variable)

Beschreibung: Kopiert eine Antwort auf einen Befehl in eine Variable (S. 112).

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

Hinweis: CPY kann nur in Makros verwendet werden.

Beispiel: Es ist möglich, den Wert einer Variable (z. B. SOURCE) in eine andere Variable (z. B. TARGET) zu kopieren:

Senden: `CPY TARGET VAR? SOURCE`

CST? (Get Assignment Of Stages To Axes)

Beschreibung: Fragt den Namen des Verstellertyps 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 Verstellertyps ist, der der Achse zugewiesen ist.

Hinweise: Den Achsen des Positionierers wird beim Einschalten oder Neustart des Controllers automatisch der passende Verstellertyp zugewiesen.

Für die optionalen Einzelachsen S1 und S2 kann die Zuweisung geändert werden, siehe "Slave-Module für Einzelachsen konfigurieren" (S. 66).

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

Hinweis: Als Antwort ist nur 2.0 (für GCS 2.0) möglich.

DEL (Delay the Command Interpreter)

Beschreibung: Verzögert um <uint> Millisekunden.

Format: DEL <uint>

Argumente: <uint> ist der Verzögerungswert in Millisekunden.

Antwort: Keine

Hinweis: DEL sollte nur in Makros verwendet werden.

DIA? (Get Diagnosis Information)

Beschreibung: Fragt den aktuellen Wert der angegebenen Messgröße ab.

Werden alle Argumente weggelassen, werden die aktuellen Werte aller Messgrößen abgefragt.

Format: DIA? [{<MeasureID>}]

Argumente: <MeasureID> ist die Kennung einer Messgröße; nähere Angaben siehe unten.

Antwort: {<MeasureID>"="<MeasuredValue> LF}

wobei

<MeasuredValue> den aktuellen Wert der Messgröße angibt; nähere Angaben siehe unten.

Hinweise: Verwenden Sie die Antwort auf den Befehl HDI? (S. 143), um Beschreibungen und physikalische Einheiten der unterstützten Messgrößen zu erhalten.

C-886 unterstützt die folgenden Messgrößen:

<MeasureID>	<Description> (mit HDI? abfragen)
1	Hexapod Powered: ohne Bedeutung für den C-886
2	Controller E-Stop Activated: ohne Bedeutung für den C-886
3	Temperature of Controller: ohne Bedeutung für den C-886
4	Index of Faulty Point in Waveform: Index des Kurvenformpunktes, an dem der letzte Fehler während der Funktionsgeneratorausgabe aufgetreten ist

DPA (Reset Settings to Default)

Beschreibung: Setzt Parameterwerte und parameterunabhängige Einstellungen auf Werkseinstellungen zurück.

Format: DPA <Pswd> [{<ItemID> <PamID>}]

Argumente: <Pswd> ist das Kennwort zum Zurücksetzen des Speichers. Nähere Angaben siehe unten.

<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, ungültiges Kennwort

Hinweise: DPA setzt im flüchtigen Speicher des C-886 die mit WPA gespeicherten Parameterwerte und die Einstellungen für Koordinatensysteme auf Werkseinstellungen zurück.

Beim C-886 entfällt für DPA die Angabe von <ItemID> und <PamID>.

Die Werkseinstellungen, die mit DPA geladen werden, sind unabhängig von den Einstellungen im permanenten Speicher, die mit WPA (S. 224) überschrieben werden können. Die (mit WPA gespeicherten) Einstellungen aus dem permanenten Speicher werden beim Einschalten oder Neustart des C-886 automatisch in den flüchtigen Speicher geladen.

Gültige Kennwörter:

100 Setzt die Werte aller Parameter und die Einstellungen für Koordinatensysteme (Details siehe Kennwort SKS) auf Werkseinstellungen zurück

SKS Einstellungen, die mit DPA SKS auf Werkseinstellungen zurücksetzt werden:

- Achsorientierungs-Koordinatensystem (Typ KSB()): PI_BASE wird aktiviert
- Korrektur-Koordinatensystem (Typ KLD() oder KLF()): PI_Levelling wird aktiviert
- Betriebs-Koordinatensystem ZERO wird aktiviert und basiert auf PI_BASE und PI_Levelling
- Pivotpunkt (siehe SPI (S. 197)), Verfahrbereichsgrenzen der Achsen (siehe NLM (S. 188), PLM (S. 190) und SSL (S. 199))

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. 143) 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.

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 für <NumberOfPoints> der Wert -1 gesetzt wird, werden alle gültigen Punkte der ausgewählten Tabellen ausgelesen

Wenn <RecTableID> weggelassen wird, werden alle Tabellen ausgelesen.

Mit HDR? (S. 143) erhalten Sie eine Liste aller verfügbaren Aufzeichnungs- und Triggeroptionen sowie Informationen über zusätzliche Parameter und Befehle für die Datenaufzeichnung.

Nähere Informationen siehe "Datenrekorder" (S. 76).

Beispiel:

```
drt 1 1 0
mov x 2
drr? 1 5 1 2
# REM data recorded with C-886 controller
#
# TYPE = 1
# SEPARATOR = 32
# DIM = 2
# NDATA = 5
# SAMPLE_TIME = 0.009009
#
# NAME0 = TARGET POSITION X
# NAME1 = REAL POSITION X
# DISP_UNIT0 = mm
# DISP_UNIT1 = mm
# RATIO_NOM0 =1
# RATIO_DENOM0 = 1
# RATIO_NOM1 =1
# RATIO_DENOM1 = 1
# END_HEADER
0.004220971838 -0.000364157866
0.064530499279 0.036015000194
0.145174950361 0.140960231423
0.232607111335 0.216114446521
```

```
0.309781700373 0.30354321003
```

Mit DRT (S. 137) wird festgelegt, dass eine Aufnahme durch den nächsten Bewegungsbefehl, z.B. durch MOV (S. 184), ausgelöst werden soll.

MOV löst die Bewegung der Achse X zur Position 2 aus.

Mit DRR? werden die ersten fünf Punkte der Datenrekordertabellen 1 und 2 abgefragt.

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:	Unabhängig von der mit <RecTableID> gewählten Datenrekordertabelle wird die mit <Triggersource> gewählte Triggeroption immer für alle Datenrekordertabellen eingestellt. Mit HDR? (S. 143) erhalten Sie eine Liste aller verfügbaren Aufzeichnungs- und Triggeroptionen sowie zusätzliche Informationen über die Datenaufzeichnung. IMP (S. 149), STE (S. 201), WGO (S. 220) und WGR (S. 222) lösen unabhängig von den DRT-Einstellungen jeweils eine Aufzeichnung durch den Datenrekorder aus (WGR nur, wenn der Funktionsgenerator aktiv ist). Die Aufzeichnung endet, wenn die maximale Punktzahl erreicht ist (vorgegeben durch den Parameter ***Data Recorder Points Per Table***, ID 0x16000201). Nähere Informationen finden Sie unter "Datenrekorder" (S. 76).

Verfügbare Triggeroptionen:

0 = No trigger. Exception: STE, IMP, WGO and WGR always trigger data recording;
Datenaufzeichnung wird nicht ausgelöst, und eine noch laufende Datenaufzeichnung wird fortgeführt, bis die maximale Punktanzahl erreicht ist. Ausnahme: STE, IMP, WGO und WGR lösen immer eine Aufnahme aus. <Value> muss ein Dummy sein.

1 = Trigger with next command that changes the position;
z.B. MOV (S. 184), MVR (S. 186); <Value> muss ein Dummy sein. Standardmäßig eingestellt.

2 = Trigger with next command;
setzt Trigger nach Ausführung auf die Triggeroption 0; <Value> muss ein Dummy sein.

4 = Trigger immediately;
setzt Trigger nach Ausführung auf die Triggeroption 0; <Value> muss ein Dummy sein

6 = Trigger with next command that changes the position; z.B. MOV, MVR;
setzt Trigger nach Ausführung auf die Triggeroption 0; <Value> muss ein Dummy sein.

Ist die Triggeroption 2, 4 oder 6 eingestellt, wird nach dem Auslösen der Aufzeichnung automatisch auf die Triggeroption 0 umgeschaltet. Dadurch läuft die Aufzeichnung so lange weiter, bis die maximale Punktanzahl der Datenrekordertabelle(n) erreicht ist. So können mehrere Bewegungen nacheinander aufgezeichnet werden. Die Triggeroptionen 2, 4 oder 6 sind z.B. dann sinnvoll, wenn das Dynamikprofil durch aufeinander folgende MOV-Befehle festgelegt werden soll (siehe MOV).

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. 137).

ECO? (Echo a String)

Beschreibung: Gibt einen String aus.

ECO? kann verwendet werden, um die Kommunikation zu testen.

Format: ECO? <String>

Argumente: <String> ist eine beliebige Zeichenkombination aus Buchstaben und Zahlen

Antwort: <String> LF

Hinweis: <String> kann aus maximal 100 Zeichen bestehen.

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. 230) 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.

FRF (Fast Reference Move To Reference Switch)

Beschreibung: Startet eine Referenzfahrt.

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 synchron bewegt.

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: Für Achsen mit inkrementellen Sensoren können Bewegungen erst nach einer erfolgreichen Referenzfahrt kommandiert werden (auch als „Initialisierung" bezeichnet).

Das Verhalten des Positionierer nach der Referenzfahrt wird durch die Parameter ***Behaviour After Reference Move*** (ID 0x07030401) und ***Target For Motion After Reference Move*** (ID 0x07030402) festgelegt. Entsprechend der Parameterwerte kann die Plattform z. B. nach der Referenzfahrt automatisch zu einer vorgegebenen Position bewegt werden.

- Wert des Parameters 0x07030401 = 0: Bewegungsplattform bleibt nach der Referenzfahrt in der Referenzposition.
- Wert des Parameters 0x07030401 = 1: Bewegungsplattform fährt nach der Referenzfahrt zur Zielposition, die durch Parameter 0x07030402 vorgegeben ist.

Für Achsen mit absolut messenden Sensoren ist keine Referenzfahrt erforderlich. Die Verwendung des Befehls FRF wird für diese Achsen

trotzdem empfohlen. FRF startet für Achsen mit absolut messenden Sensoren keine Referenzfahrt, sondern setzt die Zielpositionen auf die aktuellen Positionswerte. Zusätzlich werden die oben beschriebenen Parameterwerte wirksam, so dass die Achsen z. B. zu einer definierten "Ausgangsposition" bewegt werden können.

Während einer Referenzfahrt bewegt sich der Positionierer auf unvorhersehbare Weise.

Für die Achsen der Bewegungsplattform des Positionierers (X, Y, Z, U, V, W) erfolgt immer eine gemeinsame Referenzfahrt. Daher reicht für den Start der Referenzfahrt der Bewegungsplattform die Angabe einer einzigen Achse aus, z.B.:
FRF X

Für die Achsen der Bewegungsplattform des Positionierers (X, Y, Z, U, V, W) ist nach erfolgreicher Verwendung von FRF der Servomodus eingeschaltet.

FRF kann durch #24 (S. 127), STP (S. 202) und HLT (S. 145) abgebrochen werden.

Verwenden Sie FRF? (S. 141), um zu prüfen, ob die Referenzfahrt erfolgreich war.

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
Hinweis:	Achsen gelten als "referenziert", wenn eine Referenzfahrt erfolgreich durchgeführt wurde mit FRF (S. 140), oder wenn die Achsen mit absolut messenden Sensoren ausgestattet sind.

GWD? (Get Wave Table Data)

Beschreibung: Fragt die Kurvenform für die angegebene Kurventabelle ab.

Format: GWD? [<StartPoint> <NumberOfPoints> [{<WaveTableID>}]]

Argumente: <StartPoint> ist der Startpunkt in der Kurventabelle, beginnt mit Index 1

<NumberOfPoints> bezeichnet die Anzahl der je Tabelle zu lesenden Punkte

<WaveTableID> bezeichnet eine Kurventabelle des Controllers

Antwort: Der Inhalt der Kurventabelle (Kurvenform) im GCS-Array-Format (siehe separates Handbuch für GCS Array, SM 146E und untenstehendes Beispiel.)

Hinweise: Abhängig von der Kurvenformdefinition mit WAV (S. 214) können die Kurventabellen unterschiedliche Längen haben.

- Fragen Sie nur Kurventabellen gleicher Länge im selben Befehl ab.
- Fragen Sie den Inhalt von Kurventabellen mit unterschiedlicher Länge mit separaten Befehlen ab.

Wenn Kurventabellen unterschiedlicher Länge gemeinsam abgefragt werden, enthält die Antwort maximal so viele Punkte wie die kürzeste Kurventabelle, oder die Abfrage erzeugt einen Fehler.

Beispiel:

```
gwd? 1 10 1 2
# TYPE = 1
#
# SEPARATOR = 32
# DIM = 2
# SAMPLE_TIME = 0.009000
# NDATA = 10
# NAME0 = Wave table 1
# NAME1 = Wave table 2
# END_HEADER
-9.7773e-08 -9.7773e-08
9.9005486e-05 9.9005486e-05
0.000396286628 0.000396286628
0.000891734005 0.000891734005
0.001585328104 0.001585328104
0.002477041554 0.002477041554
0.00356683912 0.00356683912
0.004854677712 0.004854677712
0.006340506381 0.006340506381
0.008024266324 0.008024266324
```

HDI? (Get Help For Interpretation Of DIA? Response)

Beschreibung:	Zeigt Beschreibungen und physikalische Einheiten für die Messgrößen an, die mit dem Befehl DIA? (S. 132) abgefragt werden können.
Format:	HDI?
Argumente:	Keine
Antwort	{<MeasureID>"="<Description>TAB<PhysUnit> LF} wobei <MeasureID> die Kennung der Messgröße ist <Description> der Name der Messgröße ist <PhysUnit> die physikalische Einheit der Messgröße ist.

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

Beispiel:

```
hdr?
#RecordOptions
0=Nothing is recorded
1=Commanded position of axis
2=Real position of axis
80=Status register of axis
#TriggerOptions
0=No trigger. Exception: STE and IMP always trigger data recorder
1=Trigger with next command that changes the position, default setting
2=Trigger with next command, resets trigger settings to 0
4=Trigger immediately, resets trigger settings to 0
6=Trigger with next command that changes the position, resets trigger settings to 0
#Additional information
Get data recorder configuration with DRC?
Get number of recorded points with DRL?
Get recorded data values with DRR?
Set data recorder trigger source with DRT
Get data recorder trigger source with DRT?
Tell number of data recorder tables with TNR?
#Sources for Record Options
0 = X Y Z U V W 1 2 3 4 5 6 S1 S2
1 = X Y Z U V W 1 2 3 4 5 6 S1 S2
2 = X Y Z U V W 1 2 3 4 5 6 S1 S2
80 = 1 2 3 4 5 6 S1 S2
end of help
```

Die Datenquellen 1 bis 6 sind die Antriebe des Positionierers.

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

Hinweis: Die Antwort auf HLP? enthält die in der aktuellen Befehlsebene zugelassenen Befehle. Weitere Informationen siehe CCL (S. 129).

HLT (Halt Motion Smoothly)

Beschreibung: Stoppt die Bewegung der angegebenen Achsen sanft. Nähere Angaben siehe Hinweise unten.

Fehlercode 10 wird gesetzt.

#24 (S. 127) und STP (S. 202) 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 alle Achsenbewegungen, die durch Bewegungsbefehle oder Funktionsgeneratorausgabe verursacht werden, und die Referenzfahrt.

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

HLT stoppt Makros nicht.

HPA? (Get List Of Available Parameters)

Beschreibung: Antwortet mit einem Hilfe-String, der alle verfügbaren Parameter mit Kurzbeschreibungen enthält. Weitere Informationen siehe "Parameterübersicht" (S. 250).

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.

Hinweise: Der String hat folgendes Format:

<CmdLevel>TAB<MaxItem>TAB<DataType>TAB<FunctionGroupDescription>TAB<ParameterDescription>[{TAB<PossibleValue>"="<ValueDescription>}]

wobei

<CmdLevel> ist die Befehlsebene, die Schreibzugriff auf den Parameterwert erlaubt

<MaxItem> ist die maximale Anzahl der Elemente desselben Typs, auf die sich der Parameter auswirkt. (Die Bedeutung von „Element" hängt vom Parameter ab; es kann sich um eine Achse, einen Antrieb oder das gesamte System handeln.)

<DataType> ist der Datentyp des Parameterwertes, er kann INT, FLOAT oder CHAR sein

<FunctionGroupDescription> ist der Name der Funktionsgruppe, der der Parameter angehört. (Parameter werden entsprechend ihrem Zweck zu Gruppen zusammengefasst, um ihre Beziehung zueinander zu verdeutlichen.)

<ParameterDescription> ist der Name des Parameters

<PpossibleValue> ist ein Wert aus dem zulässigen Datenbereich

<ValueDescription> ist die Bedeutung des entsprechenden Wertes

Die Auflistung der Parameter variiert in Abhängigkeit vom installierten optionalen Zubehör.

Die Werte der aufgelisteten Parameter können mit SPA (S. 196) geändert werden. SPA beeinflusst die Parametereinstellungen im flüchtigen Speicher (RAM).

IFC? (Get Current Interface Parameters)

Beschreibung: Fragt die Werte der Schnittstellenparameter für die Kommunikation aus dem flüchtigen Speicher ab.

Format: IFC? [{<InterfacePam>}]

Argumente: <InterfacePam> ist der abzufragende Schnittstellenparameter, mögliche Werte siehe unten.

Antwort: {<InterfacePam>"="<PamValue> LF}

wobei

<PamValue> den Wert des Schnittstellenparameters aus dem flüchtigen Speicher angibt.

<InterfacePam> kann IPADR, IPSTART, IPMASK, IPMAXCONN, MACADR oder TERMSTR sein.

Für <InterfacePam> = IPSTART gibt <PamValue> die aktuelle Einstellung des Startup-Verhaltens für die Konfiguration der IP-Adresse für die TCP/IP-Kommunikation an,
0 = Die mit IPADR definierte IP-Adresse wird verwendet.
1 = DHCP wird verwendet, um die IP-Adresse zu erhalten;

Für <InterfacePam> = IPADR geben die ersten vier Teile von <PamValue> die IP-Adresse an, die aktuell für die TCP/IP-Kommunikation verwendet wird, der letzte Teil gibt den Port an;

Für <InterfacePam> = IPMASK gibt <PamValue> die IP-Masken-Einstellung, die aktuell für die TCP/IP-Kommunikation verwendet wird, in der Form uint.uint.uint.uint an;

Für <InterfacePam> = IPMAXCONN gibt <PamValue> die maximal zulässige Anzahl von Verbindungen für die TCP/IP-Kommunikation an;

Für <InterfacePam> = MACADR gibt <PamValue> die unveränderliche, eindeutige Adresse der Netzwerkhardware im C-886 an;

Für <InterfacePam> = TERMSTR gibt <PamValue> das Abschlusszeichen für die Befehle des GCS aus:
0 = LineFeed (ASCII-Zeichen 10)

IFS (Set Interface Parameters as Default Values)

Beschreibung: Speichert Schnittstellenparameter.

Ändert die Standardparameter für die Schnittstelle im permanenten Speicher, aber nicht die aktuell aktiven Parameter. Die mit IFS vorgenommenen Einstellungen werden beim nächsten Einschalten oder Neustart aktiviert.

Format: IFS <Pswd> {<InterfacePam> <PamValue>}

Argumente: <Pswd> ist das Passwort zum Schreiben in den permanenten Speicher, Standardwert ist "100"

<InterfacePam> ist der zu ändernde Schnittstellenparameter, siehe unten

<PamValue> gibt den Wert des Schnittstellenparameters an, siehe unten

Antwort: Keine

Folgende Schnittstellenparameter können gesetzt werden:

IPADR
Die ersten vier Teile von <PamValue> geben die Standard-IP-Adresse für die TCP/IP-Kommunikation an, der letzte Teil gibt den zu verwendenden Standardport an, Standard ist 192.168.1.28:50000; Hinweis: Während die IP-Adresse geändert werden kann, muss der Port stets 50000 sein!

IPSTART
<PamValue> definiert das Startup-Verhalten zur Konfiguration der IP-Adresse für die TCP/IP-Kommunikation,
0 = Die mit IPADR definierte IP-Adresse wird verwendet
1 = DHCP wird verwendet, um die IP-Adresse zu erhalten. (Standard);

IPMASK
<PamValue> gibt die für die TCP/IP-Kommunikation zu verwendende Subnetzmaske in der Form uint.uint.uint.uint an, Standard ist 255.255.255.0

Antwort: Keine

Hinweise: **Beachten Sie, dass die Anzahl von Schreibzyklen im permanenten Speicher begrenzt ist. Schreiben Sie Standardeinstellungen nur, wenn dies notwendig ist.**

Weitere Schnittstellenparameter des C-886 sind schreibgeschützt. Die Voreinstellungen dieser Parameter können mit IFS? (S. 148) abgefragt werden.

Weitere Informationen finden Sie in "Kommunikation über TCP/IP-Schnittstelle herstellen" (S. 57).

IFS? (Get Interface Parameters as Default Values)

Beschreibung: Fragt die im permanenten Speicher gespeicherten Parameterwerte der Schnittstellenkonfiguration ab (d. h. Standardeinstellungen)

Format: IFS? [{<InterfacePam>}]

Argumente: <InterfacePam> ist der abzufragende Schnittstellenparameter. Mögliche Werte siehe unten.

Antwort: {<InterfacePam>"="<PamValue> LF}

wobei

<PamValue> der Wert des Schnittstellenparameters im permanenten Speicher ist.

<InterfacePam> kann IPADR, IPSTART, IPMASK, IPMAXCONN, MACADR und TERMSTR sein

Die folgenden Schnittstellenparameter sind schreibgeschützt:

Für <InterfacePam> = IPMAXCONN gibt <PamValue> die maximal zulässige Anzahl von Verbindungen für die TCP/IP-Kommunikation an;

Für <InterfacePam> = MACADR gibt <PamValue> die eindeutige Adresse der Netzwerkhardware im C-886 aus;

Für <InterfacePam> = TERMSTR gibt <PamValue> das Abschlusszeichen für die Befehle des GCS aus:
0 = LineFeed (ASCII-Zeichen 10)

Für alle weiteren Ausprägungen von <InterfacePam> siehe IFS (S. 147).

IMP (Start Impulse and Response Measurement)

Beschreibung: Startet einen Impuls und zeichnet die Impulsantwort für die angegebene Achse auf.

Die Datenrekorderkonfiguration, d.h. die Zuweisung der Datenquellen und der Aufzeichnungsoptionen zu den Rekordertabellen, kann mit DRC gesetzt werden.

Die aufgezeichneten Daten können mit dem Befehl DRR? (S. 135) gelesen werden.

Format: IMP <AxisID> <Amplitude>

Argumente: <AxisID> ist eine Achse des Controllers

<Amplitude> ist die Höhe des Impulses. Nähere Angaben siehe unten.

Antwort: Keine

Fehlersuche: Die Zielposition, die aus der angegebenen Impulshöhe resultiert, ist außerhalb der Grenzwerte.

Hinweise: Ein „Impuls" besteht aus einer relativen Bewegung mit der angegebenen Amplitude, gefolgt von einer gleich großen Bewegung in die entgegengesetzte Richtung. Der Impuls wird relativ zu der aktuellen Position ausgeführt.

Die Pulsweite des Impulses resultiert aus dem Wert des Parameters ***Pulse Length Factor*** (ID 0x0E000900) multipliziert mit der C-886-Zykluszeit.

Die physikalische Einheit, in der <Amplitude> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden.

Für die Achsen des Positionierers (X, Y, Z, U, V, W) gilt Folgendes:

- Vor dem Start jeder Bewegung erfolgt eine Prüfung, ob die Bewegungsplattform die kommandierte Zielposition tatsächlich erreichen kann. Mit VMO? (S. 212) können Sie abfragen, ob die Zielposition erreicht werden kann.
- Rotationen erfolgen um den Drehpunkt.
- Wenn der Parameter ***Trajectory Source*** (ID 0x19001900) auf 1 gesetzt ist, muss das Dynamikprofil durch aufeinander folgende MOV-Befehle vorgegeben werden. IMP ist nicht zulässig.

Weitere Informationen siehe "Bewegungen des Positionierers" (S. 29).

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 Makroausführungs-Zeiger zurück zur vorherigen Zeile springt, 0 bedeutet, dass der Befehl erneut ausgeführt wird, was dem Verhalten von WAC (S. 213) 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

KCP (Copy Coordinate System)

Beschreibung: Erzeugt eine Kopie eines Koordinatensystems.

Format: KCP <CSNameSource> <CSNameCopy>

Argumente: <CSNameSource> ist der Name des Koordinatensystems, von dem eine Kopie erzeugt werden soll. PI_Base, PI_Levelling, ZERO und HEXAPOD können nicht kopiert werden.

<CSNameCopy> ist der Name der Kopie des Koordinatensystems.

Antwort: Keine

Hinweise: PI_Base, PI_Levelling, ZERO und HEXAPOD können nicht kopiert werden.

Möglichkeiten zum Anlegen der Kopie:

- <CSNameCopy> ist ein neuer Name. Die Kopie wird unter diesem Namen als neues Koordinatensystem angelegt.
- <CSNameCopy> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird dadurch überschrieben.

Die Verkettung mit dem Vorgänger in einer Kette von

Koordinatensystemen wird kopiert. Die Verkettung mit Nachfolgern wird nicht kopiert.

Die Kopie wird im flüchtigen Speicher erzeugt. Mit WPA (S. 224) kann die Kopie in den permanenten Speicher geschrieben werden.

KEN (Enable Coordinate System)

Beschreibung: Aktiviert das angegebene Koordinatensystem. Der Umfang der Einstellungen, die durch das Aktivieren beeinflusst werden, hängt vom Typ des Koordinatensystems ab, siehe unten.

Positionsangaben für die Bewegungsplattform des Positionierers (Abfrage mit POS? (S. 191)) beziehen sich auf das aktive Betriebs-Koordinatensystem.

Wenn mit dem Work-und-Tool-Konzept gearbeitet wird:

- Das Work-und-Tool-Konzept verwendet eine Kombination aus zwei aktiven Betriebs-Koordinatensystemen. In der Regel besteht die Kombination aus jeweils einem aktiven Koordinatensystem der Typen KST und KSW. Wenn nur für einen der beiden Typen ein Koordinatensystem aktiv ist, wird für den anderen Typ automatisch ein Ersatz verwendet.
- Die mit POS? abgefragte aktuelle Position der bewegten Plattform des Positionierers ist als Position des Tool-Koordinatensystems im Work-Koordinatensystem zu verstehen.

Das Aktivieren von Koordinatensystemen mit KEN löst keine Bewegung aus.

Format: KEN <CSName>

Argumente: <CSName> ist der Name des Koordinatensystems, das aktiviert werden soll.

Antwort: Keine

Hinweise: Vor dem Aktivieren eines Koordinatensystems prüft KEN die korrekte Definition dieses Koordinatensystems inklusive seiner korrekten Verkettung. Wenn das Koordinatensystem nicht korrekt definiert ist, wird es nicht aktiviert.

Koordinatensysteme der folgenden Typen können mit KEN aktiviert werden: KSD, KSF, KSW, KST, KSB(USER), KSB(PI), KLF(USER), KLF(PI), KLD(USER), KLD(PI), ZERO

Aus den folgenden Gruppen von Koordinatensystemen ist jeweils genau ein Koordinatensystem bzw. genau eine Koordinatensystem-Kombination aktiv:

- Betriebs-Koordinatensystem: ein Koordinatensystem vom Typ ZERO oder KSF oder KSD oder - für das Work-und-Tool-Konzept - eine Kombination aus Koordinatensystemen der Typen KSW/KST oder ZERO/KST oder KSW/ZERO
- Achsorientierungs-Koordinatensystem (Typ KSB(PI) oder KSB(USER))
- Korrektur-Koordinatensystem (Typ KLD(USER) oder KLD(PI) oder KLF(USER) oder KLF(PI))

Das Aktivieren eines Koordinatensystems für eine der Gruppen deaktiviert gleichzeitig das Koordinatensystem bzw. die Koordinatensystem-Kombination, welche/s zuvor für diese Gruppe aktiv war.

Durch Senden von KEN ZERO wird das werkseitig aktive Betriebs-Koordinatensystem ZERO wieder aktiviert. Wenn die Befehlsebene 1 aktiv ist (siehe CCL (S. 129)), aktiviert KEN ZERO auch wieder das werkseitig aktive Korrektur-Koordinatensystem PI_Levelling, jedoch nicht das werkseitig aktive Achsorientierungs-Koordinatensystems PI_Base (dieses kann durch Senden von KEN PI_Base wieder aktiviert werden). Durch Senden von DPA SKS (S. 133) können alle werkseitig aktiven Koordinatensysteme unabhängig von der aktuell aktiven Befehlsebene wieder aktiviert werden.

Durch das aktive Betriebs-Koordinatensystem werden Werte für die folgenden Einstellungen vorgegeben (beim Work-und-Tool-Konzept durch die Kombination aus zwei Betriebs-Koordinatensystemen):

- NLM (S. 188): Untergrenze des Verfahrbereichs der Achse
- PLM (S. 190): Obergrenze des Verfahrbereichs der Achse
- SSL (S. 199): Aktivierungszustand der Verfahrbereichsgrenzen der Achse
- SPI (S. 197): Koordinaten des Pivotpunkts (nur für Koordinatensysteme der Typen KSF und ZERO)
- Wenn vom Controller unterstützt:
  SST: Schrittweite für Bewegungen, die durch eine manuelle Bedieneinheit ausgelöst werden

Vor dem Aktivieren von Korrektur- und Achsorientierungs-Koordinatensystemen (Typen KLD(), KLF() und KSB()) ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL).

Die Koordinatensysteme werden im flüchtigen Speicher aktiviert und deaktiviert. Mit WPA (S. 224) kann der Aktivierungszustand in den permanenten Speicher geschrieben werden.

KEN? (Get Enabled Coordinate System)

Beschreibung: Listet die Namen der aktiven Koordinatensysteme auf und zeigt deren Typ an.

Format: KEN? [{<CSName>}]

Argumente: <CSName> ist der Name eines aktiven Koordinatensystems. Nicht zulässig: ZERO.

Wenn <CSName> weggelassen wird, werden alle aktiven Koordinatensysteme aufgelistet.

Antwort: {<CSName>"="<CSType>}

wobei

<CSType> den Typ des Koordinatensystems angibt.

Hinweise: KEN? fragt den flüchtigen Speicher ab.

Wenn das Betriebs-Koordinatensystem ZERO aktiv ist, wird es in der Antwort auf KEN? **nicht** angezeigt, und die Antwort enthält nur Folgendes:

- Das aktive Korrektur-Koordinatensystem, d.h. ein Koordinatensystem vom Typ KLD(PI) oder KLD(USER) oder KLF(PI) oder KLF(USER)
- Das aktive Achsorientierungs-Koordinatensystem, d.h. ein Koordinatensystem vom Typ KSB(PI) oder KSB(USER)

Wenn <CSName> in der Abfrage angegeben wird und das entsprechende Koordinatensystem nicht aktiv ist, sendet der C-886 eine leere Antwort und setzt einen Fehler (Abfrage des Fehlercodes mit ERR? (S. 139)).

KET? (Get Enabled Coordinate System Types)

Beschreibung: Listet die aktiven Typen von Koordinatensystemen auf und zeigt die Namen der entsprechenden Koordinatensysteme an.

Format: KET? [{<CSType>}]

Argumente: <CSType> ist ein aktiver Koordinatensystem-Typ. Mögliche Werte: KSW, KST, KSF, KSD, KLD(PI), KLD(USER), KLF(PI), KLF(USER), KSB(PI), KSB(USER)

Wenn <CSType> weggelassen wird, werden alle aktiven Koordinatensystem-Typen aufgelistet.

Antwort: {<CSType>"="<CSName>}

wobei

<CSName> den Namen des Koordinatensystems angibt.

Hinweise: KET? fragt den flüchtigen Speicher ab.

Wenn das Betriebs-Koordinatensystem vom Typ ZERO aktiv ist, wird dieser Typ in der Antwort auf KET? **nicht** angezeigt, und die Antwort enthält nur Folgendes:

- Den Typ des aktiven Korrektur-Koordinatensystems, d.h. KLD(PI) oder KLD(USER) oder KLF(PI) oder KLF(USER)
- Den Typ des aktiven Achsorientierungs-Koordinatensystems, d.h. KSB(PI) oder KSB(USER)

Wenn <CSType> in der Abfrage angegeben wird und kein Koordinatensystem des entsprechenden Typs aktiv ist, sendet der C-886 eine leere Antwort und setzt einen Fehler (Abfrage des Fehlercodes mit ERR? (S. 139)).

KLC? (Get Properties Of Work-And-Tool Combinations)

Beschreibung: Listet die Eigenschaften der im flüchtigen Speicher vorhandenen Kombinationen von Koordinatensystemen für das Work-und-Tool-Konzept auf.

Das Work-und-Tool-Konzept verwendet eine Kombination aus zwei aktiven Betriebs-Koordinatensystemen. In der Regel besteht die Kombination aus jeweils einem aktiven Koordinatensystem der Typen KST und KSW. Wenn nur für einen der beiden Typen ein Koordinatensystem aktiv ist, wird für den anderen Typ automatisch ein Ersatz verwendet. Ersatzweise verwendete Koordinatensysteme sind unter dem Namen "Zero" in der Antwort auf KLC? enthalten.

Eine Kombination wird im flüchtigen Speicher beim Aktivieren eines Koordinatensystems vom Typ KST oder KSW angelegt. Die Kombinationen sind im flüchtigen Speicher auch dann noch vorhanden, wenn die darin enthaltenen Koordinatensysteme vom Typ KST oder KSW nicht mehr aktiv sind.

Wenn mit KRM (S. 167) ein Koordinatensystem des Typs KST oder KSW gelöscht wird, sind die Kombinationen, in denen dieses Koordinatensystem enthalten war, nicht mehr in der Antwort auf KLC? enthalten.

Mit WPA (S. 224) können die im flüchtigen Speicher vorhandenen Kombinationen in den permanenten Speicher geschrieben werden.

Die Eigenschaften der im flüchtigen Speicher vorhandenen Koordinatensysteme können mit KLS? (S. 164) abgefragt werden.

Format: KLC? [<CSName1>[<CSName2>[<Item1>[<Item2>]]]]

Argumente: <CSName1> ist der Name eines Koordinatensystems vom Typ KST oder KSW, das Bestandteil einer Kombination im flüchtigen Speicher ist.

<CSName2> ist der Name eines Koordinatensystems vom Typ KST oder KSW, das Bestandteil einer Kombination im flüchtigen Speicher ist.

<Item1> ist eine Eigenschaft der Achsen des Controllers für die abgefragte Kombination von Koordinatensystemen. Mögliche Werte:

- NLM: Untergrenze des Verfahrbereichs der Achse
- PLM: Obergrenze des Verfahrbereichs der Achse
- SSL: Aktivierungszustand der Verfahrbereichsgrenzen der Achse
- Wenn vom Controller unterstützt:
  SST: Schrittweite für Bewegungen, die durch eine manuelle Bedieneinheit ausgelöst werden

Die Einstellungen der Eigenschaften für die aktuell aktive Kombination können mit den entsprechenden Befehlen geändert und mit WPA gespeichert werden.

<Item2> ist eine Achse des Controllers, mögliche Werte: X, Y, Z, U, V, W

Wenn die Eigenschaften aller Kombinationen von Koordinatensystemen aufgelistet werden sollen, werden alle Argumente weggelassen.

Antwort: <String>

<String> enthält im XML-Format Informationen zu den im flüchtigen Speicher vorhandenen Kombinationen von Koordinatensystemen.

Der Aufbau der Antwort hängt von der Anzahl der Argumente im gesendeten Befehl ab. Mögliche Antworten je nach Anzahl der Argumente am Beispiel der Koordinatensysteme Node1, Node2, Node3 und Node4:

Gesendet:

```
KLC?
```

Antwort:

```
<CombinedCoordinateSystem>[SP][LF]
[TAB] <NODE1.NODE2 Name="NODE1.NODE2" Work="NODE1"
Tool="NODE2">[SP][LF]
[TAB] [TAB] <NLM X="-3.0" … W="-5.0"/>[SP][LF]
[TAB] [TAB] <PLM X="3.0" … W="5.0"/>[SP][LF]
```

```
[TAB] [TAB] <SSL X="1" Y="0" … W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.1" … W="0.2"/>[SP][LF]
[TAB] </NODE1.NODE2>[SP][LF]
[TAB] [TAB] <NODE1.NODE4  Name="NODE1.NODE4"
Work="NODE1" Tool="NODE4"> [SP][LF]
[TAB] [TAB] <NLM X="-4.0" … W="-3.4"/>[SP][LF]
[TAB] [TAB] <PLM X="2.0" … W="2.0"/>[SP][LF]
[TAB] [TAB] <SSL X="1" Y="0" … W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.2" … W="0.15"/>[SP][LF]
[TAB] </NODE1.NODE4>[SP][LF]
…
</CombinedCoordinateSystem>[LF]
```

Gesendet:

```
KLC? Node1
```

Antwort:

```
<CombinedCoordinateSystem>[SP][LF]
[TAB] <NODE1.NODE2 Name="NODE1.NODE2" Work="NODE1"
Tool="NODE2"> [SP][LF]
[TAB] [TAB] <NLM X="-3.0" … W="-5.0"/>[SP][LF]
[TAB] [TAB] <PLM X="3.0"  … W="5.0"/>[SP][LF]
[TAB] [TAB] <SSL X="1" Y="0" … W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.1" … W="0.2"/>[SP][LF]
[TAB] </NODE1.NODE2 >[SP][LF]
[TAB] <NODE1.NODE4 Name="NODE1.NODE4" Work="NODE1"
Tool="NODE4"> [SP][LF]
[TAB] [TAB] <NLM X="-1.0" … W="-7.0"/>[SP][LF]
[TAB] [TAB] <PLM X="1.1"  … W="10.0"/>[SP][LF]
[TAB] [TAB] <SSL X="1" Y="0" … W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.2" … W="0.21"/>[SP][LF]
[TAB] </NODE1.NODE4 >[SP][LF]
…
</CombinedCoordinateSystem>[LF]
```

Gesendet:

```
KLC? Node1 Node2
```

Antwort:

```
<CombinedCoordinateSystem>[SP][LF]
[TAB] <NODE1.NODE2 Name="NODE1.NODE2" Work="NODE1"
Tool="NODE2"> [SP][LF]
[TAB] [TAB] <NLM X="-3.0" … W="-5.0"/>[SP][LF]
[TAB] [TAB] <PLM X="3.0" …  W="5.0"/>[SP][LF]
[TAB] [TAB] <<SSL X="1" Y="0" … W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.1" … W="0.2"/>[SP][LF]
```

```
[TAB] </NODE1.NODE2 >[SP][LF]
</CombinedCoordinateSystem>[LF]
```

Gesendet:
`KLC? Node1 Node2 PLM`
Antwort:
`<PLM X="3.0" … W="5.0"/>[LF]`

Gesendet:
`KLC? Node1 Node2 PLM X`
Antwort:
`X = 3.0 [LF]`

KLD (Define Leveling Coordinate System By Specifying Values)

Beschreibung: Definiert ein Korrektur-Koordinatensystem vom Typ KLD(USER) zur dauerhaften Korrektur von Fehlern in der Ausrichtung des Positionierers (z. B. Einbaufehler).

Das Korrektur-Koordinatensystem wird auf Basis von Messwerten definiert (z. B. unter Verwendung eines Interferometers) und korrigiert die Linearverschiebung (Achsen X, Y, Z) und Achsneigung (Achsen U, V, W) der Bewegungsplattform des Positionierers.

Wenn die Linearverschiebung und Achsneigung nicht gemessen werden kann: Verwenden Sie KLF (S. 160) zum Definieren eines Korrektur-Koordinatensystems.

Das Koordinatensystem wird im flüchtigen Speicher definiert. Mit WPA (S. 224) kann die Definition in den permanenten Speicher geschrieben werden.

Format: KLD <CSName> [{<AxisID> <Offset>}]

Argumente: <CSName> ist der Name des Koordinatensystems, das definiert werden soll.

<AxisID> ist eine Achse des Controllers. Mögliche Werte: X, Y, Z, U, V, W.

<Offset> ist ein Offset, der nach der Referenzfahrt zum aktuellen Positionswert der Achse addiert wird; in physikalischen Einheiten.

Für Achsen, die im Befehl KLD nicht angegeben werden, wird der Offset auf null gesetzt.

Antwort: Keine

Hinweise: Vor dem Definieren eines Korrektur-Koordinatensystems ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).

Möglichkeiten zum Definieren eines Koordinatensystems mit KLD:

- <CSName> ist ein neuer Name. Das Korrektur-Koordinatensystem wird unter diesem Namen neu angelegt.
- <CSName> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird durch die Definition mit KLD überschrieben.

Empfohlene Vorgehensweise zum Definieren und Aktivieren eines Korrektur-Koordinatensystems vom Typ KLD(USER):

1. Ausführen der Referenzfahrt (siehe FRF (S. 140))
2. Messen der Abweichung von Position und Orientierung der bewegten Plattform des Positionierers von derjenigen Position und Orientierung, in der künftig gelten soll X = 0, Y = 0, Z = 0, U = 0, V = 0, W = 0 (Messung mit externem Messinstrument)
3. Wechsel auf Befehlsebene 1 durch Senden von `CCL 1 advanced`
4. Definieren des Korrektur-Koordinatensystems mit KLD unter Angabe der gemessenen Abweichungen für die Achsen der Bewegungsplattform (Offsetwerte)
5. Aktivieren des Korrektur-Koordinatensystems (siehe KEN (S. 152))
6. Optional: Festlegung des Verhaltens nach der Referenzfahrt durch Setzen der Parameter ***Behaviour After Reference Move*** (ID 0x07030401) und ***Target For Motion After Reference Move*** (ID 0x07030402). Auf diese Weise kann die Plattform z. B. nach der Referenzfahrt automatisch zur Nullposition bewegt werden.
   - Wert des Parameters 0x07030401 = 0: Bewegungsplattform bleibt nach der Referenzfahrt in der Referenzposition
   - Wert des Parameters 0x07030401 = 1: Bewegungsplattform fährt nach der Referenzfahrt zur Zielposition, die durch Parameter 0x07030402 vorgegeben ist
7. Speichern der Einstellungen durch Senden von `WPA SKS`

Die Offsetwerte, die in der Antwort auf KLS? (S. 164) für Koordinatensysteme vom Typ KLD(USER) angezeigt werden, ergeben sich durch Rückrechnung aus allen aktuell aktiven Koordinatensystemen. Sie können sich deshalb beim Wechsel der aktiven Koordinatensysteme ändern. Die in der Antwort auf KLT?

(S. 166) enthaltenen Offsetwerte sind dagegen jeweils auf den angegebenen Vorgänger in der Kette bezogen und deshalb unabhängig von den aktuell aktiven Koordinatensystemen.

Für Korrektur-Koordinatensysteme, die mit KLD definiert werden, gilt Folgendes:

- Das Korrektur-Koordinatensystem ist immer direkter Nachfolger des werkseitig eingestellten Korrektur-Koordinatensystems PI-Levelling (automatische Verkettung).
- Das Korrektur-Koordinatensystem kann **nicht** mit KLN (S. 162) mit anderen Koordinatensystemen verkettet werden

DPA SKS (S. 133) aktiviert unabhängig von der aktuell aktiven Befehlsebene wieder das werkseitig aktive Korrektur-Koordinatensystem PI_Levelling, Details siehe KEN (S. 152).

KLF (Define Leveling Coordinate System At Current Position)

Beschreibung: Definiert ein Korrektur-Koordinatensystem vom Typ KLF(USER) zur dauerhaften Korrektur von Fehlern in der Ausrichtung des Positionierers (z. B. Einbaufehler).

Zum Definieren des Korrektur-Koordinatensystems wird die Bewegungsplattform nach der Referenzfahrt in die Position und Orientierung kommandiert, in der künftig gelten soll X = 0, Y = 0, Z = 0, U = 0, V = 0, W = 0. Durch Senden von KLF wird ein Koordinatensystem mit Offsetwerten definiert, die nach der Referenzfahrt zu den aktuellen Positionswerten der Achsen addiert werden; in physikalischen Einheiten.

Wenn Linearverschiebung (Achsen X, Y, Z) und Achsneigung (Achsen U, V, W) gemessen werden sollen: Verwenden Sie KLD (S. 158) zum Definieren eines Korrektur-Koordinatensystems.

Das Koordinatensystem wird im flüchtigen Speicher definiert. Mit WPA (S. 224) kann die Definition in den permanenten Speicher geschrieben werden.

Format: KLF <CSName>

Argumente: <CSName> ist der Name des Koordinatensystems, das definiert werden soll.

Antwort: Keine

Hinweise: Vor dem Definieren eines Korrektur-Koordinatensystems ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).

Möglichkeiten zum Definieren eines Koordinatensystems mit KLF:

- <CSName> ist ein neuer Name. Das Korrektur-Koordinatensystem wird unter diesem Namen neu angelegt.
- <CSName> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird durch die Definition mit KLF überschrieben.

Die Definition mit KLF ist nur möglich, wenn sich der Positionierer nicht bewegt.

Empfohlene Vorgehensweise zum Definieren und Aktivieren eines Korrektur-Koordinatensystems vom Typ KLF(USER):

1. Ausführen der Referenzfahrt (siehe FRF (S. 140))
2. Anfahren derjenigen Position und Orientierung der Bewegungsplattform des Positionierers, in der künftig gelten soll X = 0, Y = 0, Z = 0, U = 0, V = 0, W = 0
3. Wechsel auf Befehlsebene 1 durch Senden von `CCL 1 advanced`
4. Definieren des Korrektur-Koordinatensystems mit KLF
5. Aktivieren des Korrektur-Koordinatensystems (siehe KEN (S. 152))
6. Optional: Festlegung des Verhaltens nach der Referenzfahrt durch Setzen der Parameter ***Behaviour After Reference Move*** (ID 0x07030401) und ***Target For Motion After Reference Move*** (ID 0x07030402). Auf diese Weise kann die Plattform z. B. nach der Referenzfahrt automatisch zur Nullposition bewegt werden.
   - Wert des Parameters 0x07030401 = 0: Bewegungsplattform bleibt nach der Referenzfahrt in der Referenzposition
   - Wert des Parameters 0x07030401 = 1: Bewegungsplattform fährt nach der Referenzfahrt zur Zielposition, die durch Parameter 0x07030402 vorgegeben ist
7. Speichern der Einstellungen durch Senden von `WPA SKS`

Die Offsetwerte, die in der Antwort auf KLS? (S. 164) für Koordinatensysteme vom Typ KLF(USER) angezeigt werden, ergeben sich durch Rückrechnung aus allen aktuell aktiven Koordinatensystemen. Sie können sich deshalb beim Wechsel der aktiven Koordinatensysteme ändern. Die in der Antwort auf KLT? (S. 166) enthaltenen Offsetwerte sind dagegen jeweils auf den angegebenen Vorgänger in der Kette bezogen und deshalb unabhängig von den aktuell aktiven Koordinatensystemen.

Für Korrektur-Koordinatensysteme, die mit KLF definiert werden, gilt Folgendes:

- Das Korrektur-Koordinatensystem ist immer direkter Nachfolger

des werkseitig eingestellten Korrektur-Koordinatensystems PI-Levelling (automatische Verkettung).

- Das Korrektur-Koordinatensystem kann **nicht** mit KLN (S. 162) mit anderen Koordinatensystemen verkettet werden

DPA SKS (S. 133) aktiviert unabhängig von der aktuell aktiven Befehlsebene wieder das werkseitig aktive Korrektur-Koordinatensystem PI_Levelling, Details siehe KEN (S. 152).

KLN (Link Coordinate Systems)

Beschreibung: Verbindet zwei Koordinatensysteme zu einer Kette aus Vorgänger und Nachfolger.

Format: KLN <CSName1> <CSName2>

Argumente: <CSName1> ist der Name des Koordinatensystems, das in der Kette als Nachfolger an <CSName2> angehängt werden soll.

<CSName2> ist der Name des Koordinatensystems, das der Vorgänger von <CSName1> in der Kette sein soll.

Antwort: Keine

Hinweise: Jedes Koordinatensystem ist Bestandteil mindestens einer Kette. Grundsätzlicher Aufbau von Koordinatensystem-Ketten siehe "Koordinatensysteme" (S. 35).

Werkseitig sind folgende Koordinatensysteme zu einer Kette verbunden:

- Das Koordinatensystem HEXAPOD, das auf der Konfigurationsdatei mit den Geometriedaten des Positionierers basiert, ist der "Urspung" aller Ketten und Vorgänger des Korrektur-Koordinatensystems PI_Levelling (unveränderbare Verkettung)
- PI_Levelling ist Vorgänger des Achsorientierungs-Koordinatensystems PI_Base
- PI_Base ist Vorgänger des Betriebs-Koordinatensystems ZERO

Für verkettete Koordinatensysteme gilt:

- Die tatsächlichen Offsetwerte für die Position der Achsen X, Y, Z, U, V, W ergeben sich jeweils aus den Offsetwerten der mit einem Koordinatensystem verketteten Vorgänger (Details siehe KLT? (S. 166)).
- Jedes Koordinatensystem hat genau einen Vorgänger und kann mindestens einen Nachfolger haben.
- Wenn ein Koordinatensystem aktiv ist, sind alle Vorgänger in der Kette ebenfalls in Verwendung und können nicht gelöscht oder

überschrieben werden.

- Wenn ein (nicht verwendetes) Koordinatensystem gelöscht wird, werden sein Vorgänger und sein Nachfolger in der Kette miteinander verbunden.

Beschränkungen für den Aufbau von Ketten mit KLN:

- Ein Koordinatensystem kann nicht mit sich selbst verkettet werden.
- Ringverbindungen aus mindestens zwei Koordinatensystemen können zwar gebildet, aber nicht mit KEN (S. 152) aktiviert werden.
- Ein Koordinatensystem, das verwendet wird, kann nicht mit KLN als Nachfolger an ein anderes Koordinatensystem angehängt werden.
- An ein Koordinatensystem, das verwendet wird, kann mit KLN ein (nicht verwendetes) Koordinatensystem als Nachfolger angehängt werden.
- Vor dem Verketten von Achsorientierungs-Koordinatensystemen vom Typ KSB(USER) als Nachfolger ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).
- Koordinatensysteme der Typen KLD(PI), KLF(PI), KLD(USER), KLF(USER) und das Koordinatensystem HEXAPOD können nicht mit KLN verkettet werden.
- Das Koordinatensystem PI_Base kann nicht mit KLN als Nachfolger an ein anderes Koordinatensystem angehängt werden.
- Koordinatensysteme vom Typ KSB(USER) können mit KLN nur mit anderen Koordinatensystemen vom Typ KSB(USER) verkettet oder als Nachfolger an das Koordinatensystem PI_Base angehängt werden.
- Das Koordinatensystem ZERO kann nicht mit KLN als Nachfolger an ein anderes Koordinatensystem angehängt werden.

Koordinatensysteme werden mit KLN im flüchtigen Speicher verkettet. Mit WPA (S. 224) kann die Verkettung in den permanenten Speicher geschrieben werden.

KLN? (Get Coordinate System Chains)

Beschreibung:	Listet die Bestandteile der vorhandenen Ketten von Koordinatensystemen auf. Jedes Koordinatensystem ist Bestandteil mindestens einer Kette. Grundsätzlicher Aufbau von Koordinatensystem-Ketten siehe "Koordinatensysteme" (S. 35).
Format:	KLN? [{<CSName>}]
Argumente:	<CSName> ist der Name eines Koordinatensystems, dessen Vorgänger in der Kette aufgelistet werden sollen.

Wenn die Vorgänger aller Koordinatensysteme aufgelistet werden sollen, wird <CSName> weggelassen.

Antwort: {<CSName>"="<String>}

wobei

<String> die Namen der Vorgänger des Koordinatensystems in der Kette enthält. Der "Urspung" der Kette steht immer am Ende der Zeile.

Hinweise: Für eine übersichtliche Darstellung werden für Betriebs-Koordinatensysteme der Typen KSD, KSF, KSW und KST nur die Vorgänger bis zum Koordinatensystem ZERO aufgelistet. Die Vorgänger von ZERO können jedoch gezielt abgefragt werden und sind auch als separate Zeile in der Antwort enthalten, wenn <CSName> in der Abfrage weggelassen wird.

KLS? (Get Coordinate System Properties)

Beschreibung: Listet die Eigenschaften der im flüchtigen Speicher vorhandenen Koordinatensysteme auf.

Mit WPA (S. 224) können die im flüchtigen Speicher vorhandenen Koordinatensysteme in den permanenten Speicher geschrieben werden.

Die Eigenschaften der im flüchtigen Speicher vorhandenen Kombinationen von Koordinatensystemen für das Work-und-Tool-Konzept können mit KLC? (S. 155) abgefragt werden.

Format: KLS? [<CSName>[<Item1>[<Item2>]]]

Argumente: <CSName> ist der Name eines Koordinatensystems. Nicht zulässig: HEXAPOD.

<Item1> ist eine Eigenschaft der Achsen des Controllers. Mögliche Werte:

Für alle Typen von Koordinatensystemen:

- POS: Offset für Positionen der Achsen

Nur für Koordinatensysteme der Typen KSD, KSF und ZERO:

- NLM: Untergrenze des Verfahrbereichs der Achse
- PLM: Obergrenze des Verfahrbereichs der Achse
- SSL: Aktivierungszustand der Verfahrbereichsgrenzen der Achse
- Wenn vom Controller unterstützt:
  SST: Schrittweite für Bewegungen, die durch eine manuelle Bedieneinheit ausgelöst werden

Nur für Koordinatensysteme der Typen KSF und ZERO:

- SPI: Koordinaten des Pivotpunkts

<Item2> ist eine Achse des Controllers, mögliche Werte: X, Y, Z, U, V, W

Wenn die Eigenschaften aller Koordinatensysteme aufgelistet werden sollen, werden alle Argumente weggelassen.

Antwort: <String>

<String> enthält im XML-Format Informationen zu den im flüchtigen Speicher vorhandenen Koordinatensystemen.

Der Aufbau von <String> hängt von der Anzahl der Argumente im gesendeten Befehl ab.

Wenn kein Argument oder nur der Name eines Koordinatensystems angegeben wird, listet <String> zu jedem in der Antwort enthaltenen Koordinatensystem die zutreffenden Angaben für <Item1> und <Item2> und zusätzlich folgende Informationen auf:

- Name des Koordinatensystems (Name="")
- Name des direkten Vorgängers des Koordinatensystems in der Kette (Parent="")
- Aktuelle Verwendung des Koordinatensystems;
  - Used="True": Das Koordinatensystem wird verwendet, d.h. es ist entweder selbst aktiv, oder es ist ein Vorgänger des aktiven Koordinatensystems in dessen Kette.
  - Used="False": Das Koordinatensystem wird nicht verwendet.
- Typ des Koordinatensystems (Type="")

Hinweis: Die Offsetwerte, die in der Antwort auf KLS? für Korrektur-Koordinatensysteme der Typen KLD(USER) und KLF(USER) angezeigt werden, ergeben sich durch Rückrechnung aus allen aktuell aktiven Koordinatensystemen. Sie können sich deshalb beim Wechsel der aktiven Koordinatensysteme ändern.

Beispiel: Das nachfolgende Beispiel zeigt die Antwort auf KLS? für die werkseitig vorhandenen Koordinatensysteme.

```
KLS?
<SingleCoordinateSystem>
     <ZERO Name="ZERO" Parent="PI_BASE" Used="True" Type="ZERO">
          <POS X="0.000000" Y="0.000000" Z="0.000000" U="0.000000"
V="0.000000" W="0.000000"/>
          <NLM X="-10.000100" Y="-10.000100" Z="-10.000100"
U="-1.000100" V="-1.000100" W="-1.000100"/>
          <PLM X="10.000100" Y="10.000100" Z="10.000100"
U="1.000100" V="1.000100" W="1.000100"/>
          <SSL X="1" Y="1" Z="1" U="1" V="1" W="1"/>
          <SPI R="0.000000" S="0.000000" T="481.000000"/>
          <SST X="0.010000" Y="0.010000" Z="0.010000" U="0.010000"
```

```
V="0.010000" W="0.010000"/>
     </ZERO>
     <PI_BASE Name="PI_BASE" Parent="PI_LEVELLING" Used="True"
Type="KSB(PI)">
          <POS X="0.000000" Y="0.000000" Z="0.000000" U="0.000000"
V="0.000000" W="0.000000"/>
     </PI_BASE>
     <PI_LEVELLING Name="PI_LEVELLING" Parent="HEXAPOD"
Used="True" Type="KLD(PI)">
          <POS X="0.000000" Y="0.000000" Z="0.000000" U="0.000000"
V="0.000000" W="0.000000"/>
     </PI_LEVELLING></SingleCoordinateSystem>
```

KLT? (Get Offset Resulting From A Chain)

Beschreibung: Listet für ein Koordinatensystem die Offsetwerte für die Position der Achsen X, Y, Z, U, V, W auf, die aus den Offsetwerten der mit diesem Koordinatensystem verketteten Vorgänger resultieren.

Für Betriebs-Koordinatensysteme gehen nur die als Vorgänger verketteten Betriebs-Koordinatensysteme bis zum Koordinatensystem ZERO in die Berechnung ein.

Für Achsorientierungs-Koordinatensysteme gehen nur die als Vorgänger verketteten Achsorientierungs-Koordinatensysteme bis zum Koordinatensystem HEXAPOD in die Berechnung ein.

Für Korrektur-Koordinatensysteme gehen nur die als Vorgänger verketteten Korrektur-Koordinatensysteme bis zum Koordinatensystem HEXAPOD in die Berechnung ein.

Format: KLT? [<StartCS> [<EndCS>]]

Argumente: <StartCS> ist der Name des Koordinatensystems, für das die aus seinen Vorgängern resultierenden Offsetwerte abgefragt werden sollen.

<EndCS> ist der Name eines als Vorgänger von <StartCS> verketteten Koordinatensystems, das als Ausgangspunkt der Offsetberechnung verwendet werden soll. Wenn <EndCS> weggelassen wird, hängt der Ausgangspunkt der Berechnung von der Angabe für <StartCS> ab:

- <StartCS> ist ein Betriebs-Koordinatensystem (Typ KSD, KSF, KST, KSW oder ZERO): Ausgangspunkt ist ZERO
- <StartCS> ist ein Achsorientierungs- oder Korrektur-Koordinatensystem (Typen KSB(PI), KSB(USER), KLD(PI), KLD(USER), KLF(PI), KLF(USER)): Ausgangspunkt ist HEXAPOD

Wenn die resultierenden Offsetwerte für alle Koordinatensysteme aufgelistet werden sollen, werden alle Argumente weggelassen.

Antwort: <String>

<String> enthält für jedes abgefragte Koordinatensystem eine Zeile mit folgenden Angaben:

- Name: Der Name des Koordinatensystems, für das die resultierenden Offsetwerte aufgelistet werden.
- EndCoordinateSystem: Der Name des Koordinatensystems, das als Ausgangspunkt der Berechnung der Offsetwerte verwendet wurde.
- X, Y, Z, U, V, W: Resultierender Offsetwert für die entsprechende Achse.

Hinweis: KLT? fragt den flüchtigen Speicher ab.

Die Eigenschaften der im flüchtigen Speicher vorhandenen Koordinatensysteme können mit KLS? (S. 164) abgefragt werden.

Beispiel: Das nachfolgende Beispiel zeigt die Antwort auf KLT? für die werkseitig vorhandenen Koordinatensysteme.

```
KLT?
Name=ZERO  EndCoordinateSystem=ZERO   X=0.000000 Y=0.000000
     Z=0.000000 U=0.000000 V=0.000000 W=0.000000
Name=PI_BASE    EndCoordinateSystem=HEXAPOD     X=0.000000
     Y=0.000000 Z=0.000000 U=0.000000 V=0.000000 W=0.000000
Name=PI_LEVELLING    EndCoordinateSystem=HEXAPOD     X=0.000000
     Y=0.000000 Z=0.000000 U=0.000000 V=0.000000 W=0.000000
```

KRM (Remove Coordinate System)

Beschreibung: Löscht ein Koordinatensystem.

Format: KRM <CSName>

Argumente: <CSName> ist der Name des Koordinatensystems, das gelöscht werden soll.

Antwort: Keine

Hinweise: Ein Koordinatensystem, das verwendet wird (d.h. das selbst aktiv ist oder als Vorgänger mit dem aktiven Koordinatensystem verkettet ist), kann nicht gelöscht werden.

Beim Löschen eines (nicht verwendeten) Koordinatensystems werden dessen Vorgänger und Nachfolger in der Kette von Koordinatensystemen miteinander verbunden.

Vor dem Löschen eines Korrektur-Koordinatensystems der Typen KLD(USER), KLF(USER) und KSB(USER) ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).

Das Koordinatensystem wird im flüchtigen Speicher gelöscht. Mit WPA (S. 224) kann das Löschen in den permanenten Speicher übertragen werden.

KSB (Define Orientational Coordinate System)

Beschreibung: Definiert ein Achsorientierungs-Koordinatensystem vom Typ KSB(USER) zum dauerhaften Ändern der Richtung der Achsen X und/oder Y und/oder Z.

Die Richtung der Achsen wird durch Drehen des Koordinatensystems in 90°-Schritten wie folgt geändert:

- Drehen um X, d.h. Angabe eines Winkelwerts für U, ändert die Richtung der Achsen Y und Z
- Drehen um Y, d.h. Angabe eines Winkelwerts für V, ändert die Richtung der Achsen X und Z
- Drehen um Z, d.h. Angabe eines Winkelwerts für W, ändert die Richtung der Achsen X und Y

Das Koordinatensystem wird im flüchtigen Speicher definiert. Mit WPA (S. 224) kann die Definition in den permanenten Speicher geschrieben werden.

Format: KSB <CSName> [{<AxisID> <Angle>}]

Argumente: <CSName> ist der Name des Koordinatensystems, das definiert werden soll.

<AxisID> ist eine Achse des Controllers. Mögliche Werte: U, V, W.

<Angle> ist der Winkel, um den die Achse gedreht werden soll. Mögliche Werte: 0, 90, 180, 270, -90, -180, -270 (Einheit: Grad).

Für Achsen, die im Befehl KSB nicht angegeben werden, wird der Winkel auf null gesetzt.

Antwort: Keine

Hinweise: Vor dem Definieren eines Achsorientierungs-Koordinatensystems ist der Wechsel auf Befehlsebene 1 erforderlich (siehe CCL (S. 129)).

Möglichkeiten zum Definieren eines Koordinatensystems mit KSB:

- <CSName> ist ein neuer Name. Das

Achsorientierungs-Koordinatensystem wird unter diesem Namen neu angelegt und hat automatisch das aktive Achsorientierungs-Koordinatensystem als Vorgänger

- <CSName> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird durch die Definition mit KSB überschrieben und automatisch als Nachfolger an das aktive Achsorientierungs-Koordinatensystem angehängt

Achsorientierungs-Koordinatensysteme vom Typ KSB(USER) können mit KLN (S. 162) mit anderen Koordinatensystemen vom Typ KSB(User) verkettet oder als Nachfolger an das Koordinatensystem PI_Base angehängt werden.

Achsorientierungs-Koordinatensysteme vom Typ KSB(USER) können mit KEN (S. 152) aktiviert werden. DPA SKS (S. 133) aktiviert wieder das werkseitig aktive Achsorientierungs-Koordinatensystem PI_Base, Details siehe KEN.

KSD (Define Operating Coordinate System By Specifying Values)

Beschreibung: Definiert ein Betriebs-Koordinatensystem vom Typ KSD.

Ein Koordinatensystem vom Typ KSD kann beliebig im Raum platziert und ausgerichtet werden. Siehe "Das Work-und-Tool-Konzept" und insbesondere "Betrachtung von Koordinatensystemen der Typen KSD und KSF aus dem Blickwinkel von "Work" und "Tool"" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007) sowie "Begriffserklärung" (S. 3) für weiterführende Erläuterungen.

Die Platzierung des Koordinatensystems im Raum wird durch Angabe von Offsetwerten für die Achsen X, Y, Z festgelegt. Die Ausrichtung des Koordinatensystems wird durch Angabe von Offsetwerten für die Achsen U, V, W festgelegt:

- Drehen um X, d.h. Angabe eines Offsets für U, ändert die Richtung der Achsen Y und Z
- Drehen um Y, d.h. Angabe eines Offsets für V, ändert die Richtung der Achsen X und Z
- Drehen um Z, d.h. Angabe eines Offsets für W, ändert die Richtung der Achsen X und Y

Wenn ein Betriebs-Koordinatensystem vom Typ KSD mit KEN (S. 152) aktiviert wurde:

- Das Koordinatensystem bestimmt die Positionsangaben für die Bewegungsplattform des Positionierers (Abfrage der aktuellen Position mit POS? (S. 191)).
- Die Koordinaten des Drehpunkts können **nicht** mit SPI (S. 197)

geändert werden, und der mit SPI definierte Pivotpunkt wird **nicht** verwendet.

Wenn ein Koordinatensystem vom Typ KSD selbst nicht aktiv ist, jedoch als Vorgänger mit dem aktiven Betriebs-Koordinatensystem verkettet ist, gehen seine Offsetwerte in die Berechnung der Offsetwerte des aktiven Betriebs-Koordinatensystems ein (siehe KLT? (S. 166)).

Das Koordinatensystem wird im flüchtigen Speicher definiert. Mit WPA (S. 224) kann die Definition in den permanenten Speicher geschrieben werden.

Format: KSD <CSName> [{<AxisID> <Offset>}]

Argumente: <CSName> ist der Name des Koordinatensystems, das definiert werden soll.

<AxisID> ist eine Achse des Controllers. Mögliche Werte: X, Y, Z, U, V, W.

<Offset> ist ein Offsetwert für die Platzierung oder Ausrichtung der Achse; in physikalischen Einheiten.

Für Achsen, die im Befehl KSD nicht angegeben werden, wird der Offset auf null gesetzt.

Antwort: Keine

Hinweise: Wenn ein Betriebskoordinatensystem vom Typ KSD aktiv ist:

- Neben den Offsetwerten für die Achsen X, Y, Z, U, V, W gibt das Koordinatensystem Werte für die folgenden Einstellungen vor:
  - NLM (S. 188): Untergrenze des Verfahrbereichs der Achse
  - PLM (S. 190): Obergrenze des Verfahrbereichs der Achse
  - SSL (S. 199): Aktivierungszustand der Verfahrbereichsgrenzen der Achse
  - Wenn vom Controller unterstützt:
    SST: Schrittweite für Bewegungen, die durch eine manuelle Bedieneinheit ausgelöst werden
- Die aktuellen Einstellungen können mit den entsprechenden Befehlen geändert werden.
- Die aktuellen Einstellungen können mit KLS? (S. 164) abgefragt werden.
- Die aktuellen Einstellungen werden mit WPA (S. 224) für das Koordinatensystem gespeichert.
- Für Details zum Auslösen von Bewegungen siehe "Kommandieren von Bewegungen" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007).

Möglichkeiten zum Definieren eines Koordinatensystems mit KSD:

- <CSName> ist ein neuer Name. Das Betriebs-Koordinatensystem wird unter diesem Namen neu angelegt und hat automatisch das Betriebs-Koordinatensystem ZERO als Vorgänger.
- <CSName> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird durch die Definition mit KSD überschrieben.
  - Wenn das überschriebene Koordinatensystem ebenfalls vom Typ KSD war, bleibt die Verkettung mit seinem Vorgänger und Nachfolger erhalten.
  - Wenn das überschriebene Koordinatensystem nicht vom Typ KSD war, wird sein Typ in KSD geändert, und es wird an ZERO als neuen Vorgänger angehängt.

Betriebs-Koordinatensysteme der Typen KSD, KSF, KST, KSW können mit KLN (S. 162) mit anderen Betriebs-Koordinatensystemen dieser Typen verkettet oder wieder als Nachfolger an das Koordinatensystem ZERO angehängt werden.

KEN ZERO und DPA SKS aktivieren wieder das werkseitig aktive Betriebs-Koordinatensystem ZERO, Details siehe KEN (S. 152).

KSF (Define Operating Coordinate System At Current Position)

Beschreibung: Definiert ein Betriebs-Koordinatensystem vom Typ KSF an der aktuellen Position der Bewegungsplattform des Positionierers.

Siehe "Das Work-und-Tool-Konzept" und insbesondere "Betrachtung von Koordinatensystemen der Typen KSD und KSF aus dem Blickwinkel von "Work" und "Tool"" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007) sowie "Begriffserklärung" (S. 3) für weiterführende Erläuterungen.

Wenn ein Betriebs-Koordinatensystem vom Typ KSF mit KEN (S. 152) aktiviert wurde, bestimmt es die Positionsangaben für die Bewegungsplattform des Positionierers (Abfrage der aktuellen Position mit POS? (S. 191)), und der mit SPI (S. 197) definierte Pivotpunkt wird als Drehpunkt für Rotationen verwendet.

Wenn ein Koordinatensystem vom Typ KSF selbst nicht aktiv ist, jedoch als Vorgänger mit dem aktiven Betriebs-Koordinatensystem verkettet ist, gehen seine Offsetwerte in die Berechnung der Offsetwerte des aktiven Betriebs-Koordinatensystems ein (siehe KLT? (S. 166)).

Die Definition mit KSF ist nur möglich, wenn sich der Positionierer nicht bewegt.

Das Koordinatensystem wird im flüchtigen Speicher definiert. Mit WPA (S. 224) kann die Definition in den permanenten Speicher geschrieben werden.

Format: KSF <CSName>

Argumente: <CSName> ist der Name des Koordinatensystems, das definiert werden soll.

Antwort: Keine

Hinweise: Wenn ein Koordinatensystem vom Typ KSF definiert wird, werden seine Pivotpunkt-Koordinaten (R, S, T, siehe SPI (S. 197)) auf die Werte gesetzt, die für das aktuell aktive Betriebs-Koordinatensystem gültig sind. Wenn das aktive Betriebs-Koordinatensystem den mit SPI gesetzten Pivotpunkt nicht unterstützt (Typen KSD, KST/KSW), werden die Pivotpunkt-Koordinaten des KSF-Koordinatensystems auf R = S = T = 0 gesetzt.

Wenn ein Betriebskoordinatensystem vom Typ KSF aktiv ist:

- Neben den Offsetwerten für die Achsen X, Y, Z, U, V, W gibt das Koordinatensystem Werte für die folgenden Einstellungen vor:
  - NLM (S. 188): Untergrenze des Verfahrbereichs der Achse
  - PLM (S. 190): Obergrenze des Verfahrbereichs der Achse
  - SSL (S. 199): Aktivierungszustand der Verfahrbereichsgrenzen der Achse
  - SPI (S. 197): Pivotpunktkoordinaten R, S, T
  - Wenn vom Controller unterstützt:
    SST: Schrittweite für Bewegungen, die durch eine manuelle Bedieneinheit ausgelöst werden
- Die aktuellen Einstellungen können mit den entsprechenden Befehlen geändert werden. Die Pivotpunktkoordinaten können mit SPI nur geändert werden, wenn für die aktuelle Position der Plattform gilt U = V = W = 0.
- Die aktuellen Einstellungen können mit KLS? (S. 164) abgefragt werden.
- Die aktuellen Einstellungen werden mit WPA (S. 224) für das Koordinatensystem gespeichert.
- Für Details zum Auslösen von Bewegungen siehe "Kommandieren von Bewegungen" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007).

Möglichkeiten zum Definieren eines Koordinatensystems mit KSF:

- <CSName> ist ein neuer Name. Das Betriebs-Koordinatensystem wird unter diesem Namen neu angelegt und hat automatisch das Betriebs-Koordinatensystem ZERO als Vorgänger.
- <CSName> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird durch die

Definition mit KSF überschrieben und automatisch als Nachfolger an das Betriebs-Koordinatensystem ZERO angehängt.

Betriebs-Koordinatensysteme der Typen KSD, KSF, KST, KSW können mit KLN (S. 162) mit anderen Betriebs-Koordinatensystemen dieser Typen verkettet oder wieder als Nachfolger an das Koordinatensystem ZERO angehängt werden.

`KEN ZERO` und `DPA SKS` aktivieren wieder das werkseitig aktive Betriebs-Koordinatensystem ZERO, Details siehe KEN (S. 152).

KST (Define "Tool" Operating Coordinate System)

Beschreibung: Definiert ein Betriebs-Koordinatensystem vom Typ KST ("Tool-Koordinatensystem") für das Work-und-Tool-Konzept.

Siehe "Das Work-und-Tool-Konzept" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007) sowie "Begriffserklärung" (S. 3) für weiterführende Erläuterungen.

Die Platzierung des Tool-Koordinatensystems wird durch Angabe von Offsetwerten für die Achsen X, Y, Z festgelegt. Die Ausrichtung des Tool-Koordinatensystems wird durch Angabe von Offsetwerten für die Achsen U, V, W festgelegt:

- Drehen um X, d.h. Angabe eines Offsets für U, ändert die Richtung der Achsen Y und Z
- Drehen um Y, d.h. Angabe eines Offsets für V, ändert die Richtung der Achsen X und Z
- Drehen um Z, d.h. Angabe eines Offsets für W, ändert die Richtung der Achsen X und Y

Wenn ein Betriebs-Koordinatensystem vom Typ KST aktiv ist, wird mit dem Work-und-Tool-Konzept gearbeitet, und Folgendes gilt:

- Zusätzlich zum Tool-Koordinatensystem muss ein Work-Koordinatensystem aktiv sein. Wenn kein Work-Koordinatensystem vom Typ KSW aktiv ist, wird als Ersatz ein automatisch erzeugtes Work-Koordinatensystem verwendet.
- Die Koordinaten des Drehpunkts können **nicht** mit SPI (S. 197) geändert werden, und der mit SPI definierte Pivotpunkt wird **nicht** verwendet.

Wenn ein Koordinatensystem vom Typ KST selbst nicht aktiv ist, jedoch als Vorgänger mit dem aktiven Betriebs-Koordinatensystem verkettet ist, gehen seine Offsetwerte in die Berechnung der Offsetwerte des aktiven Betriebs-Koordinatensystems ein (siehe KLT? (S. 166)).

Das Koordinatensystem wird im flüchtigen Speicher definiert. Mit WPA (S. 224) kann die Definition in den permanenten Speicher geschrieben werden.

Format: KST <CSName> [{<AxisID> <Offset>}]

Argumente: <CSName> ist der Name des Koordinatensystems, das definiert werden soll.

<AxisID> ist eine Achse des Controllers. Mögliche Werte: X, Y, Z, U, V, W.

<Offset> ist ein Offsetwert für die Platzierung oder Ausrichtung der Achse; in physikalischen Einheiten.

Für Achsen, die im Befehl KST nicht angegeben werden, wird der Offset auf null gesetzt.

Antwort: Keine

Hinweise: Wenn eine Kombination aus Koordinatensystemen der Typen KSW/KST oder ZERO/KST oder KSW/ZERO aktiv ist:

- Neben den Offsetwerten für die Achsen X, Y, Z, U, V, W gibt die Kombination Werte für die folgenden Einstellungen vor:
  - NLM (S. 188): Untergrenze des Verfahrbereichs der Achse
  - PLM (S. 190): Obergrenze des Verfahrbereichs der Achse
  - SSL (S. 199): Aktivierungszustand der Verfahrbereichsgrenzen der Achse
  - Wenn vom Controller unterstützt:
    SST: Schrittweite für Bewegungen, die durch eine manuelle Bedieneinheit ausgelöst werden
- Die aktuellen Einstellungen können mit den entsprechenden Befehlen geändert werden.
- Die aktuellen Einstellungen können mit KLC? (S. 155) abgefragt werden.
- Die aktuellen Einstellungen werden mit WPA (S. 224) für die Koordinatensystem-Kombination gespeichert.
- Für Details zum Auslösen von Bewegungen siehe "Kommandieren von Bewegungen" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007).

Möglichkeiten zum Definieren eines Koordinatensystems mit KST:

- <CSName> ist ein neuer Name. Das Betriebs-Koordinatensystem wird unter diesem Namen neu angelegt und hat automatisch das Betriebs-Koordinatensystem ZERO als Vorgänger.
- <CSName> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird durch die Definition mit KST überschrieben.

- Wenn das überschriebene Koordinatensystem ebenfalls vom Typ KST war, bleibt die Verkettung mit seinem Vorgänger und Nachfolger erhalten.
- Wenn das überschriebene Koordinatensystem nicht vom Typ KST war, wird sein Typ in KST geändert, und es wird an ZERO als neuen Vorgänger angehängt.

Betriebs-Koordinatensysteme der Typen KSD, KSF, KST, KSW können mit KLN (S. 162) mit anderen Betriebs-Koordinatensystemen dieser Typen verkettet oder wieder als Nachfolger an das Koordinatensystem ZERO angehängt werden.

KEN ZERO und DPA SKS aktivieren wieder das werkseitig aktive Betriebs-Koordinatensystem ZERO, Details siehe KEN (S. 152).

KSW (Define "Work" Operating Coordinate System)

Beschreibung: Definiert ein Betriebs-Koordinatensystem vom Typ KSW ("Work-Koordinatensystem") für das Work-und-Tool-Konzept.

Siehe "Das Work-und-Tool-Konzept" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007) sowie "Begriffserklärung" (S. 3) für weiterführende Erläuterungen.

Die Platzierung des Work-Koordinatensystems wird durch Angabe von Offsetwerten für die Achsen X, Y, Z festgelegt. Die Ausrichtung des Work-Koordinatensystems wird durch Angabe von Offsetwerten für die Achsen U, V, W festgelegt:

- Drehen um X, d.h. Angabe eines Offsets für U, ändert die Richtung der Achsen Y und Z
- Drehen um Y, d.h. Angabe eines Offsets für V, ändert die Richtung der Achsen X und Z
- Drehen um Z, d.h. Angabe eines Offsets für W, ändert die Richtung der Achsen X und Y

Wenn ein Betriebs-Koordinatensystem vom Typ KSW aktiv ist, wird mit dem Work-und-Tool-Konzept gearbeitet, und Folgendes gilt:

- Zusätzlich zum Work-Koordinatensystem muss ein Tool-Koordinatensystem aktiv sein. Wenn kein Tool-Koordinatensystem vom Typ KST aktiv ist, wird als Ersatz ein automatisch erzeugtes Tool-Koordinatensystem verwendet.
- Die Koordinaten des Drehpunkts können **nicht** mit SPI (S. 197) geändert werden, und der mit SPI definierte Pivotpunkt wird **nicht** verwendet.

Wenn ein Koordinatensystem vom Typ KSW selbst nicht aktiv ist,

jedoch als Vorgänger mit dem aktiven Betriebs-Koordinatensystem verkettet ist, gehen seine Offsetwerte in die Berechnung der Offsetwerte des aktiven Betriebs-Koordinatensystems ein (siehe KLT? (S. 166)).

Das Koordinatensystem wird im flüchtigen Speicher definiert. Mit WPA (S. 224) kann die Definition in den permanenten Speicher geschrieben werden.

Format: KSW <CSName> [{<AxisID> <Offset>}]

Argumente: <CSName> ist der Name des Koordinatensystems, das definiert werden soll.

<AxisID> ist eine Achse des Controllers. Mögliche Werte: X, Y, Z, U, V, W.

<Offset> ist ein Offsetwert für die Platzierung oder Ausrichtung der Achse; in physikalischen Einheiten.

Für Achsen, die im Befehl KSW nicht angegeben werden, wird der Offset auf null gesetzt.

Antwort: Keine

Hinweise: Wenn eine Kombination aus Koordinatensystemen der Typen KSW/KST oder ZERO/KST oder KSW/ZERO aktiv ist:

- Neben den Offsetwerten für die Achsen X, Y, Z, U, V, W gibt die Kombination Werte für die folgenden Einstellungen vor:
  - NLM (S. 188): Untergrenze des Verfahrbereichs der Achse
  - PLM (S. 190): Obergrenze des Verfahrbereichs der Achse
  - SSL (S. 199): Aktivierungszustand der Verfahrbereichsgrenzen der Achse
  - Wenn vom Controller unterstützt:
    SST: Schrittweite für Bewegungen, die durch eine manuelle Bedieneinheit ausgelöst werden
- Die aktuellen Einstellungen können mit den entsprechenden Befehlen geändert werden.
- Die aktuellen Einstellungen können mit KLC? (S. 155) abgefragt werden.
- Die aktuellen Einstellungen werden mit WPA (S. 224) für die Koordinatensystem-Kombination gespeichert.
- Für Details zum Auslösen von Bewegungen siehe "Kommandieren von Bewegungen" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007).

Möglichkeiten zum Definieren eines Koordinatensystems mit KSW:

- <CSName> ist ein neuer Name. Das Betriebs-Koordinatensystem

wird unter diesem Namen neu angelegt und hat automatisch das Betriebs-Koordinatensystem ZERO als Vorgänger.

- <CSName> ist der Name eines existierenden Koordinatensystems, das nicht verwendet wird. Das Koordinatensystem wird durch die Definition mit KSW überschrieben.
  - Wenn das überschriebene Koordinatensystem ebenfalls vom Typ KSW war, bleibt die Verkettung mit seinem Vorgänger und Nachfolger erhalten.
  - Wenn das überschriebene Koordinatensystem nicht vom Typ KSW war, wird sein Typ in KSW geändert, und es wird an ZERO als neuen Vorgänger angehängt.

Betriebs-Koordinatensysteme der Typen KSD, KSF, KST, KSW können mit KLN (S. 162) mit anderen Betriebs-Koordinatensystemen dieser Typen verkettet oder wieder als Nachfolger an das Koordinatensystem ZERO angehängt werden.

`KEN ZERO` und `DPA SKS` aktivieren wieder das werkseitig aktive Betriebs-Koordinatensystem ZERO, Details siehe KEN (S. 152).

MRT (Set Target Relative In Tool Coordinate System)

Beschreibung: Bewegt die angegebene Achse relativ im Tool-Koordinatensystem.

Siehe "Das Work-und-Tool-Konzept" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007) sowie "Begriffserklärung" (S. 3) für weiterführende Erläuterungen.

Während der Ermittlung der anzufahrenden Zielposition aus den Werten für <Distance> wird zuerst die Translation berechnet und danach die Rotation.

Der Servomodus muss für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb).

Format: MRT {<AxisID> <Distance>}

Argumente: <AxisID> ist eine Achse des Controllers. Mögliche Werte: X, Y, Z, U, V, W.

<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

Fehlersuche:

- Zielposition außerhalb des aktuellen Arbeitsraums.
- Parameter ***Trajectory Source*** (ID 0x19001900) ist auf 1 gesetzt

(muss bei Verwendung von MRT jedoch auf 0 gesetzt sein).

- Servomodus ist Aus für eine der angegebenen Achsen.
- Für mindestens eine Achse wurde die Referenzfahrt nicht erfolgreich abgeschlossen.

Hinweise: Wenn nicht mit dem Work-und-Tool-Konzept gearbeitet wird, werden Bewegungen mit MRT in dem Tool-Koordinatensystem ausgeführt, das entsprechend dem aktiven Betriebs-Koordinatensystem besteht (siehe "Betrachtung von Koordinatensystemen der Typen KSD und KSF aus dem Blickwinkel von "Work" und "Tool"" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007)).

Die physikalische Einheit, in der <Distance> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden.

Um festzustellen, ob eine Bewegung abgeschlossen ist, wird empfohlen, #5 (S. 124) zu senden.

Die Bewegung kann durch #24 (S. 127), STP (S. 202) und HLT (S. 145) abgebrochen werden.

MRW (Set Target Relative In Work Coordinate System)

Beschreibung: Bewegt die angegebene Achse relativ im Work-Koordinatensystem.

Siehe "Das Work-und-Tool-Konzept" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007) sowie "Begriffserklärung" (S. 3) für weiterführende Erläuterungen.

Während der Ermittlung der anzufahrenden Zielposition aus den Werten für <Distance> wird zuerst die Translation berechnet und danach die Rotation.

Der Servomodus muss für die kommandierte Achse vor dem Einsatz dieses Befehls eingeschaltet sein (geregelter Betrieb).

Format: MRW {<AxisID> <Distance>}

Argumente: <AxisID> ist eine Achse des Controllers. Mögliche Werte: X, Y, Z, U, V, W.

<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

Fehlersuche:

- Zielposition außerhalb des aktuellen Arbeitsraums.
- Parameter ***Trajectory Source*** (ID 0x19001900) ist auf 1 gesetzt

(muss bei Verwendung von MRW jedoch auf 0 gesetzt sein).

- Servomodus ist Aus für eine der angegebenen Achsen.
- Für mindestens eine Achse wurde die Referenzfahrt nicht erfolgreich abgeschlossen.

Hinweise: Wenn nicht mit dem Work-und-Tool-Konzept gearbeitet wird, werden Bewegungen mit MRW in dem Work-Koordinatensystem ausgeführt, das entsprechend dem aktiven Betriebs-Koordinatensystem besteht (siehe "Betrachtung von Koordinatensystemen der Typen KSD und KSF aus dem Blickwinkel von "Work" und "Tool"" in der Technical Note "Koordinatensysteme für Hexapod-Mikroroboter" (C887T0007)).

Die physikalische Einheit, in der <Distance> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden.

Um festzustellen, ob eine Bewegung abgeschlossen ist, wird empfohlen, #5 (S. 124) zu senden.

Die Bewegung kann durch #24 (S. 127), STP (S. 202) und HLT (S. 145) abgebrochen werden.

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

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 FREE?
MAC NSTART <macroname> <uint> [{<String>}]
MAC START <macroname> [{<String>}]

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 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 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 FREE?
Fragt nach dem freien Speicherplatz für die Makroaufzeichnung
Antwort: <uint> ist die Zeichenanzahl in Bytes, für die noch freier Speicher zur Verfügung steht

MAC NSTART <macroname> <uint> [{<String>}]
Wiederholt das angegebene Makro <uint> Mal. Eine neue Ausführung wird gestartet, wenn die letzte beendet ist.
<String> steht für den Wert einer im Makro enthaltenen lokalen Variablen. 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. Pro Befehlszeile sind maximal 256 Zeichen zulässig. <String> kann direkt oder über den Wert einer anderen Variable angegeben werden. Nähere Angaben siehe "Variablen" (S. 112).

MAC START <macroname> [{<String>}]
Startet eine Ausführung des angegebenen Makros. <String> hat 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.

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

Makros können lokale und globale Variablen beinhalten. Die Namen der in einem Makro verwendeten lokalen Variablen müssen eine

fortlaufende Reihe bilden. Beispiel für erlaubte Benennung: 1, 2, 3, 4. Nicht erlaubt ist z.B. die Benennung mit 1, 2, 5, 6. Weitere Informationen finden Sie im Abschnitt "Variablen" (S. 112).

Ein laufendes Makro sendet keine Antworten an eine Schnittstelle.

In Abhängigkeit vom Parameterwert 0x72 (Ignore Macro Error) bestehen die folgenden Möglichkeiten, wenn ein Fehler durch ein laufendes Makro verursacht wird:

0 = Makroausführung wird gestoppt
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-886 zur Verfügung gestellten Befehle können nur in Makros verwendet werden:
ADD (S. 128), CPY (S. 131), DEL (S. 132), JRC (S. 150), MEX (S. 184) und WAC (S. 213).

Ein Makro kann ein anderes Makro starten. Die Höchstzahl der Verschachtelungsebenen beträgt 10. 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; nur der letzte Befehl wird, unabhängig von seiner Quelle, ausgeführt.

Die Makroausführung kann durch #24 (S. 127) und STP (S. 202) 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. 126) können Sie abfragen, ob ein Makro aktuell auf dem Controller ausgeführt wird.

Warnung: Die Anzahl von Schreibzyklen im permanenten Speicher ist begrenzt.

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: WPA, WAV, WTR, WGO, WAV?

Beispiel:	Senden: `MAN? WPA`
	Empfangen:
	`WPA <Password> [{<ItemID> <PamID>}] Save Parameters To Non-Volatile Memory`
	`#AvailablePasswords`
	`<Pswd> <Param_Setting>`
	`100 All Parameters, Settings of Coordinate System Configuration`
	`101 All Parameters`
	`SKS Settings of Coordinate System Configuration`
	`end of help`

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. 213).

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

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

Fehlersuche:

- Zielposition außerhalb des aktuellen Arbeitsraums.
- Unzulässige Achsenkennung
  Weitere Informationen finden Sie im Abschnitt "Kommandierbare Elemente" (S. 25).
- Servomodus ist Aus für eine der angegebenen Achsen.
- Für mindestens eine Achse wurde die Referenzfahrt nicht erfolgreich abgeschlossen.

Hinweise:

Der Servomodus muss bei Verwendung dieses Befehls eingeschaltet sein (geregelter Betrieb).

Die physikalische Einheit, in der <Position> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden.

Um festzustellen, ob eine Bewegung abgeschlossen ist, wird empfohlen, #5 (S. 124) zu senden.

Die Bewegung kann durch #24 (S. 127), STP (S. 202) und HLT (S. 145) abgebrochen werden.

Für die Achsen des Positionierers (X, Y, Z, U, V, W) gilt Folgendes:

- Vor dem Start jeder Bewegung erfolgt eine Prüfung, ob die Bewegungsplattform die kommandierte Zielposition tatsächlich erreichen kann. Mit VMO? (S. 212) können Sie abfragen, ob die Zielposition erreicht werden kann.
- Rotationen erfolgen um den Drehpunkt.
- Je nach Einstellung des Parameters ***Trajectory Source*** (ID 0x19001900) wird das Dynamikprofil für die Achsen des Positionierers (X, Y, Z, U, V, W) durch eine der beiden folgende Quellen vorgegeben:
  - Profilgenerator des C-886
  - Zyklische Übertragung von Zielpositionen durch aufeinander folgende MOV-Befehle

Weitere Informationen siehe "Bewegungen des Positionierers" (S. 29).

Beispiel 1:

Senden: MOV X 10 U 5

Hinweis: Achse X bewegt sich nach 10 (Zielposition in mm), Achse U bewegt sich nach 5 (Zielposition in °)

Beispiel 2:

Senden: MOV X 4 Y 2.3 Z -3 U -5.3 V 3 W 1

Hinweis: Mit einem einzigen Bewegungsbefehl können Zielpositionen für alle sechs Achsen des Positionierers gesetzt werden.

Beispiel 3:

Senden: MOV Z 100

Senden: ERR?

Empfangen: 7

Hinweis: Die Achse bewegt sich nicht. Der Fehlercode „7" in der Antwort auf den Befehl ERR? (S. 139) zeigt an, dass die im Bewegungsbefehl 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 unterschiedliche Quellen geändert werden, siehe "Bewegungen des Positionierers" (S. 29).
MOV? fragt die kommandierten Positionen ab. Verwenden Sie POS? (S. 191), 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

Fehlersuche:

- Zielposition außerhalb des aktuellen Arbeitsraums.
- Unzulässige Achsenkennung
  Weitere Informationen finden Sie im Abschnitt "Kommandierbare Elemente" (S. 25).
- Servomodus ist Aus für eine der angegebenen Achsen.
- Für mindestens eine Achse wurde die Referenzfahrt nicht erfolgreich abgeschlossen.

Hinweise:

Der Servomodus muss bei Verwendung dieses Befehls eingeschaltet sein (geregelter Betrieb).

Die physikalische Einheit, in der <Distance> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden.

Um festzustellen, ob eine Bewegung abgeschlossen ist, wird empfohlen, #5 (S. 124) zu senden.

Die Bewegung kann durch #24 (S. 127), STP (S. 202) und HLT (S. 145) abgebrochen werden.

Für die Achsen des Positionierers (X, Y, Z, U, V, W) gilt Folgendes:

- Vor dem Start jeder Bewegung erfolgt eine Prüfung, ob die Bewegungsplattform die kommandierte Zielposition tatsächlich erreichen kann. Mit VMO? (S. 212) können Sie abfragen, ob die Zielposition erreicht werden kann.
- Rotationen erfolgen um denDrehpunkt.
- Wenn der Parameter ***Trajectory Source*** (ID 0x19001900) auf 1 gesetzt ist, muss das Dynamikprofil durch aufeinander folgende MOV-Befehle vorgegeben werden. MVR ist nicht zulässig.

Weitere Informationen siehe "Bewegungen des Positionierers" (S. 29).

Beispiel:

Senden: MOV X 0.5
Hinweis: Dies ist eine absolute Bewegung.
Senden: POS? X
Empfangen: X=0.500000
Senden: MOV? X
Empfangen: X=0.500000
Senden: MVR X 2
Hinweis: Dies ist eine relative Bewegung.
Senden: POS? X
Empfangen: X=2.500000
Senden: MVR X 2000

Hinweis: Neue Zielposition von Achse X würde den Bewegungsbereich überschreiten. Befehl wird ignoriert, d. h. die Zielposition bleibt unverändert und die Achse bewegt sich nicht.
Senden: MOV? X
Empfangen: X=2.500000
Senden: POS? X
Empfangen: X=2.500000

NLM (Set Low Position Soft Limit)

Beschreibung:	Setzt die Untergrenze des Achsenstellwegs im geregelten Betrieb ("Verfahrbereichsgrenze").
Format:	NLM {<AxisID> <LowLimit>}
Argumente:	<AxisID> ist eine Achse des Controllers <LowLimit> ist die Position für die Untergrenze des Stellwegs in physikalischen Einheiten
Antwort:	Keine
Hinweise:	Für die Achsen der Bewegungsplattform des Positionierers werden die Standardwerte für NLM durch das aktive Betriebs-Koordinatensystem vorgegeben (beim Work-und-Tool-Konzept durch die Kombination aus zwei Betriebs-Koordinatensystemen). Der mit NLM gesetzte Wert muss kleiner als der aktuelle Positionswert sein. Für die Achsen X, Y, Z, U, V und W sind nur negative Werte erlaubt. Die Verfahrbereichsgrenzen werden mit SSL (S. 199) aktiviert und deaktiviert. Verfahrbereichsgrenzen können nur gesetzt werden, wenn sich die Achse nicht bewegt (Abfrage mit #5 (S. 124)). Die physikalische Einheit, in der <LowLimit> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden. Wenn der Pivotpunkt mit SPI (S. 198) geändert wird, werden die Verfahrbereichsgrenzen für die Rotationsachsen U, V und W nicht angepasst.

Beispiel:

Senden: NLM? X
Empfangen: X = -22.5
Senden: POS? X
Empfangen: X = -10
Senden: NLM X -5
Senden: ERR?
Empfangen: 27 - (error 27 - "Soft limit out of range")
Senden: NLM? X
Empfangen: X = -22.5

NLM? (Get Low Position Soft Limit)

Beschreibung: Fragt die Position der Verfahrbereichsgrenze ab, die die Untergrenze des Achsenstellwegs im geregelten Betrieb bestimmt.

Format: NLM? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers

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

wobei

<LowLimit> die Position für die Untergrenze des Stellwegs in physikalischen Einheiten ist.

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

PLM (Set High Position Soft Limit)

Beschreibung:	Setzt die Obergrenze des Achsenstellwegs im geregelten Betrieb ("Verfahrbereichsgrenze").
Format:	PLM {<AxisID> <HighLimit>}
Argumente:	<AxisID> ist eine Achse des Controllers <HighLimit> ist die Position für die Obergrenze des Stellwegs in physikalischen Einheiten.
Antwort:	Keine
Hinweise:	Für die Achsen der Bewegungsplattform des Positionierers werden die Standardwerte für PLM durch das aktive Betriebs-Koordinatensystem vorgegeben (beim Work-und-Tool-Konzept durch die Kombination aus zwei Betriebs-Koordinatensystemen). Der mit PLM gesetzte Wert muss größer als der aktuelle Positionswert sein. Für die Achsen X, Y, Z, U, V und W sind nur positive Werte erlaubt. Die Verfahrbereichsgrenzen werden mit SSL (S. 199) aktiviert und deaktiviert. Verfahrbereichsgrenzen können nur gesetzt werden, wenn sich die Achse nicht bewegt (Abfrage mit #5 (S. 124)). Die physikalische Einheit, in der <HighLimit> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden. Wenn der Pivotpunkt mit SPI (S. 198) geändert wird, werden die Verfahrbereichsgrenzen für die Rotationsachsen U, V und W nicht angepasst.
Beispiel:	Senden: PLM? X Empfangen: X = 22.5 Senden: POS? X Empfangen: X = 10 Senden: PLM X 5 Senden: ERR? Empfangen: 27 - (error 27 - "Soft limit out of range") Senden: PLM? X Empfangen: X = 22.5

PLM? (Get High Position Soft Limit)

Beschreibung:	Fragt die Position der Verfahrbereichsgrenze ab, die die Obergrenze des Achsenstellwegs im geregelten Betrieb bestimmt.
Format:	PLM? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers
Antwort:	{<AxisID>"="<HighLimit> LF} wobei <HighLimit> die Position für die Obergrenze des Stellwegs in physikalischen Einheiten ist.

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
Hinweise:	Dieser Befehl ist funktionsgleich mit #3 (S. 123), der bevorzugt werden sollte, wenn der Controller zeitaufwändige Aufgaben ausführt. Die aktuelle Position der Achsen X, Y, Z, U, V und W wird aus den gemessenen Positionen der einzelnen Antriebe berechnet. Die physikalische Einheit, in der die Achsenposition angegeben wird, kann mit PUN? (S. 192) abgefragt werden.

PUN? (Get Axis Unit)

Beschreibung:	Fragt die aktuelle Einheit der Achse ab. Werden alle Argumente weggelassen, so wird die aktuelle Einheit für alle Achsen abgefragt.
Format:	PUN? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers.
Antwort:	{<AxisID>"="<string> LF} wobei <string> die aktuelle Einheit der Achse ist.
Fehlersuche:	Unzulässige Achsenkennung
Hinweis:	Für die Positionsangaben der Achsen gelten folgende Einheiten: X, Y, Z, S1, S2: Millimeter U, V, W: Grad

RBT (Reboot System)

Beschreibung:	Startet das System neu. Der Controller verhält sich wie nach dem Einschalten.
Format:	RBT
Argumente:	Keine
Antwort:	Keine

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? (Get Reference Mode)

Beschreibung: Fragt den Modus der Referenzwertbestimmung der angegebenen Achsen ab.

Format: RON? [{<AxisID>}]

Argumente: <AxisID> ist eine Achse des Controllers.

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

wobei

<ReferenceOn> der aktuell für die Achse gesetzte Modus der Referenzwertbestimmung ist

Fehlersuche: Unzulässige Achsenkennung

Hinweise: RON? gibt immer 1 zurück. Für Achsen, deren Position durch inkrementelle Sensoren gemessen wird, bedeutet das:

- Bewegungsbefehle werden nur nach erfolgreicher Referenzwertbestimmung ausgeführt.
- Die Referenzwertbestimmung muss durch eine Referenzfahrt erfolgen.

Mit dem C-886 erfolgt die Referenzfahrt zum Referenzschalter (Start mit FRF (S. 140)).

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).

Hinweis: Fragt den Parameterwert ***Data Recorder Table Rate*** (ID 0x16000000) ab.

Weitere Informationen finden Sie unter "Datenrekorder" (S. 76).

SAI? (Get List Of Current Axis Identifiers)

Beschreibung: Fragt die Achsenkennung ab.

Siehe auch "Kommandierbare Elemente" (S. 25).

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.

SCT (Set Cycle Time)

Beschreibung: Legt die Zykluszeit für das Ausführen eines Dynamikprofils fest.

Wird verwendet bei der zyklischen Übertragung von Zielpositionen, siehe auch "Bewegungen des Positionierers" (S. 29).

Format: SCT "T" <CycleTime>

Argumente: "T" ist das erforderliche Schlüsselwort für das Argument <CycleTime>

<CycleTime> ist die Zykluszeit in ms, Format: float.

Fehlersuche: Zulässiger Wertebereich überschritten

Hinweise: Der zulässige Wertebereich für <CycleTime> ist 1 bis 100000 ms; Standardwert ist 100 ms.

Die mit SCT eingestellte Zykluszeit ist nur wirksam, wenn der Parameter ***Trajectory Source*** (ID 0x19001900) den Wert 1 hat (= das Dynamikprofil wird durch aufeinander folgende MOV-Befehle festgelegt).

Die Zykluszeit wird verwendet, um während der Bewegung die Geschwindigkeit so zu berechnen, dass die vorgegebenen Punkte des Dynamikprofils jeweils genau am Ende des Zeitintervalls erreicht werden.

Wenn der Parameter ***Trajectory Execution*** (ID 0x19001901) den Wert 1 hat, ist durch die Zwischenspeicherung der Dynamikprofilpunkte sichergestellt, dass das Dynamikprofil im entsprechenden Zeittakt ausgeführt wird.

Wenn der Parameter ***Trajectory Execution*** (ID 0x19001901) den Wert 0 hat, werden die MOV-Befehle sofort nach dem Senden ausgeführt.

- Stellen Sie sicher, dass die MOV-Befehle im der Zykluszeit entsprechenden Zeittakt gesendet werden, um das Dynamikprofil einzuhalten.

Beispiel: Senden: SCT T 200

Setzt die Zykluszeit für das Ausführen eines Dynamikprofils, das durch aufeinander folgende MOV-Befehle vorgegeben wird, auf 200 ms.

SCT? (Get Cycle Time)

Beschreibung: Fragt die aktuelle Zykluszeit für das Abfahren eines definierten Dynamikprofils ab.

Format: SCT? [<T>]

Argumente: "T" dient als Schlüsselwort und kann für die Abfrage weggelassen werden.

Antwort: T=<float> LF

<float> ist die Zykluszeit in ms.

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 **Beachten Sie, dass dieser Befehl für die Einstellung hardwarespezifischer Parameter gilt. Falsche Werte können eventuell zu fehlerhaftem Betrieb oder zur Beschädigung Ihrer Hardware führen!** Mit HPA? (S. 145) erhalten Sie eine Liste der verfügbaren Parameter.
Fehlersuche:	Unzulässige Elementkennung, falsche Parameter-ID, Wert im unzulässigen Bereich, zu niedrige Befehlsebene für Schreibzugriff Parameterwerte können nur gesetzt werden, wenn sich die Achse nicht bewegt (Abfrage mit #5 (S. 124)).
Verfügbare Element-IDs und Parameter-IDs:	Das Element kann eine Achse, ein Antrieb oder das gesamte System sein; der Elementtyp ist vom Parameter abhängig. Siehe "Parameterübersicht" (S. 250) für den entsprechenden Elementtyp, bezüglich der Elementkennungen siehe "Kommandierbare Elemente" (S. 25). Die gültigen Parameter-IDs werden in der "Parameterübersicht" (S. 250) angegeben.

Beispiel:	Mit den folgenden Befehlen wechseln Sie auf Befehlsebene 1 und setzen den Namenszusatz für den C-886 auf den Wert *testfarm*, Parameter-ID im Hexadezimalformat geschrieben:
	`CCL 1 advanced`
	`SPA 1 0x0D001000 testfarm`

SPA? (Get Volatile Memory Parameters)

Beschreibung:	Fragt den Wert eines Parameters für ein angegebenes Element aus dem flüchtigen Speicher (RAM) ab. Mit HPA? (S. 145) 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
Verfügbare Element-IDs und Parameter-IDs:	Das Element kann eine Achse, ein Antrieb oder das gesamte System sein; der Elementtyp ist vom Parameter abhängig. Siehe "Parameterübersicht" (S. 250) für den entsprechenden Elementtyp, bezüglich der Elementkennungen siehe "Kommandierbare Elemente" (S. 25). Die gültigen Parameter-IDs werden in der "Parameterübersicht" (S. 250) angegeben.

SPI (Set Pivot Point)

Beschreibung:	Setzt die Pivotpunktkoordinaten im flüchtigen Speicher. Kann nur gesetzt werden, wenn für die Rotationskoordinaten der bewegten Plattform gilt: U = V = W = 0
Format:	SPI {<PPCoordinate> <Position>}
Argumente:	<PPCoordinate> ist eine Pivotpunktkoordinate, siehe unten. <Position> ist der Wert der Pivotpunktkoordinate, siehe unten.
Antwort:	Keine
Fehlersuche:	Mindestens eine der Rotationskoordinaten U, V und W ist ungleich 0

Hinweise: <PPCoordinate> kann R, S und T sein. Als Alias-Kennungen für R, S und T können auch X, Y und Z verwendet werden.

<Position> wird in mm angegeben. Die Standardwerte werden durch das aktive Betriebs-Koordinatensystem vorgegeben (nur für Koordinatensysteme der Typen KSF und ZERO).

Weitere Informationen zum Pivotpunkt siehe "Begriffserklärung" (S. 3).

Bei einer Verlegung des Pivotpunkts mit SPI ändern sich die möglichen Stellwege für die Rotationsachsen U, V und W und damit der Arbeitsraum. Folgende Werte werden jedoch **nicht** an die geänderten Stellwege angepasst:

- Antworten auf TMN? (S. 204) und TMX? (S. 205)
- Verfahrbereichsgrenzen, die mit NLM (S. 188) und PLM (S. 190) gesetzt sind

Verwenden Sie VMO? (S. 212), um abzufragen, ob die Zielposition erreicht werden kann.

Beispiel:

```
Senden: SPI?
Empfangen: R=0
S=0
T=0
Senden: SPI S 2
Senden: SPI?
Empfangen: R=0
S=2
T=0
Senden: SPI Z 2
Senden: SPI?
Empfangen: R=0
S=2
T=2
```

SPI? (Get Pivot Point)

Beschreibung: Fragt die Pivotpunktkoordinaten ab.

Format: SPI? [{<PPCoordinate>}]

Argumente: <PPCoordinate> ist eine Pivotpunktkoordinate, siehe unten

Antwort: {<PPCoordinate>"="<Position> LF}

wobei

<Position> der Wert der Pivotpunktkoordinate in physikalischen Einheiten ist.

Hinweis: <PPCoordinate> kann R, S und T sein. Als Alias-Kennungen für R, S und T können auch X, Y und Z verwendet werden.

SSL (Set Soft Limit)

Beschreibung: Aktiviert bzw. deaktiviert die Verfahrbereichsgrenzen, die mit NLM (S. 188) und PLM (S. 190) gesetzt werden.

Format: SSL {<AxisID> <SoftLimitsOn>}

Argumente: <AxisID> ist eine Achse des Controllers

<SoftLimitsOn> ist der Status der Verfahrbereichsgrenzen:

0 = Verfahrbereichsgrenzen deaktiviert
1 = Verfahrbereichsgrenzen aktiviert

Antwort: Keine

Hinweis: Für die Achsen der Bewegungsplattform des Positionierers werden die Standardwerte für SSL durch das aktive Betriebs-Koordinatensystem vorgegeben (beim Work-und-Tool-Konzept durch die Kombination aus zwei Betriebs-Koordinatensystemen).

Verfahrbereichsgrenzen können nur aktiviert/deaktiviert werden, wenn sich die Achse nicht bewegt (Abfrage mit #5 (S. 124)).

Beispiel: Senden: SSL X 1
Die Verfahrbereichsgrenzen für Achse X sind aktiviert.

Senden: SSL Y 0 Z 1 W 1
Die Verfahrbereichsgrenzen sind für die Achse Y deaktiviert und für die Achsen Z und W aktiviert.

SSL? (Get Soft Limit Status)

Beschreibung:	Fragt den Status der Verfahrbereichsgrenzen ab, die mit NLM (S. 188) und PLM (S. 190) gesetzt werden. Werden alle Argumente weggelassen, so wird der Status für alle Achsen abgefragt.
Format:	SSL? [{<AxisID>}]
Argumente:	<AxisID> ist eine Achse des Controllers
Antwort:	{<AxisID>"="<SoftLimitsOn> LF} wobei <SoftLimitsOn> der Status der Verfahrbereichsgrenzen ist: 0 = Verfahrbereichsgrenzen deaktiviert 1 = Verfahrbereichsgrenzen aktiviert
Fehlersuche:	Unzulässige Achsenkennung

SSN? (Get Device Serial Number)

Beschreibung:	Fragt die Seriennummer des C-886 ab.
Format:	SSN?
Argumente:	Keine
Antwort:	<SerialNumber> ist die Seriennummer des Geräts.

STA? (Query Status Register Value)

Beschreibung:	Fragt die Systemstatus-Information ab.
Format:	STA?
Argumente:	Keine
Antwort:	Die Antwort ist bit-codiert. Für die individuellen Codes siehe unten.

Hinweise: Dieser Befehl ist funktionsgleich mit #4 (S. 123).

Die Antwort ist die Summe der untenstehenden Codes in Hexadezimalformat. Bei der Auswertung der Antwort ist Folgendes zu beachten:

- Die Bits 14 und 15 für den Bewegungsstatus der optionalen Einzelachsen S1 und S2 werden nur dann gesetzt, wenn die Bewegung durch einen Befehl ausgelöst wurde.
- Unbelegte Bits haben den Wert 0.

Bit:	23	22	21	20	19	18	17	16
	-	-	-	-	Referenz-fahrt wird ausgeführt	Referenz-fahrt Achse S2 erfolg-reich	Referenz-fahrt Achse S1 erfolg-reich	Referenz-fahrt Posi-tionierer erfolgreich
Bit:	15	14	13	12	11	10	9	8
	Achse S2 in Bewegung	Achse S1 in Bewegung	Antrieb 6 in Bewegung	Antrieb 5 in Bewegung	Antrieb 4 in Bewegung	Antrieb 3 in Bewegung	Antrieb 2 in Bewegung	Antrieb 1 in Bewegung
Bit:	7	6	5	4	3	2	1	0
	Bewegungs-fehler Achse S2	Bewegungs-fehler Achse S1	Bewe-gungsfehler Antrieb 6	Bewe-gungsfehler Antrieb 5	Bewe-gungsfehler Antrieb 4	Bewe-gungsfehler Antrieb 3	Bewe-gungsfehler Antrieb 2	Bewe-gungsfehler Antrieb 1

Beispiel: Senden: STA?
Empfangen: 0x71804
Hinweis: Die Antwort wird im Hexadezimalformat angegeben. Sie besagt: Für Antrieb 3 wurde ein Bewegungsfehler gemeldet, die Antriebe 4 und 5 sind in Bewegung. Die Referenzfahrt des Positionierers und der optionalen Einzelachsen S1 und S2 wurde erfolgreich abgeschlossen.

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 gesetzt werden.

Die aufgezeichneten Daten können mit dem Befehl DRR? (S. 135) 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: Die Zielposition, die aus der angegebenen Sprunggröße resultiert, ist außerhalb der Grenzwerte.

Hinweise: Ein "Sprung" besteht aus einer relativen Bewegung mit der angegebenen Amplitude. Der Schritt wird relativ zu der aktuellen Position ausgeführt.

Die physikalische Einheit, in der <Amplitude> anzugeben ist, kann mit PUN? (S. 192) abgefragt werden.

Für die Achsen des Positionierers (X, Y, Z, U, V, W) gilt Folgendes:

- Vor dem Start jeder Bewegung erfolgt eine Prüfung, ob die Bewegungsplattform die kommandierte Zielposition tatsächlich erreichen kann. Mit VMO? (S. 212) können Sie abfragen, ob die Zielposition erreicht werden kann.
- Rotationen erfolgen um den Drehpunkt.
- Wenn der Parameter ***Trajectory Source*** (ID 0x19001900) auf 1 gesetzt ist, muss das Dynamikprofil durch aufeinander folgende MOV-Befehle vorgegeben werden. STE ist nicht zulässig.

Weitere Informationen siehe "Bewegungen des Positionierers" (S. 29).

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. 127).

Format: STP

Argumente: Keine

Antwort: Keine

Fehlersuche: Kommunikationsstörung

Hinweise: STP stoppt alle Achsenbewegungen, die durch Bewegungsbefehle oder Funktionsgeneratorausgabe verursacht werden, und die Referenzfahrt.

STP stoppt Makros.

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

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: Für die Achsen der bewegten Plattform des Positionierers (X, Y, Z, U, V, W) wird der Servomodus immer gemeinsam ein- oder ausgeschaltet. Daher reicht zum Setzen des Servomodus für die Achsen der bewegten Plattform die Angabe einer einzigen Achse aus, z.B.:
SVO X 1

Bewegungen der Achsen können ausschließlich bei eingeschaltetem Servomodus ausgelöst werden.

Der Servomodus kann nur ausgeschaltet werden, wenn sich die Achse nicht bewegt (Abfrage mit #5 (S. 124)).

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

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.

Hinweise: Die Stellwege in X, Y, Z, U, V, W sind voneinander abhängig. Je nach der aktuellen Position der Bewegungsplattform des Positionierers kann der tatsächlich verfügbare Stellweg für die Achsen X, Y, Z, U, V und W geringer ausfallen als in der Antwort auf TMN? angegeben. Die Antwort auf TMN? entspricht dem tatsächlich verfügbaren Stellweg einer Achse nur dann, wenn die beiden folgenden Bedingungen erfüllt sind:

- Alle anderen Achsen stehen auf Nullposition.
- Die Standardeinstellungen für die Pivotpunktkoordinaten gelten.

Bei einer Verlegung des Pivotpunkts mit SPI (S. 197) ändern sich die verfügbaren Stellwege für die Rotationsachsen U, V und W. Die Werte für die kleinsten kommandierbaren Positionen werden jedoch **nicht** an die geänderten Stellwege angepasst.

Verwenden Sie VMO? (S. 212), um abzufragen, ob eine Zielposition erreicht werden kann.

Verwenden Sie TRA? (S. 206), um die absolute Position abzufragen, die maximal kommandiert werden kann, wenn sich die Plattform des Positionierers entlang eines vorgegebenen Richtungsvektors bewegt.

Die physikalische Einheit, in der die kleinste kommandierbare Position angegeben wird, kann mit PUN? (S. 192) abgefragt werden.

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.

Hinweise: Die Stellwege in X, Y, Z, U, V, W sind voneinander abhängig. Je nach der aktuellen Position der Bewegungsplattform des Positionierers kann der tatsächlich verfügbare Stellweg für die Achsen X, Y, Z, U, V und W geringer ausfallen als in der Antwort auf TMX? angegeben. Die Antwort auf TMX? entspricht dem tatsächlich verfügbaren Stellweg einer Achse nur dann, wenn die beiden folgenden Bedingungen erfüllt sind:

- Alle anderen Achsen stehen auf Nullposition.
- Die Standardeinstellungen für die Pivotpunktkoordinaten gelten.

Bei einer Verlegung des Pivotpunkts mit SPI (S. 197) ändern sich die verfügbaren Stellwege für die Rotationsachsen U, V und W. Die Werte für die größten kommandierbaren Positionen werden jedoch **nicht** an die geänderten Stellwege angepasst.

Verwenden Sie VMO? (S. 212), um abzufragen, ob eine Zielposition erreicht werden kann.

Verwenden Sie TRA? (S. 206), um die absolute Position abzufragen, die

maximal kommandiert werden kann, wenn sich die Plattform des Positionierers entlang eines vorgegebenen Richtungsvektors bewegt.

Die physikalische Einheit, in der die größte kommandierbare Position angegeben wird, kann mit PUN? (S. 192) abgefragt werden.

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:	Die Antwort gibt den Wert des Parameters ***Data Recorder Channel Number*** (ID 0x16000300) an. Weitere Informationen finden Sie unter "Datenrekorder" (S. 76).

TRA? (Get Maximum Commandable Position For Direction Vector)

Beschreibung:	Fragt die absolute Position ab, die maximal kommandiert werden kann, wenn sich die Plattform des Positionierers entlang des Richtungsvektors bewegt, der durch die angegebenen Achsenanteile definiert wird. Die maximal kommandierbare Position wird ausgehend von der aktuellen Position berechnet und kann nur abgefragt werden, wenn sich die Plattform des Positionierers nicht bewegt. Anmerkung: „Maximal“ bezieht sich auf den Betrag des Positionswerts. Deshalb wird in dieser Beschreibung auch die größtmögliche Auslenkung in negative Richtung als „maximale“ Position bezeichnet (und nicht als „minimale“ Position).
Format:	TRA? {<AxisID> <Component>}
Argumente:	<AxisID> ist eine Achse des Controllers. Mögliche Werte: X, Y, Z, U, V, W. <Component> ist der Anteil der Achse am Richtungsvektor. Muss für mindestens eine abgefragte Achse verschieden von null sein. Kann ein negatives Vorzeichen haben. Achsen, die in der Abfrage nicht angegeben werden, haben keinen Anteil am Richtungsvektor und sind in der Antwort nicht enthalten.

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

wobei

<float> die für die Achse maximal kommandierbare absolute Position ist, wenn sich die Plattform des Positionierers entlang des angegebenen Richtungsvektors bewegt; in physikalischen Einheiten.

Hinweise: Die aktuellen Einstellungen für die Verfahrbereichsgrenzen (siehe NLM (S. 188), PLM (S. 190), SSL (S. 199)) und, falls vom aktiven Betriebs-Koordinatensystem verwendet, für den mit SPI definierten Pivotpunkt (siehe SPI (S. 197)) werden in die Berechnung einbezogen.

Die physikalische Einheit, in der die maximal kommandierbare Position angegeben wird, kann mit PUN? (S. 192) abgefragt werden.

Beispiel: Die Bewegungsplattform soll sich in Richtung des Vektors (X, Z) = (2, 4) bewegen. Mit dem Befehl PLM ist die obere Verfahrbereichsgrenze für die Achse X auf den Wert 1 gesetzt.

Senden Sie:

TRA? X 2 Z 4

Die Antwort gibt die maximale absolute Position an, die von der aktuellen Position aus in Richtung des Vektors (X, Z) = (2, 4) angefahren werden kann:

X=1.00000

Z=1.99869

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

TWG? (Get Number of Wave Generators)

Beschreibung:	Fragt die Anzahl der im Controller verfügbaren Funktionsgeneratoren ab.
Format:	TWG?
Argumente:	Keine
Antwort	<uint> ist die Anzahl der verfügbaren Funktionsgeneratoren

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. 112).

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. 112).

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. 128).

VAR? (Get Variable Values)

Beschreibung: Gibt Variablenwerte zurück.

Wird VAR? mit CPY (S. 131), JRC (S. 150), MEX (S. 184) oder WAC (S. 213) kombiniert, muss die Antwort auf VAR? ein einzelner Wert sein (und nicht mehr).

Nähere Angaben zu lokalen und globalen Variablen siehe "Variablen" (S. 112).

Format: VAR? [{<Variable>}]

Argumente: <Variable> ist der Name der abzufragenden Variablen. Nähere Angaben zu Namenskonventionen siehe "Variablen" (S. 112).

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.

Hinweis: Innerhalb eines Makros kann VAR? nur in Kombination mit CPY (S. 131), JRC (S. 150), MEX (S. 184) oder WAC (S. 213) sinnvoll verwendet werden.

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:	Der Befehl ist nur für die optionalen Einzelachsen S1 und S2 zulässig. <Velocity> muss ≥ 0 sein. Die mit VEL gesetzte Geschwindigkeit wird nur im flüchtigen Speicher (RAM) gespeichert.

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.

VER? (Get Versions Of Firmware And Drivers)

Beschreibung:	Fragt die Versionen der Firmware des C-886 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.

Beispiel: Für C-886 antwortet VER? etwa Folgendes:

```
ver?
2: (c)2016 Physik Instrumente (PI) GmbH & Co. KG,
E-873.10C885, 117004372, 02.012
3: (c)2016 Physik Instrumente (PI) GmbH & Co. KG,
E-873.10C885, 117007103, 02.012
4: (c)2016 Physik Instrumente (PI) GmbH & Co. KG,
E-873.10C885, 117004369, 02.012
5: (c)2016 Physik Instrumente (PI) GmbH & Co. KG,
E-873.10C885, 117007100, 02.012
6: (c)2016 Physik Instrumente (PI) GmbH & Co. KG,
E-873.10C885, 117007101, 02.012
7: (c)2016 Physik Instrumente (PI) GmbH & Co. KG,
E-873.10C885, 117007102, 02.012
8: (c)2014 Physik Instrumente(PI)
Karlsruhe,E-861.11C885, 000000000, 00.012
FW: V 2.3.3.186
Macro: V 0.15.0.0
OS: V #13 SMP PREEMPT Fri Sep 29 09:53:55 CEST 2017
Hexdata: V 1.0.0.0
C886:  (c)2017 Physik Instrumente(PI) Karlsruhe,
C-886.M2 Master
```

VLS (Set System Velocity)

Beschreibung: Setzt die Geschwindigkeit für die Bewegungsplattform des Positionierers.

Format: VLS <SystemVelocity>

Argumente: <SystemVelocity> ist der Geschwindigkeitswert in physikalischen Einheiten.

Antwort: Keine

Hinweise: Die Einheit für <SystemVelocity> ist mm/s.

Die Untergrenze für <SystemVelocity> ist durch die kleinste Schrittweite des Positionierers bedingt (modellabhängig) und wird durch den Wert des Parameters ***Minimum System Velocity*** (ID 0x19001501) vorgegeben. Der Parameter kann mit SPA? (S. 197) abgefragt werden.

Die Geschwindigkeit kann mit VLS nur gesetzt werden, wenn sich der Positionierer nicht bewegt (Achsen X, Y, Z, U, V, W; Abfrage mit #5 (S. 124)).

Für die optionalen Einzelachsen S1 und S2 kann die Geschwindigkeit mit VEL (S. 209) gesetzt werden.

VLS? (Get System Velocity)

Beschreibung:	Fragt die Geschwindigkeit der Bewegungsplattform des Positionierers ab, die mit VLS (S. 211) gesetzt ist.
Format:	VLS?
Argumente:	Keine
Antwort:	<SystemVelocity> ist der Geschwindigkeitswert in physikalischen Einheiten, siehe VLS.

VMO? (Virtual Move)

Beschreibung:	Prüft, ob die Bewegungsplattform des Positionierers eine vorgegebene Position von der aktuellen Position aus anfahren kann. VMO? löst keine Bewegung aus.
Format:	VMO? {<AxisID> <Position>}
Argumente:	<AxisID> ist eine Achse des Controllers, siehe unten <Position> ist ein zu prüfender Zielpositionswert
Antwort:	<uint> gibt an, ob die Bewegungsplattform die aus den angegebenen Zielpositionswerten resultierende Position anfahren kann: 0 = vorgegebene Position kann nicht angefahren werden 1 = vorgegebene Position kann angefahren werden

Hinweise: Zulässig sind die Achsen X, Y, Z, U, V, W.

VMO? prüft Folgendes:

- Liegen die Stützstellen des berechneten Profils und die Zielposition außerhalb der Stellwegsgrenzen, die mit TMN? (S. 204) und TMX? (S. 205) oder TRA? (S. 206) abgefragt werden können?
- Sind die mit NLM (S. 188) und PLM (S. 190) gesetzten Verfahrbereichsgrenzen mit SSL (S. 199) aktiviert, und wenn ja, liegen die Stützstellen und die Zielposition außerhalb dieser Verfahrbereichsgrenzen?
- Sind die einzelnen Antriebe in der Lage, die Plattform zu den notwendigen Stützstellen und zur vorgegebenen Zielposition zu bewegen?

Um eine zuverlässige Antwort zu erhalten, senden Sie VMO? erst nach einer erfolgreichen Referenzfahrt (Start mit FRF (S. 140)) und nur dann, wenn sich der Positionierer nicht bewegt (Abfrage mit #5) (S. 124).

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. 184).

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.

WAV (Set Waveform Definition)

Beschreibung:

Definiert eine Kurvenform des angegebenen Typs für die angegebene Kurventabelle.

Um eine flexible Definition zu ermöglichen, kann eine Kurvenform (Kurventabelleninhalt) durch Aneinanderreihen von "Segmenten" aufgebaut werden. Jedes Segment wird mit einem separaten WAV-Befehl definiert. Mit dem Argument <AppendWave> (siehe unten) wird ein Segment an den bestehenden Kurventabelleninhalt angehängt. Um einzelne Segmente oder deren Reihenfolge zu ändern, muss die gesamte Kurvenform Segment für Segment neu erstellt werden.

Ein Segment kann auf vordefinierten Kurvenformen basieren (siehe Argument <WaveType> unten).

Kurvenformen können nicht geändert werden, während sie vom Funktionsgenerator ausgegeben werden. Vor dem Ändern einer Kurvenform mit WAV muss die Funktionsgeneratorausgabe aus der zugehörigen Kurventabelle gestoppt werden.

Die Kurvenformwerte sind absolute Werte.

Die Dauer eines Ausgabezyklus für die Kurvenform lässt sich folgendermaßen berechnen:

Ausgabedauer = Zykluszeit des C-886 * WTR-Wert * Anzahl der Punkte

Wobei

die Zykluszeit des C-886 durch den Parameter 0x0E000200 angegeben

wird (in Sekunden)

der WTR-Wert (Ausgaberate des Funktionsgenerators) die Anzahl der C-886-Zyklen angibt, über die sich die Ausgabe eines Kurvenpunktes zeitlich erstreckt; Standard ist 1

die Anzahl der Punkte der Länge der Kurventabelle entspricht (Summe der Längen aller Segmente in dieser Tabelle)

Weitere Informationen siehe "Funktionsgenerator" (S. 79).

Format: WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>

Argumente: <WaveTableID> ist die Kurventabellenkennung.

<AppendWave> kann „X“ oder „&“ sein:
„X“ löscht die Kurventabelle und beginnt am ersten Punkt der Tabelle zu schreiben.
„&“ hängt das definierte Segment an den vorhandenen Kurventabelleninhalt an, um die Kurvenform zu verlängern.

<WaveType>: Der Typ der zum Definieren des Segments verwendeten Kurve. Dies kann einer der folgenden Kurventypen sein:
„PNT“ (benutzerdefinierte Kurve)
„SIN_P“ (invertierte Kosinuskurve)
„RAMP“ (Rampenkurve)
„LIN“ (Kurve in Form einer einzelnen Abtastzeile)

<WaveTypeParameters> steht für die Parameter der Kurve:

Für „PNT“:

<WaveStartPoint> <WaveLength> {<WavePoint>}

<WaveStartPoint>: Der Index des Startpunkts. Muss 1 sein.

<WaveLength>: Die Anzahl der in die Kurventabelle zu schreibenden Punkte (= Segmentlänge).

<WavePoint>: Der Wert eines einzelnen Punktes.

Für „SIN_P“:

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <CurveCenterPoint>

<SegLength>: Die Länge des Kurventabellensegments in Punkten. Nur die durch <SegLength> angegebene Anzahl von Punkten wird in die Kurventabelle geschrieben. Ist der Wert von <SegLength> größer als der Wert von <WaveLength>, werden die fehlenden Punkte im Segment mit dem Endpunktwert der Kurve aufgefüllt.

<Amp>: Die Amplitude der Sinuskurve.

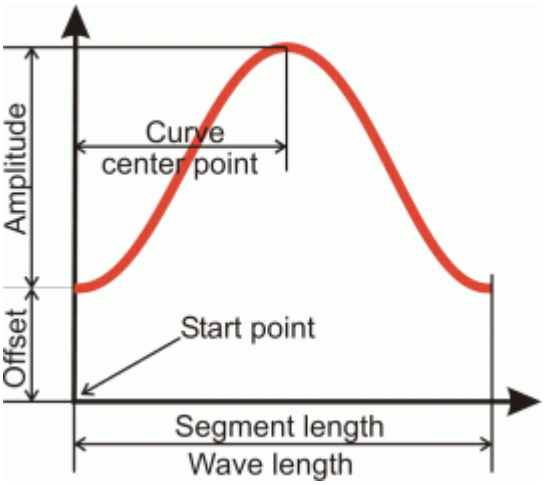
<Offset>: Der Offset der Sinuskurve.

<WaveLength>: Die Länge der Sinuskurve in Punkten.

<StartPoint>: Der Index des Startpunkts der Sinuskurve im Segment. Gibt die Phasenverschiebung an. Der kleinstmögliche Wert ist 0.

<CurveCenterPoint>: Der Index des Mittelpunkts der Sinuskurve. Bestimmt, ob die Kurve symmetrisch ist oder nicht. Der kleinstmögliche Wert ist 0.

Beispiel (weitere Beispiele siehe "Kurvenform definieren" (S. 82)):



Für „RAMP“:

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <SpeedUpDown> <CurveCenterPoint>

<SegLength>: Die Länge des Kurventabellensegments in Punkten. Nur die durch <SegLength> angegebene Anzahl von Punkten wird in die Kurventabelle geschrieben. Ist der Wert von <SegLength> größer als der Wert von <WaveLength>, werden die fehlenden Punkte im Segment mit dem Endpunktwert der Kurve aufgefüllt.

<Amp>: Die Amplitude der Rampenkurve.

<Offset>: Der Offset der Rampenkurve.

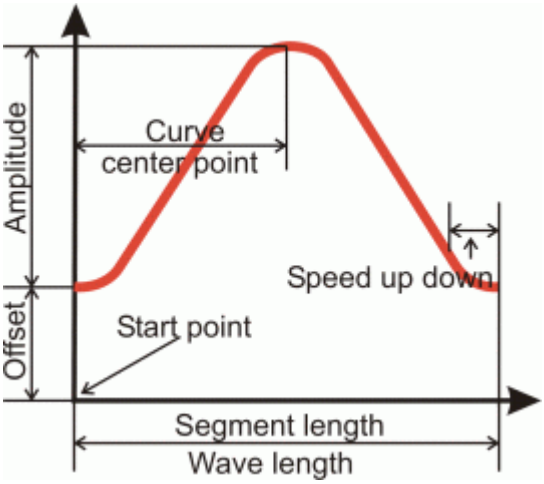
<WaveLength>: Die Länge der Rampenkurve in Punkten.

<StartPoint>: Der Index des Startpunkts der Rampenkurve im Segment. Gibt die Phasenverschiebung an. Der kleinstmögliche Wert ist 0.

<SpeedUpDown>: Die Anzahl der Punkte für Beschleunigung und Verlangsamung.

<CurveCenterPoint>: Der Index des Mittelpunkts der Rampenkurve. Bestimmt, ob die Kurve symmetrisch ist oder nicht. Der kleinstmögliche Wert ist 0.

Beispiel (weitere Beispiele siehe "Kurvenform definieren" (S. 82)):



Für „LIN“:

<SegLength> <Amp> <Offset> <WaveLength> <StartPoint> <SpeedUpDown>

<SegLength>: Die Länge des Kurventabellensegments in Punkten. Nur die durch <SegLength> angegebene Anzahl von Punkten wird in die Kurventabelle geschrieben. Ist der Wert von <SegLength> größer als der Wert von <WaveLength>, werden die fehlenden Punkte im Segment mit dem Endpunktwert der Kurve aufgefüllt.

<Amp>: Die Amplitude der Abtastzeile.

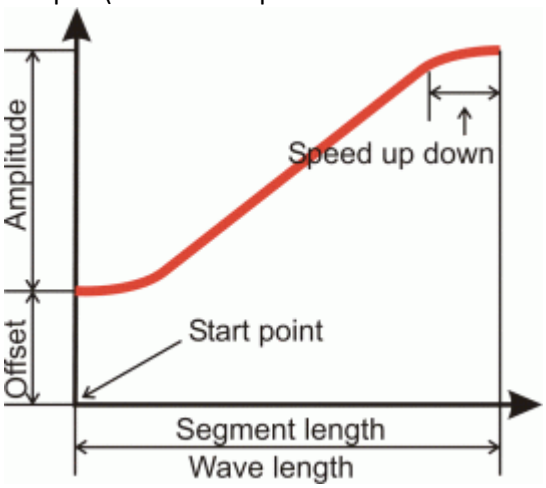
<Offset>: Der Offset der Abtastzeile.

<WaveLength>: Die Länge der Kurve (einzelne Abtastzeile) in Punkten.

<StartPoint>: Der Index des Startpunkts der Abtastzeile im Segment. Der kleinstmögliche Wert ist 0.

<SpeedUpDown>: Die Anzahl der Punkte für Beschleunigung und Verlangsamung.

Beispiel (weitere Beispiele siehe "Kurvenform definieren" (S. 82)):



Antwort: Keine

Fehlersuche: Ungültige Kurventabellenkennung

Die Gesamtanzahl der Punkte für die Kurvenform (die aus mehreren Segmenten bestehen kann) überschreitet die Anzahl der verfügbaren Speicherpunkte.

Hinweise: Bei der Definition einer Kurvenform mit WAV werden die Zielpositionen und die resultierenden Geschwindigkeiten nicht geprüft. Die Prüfung erfolgt erst während der Funktionsgeneratorausgabe.

Die Kurvenform beeinflusst die Geschwindigkeit während der Bewegungen.

Die Geschwindigkeit wird unter anderem durch folgende Faktoren begrenzt:

- Typ der Mechanik
- Kombination der zu bewegenden Achsen
- Aktuelle Einstellungen für Koordinatensystem und Drehpunkt
- Amplitude der Bewegung

➢ Definieren Sie die Kurvenform so, dass die Spezifikationen der angeschlossenen Mechanik während der Funktionsgeneratorausgabe eingehalten werden.

WAV? (Get Waveform Definition)

Beschreibung:	Fragt den Wert eines Kurvenparameters für eine angegebene Kurventabelle ab. Weitere Informationen siehe "Funktionsgenerator" (S. 79).
Format:	WAV? [{<WaveTableID> <WaveParameterID>}]
Argumente:	<WaveTableID> ist die Kurventabellenkennung. <WaveParameterID> ist die Kurvenparameter-ID: 1 = aktuelle Kurventabellenlänge als Anzahl der Punkte
Antwort:	{<WaveTableID> <WaveParameterID>"="<float> LF} wobei <float> von <WaveParameterID> abhängt; gibt die aktuelle Anzahl der Kurvenformpunkte in der Kurventabelle für <WaveParameterID> = 1 an
Fehlersuche:	Ungültige Kurventabellenkennung

WCL (Clear Wave Table Data)

Beschreibung:	Löscht den Inhalt der angegebenen Kurventabelle. Solange ein Funktionsgenerator aktiv ist, ist es nicht möglich, die damit verbundene Kurventabelle zu löschen. Weitere Informationen siehe "Funktionsgenerator" (S. 79).
Format:	WCL {<WaveTableID>}
Argumente:	<WaveTableID> ist die Kurventabellenkennung.
Antwort:	Keine

WGC (Set Number Of Wave Generator Cycles)

Beschreibung: Setzt die Anzahl der Ausgabezyklen für den angegebenen Funktionsgenerator (die Ausgabe selbst wird mit WGO (S. 220) gestartet).

Weitere Informationen siehe "Funktionsgenerator" (S. 79).

Format: WGC {<WaveGenID> <Cycles>}

Argumente: <WaveGenID> ist die Funktionsgeneratorkennung

<Cycles> ist die Anzahl der Funktionsgenerator-Ausgabezyklen.

Antwort: Keine

Hinweise: Mit <WaveGenID> = 0 werden alle Funktionsgeneratoren addressiert.

Wenn Cycles = 0, erfolgt die Ausgabe der Kurvenform ohne Begrenzung, bis sie mit WGO oder #24 (S. 127) oder STP (S. 202) oder HLT (S. 145) gestoppt wird.

WGC? (Get Number Of Wave Generator Cycles)

Beschreibung: Fragt die Anzahl der Ausgabezyklen ab, die für den angegebenen Funktionsgenerator gesetzt sind.

Weitere Informationen siehe "Funktionsgenerator" (S. 79).

Format: WGC? [{<WaveGenID>}]

Argumente: <WaveGenID> ist die Funktionsgeneratorkennung

Antwort: {<WaveGenID>"="<Cycles> LF}

wobei

<Cycles> die Anzahl der Funktionsgenerator-Ausgabezyklen ist, die mit WGC (S. 220) gesetzt sind.

WGO (Set Wave Generator Start/Stop Mode)

Beschreibung: Startet und stoppt den angegebenen Funktionsgenerator. Beim Start der Funktionsgeneratorausgabe wird automatisch ein Datenaufzeichnungszyklus gestartet.

Alle Funktionsgeneratoren, deren Ausgabe gleichzeitig aktiv sein soll, müssen im selben Befehl gestartet werden.

Für die Achsen der Bewegungsplattform des Positionierers, deren Funktionsgenerator **nicht** gestartet wurde, wird immer die zuletzt gültige Zielposition kommandiert.

Die Anzahl der Ausgabezyklen kann durch WGC (S. 220) begrenzt werden.

Mit WTR (S. 228) können Sie die einzelnen Ausgabezyklen der Kurvenform verlängern.

Die Funktionsgeneratorausgabe wird auch fortgesetzt, wenn die PC-Software, mit der sie gestartet wurde, beendet wird.

Der Befehl #9 kann zum Abfragen des aktuellen Aktivierungszustands der Funktionsgeneratoren verwendet werden. Mit WGO? können die zuletzt kommandierten Startoptionen für den Funktionsgenerator abgefragt werden. Mit WGS? (S. 223) können weitere Statusinformationen über den Funktionsgenerator abgefragt werden.

Weitere Informationen siehe "Funktionsgenerator" (S. 79).

Format: WGO {<WaveGenID> <StartMode>}

Argumente: <WaveGenID> ist die Funktionsgeneratorkennung

<StartMode> ist der Startmodus für den angegebenen Funktionsgenerator, kann im Hexadezimal- oder Dezimalformat angegeben werden. Mögliche Werte:

0: Funktionsgeneratorausgabe wird gestoppt. Sie können die Funktionsgeneratorausgabe auch mit #24 (S. 127) oder STP (S. 202) oder HLT (S. 145) stoppen.

Bit 0 = 0x1 (Hexadezimalformat) oder 1 (Dezimalformat): sofortiger Start der Funktionsgeneratorausgabe, synchronisiert durch Servozyklus

Antwort: Keine

Fehlersuche: Ungültige Funktionsgeneratorkennung

Mit dem Funktionsgenerator ist keine Kurventabelle verbunden. Verbinden Sie mit WSL (S. 227) eine Kurventabelle.

Wenn die Funktionsgeneratorausgabe aktiv ist, sind Befehle zum Starten oder Konfigurieren von Bewegungen sowie das Ausführen entsprechender Makros **nicht** zulässig.

Während der Funktionsgeneratorausgabe prüft der C-886 ständig, ob die Bewegung noch möglich ist. In folgenden Fällen stoppt der C-886 die Bewegung abrupt und setzt einen Fehlercode:

- Die auszugebenden Zielpositionen können nicht erreicht werden.
- Die erforderliche Geschwindigkeit kann nicht erreicht werden.
- Die Bewegung würde eine Kollision verursachen.

➢ Fragen Sie mit dem Befehl `WGS?` den aktuellen Status der Funktionsgeneratoren ab, insbesondere den Index der Kurvenformpunkte, an denen ein Fehler aufgetreten ist.

➢ Fragen Sie mit dem Befehl `ERR?` (S. 139) den Fehlercode des zuletzt aufgetretenen Fehlers ab.

WGO? (Get Wave Generator Start/Stop Mode)

Beschreibung:	Fragt den Start-/Stoppmodus des angegebenen Funktionsgenerators ab. Weitere Informationen siehe "Funktionsgenerator" (S. 79).
Format:	WGO? [{<WaveGenID>}]
Argumente:	<WaveGenID> ist die Funktionsgeneratorkennung
Antwort:	{<WaveGenID>"="<StartMode> LF} wobei <StartMode> der zuletzt kommandierte Startmodus des Funktionsgenerators im Dezimalformat ist.Weitere Informationen siehe WGO (S. 220).
Hinweise:	<StartMode> ist nicht nur 0, wenn `WGO <WaveGenID> 0` gesendet wurde, sondern auch, wenn die Funktionsgeneratorausgabe geendet hat oder mit #24 (S. 127) oder STP oder HLT (S. 145) gestoppt wurde.

WGR (Starts Recording In Sync With Wave Generator)

Beschreibung:	Startet die Datenaufzeichnung, wenn der Funktionsgenerator aktiv ist. Weitere Informationen siehe "Funktionsgenerator" (S. 79) und "Datenrekorder" (S. 76).
Format:	WGR
Argumente:	Keine

Antwort: Keine

Hinweise: Nach dem Senden von WGR startet die Datenaufzeichnung mit dem nächsten Ausgabezyklus des Funktionsgenerators.

Die aufgezeichneten Daten können mit DRR? (S. 135) gelesen werden.

Das Starten der Funktionsgeneratorausgabe mit WGO (S. 220) startet gleichzeitig einen ersten Datenaufzeichnungszyklus.

Weitere Triggeroptionen zum Starten der Datenaufzeichnung siehe DRT (S. 138).

WGS? (Get Status Information On Wave Generator)

Beschreibung: Fragt Statusinformationen über den angegebenen Funktionsgenerator ab.

Weitere Informationen siehe "Funktionsgenerator" (S. 79).

Format: WGS? [<WaveGenID> [<ItemID>]]

Argumente: <WaveGenID> ist die Funktionsgeneratorkennung

<ItemID> ist die Kennung einer abzufragenden Größe. Mögliche Kennungen:
STATUS
Fragt den Status des Funktionsgenerators ab.
Mögliche Antwortwerte, in hexadezimalem Format:
0x0 = Funktionsgenerator nicht aktiv (Ausgabe läuft nicht)
0x1 = Funktionsgenerator aktiv (Ausgabe läuft)
ITERATIONS
Fragt die Anzahl der Ausgabezyklen seit dem letzten Start des Funktionsgenerators ab.
Das Stoppen des Funktionsgenerators hält den Zähler an. Das Zurücksetzen des Zählers erfolgt beim erneuten Starten des Funktionsgenerators mit dem Befehl WGO.
ERRORTYPE
Fragt den Fehlercode des zuletzt während der Ausgabe aufgetretenen Fehlers ab.
Mögliche Antwortwerte:
0 = No error occurred
7 = Position out of limits
8 = Velocity out of limits
91 = Move not possible, would cause collision
ERRORINDEX
Fragt den Index des Kurvenformpunktes ab, an dem der Fehler aufgetreten ist.

Antwort: {<WaveGenID> <ItemID>"="<Value> LF}

wobei

<Value> den aktuellen Wert der abgefragten Größe angibt.

WMS? (Get Maximum Number of Wave Table Points)

Description: Fragt die Anzahl der verfügbaren Speicherpunkte für die angegebene Kurventabelle ab.

Format: WMS? [{<WaveTableID>}]

Arguments: <WaveTableID> ist die Kurventabellenkennung.

Response {<WaveTableID>"="<NumberOfPoints> LF}

wobei

<NumberOfPoints> ist die Anzahl der Speicherpunkte, die für die Kurventabelle verfügbar sind (Summe der bereits mit WAV (S. 214) definierten Punkte und der noch ungenutzten Punkte).

WPA (Save Parameters To Non-Volatile Memory)

Beschreibung: Schreibt die aktuellen Einstellungen aus dem flüchtigen in den permanenten Speicher.

Die mit WPA gespeicherten Einstellungen werden beim Einschalten oder Neustart des C-886 automatisch aus dem permanenten Speicher in den flüchtigen Speicher geladen.

Hinweis: Fehlerhafte Einstellungen können zur Fehlfunktion des Systems führen. Vergewissern Sie sich, dass die aktuellen Einstellungen korrekt sind, bevor Sie den Befehl WPA ausführen.

Einstellungen im flüchtigen Speicher, die nicht mit WPA gespeichert wurden, gehen verloren, wenn der C-886 ausgeschaltet oder neugestartet wird bzw. wenn Einstellungen zurückgesetzt werden.

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: Wenn Parametereinstellungen gespeichert werden sollen:

- Parameterwerte können im flüchtigen Speicher mit dem Befehl SPA (S. 196) geändert werden.
- Ein Element kann eine Achse, ein Antrieb oder das gesamte System sein. Der Elementtyp hängt vom Parameter ab. Weitere Informationen siehe "Anpassen von Einstellungen" (S. 247).
- Mit der Abfrage HPA? (S. 145) erhalten Sie eine Liste aller verfügbaren Parameter. Gültige Parameter-IDs finden Sie auch in der Parameterübersicht (S. 250).

Neben den Einstellungen der Parameter können mit WPA die Einstellungen für Koordinatensysteme in den permanenten Speicher geschrieben werden (Details siehe untenstehende Tabelle).
Das Speichern mit WPA überschreibt **nicht** die Werkseinstellungen, die mit DPA (S. 133) wiederhergestellt werden können.

Hinweis: Vermeiden Sie es, den C-886 während der WPA-Prozedur auszuschalten.

Gültige Kennwörter zum Schreiben in den permanenten Speicher:

100	Speichert die aktuell gültigen Werte aller Parameter und die aktuell gültigen Einstellungen für Koordinatensysteme (Details siehe Kennwort SKS)
101	Speichert die aktuell gültigen Parameterwerte. Die Angabe von <ItemID> und <PamID> ist optional.
SKS	Die Angabe von <ItemID> und <PamID> entfällt bei Verwendung des Passworts SKS. Speichert die aktuell gültigen Einstellungen für Koordinatensysteme: ▪ Eigenschaften der im flüchtigen Speicher vorhandenen Koordinatensysteme und Kombinationen von Koordinatensystemen, siehe KLS? und KLC? ▪ Aktivierungszustand von Koordinatensystemen, siehe KEN ▪ Verkettung von Koordinatensystemen, siehe KLN Wenn ZERO aktiv ist: Die aktuellen Werte für NLM, PLM, SSL und SPI werden nicht gespeichert. Damit wird sichergestellt, dass KEN ZERO die Werkseinstellungen für das Betriebs-Koordinatensystem wieder im vollen Umfang aktiviert.

WSL (Set Connection Of Wave Table To Wave Generator)

Beschreibung: Auswahl der Kurventabelle: Verbindet eine Kurventabelle mit einem Funktionsgenerator oder trennt die Verbindung des ausgewählten Generators zu einer Kurventabelle.

Zwei oder mehr Generatoren können mit derselben Kurventabelle verbunden sein, ein Generator kann jedoch nicht mit mehreren Kurventabellen verbunden sein.

Das Löschen des Kurventabelleninhalts mit WCL (S. 219) hat keine Auswirkung auf die WSL-Einstellungen.

Solange ein Funktionsgenerator aktiv ist, ist es nicht möglich, seine Kurventabellenverbindung zu ändern.

Weitere Informationen siehe "Funktionsgenerator" (S. 79).

Format: WSL {<WaveGenID> <WaveTableID>}

Argumente: <WaveGenID> ist die Funktionsgeneratorkennung

<WaveTableID> ist die Kurventabellenkennung. Wenn <WaveTableID> = 0, wird die Verbindung des ausgewählten Generators zu einer Kurventabelle getrennt.

Antwort: Keine

WSL? (Get Connection Of Wave Table To Wave Generator)

Beschreibung: Fragt die aktuellen Einstellungen der Kurventabellenverbindung für den angegebenen Funktionsgenerator ab.

Weitere Informationen siehe "Funktionsgenerator" (S. 79).

Format: WSL? [{<WaveGenID>}]

Argumente: <WaveGenID> ist die Funktionsgeneratorkennung

Antwort: {<WaveGenID>"="<WaveTableID> LF}

wobei

<WaveTableID> die Kurventabellenkennung ist. Wenn <WaveTableID> = 0, ist keine Kurventabelle mit dem Funktionsgenerator verbunden.

WTR (Set Wave Generator Table Rate)

Beschreibung:	Setzt die Ausgaberate des Funktionsgenerators und den Interpolationstyp.
Format:	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}
Argumente:	<WaveGenID> ist die Funktionsgeneratorkennung. Nähere Angaben siehe unten. <WaveTableRate> ist Ausgaberate des Funktionsgenerators (Einheit: Anzahl der Servozyklen); muss ein Ganzzahlwert sein, der größer als null ist <InterpolationType> Verfügbare Interpolationstypen siehe unten.
Antwort:	Keine
Hinweise:	Für die einzelnen Funktionsgeneratoren des C-886 können unterschiedliche Ausgaberaten eingestellt werden. Die Ausgaberate wird für alle Funktionsgeneratoren auf denselben Wert gesetzt, wenn <WaveGenID> den Wert null hat. Mit WTR können die einzelnen Ausgabezyklen der Kurvenform verlängert werden. Die Dauer eines Ausgabezyklus für die Kurvenform kann wie folgt berechnet werden: Ausgabedauer = Zykluszeit des C-886 * WTR-Wert * Anzahl der Punkte wobei die Zykluszeit des C-886 durch den Parameter 0x0E000200 angegeben wird (in Sekunden) der WTR-Wert die Anzahl der C-886-Zyklen angibt, über die sich die Ausgabe eines Kurvenpunktes zeitlich erstreckt; Standard ist 1 die Anzahl der Punkte der Länge der Kurvenform (d. h. der Länge der Kurventabelle) entspricht WTR setzt auch den Interpolationstyp, der für die Funktionsgeneratorausgabe verwendet werden soll. Wenn die Ausgaberate des Funktionsgenerators größer als 1 ist, trägt Interpolation zur Vermeidung plötzlicher Positionssprünge einer durch den Funktionsgenerator gesteuerten Achse bei. Weitere Informationen siehe "Funktionsgenerator" (S. 79). Ein Anwendungsbeispiel finden Sie unter "Funktionsgenerator konfigurieren" (S. 90).

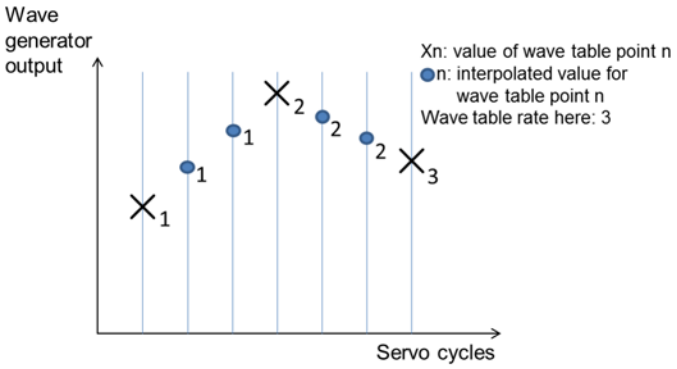
Verfügbare Interpolationstypen:

Folgende Interpolationstypen stehen zur Verfügung:
0 = keine Interpolation
1 = Gerade (Standard)

Beispiele:

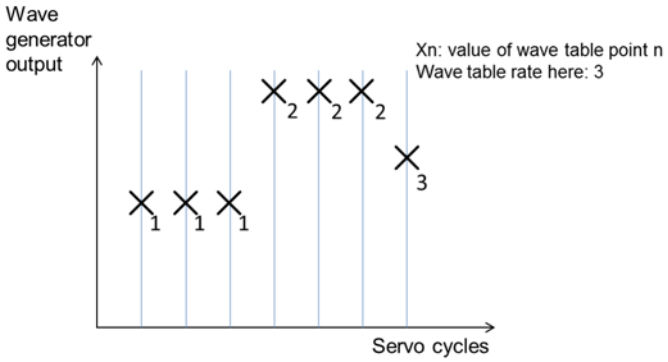
Interpolationstyp: Gerade (1; Standard)
Die Ausgaberate für Funktionsgenerator 5 wird auf 3 gesetzt, mit linearer Interpolation:

WTR 5 3 1



Interpolationstyp: keine Interpolation (0)
Die Ausgaberate für Funktionsgenerator 5 wird auf 3 gesetzt, ohne Interpolation:

WTR 5 3 0



WTR? (Get Wave Generator Table Rate)

Beschreibung:	Fragt die aktuelle Ausgaberate des Funktionsgenerators und den verwendeten Interpolationstyp ab. Weitere Informationen siehe "Funktionsgenerator" (S. 79). Ein Anwendungsbeispiel finden Sie unter "Funktionsgenerator konfigurieren".
Format:	WTR? [{<WaveGenID>}]
Argumente:	<WaveGenID> ist die Funktionsgeneratorkennung
Antwort:	{<WaveGenID>"="<WaveTableRate> <InterpolationType> LF} wobei <WaveTableRate> die Ausgaberate des Funktionsgenerators ist (Einheit: Anzahl der Servozyklen) <InterpolationType> der Interpolationstyp ist, der auf Ausgaben zwischen Kurventabellenpunkten angewendet wird, wenn die Ausgaberate größer als 1 ist. Verfügbare Interpolationstypen siehe WTR (S. 228).

8.5 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_SERVO_ON	Open-loop commands (SVA, 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 enabled.
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 LabVIEW driver 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?
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
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 answer was received while executing WAC/MEX/JRC/...
1011	PI_CMD_NOT_VALID_IN_MACRO_MODE	Command not valid during macro execution
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
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
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 can not 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

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
-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.

9 Anpassen von Einstellungen

In diesem Kapitel

Überblick über die Einstellungen des C-886 247
Parameterwerte des C-886-Master-Moduls ändern 248
Parameterwerte in Textdatei sichern 249
Parameterübersicht 250

9.1 Überblick über die Einstellungen des C-886

Für den C-886 können verschiedene Einstellungen als Standardwerte im permanenten Speicher hinterlegt werden, so dass sie beim Ausschalten oder Neustart des C-886 erhalten bleiben.

Einstellungen des Master-Moduls

Die Einstellungen des Master-Moduls bestimmen zentrale Eigenschaften und Funktionen des Systems. Die Standardwerte für die Einstellungen des Master-Moduls stammen aus verschiedenen Quellen:

- Konfigurationsdateien: nur für den Kundendienst (S. 267) zugänglich
- Schnittstellenparameter: Anpassung im permanenten Speicher mit dem Befehl `IFS` (S. 147) möglich, weitere Informationen in "Kommunikation über TCP/IP-Schnittstelle herstellen" (S. 57) und "Kommunikation über USB-Schnittstelle herstellen" (S. 64)
- Befehle für die Arbeit mit anwenderdefinierten Koordinatensystemen: weitere Informationen in "Koordinatensysteme" (S. 35)
- Parameter, die mit dem Befehl `SPA` (S. 196) geändert und mit dem Befehl `WPA` (S. 224) gespeichert werden können. Diese 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 auf die Parameter wird durch Befehlsebenen (S. 129) festgelegt.

  Weitere Informationen in "Parameterwerte des C-886-Master-Moduls ändern" (S. 248).

Einstellungen der Slave-Module für den parallelkinematischen Positionierer

Die Einstellungen der Slave-Module für den parallelkinematischen Positionierer sind bei Auslieferung des C-886 für den verwendeten Positionierer konfiguriert. Der Zugriff auf die Einstellungen der Module darf nur nach Rücksprache mit PI erfolgen.

Einstellungen der Slave-Module für die optionalen Einzelachsen

Die Einstellungen der Slave-Module für die optionalen Einzelachsen können durch den Anwender konfiguriert werden, weitere Informationen in "Slave-Module für Einzelachsen konfigurieren" (S. 66) und in den Handbüchern der entsprechenden Module (S. 6).

9.2 Parameterwerte des C-886-Master-Moduls ändern

Jeder Parameter ist sowohl im flüchtigen als auch im permanenten Speicher des C-886-Master-Moduls vorhanden. Die Werte im permanenten Speicher werden als Standardwerte beim Einschalten oder Neustart des C-886 in den flüchtigen Speicher geladen. Die Werte im flüchtigen Speicher bestimmen das aktuelle Verhalten des Systems.

Verfügbare Parameter ermitteln

- Senden Sie den Befehl `HPA?` (S. 145), um eine Liste aller verfügbaren Parameter des Master-Moduls mit Kurzbeschreibung zu erhalten.

Parameterwerte im flüchtigen Speicher ändern

INFORMATION

Das Ändern von Parameterwerten kann zu unerwünschten Ergebnissen führen.

- Legen Sie vor dem Ändern der Parametereinstellungen des C-886 eine Sicherungskopie auf dem PC an, siehe "Parameterwerte in Textdatei sichern" (S. 249). Sie können dann jederzeit die Originaleinstellungen wiederherstellen.

- Senden Sie den Befehl `SPA?` (S. 197), um eine Liste der Parameterwerte im flüchtigen Speicher des Master-Moduls zu erhalten.
- Ändern Sie die Parameterwerte im flüchtigen Speicher:
  a) Wenn für den Schreibzugriff auf die Parameterwerte notwendig, senden Sie den Befehl `CCL 1 advanced`, um auf die Befehlsebene 1 zu wechseln.
  b) Ändern Sie die Parameterwerte im flüchtigen Speicher des Master-Moduls mit dem Befehl `SPA` (S. 196).

Parameterwerte vom flüchtigen Speicher in den permanenten Speicher schreiben

INFORMATION

Für das Speichern von Parameterwerten im permanenten Speicher mit dem Befehl `WPA` ist die Eingabe eines Kennworts erforderlich. Verwendbare Kennwörter:

100	Speichert die aktuell gültigen Werte aller Parameter und die aktuell gültigen Einstellungen für Koordinatensysteme
101	Speichert die aktuell gültigen Werte aller Parameter. Parameter können einzeln ausgewählt werden.
SKS	Speichert die aktuell gültigen Einstellungen für Koordinatensysteme

➢ Schreiben Sie die aktuellen Parameterwerte mit dem Befehl `WPA` (S. 224) in den permanenten Speicher des Master-Moduls.

Alternatives Vorgehen, wenn Sie mit PIMikroMove® arbeiten:

a) Wählen Sie im Hauptfenster von PIMikroMove® den Menüeintrag ***C-886 > Save parameters to non-volatile memory***. Der Dialog ***Save Parameters to Non-Volatile Memory*** öffnet sich.

b) Geben Sie in das Auswahlfeld im Dialog ***Save Parameters to Non-Volatile Memory*** entweder das passende Kennwort ein, oder wählen Sie den passenden Eintrag.

c) Klicken Sie auf ***OK***, um das Speichern auszuführen und den Dialog zu schließen.

9.3 Parameterwerte in Textdatei sichern

INFORMATION

Um die Einstellungen der Slave-Module für die optionalen Einzelachsen zu sichern, ist die direkte Kommunikation mit den Slave-Modulen erforderlich, siehe "Slave-Module für Einzelachsen konfigurieren" (S. 66) und "Kennungen für direkten Zugriff auf Slave-Module" (S. 26). Weitere Informationen zu den Einstellungen der Slave-Module finden Sie in den Handbüchern der entsprechenden Module (S. 6).

Voraussetzungen

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

✓ Sie haben die Kommunikation zwischen dem C-886 und dem PC mit PIMikroMove® oder PITerminal über TCP/IP (S. 61) oder USB (S. 64) hergestellt.

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 mit dem Befehl `SPA?` die aktuellen Parameterwerte ab.

   Die Antwort enthält standardmäßig die Parameterwerte des C-886-Master-Moduls.

   Wenn Sie in PIMikroMove® direkt mit Slave-Modulen kommunizieren (S. 66), enthält die Antwort die Parameterwerte für das Slave-Modul, das im Fenster zum Senden von Befehlen ausgewählt ist.
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.

9.4 Parameterübersicht

INFORMATION

Der Schreibzugriff auf die Parameter des C-886 ist durch Befehlsebenen festgelegt. Nach dem Einschalten oder Neustart des Controllers ist die aktive Befehlsebene immer 0. Für bestimmte Parameter ist der Schreibzugriff nur auf der Befehlsebene 1 zugelassen. Auf Befehlsebenen > 1 besteht Schreibzugriff nur für PI-Servicepersonal.

- Wenn notwendig, senden Sie den Befehl `CCL 1 advanced` oder geben Sie das Kennwort `advanced` ein, um auf die Befehlsebene 1 zu wechseln.
- Wenn Sie Probleme mit Parametern der Befehlsebene 2 oder höher haben, wenden Sie sich an den Kundendienst (S. 267).

Parameter-ID (hexadezimal)	Befehlsebene für den Schreibzugriff	Betroffener Elementtyp	Datentyp	Parametername (Einheit)	Beschreibung
0x3C	0	Positionierer-Antrieb	CHAR	Stage Name	Name des parallelkinematischen Positionierers, für den das Slave-Modul des Antriebs konfiguriert ist

Parameter-ID (hexadezimal)	Befehlsebene für den Schreibzugriff	Betroffener Elementtyp	Datentyp	Parametername (Einheit)	Beschreibung
0x72	0	System	INT	Ignore Macro Error?	Makrofehler ignorieren? Legt fest, ob das Controllermakro gestoppt wird, wenn bei dessen Ausführung ein Fehler auftritt. 0 = Bei Fehler Makro anhalten (Standard) 1 = Makrofehler ignorieren
0x07030401	0	Positionierer-Achse	INT	Behaviour After Reference Move	Verhalten der Bewegungsplattform des Positionierers nach der Referenzfahrt Details siehe "Bewegungen des Positionierers" (S. 29).
0x07030402	0	Positionierer-Achse	FLOAT	Target For Motion After Reference Move	
0x0D000700	2	System	CHAR	Device Name	Produktbezeichnung des C-886
0x0D001000	1	System	CHAR	Customer Device Name	Frei wählbarer Namenszusatz für den C-886 Kann durch den Anwender angepasst werden, um mehrere C-886 im selben Netzwerk oder am selben PC unterscheiden zu können. Wird z. B. beim Herstellen der Kommunikation über TCP/IP in der Liste der gefundenen Controller angezeigt.
0x0E000200	3	System	FLOAT	Servo Update Time	Zykluszeit des C-886 in Sekunden
0x0E000900	0	System	INT	Pulse Length Factor	Wird mit der Zyklusdauer multipliziert Produkt entspricht der Pulsweite des Impulses Standard: 5
0x13000004	3	System	INT	Max Wave Points	Gesamtanzahl der für Kurvenformen verfügbaren Punkte Details siehe "Funktionsgenerator" (S. 79).
0x1300010A	3	System	INT	Number Of Wave Tables	Anzahl der Kurventabellen zum Speichern von Kurvenformen Details siehe "Funktionsgenerator" (S. 79).
0x16000000	3	System	INT	Data Recorder Table Rate	Aufzeichnungsrate des Datenrekorders (= 1)

Parameter-ID (hexadezimal)	Befehlsebene für den Schreibzugriff	Betroffener Elementtyp	Datentyp	Parametername (Einheit)	Beschreibung
0x16000100	3	System	INT	Max. Number of Data Recorder Channels	Maximale Anzahl der Datenrekordertabellen
0x16000200	3	System	INT	Data Recorder Max. Points	Maximale Anzahl aller Punkte der Datenrekordertabellen
0x16000201	0	System	INT	Data Recorder Points Per Table	Anzahl der Punkte pro Datenrekordertabelle 1 bis 8192 Standard: 8192
0x16000300	3	System	INT	Channel Number	Anzahl der Datenrekordertabellen
0x19001500	3	System	FLOAT	Maximum System Velocity (mm/s)	Maximale Systemgeschwindigkeit Siehe Spezifikationen im Benutzerhandbuch des Positionierers und "Bewegungen des Positionierers" (S. 29).
0x19001501	3	System	FLOAT	Minimum System Velocity (mm/s)	Minimale Systemgeschwindigkeit Details siehe "Bewegungen des Positionierers" (S. 29).
0x19001502	3	System	FLOAT	Maximum System Acceleration (mm/s^2)	Maximale Systembeschleunigung Details siehe "Bewegungen des Positionierers" (S. 29).
0x19001504	3	System	FLOAT	Path Control Step Size (mm)	Schrittweite für die Berechnung des Dynamikprofils der Plattformbewegung Details siehe "Bewegungen des Positionierers" (S. 29).
0x19001510	0	System	FLOAT	Trajectory Velocity (Phys. Unit/s)	Geschwindigkeit, Beschleunigung und Ruck für die Bewegungsplattform des Positionierers Details siehe "Profilgenerator für Punkt-zu-Punkt-Bewegungen" (S. 31).
0x19001511	0	System	FLOAT	Trajectory Acceleration (Phys. Unit/s)	
0x19001512		System	FLOAT	Trajectory Jerk (Phys. Unit/s)	

Parameter-ID (hexadezimal)	Befehlsebene für den Schreibzugriff	Betroffener Elementtyp	Datentyp	Parametername (Einheit)	Beschreibung
0x19001800	0	System	INT	Coordination Mode	Koordinationsmodus 0 = Multiachsenmodus 1 = Hexapod-Modus (= Parallelkinematik-Modus; Standard) Der Parameterwert darf nicht geändert werden.
0x19001900	0	System	INT	Trajectory Source	Quelle des Dynamikprofils für MOV-Befehle 0 = Dynamikprofil wird durch Profilgenerator festgelegt (Standard) 1 = Dynamikprofil wird durch aufeinander folgende `MOV`-Befehle festgelegt Details siehe "Bewegungen des Positionierers" (S. 29).
0x19001901	0	System	INT	Trajectory Execution	Ausführung des Dynamikprofils 0 = Dynamikprofil wird sofort ausgeführt (Standard) 1 = Dynamikprofil wird vor Ausführung in einem Zwischenspeicher abgelegt Details siehe "Zyklische Übertragung von Zielpositionen" (S. 33).
0x19001902	3	System	INT	Maximum Number of Trajectory Points	Maximale Anzahl der Dynamikprofilpunkte Details siehe "Zyklische Übertragung von Zielpositionen" (S. 33).
0x19001903	0	System	INT	Threshold for Trajectory Execution	Schwellenwert für Ausführung des Dynamikprofils Details siehe "Zyklische Übertragung von Zielpositionen" (S. 33).
0x19001904	3	System	INT	Current Number of Trajectory Points	Zeigt die aktuelle Anzahl der Dynamikprofilpunkte im Zwischenspeicher an. Details siehe "Zyklische Übertragung von Zielpositionen" (S. 33).

Parameter-ID (hexadezimal)	Befehlsebene für den Schreibzugriff	Betroffener Elementtyp	Datentyp	Parametername (Einheit)	Beschreibung
0x19002001	0	System	INT	Start All Hexapod Wave Generators	Startverhalten der Funktionsgeneratoren für die Achsen der Bewegungsplattform des Positionierers (X, Y, Z, U, V, W). 0 = Jeder Funktionsgenerator, der gestartet werden soll, muss im WGO-Befehl addressiert werden (Standard). 1 = Das Starten eines Funktionsgenerators startet alle Funktionsgeneratoren, die mit einer Kurventabelle verbunden sind. Diese Option ist nur aus Kompatibilitätsgründen vorhanden.
0x19003000	2	Positionierer-Antrieb	FLOAT	Hexapod Position Vector A0_X	Positions- und Richtungsvektoren für die Antriebe und Gelenke des Positionierers
0x19003001	2	Positionierer-Antrieb	FLOAT	Hexapod Position Vector A0_Y	
0x19003002	2	Positionierer-Antrieb	FLOAT	Hexapod Position Vector A0_Z	
0x19003010	2	Positionierer-Antrieb	FLOAT	Hexapod Joint Vector A0_X	
0x19003011	2	Positionierer-Antrieb	FLOAT	Hexapod Joint Vector A0_Y	
0x19003012	2	Positionierer-Antrieb	FLOAT	Hexapod Joint Vector A0_Z	
0x19003100	2	Positionierer-Antrieb	FLOAT	Hexapod Position Vector B0_X	
0x19003101	2	Positionierer-Antrieb	FLOAT	Hexapod Position Vector B0_Y	
0x19003102	2	Positionierer-Antrieb	FLOAT	Hexapod Position Vector B0_Z	
0x19003110	2	Positionierer-Antrieb	FLOAT	Hexapod Joint Vector B0_X	

Parameter-ID (hexadezimal)	Befehlsebene für den Schreibzugriff	Betroffener Elementtyp	Datentyp	Parametername (Einheit)	Beschreibung
0x19003111	2	Positionierer-Antrieb	FLOAT	Hexapod Joint Vector B0_Y	
0x19003112	2	Positionierer-Antrieb	FLOAT	Hexapod Joint Vector B0_Z	

10 Wartung

In diesem Kapitel

C-886 reinigen 257
Firmware aktualisieren 258

10.1 C-886 reinigen

HINWEIS

Kurzschlüsse oder Überschläge!

Der C-886 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-886 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-886 mit einem Tuch, das leicht mit einem milden Reinigungs- oder Desinfektionsmittel angefeuchtet wurde.

10.2 Firmware aktualisieren

HINWEIS

Schäden durch ungewollte Positionsänderungen!

Die maximale Haltekraft bei ausgeschaltetem Servomodus basiert auf der Selbsthemmung der Antriebe und kann geringer ausfallen als die maximale Belastbarkeit bei eingeschaltetem Servomodus (siehe Handbuch des Positionierers).

Wenn die tatsächliche Belastung des Positionierers die auf der Selbsthemmung der Antriebe basierende maximale Haltekraft überschreitet, können in folgenden Fällen ungewollte Positionsänderungen des Positionierers auftreten:

- Ausschalten des C-886
- Neustart des C-886
- Ausschalten des Servomodus für die Achsen derBewegungsplattform des Positionierers

Dadurch sind Kollisionen zwischen Positionierer, zu bewegender Last und Umgebung möglich. Kollisionen können den Positionierer, die zu bewegende Last oder die Umgebung beschädigen.

➢ Stellen Sie sicher, dass die tatsächliche Belastung der Bewegungsplattform des Positionierers die auf der Selbsthemmung der Antriebe basierende maximale Haltekraft nicht überschreitet, bevor Sie den Servomodus ausschalten, den C-886 neu starten oder ausschalten.

INFORMATION

Die Firmware des C-886 besteht aus mehreren Komponenten, die separat aktualisiert werden können.

INFORMATION

Die Firmware der Slave-Module des C-886 kann **nicht** mit dem PIFirmwareManager aktualisiert werden.

➢ Wenn eine Aktualisierung der Firmware der Slave-Module erforderlich ist, kontaktieren Sie unseren Kundendienst (S. 267).

INFORMATION

Für die Aktualisierung der Firmware müssen C-886 und PC über die TCP/IP-Schnittstelle kommunizieren.

Ident-Bezeichnung des C-886

Der Befehl `*IDN?` liest die Ident-Bezeichnung des C-886 aus, die unter anderem die Version der Firmware des Master-Moduls enthält.

Beispiel: `(c)2011-2017 Physik Instrumente (PI) GmbH & Co. KG,C-886,117567891,2.3.3.30`

- `C-886` Gerätebezeichnung
- `117567891` Seriennummer des C-886, enthält verschlüsselt das Herstellungsdatum
- `2.3.3.30` Version der Firmware des Master-Moduls (Firmwarekomponente FW)

Firmwarekomponenten

Die Versionen der Firmwarekomponenten können mit dem Befehl `VER?` ausgelesen werden.

Beispiel:

- Ident-Bezeichnungen aller Slave-Module des C-886, enthalten unter anderem jeweils die Version der Firmware des Slave-Moduls:
  - `2: (c)2016 Physik Instrumente (PI) GmbH & Co. KG, E-873.10C885, 117004372, 02.012`
  - `3: (c)2016 Physik Instrumente (PI) GmbH & Co. KG, E-873.10C885, 117007103, 02.012`
  - `4: (c)2016 Physik Instrumente (PI) GmbH & Co. KG, E-873.10C885, 117004369, 02.012`
  - `5: (c)2016 Physik Instrumente (PI) GmbH & Co. KG, E-873.10C885, 117007100, 02.012`
  - `6: (c)2016 Physik Instrumente (PI) GmbH & Co. KG, E-873.10C885, 117007101, 02.012`
  - `7: (c)2016 Physik Instrumente (PI) GmbH & Co. KG, E-873.10C885, 117007102, 02.012`
  - `8: (c)2014 Physik Instrumente(PI) Karlsruhe,E-861.11C885, 000000000, 00.012`
- `FW: V 2.3.3.30` Version der Firmware des Master-Moduls
- `Macro: V 0.15.0.0` Version der Makrofunktionalität
- `OS: V #13 SMP PREEMPT Fri Sep 29 09:53:55 CEST 2017` Version des Betriebssystems des C-886
- `Hexdata: V 1.0.0.0` Version der Konfigurationsdatei mit den Geometriedaten für den parallelkinematischen Positionierer
- `C886:  (c)2017 Physik Instrumente(PI) Karlsruhe, C-885.M2 Master` Hardware des Master-Moduls

Aktuelle Firmware des C-886 beziehen

➢ Wenden Sie sich an unseren Kundendienst (S. 267), um aktuelle Versionen der Firmwarekomponenten zu erhalten.

Werkzeug und Zubehör für das Aktualisieren der Firmware

- PC mit Windows Betriebssystem, der wie folgt vorbereitet wurde:
  - Das Programm PIFirmwareManager ist installiert. Weitere Informationen siehe "PC-Software installieren" (S. 43).
  - Die aktuellen Dateien der Firmwarekomponenten, die Sie von unserem Kundendienst erhalten haben (Dateityp Update Package; Dateiendung IPK), liegen in einem Verzeichnis auf dem PC.

Voraussetzungen für das Aktualisieren der Firmware

✓ Sie haben alle erforderlichen Vorbereitungen für die Kommunikation über die TCP/IP-Schnittstelle getroffen, siehe "Kommunikation über TCP/IP in der PC-Software herstellen" (S. 61).

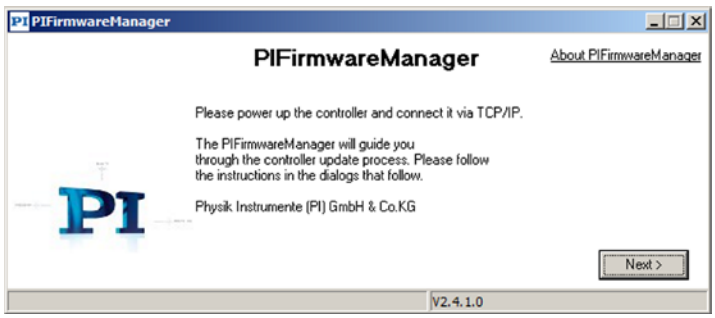
✓ Der C-886 ist eingeschaltet, und der Startvorgang des C-886 ist beendet (S. 56).

✓ Der PC ist eingeschaltet.

Firmwarekomponenten des C-886 aktualisieren

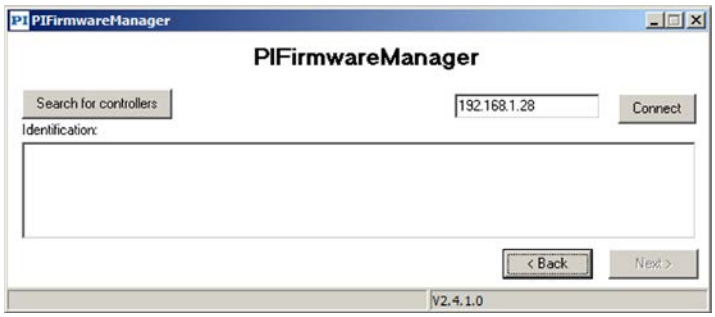
1. Starten Sie das Programm PIFirmwareManager am PC über den Startmenü-Eintrag ***Alle Programme > PI > PIFirmwareManager.***

   Das Fenster ***PIFirmwareManager*** öffnet sich.



2. Klicken Sie auf die Schaltfläche ***Next >***.

   Die Schaltfläche ***Search for controllers*** wird angezeigt.



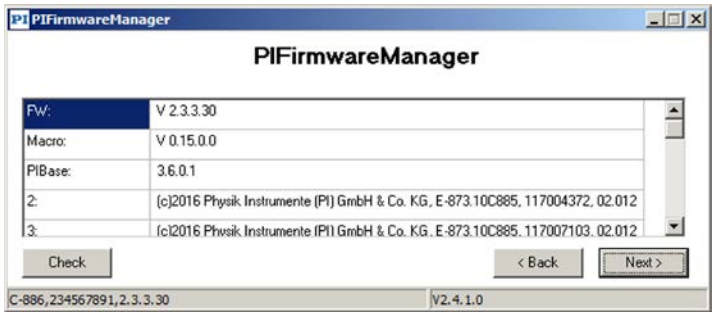
3. Klicken Sie auf die Schaltfläche ***Search for controllers***.

   Im Feld ***Identification:*** wird der gefundene C-886 angezeigt (Seriennummer, Port, IP-Adresse).

4. Markieren Sie die Zeile des C-886 im Feld ***Identification:***.

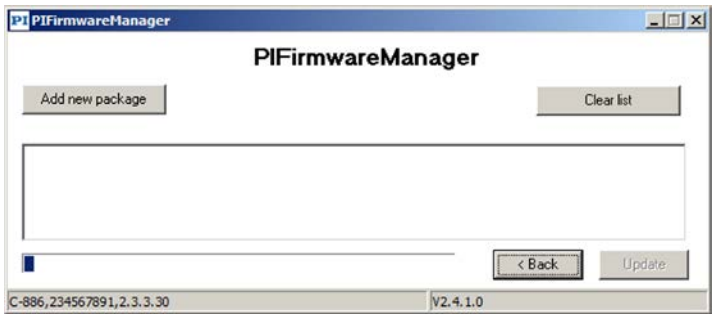
5. Klicken Sie auf die Schaltfläche ***Next >***.

   Die momentan aktiven Firmwarekomponenten des C-886 sowie einige auf dem PC installierte Software-Komponenten werden mit ihren Versionsnummern aufgelistet.



6. Klicken Sie auf die Schaltfläche ***Next >***.

   Die Schaltfläche ***Add new package*** für die Auswahl der zu aktualisierenden Firmwarekomponenten wird angezeigt.

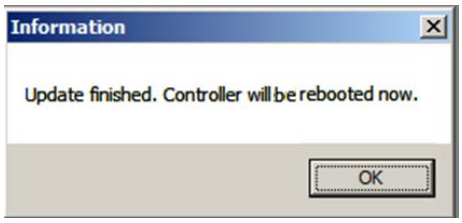


7. Klicken Sie auf die Schaltfläche ***Add new package***.

   Ein Dateiauswahlfenster öffnet sich.

8. Wechseln Sie im Dateiauswahlfenster in das Verzeichnis, in dem die Dateien liegen, die Sie vom Kundendienst erhalten haben.
9. Wählen Sie im Dateiauswahlfenster die Dateien für die zu aktualisierenden Firmwarekomponenten aus, z. B.:
   - Komponente FW: hexapod-firmware_c886_x.x.x.x.ipk
   - Komponente Macro: hexapod-macro-manager_c886_x.x.x.x.ipk
10. Klicken Sie im Dateiauswahlfenster auf die Schaltfläche ***Öffnen***, um die Auswahl zu bestätigen.
11. Klicken Sie auf die Schaltfläche ***Update***, um die Aktualisierung der ausgewählten Firmwarekomponenten auf dem C-886 zu starten.

    Das Fenster ***Information*** wird angezeigt.

12. Klicken Sie auf die Schaltfläche ***OK***.

    Der C-886 führt einen Neustart durch.

    Die Aktualisierung der Firmwarekomponenten ist beendet.

11 Störungsbehebung

Störung	Mögliche Ursachen	Behebung
Positionierer bewegt sich nicht oder unkontrolliert	Kabel defekt oder nicht korrekt angeschlossen	➢ Prüfen Sie die Kabelanschlüsse.
	Falscher Befehl oder falsche Syntax	➢ Senden Sie den Befehl ERR? (S. 139) und prüfen Sie den zurückgemeldeten Fehlercode.
	Falsche Konfiguration	Die vom C-886 verwendeten Konfigurationsdaten müssen auf den Positionierer abgestimmt sein. ➢ Prüfen Sie mit dem Befehl CST? (S. 131), auf welchen Positionierer der C-886 abgestimmt ist. ➢ Wenn der Verstellertyp in der Antwort auf CST? nicht dem angeschlossenen Positionierer entspricht, kontaktieren Sie unseren Kundendienst (S. 267). Ändern Sie die Einstellungen der Slave-Module für die Antriebe des Positionierers nur nach Rücksprache mit PI.
	Fehler in einem Slave-Modul, z. B.: ▪ Modul überhitzt ▪ Antriebs-störung ▪ Störung des Positions-sensors	Bei Störungen schaltet ein Slave-Modul den Antrieb ab. 1. Prüfen Sie die LEDs des Master-Moduls und der Slave-Module. Details siehe "Vorderwand" (S. 18). 2. Senden Sie den Befehl ERR? (S. 139) und prüfen Sie den zurückgemeldeten Fehlercode. 3. Schalten Sie den C-886 aus und lassen Sie ihn abkühlen. 4. Überprüfen Sie das System und vergewissern Sie sich, dass alle Achsen gefahrlos bewegt werden können. 5. Nehmen Sie den C-886 wieder in Betrieb, siehe "Inbetriebnahme" (S. 53). 6. Wenn die Störung erneut auftritt: Kontaktieren Sie unseren Kundendienst (S. 267).

Störung	Mögliche Ursachen	Behebung
Verringerte Genauigkeit des Positionierers	Verspannte Grundplatte	➢ Montieren Sie den Positionierer auf ebener Grundfläche (siehe Benutzerhandbuch des Positionierers).
	Verschleiß wegen kleiner Bewegungen über einen langen Zeitraum	➢ Wenn über einen langen Zeitraum nur kleine Bewegungen ausgeführt werden: Führen Sie in regelmäßigen Abständen eine Wartungsfahrt über den gesamten Stellweg durch.

Störung	Mögliche Ursachen	Behebung
Kommunikation mit dem C-886 funktioniert nicht	Falsches Kommunikationskabel wird verwendet oder es ist defekt	➢ Prüfen Sie das Kabel. – Verwenden Sie für die TCP/IP-Verbindung über einen Hub oder einen Router (mit DHCP-Server) ein Straight-Through-Netzwerkkabel. – Verwenden Sie für die direkte Verbindung mit der Ethernet-Anschlussbuchse des PC ein Crossover-Netzwerkkabel. ➢ Prüfen Sie gegebenenfalls, ob das Kabel an einem fehlerfreien System funktioniert.
	Kommunikationsschnittstelle ist nicht richtig konfiguriert	Bei Verwendung der TCP/IP-Verbindung: ➢ Schließen Sie den Controller an das Netzwerk an, **bevor** Sie ihn einschalten. Andernfalls müssen Sie den Controller aus- und wiedereinschalten. ➢ Prüfen Sie die Netzwerk-Einstellungen (S. 57). ➢ Stellen Sie sicher, dass das Netzwerk **nicht** für unbekannte Geräte gesperrt ist. ➢ Stellen Sie sicher, dass **nicht** mehrere PC-Software-Anwendungen gleichzeitig auf den C-886 zugreifen können. ➢ Stellen Sie sicher, dass Sie beim Herstellen der Kommunikation den richtigen C-886 ausgewählt haben. ➢ Wenn Sie die Probleme nicht beheben können, wenden Sie sich an Ihren Netzwerkadministrator.
	Startvorgang des C-886 ist noch nicht beendet	1. Warten Sie nach dem Einschalten oder nach einem Neustart des C-886 einige Sekunden. 2. Versuchen Sie erneut, die Kommunikation aufzubauen oder Befehle zu senden.

Störung	Mögliche Ursachen	Behebung
	Ein anderes Programm greift auf die Schnitt-stelle zu	➢ Schließen Sie das andere Programm.
	Probleme mit spezieller PC-Software	➢ Prüfen Sie, ob das System mit einer anderen PC-Software, wie z. B. einem Terminal-Programm oder einer Entwicklungsumgebung, funktioniert. Sie können die Kommunikation testen, indem Sie ein Terminal-Programm (z. B. PI Terminal) 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	Mögliche Ursachen	Behebung
Kunden-Soft-ware läuft nicht mit den PI-Treibern	Falsche Kombination der Treiberroutinen/ VIs	➢ Prüfen Sie, ob das System mit einem Terminal-Programm läuft. Wenn ja: ➢ Lesen Sie die Angaben im Handbuch der zugehörigen PC-Software und vergleichen Sie den Beispielcode auf der C-886.CD mit Ihrem Programmcode.

Wenn die Störung Ihres Systems nicht in der Tabelle angeführt ist oder wenn sie nicht wie beschrieben behoben werden kann, kontaktieren Sie unseren Kundendienst (S. 267).

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.

Die aktuellen Versionen der Benutzerhandbücher stehen auf unserer Website zum Herunterladen (S. 7) bereit.

13 Technische Daten

In diesem Kapitel

Spezifikationen 269
Systemanforderungen 273
Abmessungen 274
Pinbelegung 275

13.1 Spezifikationen

13.1.1 Datentabelle

	C-886.1
Funktion	Controller für parallelkinematische Positionierer
Achsen	6 Optional: 2 zusätzliche Einzelachsen
Bewegung und Regler	
Antriebsart	DC-Motor Antriebsart Einzelachsen optional
Motoranschluss	D-Sub 15 (w)
Reglertyp	PID-Regler
Servozykluszeit Slave-Module	50 µs
Zykluszeit des C-886	10 ms
Encodereingang	A/B-Quadratur TTL-Pegel, einseitig geerdet oder differenziell gem. RS-422; 60 MHz
Blockiererkennung	Regelung deaktiviert bei Überschreiten eines Positionsfehlers
Endschalter	2 × TTL je Antrieb (Polarität programmierbar)
Referenzschalter	1 × TTL je Antrieb
Eigenschaften Einzelachsen	Abhängig von der Antriebsart
Elektrische Eigenschaften	
Ausgangsspannung	0 bis 24 VDC
Strombegrenzung	3 A je Antrieb
Eigenschaften Einzelachsen	Abhängig von der Antriebsart

	C-886.1
Schnittstellen und Bedienung	
Kommunikations-Schnittstellen	TCP/IP: RJ45/Ethernet; USB: Mini-USB Typ B
Befehlssatz	PI General Command Set (GCS)
Bedienersoftware	PIMikroMove®
Schnittstellen zur Anwendungsprogrammierung	API für C / C++ / C# / VB.NET / MATLAB / python, Treiber für LabVIEW
Unterstützte Funktionen	Anwenderdefinierte Koordinatensysteme. Startup-Makro. Datenrekorder zur Aufnahme von Betriebsgrößen. Funktionsgenerator für periodische Bewegungen.
Umgebung	
Betriebsspannung	Externes Netzteil 24 V / 10 A im Lieferumfang enthalten
Max. Stromaufnahme	32 A
Betriebstemperaturbereich	10 bis 40 °C
Masse	4,4 kg ohne Antriebsmodule für Einzelachsen
Abmessungen	482,6 mm × 132,55 mm× 278,55 mm

	C-886.2
Funktion	Controller für parallelkinematische Positionierer
Achsen	6 Optional: 2 zusätzliche Einzelachsen
Bewegung und Regler	
Antriebsart	2-Phasen-Schrittmotor Antriebsart Einzelachsen optional
Motoranschluss	HD D-Sub 26 (w)
Reglertyp	PID-Regler
Mikroschrittauflösung	1/2048 Vollschritt
Servozykluszeit Slave-Module	50 µs
Zykluszeit des C-886	10 ms
Encodereingang	A/B-Quadratur, TTL, RS-422; 60 MHz
Blockiererkennung	Regelung deaktiviert bei Überschreiten eines Positionsfehlers
Endschalter	2 × TTL je Antrieb (Polarität programmierbar)
Referenzschalter	1 × TTL je Antrieb
Indexschalter	1 x RS-422 je Antrieb, für Indexpuls
Eigenschaften Einzelachsen	Abhängig von der Antriebsart

	C-886.2
Elektrische Eigenschaften	
Max. Ausgangsspannung	48 VDC
Max. Leistungsaufnahme Volllast	48 W je Antrieb
Leistungsaufnahme Leerlauf	3 W je Antrieb
Max. Ausgangsleistung (< 2 ms)	100 W je Antrieb
Dauerausgangsleistung	< 48 W je Antrieb
Strombegrenzung pro Motorphase	2,5 A
Eigenschaften Einzelachsen	Abhängig von der Antriebsart
Schnittstellen und Bedienung	
Kommunikations-Schnittstellen	TCP/IP: RJ45/Ethernet; USB: Mini-USB Typ B
Befehlssatz	PI General Command Set (GCS)
Bedienersoftware	PIMikroMove®
Schnittstellen zur Anwendungsprogrammierung	API für C / C++ / C# / VB.NET / MATLAB / python, Treiber für LabVIEW
Unterstützte Funktionen	Anwenderdefinierte Koordinatensysteme. Startup-Makro. Datenrekorder zur Aufnahme von Betriebsgrößen. Funktionsgenerator für periodische Bewegungen.
Umgebung	
Betriebsspannung	Externes Netzteil 24 V / 10 A im Lieferumfang enthalten
Max. Stromaufnahme	32 A
Betriebstemperaturbereich	10 bis 40 °C
Masse	4,4 kg ohne Antriebsmodule für Einzelachsen
Abmessungen	482,6 mm × 132,55 mm× 278,55 mm

	C-886.31
Funktion	Controller für parallelkinematische Positionierer
Achsen	6 Optional: 2 zusätzliche Einzelachsen
Bewegung und Regler	
Antriebsart	Q-Motion® Piezoträgheitsantrieb Antriebsart Einzelachsen optional
Motoranschluss	D-Sub 15 (w)
Reglertyp	PID-Regler

	C-886.31
Servozykluszeit Slave-Module	50 µs
Zykluszeit des C-886	10 ms
Encodereingang	Sin/Cos (differenziell), BiSS-Schnittstelle
Blockiererkennung	Regelung deaktiviert bei Überschreiten eines Positionsfehlers
Endschalter	2 × TTL je Antrieb (Polarität programmierbar)
Referenzschalter	1 × TTL je Antrieb
Eigenschaften Einzelachsen	Abhängig von der Antriebsart
Elektrische Eigenschaften	
Ausgangsspannung	0 bis 48 V
Max. Ausgangsleistung	30 W je Antrieb
Eigenschaften Einzelachsen	Abhängig von der Antriebsart
Schnittstellen und Bedienung	
Kommunikations-Schnittstellen	TCP/IP: RJ45/Ethernet; USB: Mini-USB Typ B
Befehlssatz	PI General Command Set (GCS)
Bedienersoftware	PIMikroMove®
Schnittstellen zur Anwendungsprogrammierung	API für C / C++ / C# / VB.NET / MATLAB / python, Treiber für LabVIEW
Unterstützte Funktionen	Anwenderdefinierte Koordinatensysteme. Startup-Makro. Datenrekorder zur Aufnahme von Betriebsgrößen. Funktionsgenerator für periodische Bewegungen.
Umgebung	
Betriebsspannung	Externes Netzteil 24 V / 10 A im Lieferumfang enthalten
Max. Stromaufnahme	32 A
Betriebstemperaturbereich	10 bis 40 °C
Masse	4,4 kg ohne Antriebsmodule für Einzelachsen
Abmessungen	482,6 mm × 132,55 mm× 278,55 mm

13.1.2 Bemessungsdaten

Der C-886 ist für folgende Betriebsgrößen ausgelegt:

Eingang an:	Maximale Betriebsspannung	⚠	Betriebs-frequenz	⚠	Maximale Stromaufnahme	⚠
Durchführungs-klemmen **GND** und **+24V**	24 V		⎓		32 A	

13.1.3 Umgebungsbedingungen und Klassifizierungen

Folgende Umgebungsbedingungen und Klassifizierungen sind für den C-886 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

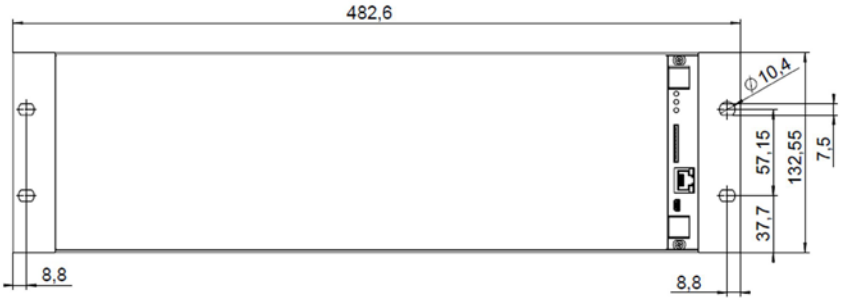
13.2 Systemanforderungen

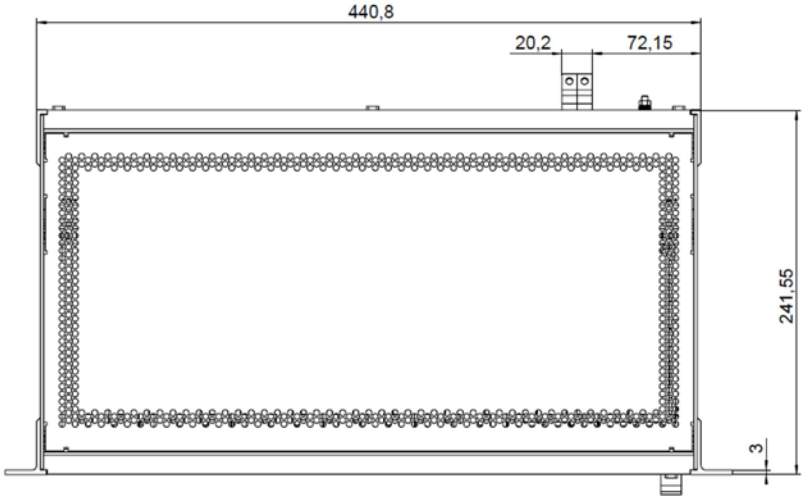
Für den Betrieb des Systems müssen folgende Systemanforderungen erfüllt sein:

- Geeigneter <parallelkinematischer Positionierer von PI
- C-886 mit Netzteil
- PC mit mindestens 30 MB freiem Speicherplatz und einem der folgenden Betriebssysteme:
  - Windows: Vista Service Pack 1, Windows 7, 8 und 10 (32 Bit, 64 Bit)
  - Linux
- Kommunikationsschnittstelle zum PC:
  USB-Schnittstelle am PC
  oder
  Ethernet-Anschluss im PC oder freier Zugangspunkt im Netzwerk, an das der PC über TCP/IP angeschlossen ist
- USB- oder Netzwerkkabel zur Verbindung des C-886 mit dem PC oder mit dem Netzwerk
- Produkt-CD mit PC-Software

13.3 Abmessungen

C-886, hier ohne Antriebsmodule. Abmessungen in mm.





13.4 Pinbelegung

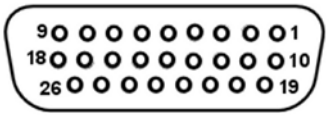
13.4.1 Axis 1, Axis 2 (nur C-886.1)

D-Sub-Buchse, 15-polig, weiblich

Pin	Funktion
1	Ausgang: Programmierbare Motorbremse (0 oder + 5 V)
2	Ausgang: Motor + (differentiell; Power PWM); für Versteller ohne PWM-Verstärker
3	Ausgang: PWM Magnitude (TTL); für Versteller mit PWM-Verstärker
4	Ausgang: +5 V
5	Eingang: Positiver Endschalter
6	GND Endschalter
7	Eingang: Encoder: A (-)
8	Eingang: Encoder: B (-)
9	Ausgang: Motor – (differentiell; Power PWM); für Versteller ohne PWM-Verstärker
10	Power GND
11	Ausgang: PWM Sign (TTL); für Versteller mit PWM-Verstärker
12	Eingang: Negativer Endschalter
13	Eingang: Referenzschalter
14	Eingang: Encoder: A (+) / ENCA
15	Eingang: Encoder: B (+) / ENCB

13.4.2 Motor (nur C-886.2)

HD D-Sub 26 (w)

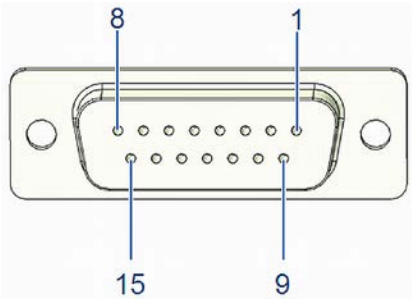


Pin	Signal	Richtung	Funktion
1	OUT0a	Ausgang	Phase I, pos. (48 V, max. 2 A, 20 kHz PWM)
2	OUT0b	Ausgang	Phase I, pos. (48 V, max. 2 A, 20 kHz PWM)
3	OUT1a	Ausgang	Phase I, neg. (48 V, max. 2 A, 20 kHz PWM)
4	OUT1b	Ausgang	Phase I, neg. (48 V, max. 2 A, 20 kHz PWM)
5	OUT2a	Ausgang	Phase II, pos. (48 V, max. 2 A, 20 kHz PWM)
6	OUT2b	Ausgang	Phase II, pos. (48 V, max. 2 A, 20 kHz PWM)
7	OUT3a	Ausgang	Phase II, neg. (48 V, max. 2 A, 20 kHz PWM)
8	OUT3b	Ausgang	Phase II, neg. (48 V, max. 2 A, 20 kHz PWM)
9	-	-	reserviert
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	-	-	reserviert
14	-	-	reserviert
15	-	-	reserviert
16	-	-	reserviert
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)
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.

13.4.3 Motor & Sensor (nur C-886.31)

Buchse D-Sub 15



Pin	Signal	Funktion
1	REF-	Referenzschalter, differenziell (-)
2	PIEZO-	Motorsignal (-)
3	PIEZO+	Motorsignal (+)
4	5V	Versorgungsspannung, +5V
5	PLIM	Positiver Endschalter
6	ID_CHIP	ID-Chip-Daten
7	ENCA-	Encoderkanal A, differenziell (-)
8	ENCB-	Encoderkanal B, differenziell (-)
9	PIEZO-	Motorsignal (-)
10	GND	GND
11	PIEZO+	Motorsignal (+)
12	NLIM	Negativer Endschalter
13	REF+	Referenzschalter, differenziell (+)
14	ENCA+	Encoderkanal A, differenziell (+)
15	ENCB+	Encoderkanal B, differenziell (+)

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.

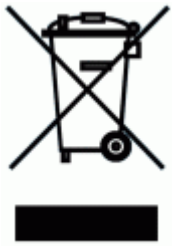
Um der Produktverantwortung als Hersteller gerecht zu werden, übernimmt die Physik Instrumente (PI) GmbH & Co. KG kostenfrei die umweltgerechte Entsorgung eines PI-Altgerätes, sofern es nach dem 13. August 2005 in Verkehr gebracht wurde.

Falls Sie ein solches Altgerät von PI besitzen, können Sie es versandkostenfrei an folgende Adresse senden:

Physik Instrumente (PI) GmbH & Co. KG

Auf der Römerstr. 1

D-76228 Karlsruhe



15 EU-Konformitätserklärung

Für den C-886 wurde eine EU-Konformitätserklärung gemäß den folgenden europäischen Richtlinien 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 50581