



# User Manual

C-884.XDC DIGITAL CONTROLLER FOR POSITIONERS WITH DC MOTOR

# Contents

<b>1</b>	<b>Legal Information.....</b>	<b>6</b>
<b>2</b>	<b>Introduction.....</b>	<b>8</b>
2.1	About this Document.....	8
2.1.1	Objective and Target Group.....	8
2.1.2	Other Applicable Documents.....	8
2.1.3	Explanation of Symbols.....	8
2.1.4	Figures.....	10
2.1.5	Downloading Manuals.....	10
2.2	European Declarations of Conformity.....	10
2.3	Customer Service Department.....	10
2.4	Old Equipment Disposal.....	11
<b>3</b>	<b>Safety.....</b>	<b>12</b>
3.1	Intended Use.....	12
3.2	General Safety Instructions.....	12
3.3	Organizational Measures.....	12
3.3.1	User Manual.....	12
3.3.2	General Personnel Qualification.....	12
<b>4</b>	<b>Product Description.....</b>	<b>13</b>
4.1	Model Overview.....	13
4.2	Front Panel.....	13
4.3	Type Plate.....	15
4.4	Scope of Delivery.....	16
4.5	Optional Accessories.....	16
4.6	Communication Interfaces.....	16
4.6.1	Controlling PI Systems.....	16
4.6.2	C-884.xDC Interfaces.....	17
4.7	Software Overview.....	18
4.8	Positioner database.....	21
4.9	ID Chip Detection.....	22
4.10	Functional Principles of the C-884.xDC.....	22
4.10.1	Block Diagram.....	22
4.10.2	Important Firmware Components.....	23
4.10.3	Commandable Items.....	25
4.10.4	Physical Units.....	27
4.10.5	Operating Modes.....	29
4.10.6	Generating a Dynamics Profile.....	30
4.10.7	Servo Algorithm and Other Control Value Corrections.....	33
4.10.8	Triggering Motion.....	36
4.10.9	On-Target State.....	38
4.10.10	Reference Switch Detection.....	39

4.10.11	Detecting Limit Switches.....	40
4.10.12	Travel Range and Soft Limits.....	41
4.10.13	Referencing.....	42
<b>5</b>	<b>Unpacking / Transportation.....</b>	<b>45</b>
5.1	Unpacking.....	45
5.2	Transportation.....	45
<b>6</b>	<b>Installation.....</b>	<b>46</b>
6.1	Mounting the C-884.xDC.....	46
6.2	Connecting the C-884.xDC to the Protective Earth Conductor.....	47
6.3	Connecting the Power Supply to the C-884.xDC.....	48
6.4	Connecting the Positioner to the C-884.xDC.....	48
6.5	Installing the PC Software.....	49
6.5.1	Installing the PC Software for the First Time.....	49
6.5.2	Updating the PC Software.....	50
6.6	Connecting the PC.....	55
6.6.1	Connecting the C-884.xDC to a PC.....	56
6.6.2	Integrating the C-884.xDC into a Network.....	56
<b>7</b>	<b>Startup and Operation.....</b>	<b>57</b>
7.1	Switching on the C-884.xDC.....	57
7.2	Establishing Communication with the PC.....	57
7.2.1	Establishing Communication via RS-232.....	57
7.2.2	Establishing Communication via USB.....	58
7.2.3	Establishing Communication via TCP/IP.....	59
7.3	Starting Motion.....	62
7.4	Adapting the Dynamic Characteristics of the System.....	66
7.4.1	Adjusting the Notch Filter.....	66
7.4.2	Optimizing the Servo Control Parameters.....	70
7.5	Making Data Backups.....	73
7.5.1	Saving Parameter Values.....	73
7.5.2	Saving Controller Macros.....	74
<b>8</b>	<b>C-884.xDC Functions.....</b>	<b>76</b>
8.1	Protective Functions of the C-884.xDC.....	76
8.1.1	Behavior with Motion Errors.....	76
8.1.2	Restoring Operational Readiness.....	77
8.1.3	Commands.....	77
8.1.4	Parameters .....	77
8.2	Trajectories for Motion Paths.....	77
8.2.1	Operating Principle of the Trajectory Buffer.....	77
8.2.2	Working with Trajectories.....	78
8.2.3	Commands.....	80
8.2.4	Parameters .....	80
8.3	Data Recorder.....	80

8.3.1	Setting up the Data Recorder.....	80
8.3.2	Starting Data Recording.....	81
8.3.3	Reading Out Recorded Data.....	81
8.3.4	Commands.....	81
8.3.5	Parameters .....	82
8.4	Digital Inputs and Outputs.....	83
8.4.1	Connecting the Digital Outputs.....	83
8.4.2	Connecting the Digital Inputs.....	83
8.4.3	Digital Output Signals.....	83
8.4.4	Digital Input Signals.....	88
8.4.5	Commands.....	89
8.4.6	Parameters .....	90
8.5	Analog Input Signals.....	90
8.5.1	Connecting Analog Signal Sources.....	91
8.5.2	Commands.....	91
8.5.3	Parameters .....	91
8.6	Controlling with HID.....	92
8.6.1	How it Works.....	92
8.6.2	HID Control Configuration.....	92
8.6.3	Programming HID Control.....	93
8.6.4	Connecting an HID.....	93
8.6.5	Testing the HID and Calibrating the Axes.....	93
8.6.6	Setting and Activating HID Control.....	97
8.6.7	Commands.....	98
8.6.8	Parameters .....	99
8.7	Controller Macros.....	99
8.7.1	Recording Macros.....	100
8.7.2	Running the Macros.....	101
8.7.3	Stopping the Macros.....	103
8.7.4	Configuring a Startup Macro.....	103
8.7.5	Deleting Macros.....	104
8.7.6	Saving and Loading Macros.....	104
8.7.7	Example Macros.....	104
8.7.8	Commands.....	108
8.7.9	Parameters .....	109
8.8	Working with GCS Commands.....	109
8.8.1	GCS Command Syntax.....	109
8.8.2	Variables.....	110
8.9	Adapting Settings.....	111
8.9.1	Parameter Commands.....	111
8.9.2	Saving Parameter Values in a Text File.....	112
8.9.3	Changing Parameter Values.....	113
8.9.4	Creating or Changing Parameter Sets for Positioners.....	114



- 9 Command Reference..... 120
- 10 Parameter Reference.....186
- 11 Maintenance.....224
  - 11.1 Cleaning.....224
  - 11.2 Updating the Firmware..... 224
- 12 Troubleshooting.....226
- 13 Technical Data..... 229
  - 13.1 Specifications.....229
  - 13.2 Maximum Ratings..... 230
  - 13.3 Ambient Conditions and Classifications.....230
  - 13.4 Dimensions..... 231
- 14 Appendix..... 232
  - 14.1 Pin Assignment.....232
    - 14.1.1 Axis Connector.....232
    - 14.1.2 I/O Connector..... 233
    - 14.1.3 RS-232 Connector..... 234
    - 14.1.4 Power Adapter Connector..... 234
  - 14.2 GCS Error Codes.....234
- 15 Glossary.....262

# 1 Legal Information



The following trademarks are the intellectual property of Physik Instrumente (PI) SE & Co. KG ("PI") and have been entered in the trademark register of the German Patent and Trade Mark Office and, in some cases, also in other trademark registers under the company name of Physik Instrumente (PI) GmbH & Co. KG: PI®, PIC®, PICMA®, PLine®, PIFOC®, PiezoWalk®, NEXACT®, NEXLINE®, PInano®, NanoCube®, Picoactuator®, PicoCube®, PIMikroMove®, PIMag®, PIHera®

The software products provided by PI are subject to the General Software License Terms of Physik Instrumente (PI) SE & Co. KG and may contain and/or use third-party software components. Further information can be found in the [General Software License Terms](#) and in the [Third-Party Software Notes](#) on our website.

Notes on brand names and third-party trademarks:

Microsoft® and Windows® are registered trademarks or trademarks of Microsoft Corporation in the USA and/or other countries.

LabVIEW National Instruments and NI are trademarks of National Instruments. Neither the driver software nor the software programs offered by PI or other goods and services are connected to or sponsored by National Instruments.

Python® is a registered trademark of Python Software Foundation.

BiSS is a trademark of iC-Haus GmbH.

The following designations are protected company names, trademarks or registered trademarks of other owners:

Linux, MATLAB, MathWorks

© 2024 Physik Instrumente (PI) SE & Co. KG, Karlsruhe, Germany. The text, photographs, and drawings in this manual are protected by copyright. With regard thereto, Physik Instrumente (PI) SE & Co. KG reserves all rights. The use of any text, images and drawings is permitted only in part and only when indicating the source.

Original instructions

First print: 10/24/2024

Document number: MS243en, ASt

Subject to change. This manual is superseded by any new release. The latest respective release is available for download on our website ([www.pi.ws](http://www.pi.ws)).

**Publisher:**

Physik Instrumente (PI) SE & Co. KG  
Auf der Römerstraße 1  
76228 Karlsruhe  
Germany

[info@pi.de](mailto:info@pi.de)  
[www.pi.de](http://www.pi.de)

**Customer service department:**

Physik Instrumente (PI) SE & Co. KG  
Auf der Römerstraße 1  
76228 Karlsruhe  
Germany

[service@pi.de](mailto:service@pi.de)  
[www.pi.de](http://www.pi.de)

## 2 Introduction

### 2.1 About this Document

#### 2.1.1 Objective and Target Group

This user manual contains the information needed for the intended use of the C-884.xDC. Basic knowledge of closed-loop systems, motion control concepts, and applicable safety measures is assumed.

#### 2.1.2 Other Applicable Documents

The devices and software tools that are mentioned in this documentation are described in separate manuals.

Document number	Document type	Product
SM148E	Software Manual	PIMikroMove
SM146E	Software Manual	GCS Array Data Format Description
SM151E	Software Manual	PI GCS 2.0 DLL0
SM155E	Software Manual	PI MATLAB Driver GCS 2.0
SM158E	Software Manual	PI GCS2 driver library for use with NI LabVIEW software
SM156E	Software Manual	PIStages3Editor: Software for managing the positioner database
SM164E	Software Manual	PIFirmwareManager for updating the firmware
A000T0028	User Manual	PIUpdateFinder: Updating PI Software
A000T0089	Technical Note	PI Software on ARM-Based Platforms
A000T0081	Technical Note	Downloading manuals from PI: PDF file with links to the manuals for digital electronics and software from PI. Supplied with the PI software.
SM157E	User Manual	PIPython - working with Python on PI controllers

The latest versions of the user manuals can be [downloaded \(p. 10\)](#) at [www.pi.ws](http://www.pi.ws).



#### 2.1.3 Explanation of Symbols

This chapter explains the symbols and markings used by PI in their user manuals.

### Typographic Conventions

Symbol / label	Meaning
1. 2.	Action consisting of one or several steps with strict sequential order
►	Action consisting of one or more steps without relevant sequential order
■	Lists
p. 5	Cross-reference to page 5
<b>RS-232</b>	Label on the product indicating an operating element (example: RS-232 interface socket)
<b><i>Start &gt; Settings</i></b>	Menu path in the PC software (example: to open the menu, <b><i>Start</i></b> and <b><i>Settings</i></b> must be clicked successively)
POS?	Command line or a command from PI's General Command Set (GCS) (example: command to get the axis position)
<b><i>Device S/N</i></b>	Parameter name (example: parameter where the serial number is stored)
5	Value that must be entered or selected via the PC software

### Symbols Used

Symbol / Label	Meaning
	General hazard symbol
	Electrical voltage



### **DANGER**

#### **Dangerous situation**

Failure to comply could lead to death or serious injury.

► Precautionary measures for avoiding the risk.



### **WARNING**

#### **Dangerous situation**

Failure to comply could lead to serious injury.

► Precautionary measures for avoiding the risk.



### **CAUTION**

#### **Dangerous situation**

Failure to comply could lead to minor injury.

► Precautionary measures for avoiding the risk.

### **NOTICE**



#### **Dangerous situation**

Failure to comply could lead to material damage.

► Precautionary measures for avoiding the risk.

### Information

Additional information on the C-884.xDC that can affect your application.

## 2.1.4 Figures

For better understandability, the colors, proportions and degree of detail in illustrations can deviate from the actual circumstances. Photographic illustrations may also differ and must not be seen as guaranteed properties.

## 2.1.5 Downloading Manuals

### Information

If a manual is missing or problems occur with downloading:

- Contact our [customer service department \(p. 10\)](#).

### Downloading Manuals

1. Open the website [www.pi.ws](http://www.pi.ws).
2. Search the website for the product number (e.g., C-884).
3. In the search results, select the product to open the product detail page.
4. Select **Downloads**.  
→ *The manuals are shown under **Documentation**. Software manuals are shown under **General Software Documentation**.*
5. For the desired manual, select **ADD TO LIST** and then **REQUEST**.
6. Fill out the request form and select **SEND REQUEST**.  
→ *The download link will be sent to the email address entered in the form.*

## 2.2 European Declarations of Conformity

For the C-884.xDC, declarations of conformity were issued according to the following European statutory requirements:

- EMC Directive
- RoHS Directive

The applied standards certifying the conformity are listed below.

- EMC: EN 61326-1
- Safety: EN 61010-1
- RoHS: EN IEC 63000

## 2.3 Customer Service Department

For enquiries and orders, contact your PI representative or send us an [email](#).

If you have any questions concerning your system, provide the following information:

- Product and serial numbers of all products in the system
- Firmware version of the controller (if applicable)
- Version of the driver or the software (if applicable)
- Operating system on the PC (if applicable)

If possible: Take photographs or make videos of your system that can be sent to our customer service department if requested.

**Customer service address:**

Physik Instrumente (PI) SE & Co. KG  
Auf der Römerstraße 1  
76228 Karlsruhe  
Germany

[service@pi.de](mailto:service@pi.de)

[www.pi.de](http://www.pi.de)

## 2.4 Old Equipment Disposal

In accordance with EU law, electrical and electronic equipment may not be disposed of in EU member states via the municipal residual waste.

Dispose of your old equipment according to international, national, and local rules and regulations.

In order to fulfill the responsibility as the product manufacturer, PI undertakes environmentally correct disposal of all PI equipment free of charge, if it was made available to the market after August 13, 2005.

Any old PI equipment can be sent free of charge to the following address:

Physik Instrumente (PI) SE & Co. KG  
Auf der Römerstraße 1  
76228 Karlsruhe  
Germany

[info@pi.de](mailto:info@pi.de)

[www.pi.de](http://www.pi.de)



## 3 Safety

### 3.1 Intended Use

The C-884.xDC is a laboratory device as defined by DIN EN 61010-1. It is intended for indoor use and use in an environment that is free of dirt, oil, and lubricants.

In accordance with its design, the C-884.xDC is intended for operating PI positioners equipped with DC motors or voice coil drives. The C-884.xDC is intended for closed-loop operation with incremental or absolute-measuring position sensors. In addition, it can read and process the reference and limit switch signals from the connected positioner.

The C-884.xDC may not be used for purposes other than those stated in this user manual. The C-884.xDC may only be used in compliance with the technical specifications and instructions in this user manual.

### 3.2 General Safety Instructions

The C-884.xDC is built according to state-of-the-art technology and recognized safety standards. Improper use of the C-884.xDC may result in personal injury and/or damage to the C-884.xDC.

- ▶ Use the C-884.xDC only for its intended purpose and when it is in perfect condition.
- ▶ Read the user manual.
- ▶ Eliminate any malfunctions that may affect safety immediately.

The operator is responsible for installing and operating the C-884.xDC correctly.

- ▶ Install the C-884.xDC near the power adapter so that the power plug can be quickly and easily disconnected from the mains.
- ▶ Use sufficiently dimensioned components to connect the C-884.xDC to the power supply.
- ▶ Only use cables and connections that comply with local safety regulations.

### 3.3 Organizational Measures

#### 3.3.1 User Manual

- ▶ Always keep this user manual available with the C-884.xDC. The latest versions of the user manuals can be [downloaded \(p. 10\)](#) at [www.pi.ws](http://www.pi.ws).
- ▶ Add all information from the manufacturer such as supplements or technical notes to the user manual.
- ▶ If you give the C-884.xDC to a third party, also include this user manual as well as other relevant information provided by the manufacturer.
- ▶ Only use the device on the basis of the complete user manual. Missing information due to an incomplete user manual can result in minor injury and damage to equipment.
- ▶ Only install and operate the C-884.xDC after you have read and understood this user manual.

#### 3.3.2 General Personnel Qualification

The C-884.xDC may only be installed, started up, operated, maintained, and cleaned by authorized and appropriately qualified personnel.



# 4 Product Description

## 4.1 Model Overview

Digital controller for positioners with DC motor

Product number	Description
C-884.4DC	Controller for DC motors, 4 axes, USB, RS-232, Ethernet, SPI, I/O, joystick connection
C-884.6DC	Controller for DC motors, 6 axes, USB, RS-232, Ethernet, SPI, I/O, joystick connection

## 4.2 Front Panel



Figure 1: C-884.4DC front panel

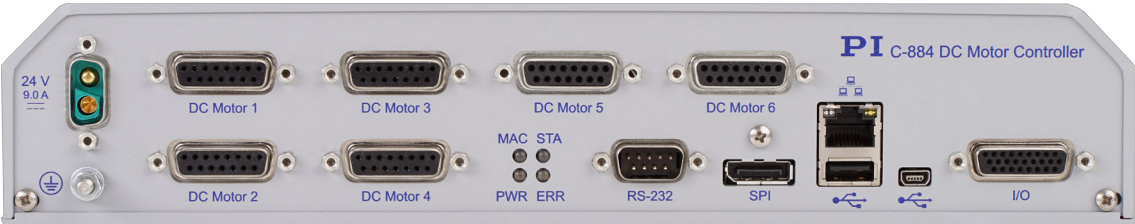




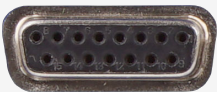






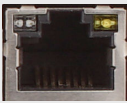
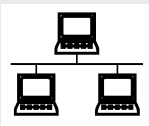







Figure 2: C-884.6DC front panel

Element	Labeling	Type	Function
	24 V 9.0 A 	<a href="#">D-sub 2W2C (p. 234)</a>	Connector for the supply voltage
		M4 threaded pin	<a href="#">Ground connection (p. 47)</a> If potential equalization is required, the C-884.xDC can be connected to the grounding system.
	DC Motor 1 to DC Motor 4 or DC motor 6	<a href="#">D-sub 15 (f) (p. 232)</a>	Connectors for positioner axes. For DC motors and voice coil drives only! <ul style="list-style-type: none"> <li>■ PWM signal output for the positioner</li> <li>■ Input of the signals of the position sensor</li> <li>■ Signal input from the limit switches and reference switch</li> <li>■ ID chip signal input</li> </ul>
	MAC	LED green/red/off	Macro: <ul style="list-style-type: none"> <li>■ Off: Macro is not running.</li> <li>■ Green: Macro is running</li> <li>■ Red: Macro error</li> </ul>
	PWR	LED green/off	Power: <ul style="list-style-type: none"> <li>■ Flashing: Linux firmware components booting</li> <li>■ Lights up continuously: Linux firmware component booting is completed.</li> <li>■ Off: C-884.xDC is not connected to the supply voltage.</li> </ul>
	STA	LED green/off	State: <ul style="list-style-type: none"> <li>■ Flashing: DSP and FPGA firmware is booting</li> <li>■ Lights up continuously: Booting the DSP and FPGA firmware is complete and the controller is ready for normal operation.</li> <li>■ Off: C-884.xDC is not connected to the supply voltage.</li> </ul>
	ERR	LED red/off	Error indicator: <ul style="list-style-type: none"> <li>■ Lights up continuously: Error (error code <math>\neq 0</math>)</li> <li>■ Off: No error (error code = 0)</li> </ul>

Element	Labeling	Type	Function
	RS-232	<a href="#">D-sub 9 (m) (p. 234)</a>	Serial connection to PC
	SPI	Display port	Serial connection to a serial peripheral interface (SPI) master unit
		RJ45	Ethernet interface for communication via TCP/IP
		USB type A	USB connection for HID (e.g., joystick or gamepad)
		Mini B USB	Universal serial bus for connecting to the PC
	I/O	<a href="#">HD D-sub 26 (f) (p. 233)</a>	Digital inputs/outputs: <ul style="list-style-type: none"><li>■ Outputs: Trigger external devices</li><li>■ Inputs: Use in macros as switch signals or for HID control</li></ul> Analog inputs: <ul style="list-style-type: none"><li>■ Use in macros or for scanning processes</li></ul>

4.3 Type Plate

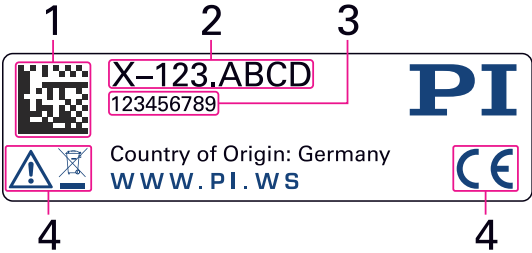


Figure 3: Type plate of the C-884.xDC

- 1. Data matrix code (example; contains the serial number)
- 2. Product number (example)
- 3. Serial number (example), individual for each C-884.xDC
- 4. Warning and conformity symbols ([old equipment disposal \(p. 11\)](#), [CE mark \(p. 10\)](#))

## 4.4 Scope of Delivery

Product number	Description
C-884.xDC	Digital controller according to the <a href="#">order (p. 13)</a>
000023194	24 V 120 W wide input range power supply with barrel connector socket; for use with line voltages from 100 to 240 V AC and voltage frequencies of 50 or 60 Hz
K050B0004	Power adapter, barrel connector to D-sub 2W2C
3763	Power cord
C-815.34	RS-232 null modem cable, 3 m, 9/9-pin
000036360	USB cable (type A to Mini-B) for connection to the PC
C-815.553	Straight-through network cable for connecting the PC via a TCP/IP network
C-815.563	Crossover network cable for direct connection to the PC via TCP/IP
C-990.CD1	Data storage medium with PC software from PI
MS242EK	Short instructions for digital motor controllers and drivers

## 4.5 Optional Accessories

The following articles are not in the scope of delivery of the C-884.xDC and must be ordered separately, if required.

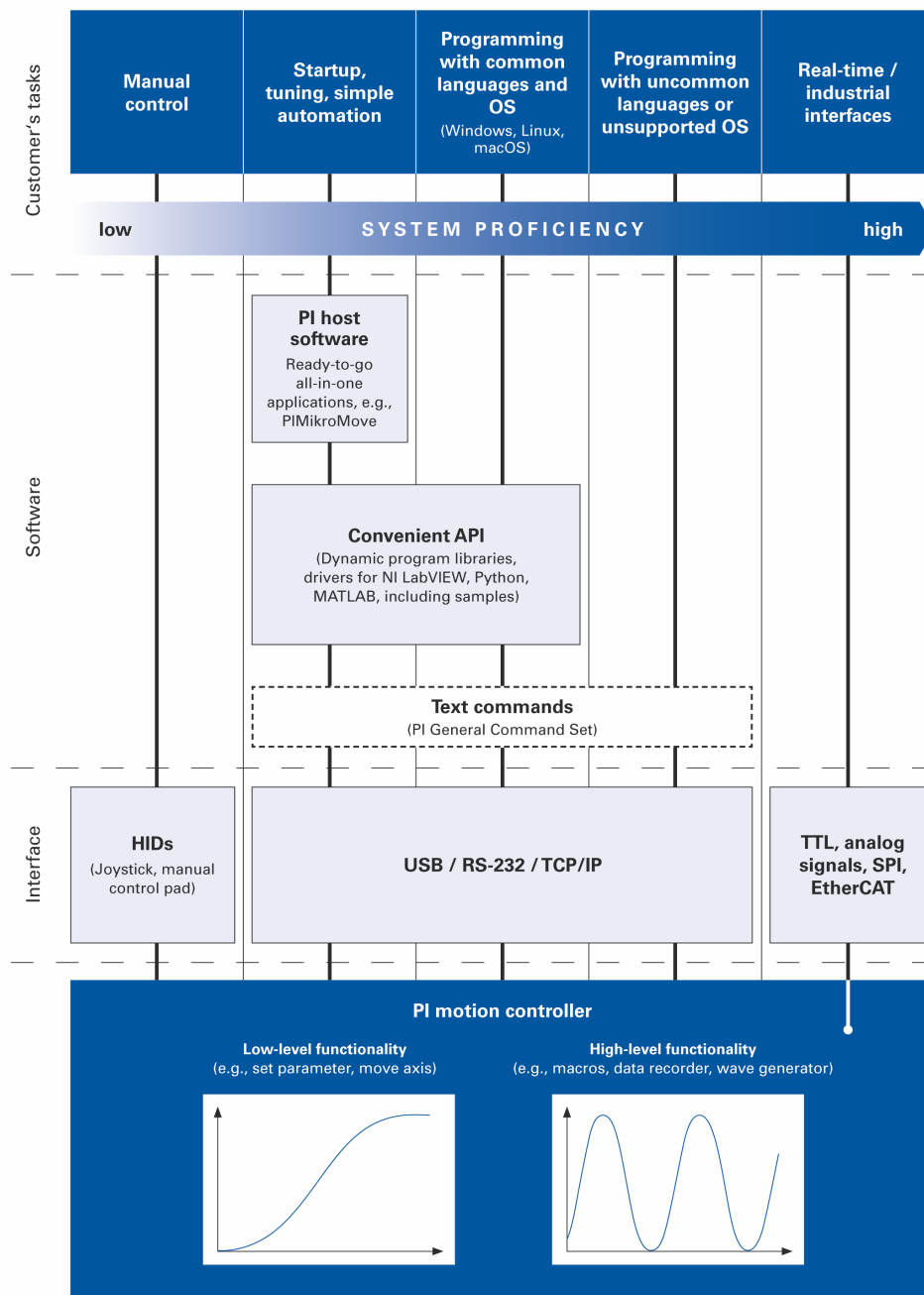
Product number	Description
C-815.38	Drive cable, DC motor, D-sub 15 (m) to D-sub 15 (f), 3 m
C-815.LDM1	Drive cable, DC motor, HD D-sub 26 (f) to D-sub 15 (m), 3 m

To order, contact our [customer service department \(p. 10\)](#).

## 4.6 Communication Interfaces

### 4.6.1 Controlling PI Systems

Basically, PI systems can be controlled as follows:



#### 4.6.2 C-884.xDC Interfaces

The C-884.xDC can be controlled via the following communication interfaces:

- PC interfaces, via [software \(p. 18\)](#) or PI General Command Set:
  - TCP/IP
  - USB
  - RS-232
- SPI (if you wish to use the SPI interface, contact the [customer service department \(p. 10\).](#))
- Analog signals
- Digital Signals: TTL

The interface parameters in the C-884.xDC's volatile memory can be queried with the [IFC?](#) command and changed with [IFC](#) command. The [IFS?](#) and [IFS](#) commands are available for querying and changing the interface parameters in the C-884.xDC's nonvolatile memory.

The number of input lines available on the C-884.xDC can be queried with the [TAC?](#) (p. 176) command. The voltage at the analog inputs can be queried with the [TAV?](#) (p. 176) command.

The number of available digital input and output lines to and from the C-884.xDC can be queried with the [TIO?](#) (p. 179) command. The status of the digital input lines can be queried with the [DIO?](#) (p. 136) command.

### TCP/IP

C-884.xDC interface parameters for TCP/IP communication:

Interface property (parameter name)	Factory setting	Note
Default IP address (IPADR)	192.168.0.75	Is not used when an IP address is assigned to the C-884.xDC by a DHCP server (IP-START).
Port for communication with the C-884.xDC	:50000	Not editable.
Subnet mask (IPMASK)	255.255.255.0	
Max. connections (IPMAXCONN)	1	Number of permissible simultaneous TCP/IP connections to the C-884.xDC
Startup behavior for configuring the IP address for TCP/IP communication (IPSTART)	DHCP active	The IP address of the C-884.xDC is assigned via DHCP by the default setting of the startup behavior. The default setting for the startup behavior only needs to be changed if the network devices are to use static addresses instead.
Auto MDI-X	No (MDI)	Not editable. If the C-884.xDC is to be connected directly to the PC via TCP/IP, a crossover cable must be used.

### RS-232

C-884.xDC's interface settings for RS-232 communication:

Interface setting	Factory setting	Note
Baud rate (RSBAUD)	115200	Possible values: 9600, 19200, 38400, 115200 To establish communication successfully, the baud rates of the C-884.xDC and PC must match.

## 4.7 Software Overview

A data storage device with the PI Software Suite is included in the C-884.xDC's [scope of delivery](#) (p. 16). Some components of the PI Software Suite are described in the table below. For information on the compatibility of the software with PC operating systems see the C-990.CD1 Release News in the root directory of the data storage device.

### Libraries, drivers

PC software	Operating system	Short description	Recommended use
Dynamic program-libraries for GCS	Windows, Linux	Allows software programming of the C-884.xDC with programming languages such as C++. The functions in dynamic program libraries are based on the PI General Command Set (GCS).	For users who would like to use a dynamic program library for their application. Is required for PIMikroMove. Is required for NI LabVIEW drivers if communication is to be established via USB (with Linux only via virtual COM port) or a daisy chain network.
Drivers for use with NI LabVIEW software	Windows, Linux	NI LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments). The NI LabVIEW software from PI is a collection of virtual instrument drivers (VI drivers) for PI controllers. These drivers support GCS commands.	For users who wish to use NI LabVIEW to program their application.
MATLAB drivers	Windows	MATLAB is a development environment and programming language for numerical calculations (must be ordered separately from MathWorks).  The PIMATLAB driver consists of a MATLAB class that can be included in any MATLAB script. This class supports the PI General Command Set. The PI MATLAB driver does not require any additional MATLAB toolboxes.	For users who wish to use MATLAB to program their application.
USB driver	Windows	Driver for the USB interface	For users who want to connect the controller to the PC via the USB interface.
PIPython	Windows, Linux, OSX	Python is a programming language that is also used as a script language (obtainable separately as open-source software).	For users who want to use Python for programming scripts for their applications.

PC software	Operating system	Short description	Recommended use
		<p>PIPython is a collection of Python modules that support GCS.</p> <p>PIPython modules can be used with Python version 3.6 or higher.</p> <p>Use on other operating systems is also possible via sockets.</p>	



### User software

PC software	Operating system	Short description	Recommended use
PIMikroMove	Windows	<p>Graphical user interface for Windows, which can be used for controllers from PI:</p> <ul style="list-style-type: none"> <li>■ Start the system without programming effort</li> <li>■ Graphic representation of the motion</li> <li>■ Macro functionality for storing command sequences on the PC (host macros)</li> <li>■ Complete environment for command entry</li> </ul> <p>PIMikroMove uses the dynamic program library to supply commands to the controller.</p>	<p>For users who want to perform simple automation tasks or test their equipment before or instead of programming an application.</p> <p>No command knowledge is necessary to operate PIMikroMove.</p> <p>A log window showing the commands sent makes it possible to learn how to use the commands.</p>
PIStages3Editor	Windows	Program opening and editing positioner databases in .db format.	For users who want to deal with the contents of positioner databases more intensively.
PITerminal	Windows, Linux	Simple user interface that can be used for nearly all PI controllers.	For users who want to send GCS commands directly to the controller.
PIFirmwareManager	Windows	Program for updating the firmware of the C-884.xDC	For users who want to update the firmware
PIUpdateFinder	Windows	Checks the PI software installed on the PC. If newer versions of the PC software are available on the PI server, they are offered for download.	For users who want to update the PC software.

## 4.8 Positioner database

You can select a parameter set appropriate for your positioner from a positioner database in the PC software from PI. The software transfers the values of the selected parameter set to the controller's volatile or nonvolatile memory.

Database file name	Description
PISTAGES3.DB	Delivery includes parameter sets for all standard positioners from PI and PI miCos, and is saved to the PC automatically during installation of the PC software. New parameter sets can be created, edited, and saved.
Product code.db	Includes the parameter set for the custom positioner "product code". In order for the parameter set to be selectable in the PC software, it must be <a href="#">imported (p. 50)</a> into PISTAGES3.DB first.

The positioner database only contains some of the information that is required to operate a positioner with the C-884.xDC. When the positioner connected to the C-884.xDC is equipped with an ID chip: Further information is loaded as parameter values from the ID chip to the volatile memory of the C-884.xDC when the C-884.xDC is switched on or rebooted.

4.9 ID Chip Detection

PI positioners with DC motor have an ID chip in the connector where the following data is stored as parameters:

- Information on the positioner: Type, serial number, date of manufacture, version of the hardware
- Signal type output by the position sensor

The data of the connected positioner is loaded from the ID chip into the volatile memory of the C-884.xDC when the C-884.xDC is switched on or rebooted.

The ID chip only contains some of the information that is required to operate the positioner with the C-884.xDC. When you use the PC software from PI, further information is loaded as parameter values from a positioner database into the volatile memory of the C-884.xDC.

The parameter values in the C-884.xDC's volatile memory can be queried and written to the nonvolatile memory (see: [Adapting Settings \(p. 111\)](#)).

4.10 Functional Principles of the C-884.xDC

4.10.1 Block Diagram

The C-884.xDC control motion of the positioner's logical axes. The following block diagram shows how the C-884.xDC generates the output signal for the axis connected:

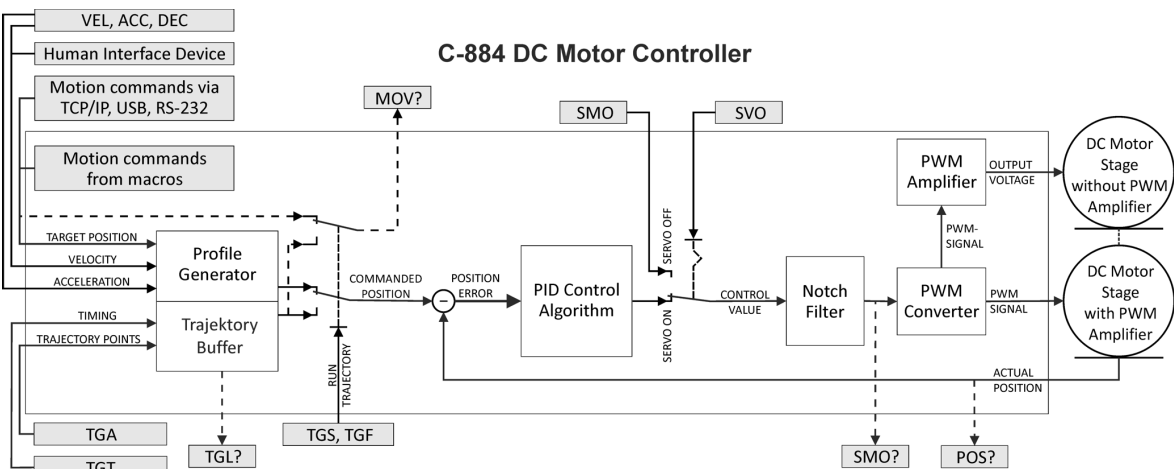


Figure 4: C-884: Generating the output signal

The C-884.xDC supports positioners with and without PWM amplifier. The output voltage level for positioners without PWM amplifier depends on the supply voltage provided by the power adapter connected to the C-884.xDC.

#### 4.10.2 Important Firmware Components

The functional units of the C-884.xDC's firmware are described in the following.

The firmware can be updated with a tool. The current firmware version can be ascertained with the [VER?](#) command.

Component	Description
Parameters	<p>Parameters reflect the properties of the positioner connected (e.g., travel range) and specify the behavior of the C-884.xDC (e.g., settings for the servo algorithm).</p> <p>The parameters can be divided into the following categories:</p> <ul style="list-style-type: none"> <li>■ Protected parameters whose default settings cannot be changed</li> <li>■ Parameters that must be set by the user to adapt to the application</li> </ul> <p>In the case of positioners with ID chip, the values of some parameters are stored on the ID chip. They are loaded to the volatile memory when switching on or rebooting the C-884.xDC.</p> <p>Command levels determine the write permission for the parameters. The current command level can be queried with the <a href="#">CCL?</a> command and changed with the <a href="#">CCL</a> command. This may require entering a password.</p> <p>The list of parameters available in the C-884.xDC can be queried with the <a href="#">HPA?</a> command.</p> <p>Refer to <a href="#">Adapting Settings (p. 111)</a> for more information on parameters</p>
ASCII commands (GCS)	<p>Communication with the C-884.xDC can be made with <a href="#">commands (p. 120)</a> from the PI General Command Set (GCS). The <a href="#">GCS syntax version (p. 109)</a> can be queried with the <a href="#">CSV?</a> command.</p> <p>Examples of the use of GCS:</p> <ul style="list-style-type: none"> <li>■ Configuring the C-884.xDC</li> <li>■ Setting the operating mode</li> <li>■ Starting positioner motion</li> <li>■ Getting system and position values</li> </ul> <p>The list of commands available in the C-884.xDC can be queried with the <a href="#">HLP?</a> command.</p>
Profile generator	<p>The profile generator does calculations for point-to-point motion in closed-loop operation to specify the target position, velocity, and acceleration of an axis to any point during motion. The result is the dynamics profile.</p> <p>Refer to <a href="#">Generating the Dynamics Profile (p. 30)</a> for further information</p>
Trajectory buffer	<p>For motion along freely definable paths, externally calculated trajectory points (target points) are loaded into the trajectory buffer of the C-884.xDC. The trajectory points are followed in closed-loop operation according to a specified chronological interval.</p> <p>Refer to: <a href="#">Trajectories for Motion Paths (p. 77)</a> for further information</p>
Servo algorithm	<p>Closed-loop operation: The position error that results from the difference between the commanded target position and the actual position (sensor feedback) runs through a servo algorithm.</p> <p>For information, see: <a href="#">Servo Algorithm and Other Control Value Corrections (p. 33)</a></p>

Component	Description
Data recorder	The C-884.xDC contains a real-time data recorder. The data recorder can record various signals (e.g., position, control value) from different data sources (e.g., logical axes). Refer to <a href="#">Data Recorder (p. 80)</a> for information
Macros	The C-884.xDC can save macros. Command sequences can be defined and stored permanently in the nonvolatile memory of the device via the macro function. A startup macro can be defined that runs each time the C-884.xDC is switched on or rebooted. The startup macro simplifies stand-alone operation (operation without a connection to the PC). Refer to <a href="#">Controller Macros (p. 99)</a> for information

### Commands

C		Page
CCL	Set Command Level	129
CCL?	Get Command Level	129
CSV?	Get Current Syntax Version	130
H		Page
HLP?	Get List Of Available Commands	151
HPA?	Get List Of Available Parameters	152
V		Page
VER?	Get Versions Of Firmware And Drivers	183

### Parameters

#### 4.10.3 Commandable Items

The following table contains the elements of the C-884.xDC that can be commanded with GCS commands.

Element	Number	ID	Description
Logical axes	4 / 6	1 to 4 / 6	<p>A logical axis represents the motion of the mechanics in C-884.xDC's firmware. It corresponds to an axis of a linear coordinate system. Motion for logical axes is commanded in the C-884.xDC's firmware (i.e., for the positioner's direction of motion).</p> <p>The axis identifier can be queried with the <a href="#">SAI?</a> command and modified with the <a href="#">SAI</a> command. It can consist of up to 8 characters; valid characters are:</p> <p>1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ_</p> <p>The positioner type connected to an axis can be queried with the <a href="#">CST?</a> command. If the <b>Stage Name</b> parameter (0x3C) has the value NOSTAGE, the axis is "deactivated". A deactivated axis is not accessible for axis-related commands (e.g., motion commands or position queries). The identifier of a deactivated axis can only be queried with <a href="#">SAI? ALL</a>.</p>
Trajectories	4 / 6	1 to 4 / 6	<p>The number of trajectories corresponds to the number of logical axes. Each trajectory is permanently allocated to a logical axis.</p> <p>Trajectories are commanded with TG* commands. Refer to: <a href="#">Trajectories for Motion Paths (p. 77)</a> for further information</p>
Analog inputs	4	1 to 4	<p>The analog input lines with the identifiers 1 to 4 are the inputs 1 to 4 of the <b>I/O</b> socket. Their number is displayed with the <a href="#">TAC?</a> command and their values can be queried with the <a href="#">TAV?</a> command.</p> <p>The values of all inputs can be recorded via record option 81 of the <a href="#">DRC</a> command.</p> <p>Refer to: <a href="#">Analog Input Signals (p. 90)</a> for further information</p>
Digital outputs	4	1 to 4	<p>1 to 4 identify digital output lines 1 to 4 of the <b>I/O</b> socket.</p> <p>For information, see: <a href="#">Digital Output Signals (p. 83)</a></p>
Digital inputs	4	1 to 4	<p>1 to 4 identify digital input lines 1 to 4 of the <b>I/O</b> socket, which can also be used as analog inputs (see above).</p> <p>For information, see: <a href="#">Digital Input Signals (p. 88)</a></p>
HID (Human Interface Device)	5	1 to 5	<p>Up to five human interface devices can be connected to the USB socket (type A) of the C-884.xDC via a USB hub.</p> <p>The number of commandable axes, buttons, and LEDs per HID depends on the HID connected. After an HID has been connected, information on the commandable axes, buttons, and LEDs can be queried with the <a href="#">HIS? command</a>.</p>

Element	Number	ID	Description
			Refer to: <a href="#">Controlling with an HID (p. 92)</a> for further information
Data recorder tables	8	1 to 8	The C-884.xDC has 8 data recorder tables (query with <a href="#">TNR?</a> ) with 8192 data points per table. For information, see: <a href="#">Data Recorder (p. 80)</a>
Overall system	1	1	C-884.xDC as an overall system Information on name, serial number, and firmware version of the C-884.xDC can be queried with the <a href="#">*IDN?</a> command. The ready state of the C-884.xDC can be queried with the <a href="#">#7</a> command.

### Commands

#		Page
#7	Request Controller Ready Status	125
*		Page
*IDN?	Get Device Information	126
C		Page
CST?	Get Assignment Of Stages To Axes	130
D		Page
DRC	Set Data Recorder Configuration	137
H		Page
HIS?	Get Configuration Of HI Device	150
S		Page
SAI	Set Current Axis Identifiers	170
SAI?	Get List Of Current Axis Identifiers	171
T		Page
TAC?	Tell Analog Channels	176
TAV?	Get Analog Input Voltage	176
TNR?	Get Number Of Record Tables	180

### Parameters

0x3C	Stage Name	Positioner name. Default value: NOSTAGE String up to 20 characters
------	------------	--

#### 4.10.4 Physical Units

The C-884.xDC supports various units of length for positions. Adapting is done by a factor that converts the sensor counts into the desired physical unit of length. The conversion

factor is set with the **Numerator Of The Counts-Per-Physical-Unit Factor** parameter (0xE) and **Denominator Of The Counts-Per-Physical-Unit Factor** parameter (0xF).

The unit symbol can be customized for display purposes with the **Axis Unit** parameter (0x7000601).

Examples:

- 1 Encoder count = 100 nm  
Counts per physical unit of length: 10000:1  
→ unit symbol: mm
- 1 Encoder count = 0.254 mm  
Counts per physical unit of length: 100:1  
→ unit symbol: inch

## Commands

## Parameters

0xE	Numerator Of The Counts-Per-Physical-Unit Factor	<p>Numerator of the factor for converting impulses to physical units.</p> <p>The factor for the counts per physical unit of length specifies the unit of length for position queries and motion commands in closed-loop operation. The values of every parameter, whose unit is either the physical unit of length itself or a unit of measurement based on it, are automatically adapted to the set factor. The factor for the counts per physical unit of length has no impact on the stability of the servo loop but is used for the input and output scaling of position values.</p> <p>Decimal number</p>
0xF	Denominator Of The Counts-Per-Physical-Unit Factor	<p>Denominator of the factor for converting impulses to physical units.</p> <p>The factor for the counts per physical unit of length specifies the unit of length for position queries and motion commands in closed-loop operation. The values of every parameter, whose unit is either the physical unit of length itself or a unit of measurement based on it, are automatically adapted to the set factor. The factor for the counts per physical unit of length has no impact on the stability of the servo loop but is used for the input and output scaling of position values.</p> <p>Decimal number</p>
0x7000601	Axis Unit	<p>Unit symbol.</p> <p>For example, the unit symbol is "mm", if the factor for the counts per physical unit of length is set with parameters 0xE and 0xF so that the encoder counts are converted into millimeters. The unit for rotation stages is normally "deg".</p> <p>String up to 20 characters</p>



### 4.10.5 Operating Modes

The C-884.xDC supports the following operating modes:

- Closed-loop operation (servo mode On)
- Open-loop operation (servo mode Off)

#### Closed-loop operation

The commanded position for the axes comes from one of the two following sources:

- [Dynamics profile \(p. 30\)](#)

A profile generator calculates the dynamics profile from the values specified for target position, velocity, acceleration, and deceleration.

- [Trajectory buffer \(p. 77\):](#)

The motion follows a path made up of points that were externally calculated and loaded to the C-884.xDC (target positions) and that are travelled according to a specified chronological interval.

The position error resulting from the difference between the commanded target position and the actual position (sensor feedback) runs through a P-I-D servo algorithm (proportional-integral-derivative). Additional corrections can be made as well.

The result is the control value for the PWM amplifier integrated in the C-884.xDC.

Further information: [Servo Algorithm and Other Control Value Corrections \(p. 33\)](#)

#### Open-loop operation

The C-884.xDC does not calculate a dynamics profile in open-loop operation and a trajectory cannot be followed.

The C-884.xDC does **not** evaluate the signals of the position sensor. As a result, the positioner can move unbraked to the end of the travel range and, despite the limit switch function, strike the hard stop.

#### Information

The C-884.xDC is intended for closed-loop operation with incremental or absolute position sensors (servo mode On). After switch-on, open-loop operation is active by default (servo mode Off).

- ▶ Query the current operating mode with the [SV0?](#), [#4](#) or [SRG?](#) commands.
- ▶ Activate closed-loop operation with the [SV0](#) command.
- ▶ If necessary, program a startup macro that starts the C-884.xDC in closed-loop operation via the [SV0](#) command.
- ▶ Avoid motion in open-loop operation.

#### Commands

#		Page
#4	Request Status Register	124
S		Page
SRG?	Query Status Register Value	174
SV0	Set Servo Mode	175
SV0?	Get Servo Mode	176

#### Parameters

### 4.10.6 Generating a Dynamics Profile

The profile generator of the C-884.xDC is used for point-to-point motion in closed-loop operation. The profile generator does calculations to specify the target position, velocity, and acceleration of the axis for each point in time during motion (dynamics profile). The values calculated are called commanded values.

The dynamics profile generated by the profile generator of the C-884.xDC depends on the motion variables that are specified by parameters and/or HID.

#### Information

When following trajectories, the profile generator does not calculate a dynamics profile.

#### Variables for Generating a Dynamics Profile

##### Acceleration (A)

Com- mands	Parameters	Remarks
<a href="#">ACC</a> <a href="#">ACC?</a>	0xB <b>Closed-Loop Acceleration (Phys. Unit/s<sup>2</sup>)</b> Change using the ACC command or <a href="#">SPA</a> / <a href="#">SEP</a> ; query with ACC?	Is limited by the <b>Maximum Closed-Loop Acceleration (Phys. Unit/s<sup>2</sup>)</b> parameter (0x4A). The maximum acceleration during HID control is specified by the <b>Closed-Loop Acceleration For HI Control (Phys. Unit/s<sup>2</sup>)</b> parameter (0x75).

##### Deceleration (D)

Com- mands	Parameters	Remarks
<a href="#">DEC</a> <a href="#">DEC?</a>	0xC <b>Closed-Loop Deceleration (Phys. Unit/s<sup>2</sup>)</b> Change with the DEC command or with <a href="#">SPA</a> / <a href="#">SEP</a> ; query with DEC?	Wird durch Parameter <b>Maximum Closed-Loop Deceleration (Phys. Unit/s<sup>2</sup>)</b> (0x4B) begrenzt. The maximum deceleration during HID control is specified by the <b>Closed-Loop Deceleration For HI Control (Phys. Unit/s<sup>2</sup>)</b> parameter (0x76).

##### Velocity (V)

Com- mands	Parameters	Remarks
<a href="#">VEL</a> <a href="#">VEL?</a>	0x49 <b>Closed-Loop Velocity (Phys. Unit/s)</b> Change with the VEL command or with <a href="#">SPA</a> / <a href="#">SEP</a> ; query with VEL?	Is limited by the <b>Maximum Closed-Loop Velocity (Phys. Unit/s)</b> (0xA) parameter. The maximum velocity during HID control is always specified by the <b>Closed-Loop Velocity For HI Control (Phys. Unit/s)</b> parameter (0x74).

#### Target position at the end of the motion

Com-mands	Parameters	Remarks
<a href="#">MOV</a> <a href="#">MVR</a> <a href="#">MVE</a> <a href="#">STE</a>	--	<p>The soft limits are set as the respective target position during HID control of the velocity.</p> <p>The C-884.xDC sets the target position to the current position of the axis in the following cases:</p> <ul style="list-style-type: none"><li>■ Disabling HID control for the axis</li><li>■ Switching servo mode on with the <i>SV0</i> command</li><li>■ Stopping the motion with the <i>#24</i>, <i>STP</i>, or <i>HLT</i> commands</li></ul>

Explanation of the Dynamics Profile

The profile generator of the C-884.xDC only supports trapezoidal velocity profiles: The axis accelerates linearly (based on the acceleration value specified) until it reaches the specified velocity. It continues to move at this velocity until it decelerates linearly (based on the deceleration value specified) and stops at the specified target position.

The following applies in the figures: A = acceleration, D = deceleration, V = velocity

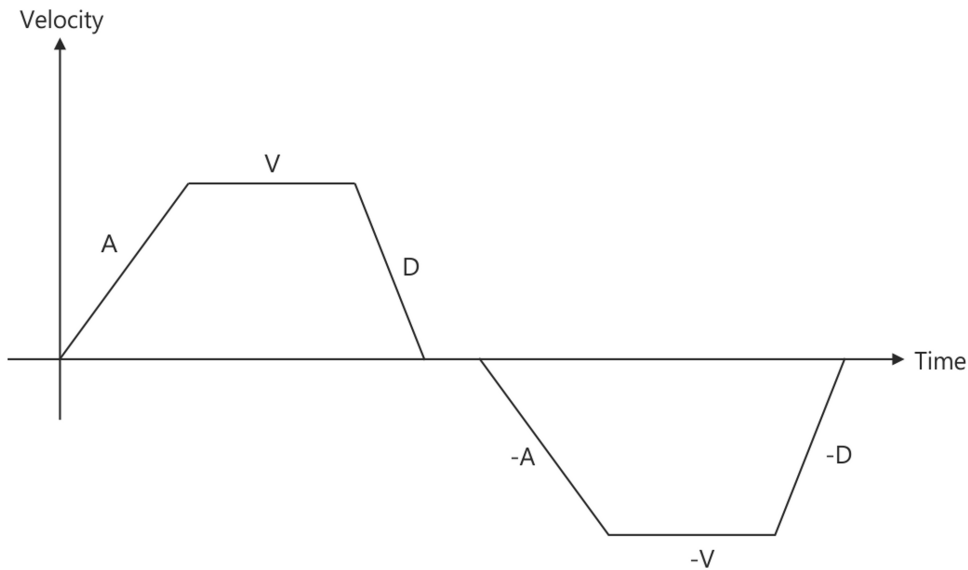


Figure 5: Simple trapezoid velocity profile

If deceleration has to begin before the axis reaches the specified velocity, the profile will not have a constant velocity portion and the trapezoid becomes a triangle.

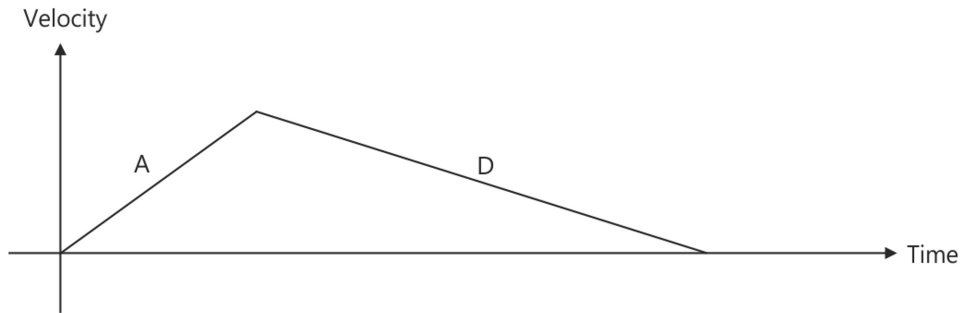


Figure 6: Simple trapezoid velocity profile without constant velocity

The edges for acceleration and deceleration can be symmetrical (acceleration = deceleration) or asymmetrical (acceleration  $\neq$  deceleration). The acceleration value is always used at the start of the motion. After that, the acceleration value is used during an increase in the absolute velocity and the deceleration value during a decrease in the absolute velocity. If none of the motion variables are changed during the course of motion, the acceleration value is used until the maximum velocity is reached and the deceleration value is used for decreasing the velocity down to zero.

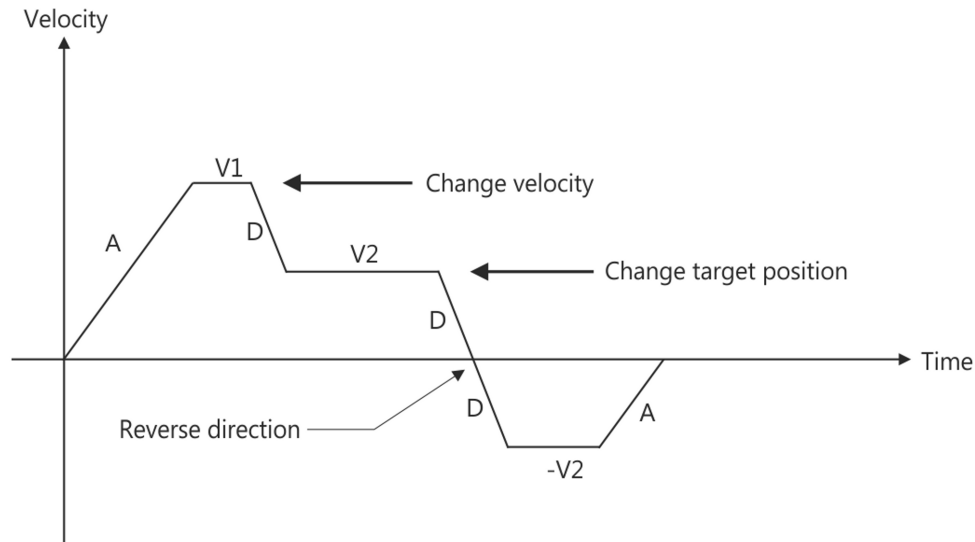


Figure 7: Complex trapezoidal profile with parameter changes

All motion variables can be changed while the axis is in motion. The profile generator will always attempt to stay within the permissible motion limits specified by the motion variables. If the target position is changed during motion so that overshooting is unavoidable, the profile generator will decelerate to a complete stop and reverse the direction of motion in order to reach the specified position.

## Commands

A		Page
ACC	Set Closed-Loop Acceleration	127
ACC?	Get Closed-Loop Acceleration	127
D		Page
DEC	Set Closed-Loop Deceleration	134
DEC?	Get Closed-Loop Deceleration	134
M		Page
MOV	Set Target Position	164
MVE	Set Target Position For Vectorial Move	165
MVR	Set Target Relative To Current Position	166
S		Page
SEP	Set Nonvolatile Memory Parameters	171
SPA	Set Volatile Memory Parameters	173
STE	Start Step And Response Measurement	175

V		Page
VEL	Set Closed-Loop Velocity	183
VEL?	Get Closed-Loop Velocity	183

#### Parameters

0xB	Closed-Loop Acceleration (Phys. Unit/s <sup>2</sup> )	Acceleration with dynamics profile in closed-loop operation. Limited by parameter 0x4A. 0 to value of 0x4A [phys. units. / s <sup>2</sup> ]
0x4A	Maximum Closed-Loop Acceleration (Phys. Unit/s <sup>2</sup> )	Maximum acceleration in closed-loop operation with dynamics profile. Specifies the maximum value for parameter 0xB.
0x75	Closed-Loop Acceleration for HI Control (Phys. Unit/s <sup>2</sup> )	Maximum acceleration during HID control.
0xC	Closed-Loop Deceleration (Phys. Unit/s <sup>2</sup> )	Deceleration with dynamics profile in closed-loop operation. Limited by parameter 0x4B. 0 to value of 0x4B [phys. units. / s <sup>2</sup> ]
0x4B	Maximum Closed-Loop Deceleration (Phys. Unit/s <sup>2</sup> )	Maximum deceleration in closed-loop operation with dynamics profile. Specifies the maximum value for parameter 0xC.
0x76	Closed-Loop Deceleration for HI Control (Phys. Unit/s <sup>2</sup> )	Maximum deceleration during HID control.
0x49	Closed-Loop Velocity (Phys. Unit/s)	Velocity in closed-loop operation with dynamics profile. Limited by parameter 0xA. 0 to value of 0xA [phys. units / s]
0xA	Maximum Closed-Loop Velocity (Phys. Unit/s)	Maximum velocity with dynamics profile in closed-loop operation. Specifies the maximum value for parameter 0x49.
0x74	Closed-Loop Velocity For HI Control (Phys. Unit/s)	Maximum velocity during HID control.

#### 4.10.7 Servo Algorithm and Other Control Value Corrections

In closed-loop operation, the control value for the PWM amplifier integrated in the C-884.xDC and therefore the settling behavior of each axis in the system is optimized by the following corrections:

- Servo algorithm: The position error resulting from the difference between the commanded position (from [dynamics profile \(p. 30\)](#) or [trajectory \(p. 77\)](#)) and the actual position (sensor feedback), runs through a PID servo algorithm (proportional-integral-derivative).
- Control value corrections: The dynamics profile or the trajectory can be subjected to offset correction and feed-forward velocity control.

Regardless of the operating mode, the control value can be subjected to an additional correction via the notch filter.

### Settings for the servo algorithm

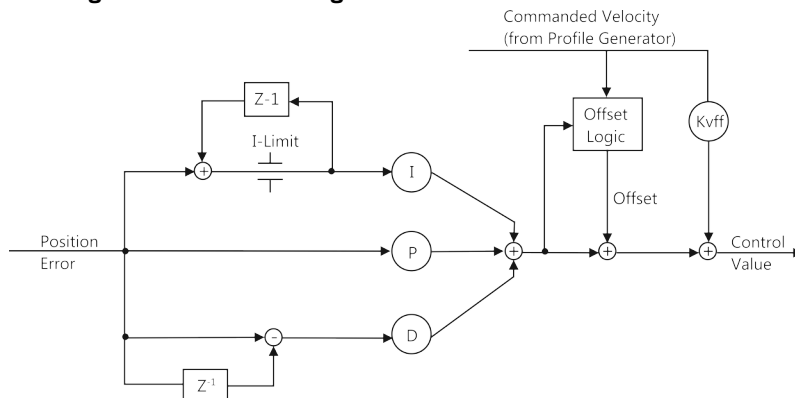


Figure 8: PID algorithm, offset correction, and feed-forward velocity control (KVff); the notch filter is not shown here

The PID servo algorithm uses the following servo control parameters. The optimal servo control parameter setting depends on your application and your requirements (refer to ["Optimizing Servo Control Parameters" \(p. 70\)](#)).

- **P Term** (0x1)
- **I Term** (0x2)
- **D Term** (0x3)
- **I-Limit** (0x4)
- **D-Term Delay (No. Of Servo Cycles)** (0x71)

Servo algorithm input can be configured for the C-884.xDC with the following parameters:

- **Numerator Of The Servo-Loop Input Factor** (0x5A)
- **Denominator Of The Servo-Loop Input Factor** (0x5B)

### Settings for the servo algorithm

Offset correction for the control value for closed-loop operation can be configured via the parameters listed below:

- **Motor Offset Positive** (0x33)
- **Motor Offset Negative** (0x34)
- **Motor Drive Offset** (0x48)

Velocity feed-forward can be configured via the following parameter:

- **Kvff** (0x5)

### Settings for the Notch Filter

The notch filter corrects the control value. It filters the part of the frequency that would cause undesired resonance in the mechanics. The notch filter can be configured with the following parameters: (refer to [Setting the Notch Filter \(p. 66\)](#)):

- **Notch Filter Frequency 1 (Hz)** (0x94)
- **Notch Filter Edge 1** (0x95)

### Commands

### Parameters

0x1	P term	Proportional constant of the PID servo algorithm. Is used for fast correction of the position error. 0 to 65535
0x2	I term	Integration constant of the PID servo algorithm. Used for reducing static position error. 0 to 65535
0x3	D term	Differential constant of the PID servo algorithm. Used for damping rapid control oscillation. The D term can be calculated as a floating average over several servo cycles. The parameter 0x71 (D-Term Delay) specifies how many values (i.e., servo cycles) are to be used for calculating the average. 0 to 65535 The preset value of this parameter should not be changed.
0x4	I limit	Limit of the integration constant. 0 to 1048575
0x71	D-Term Delay (No. Of Servo Cycles)	D term delay. Determines how many values (i.e., servo cycles) are used for calculating the mean value of the D term. 0 to 15. The preset value of this parameter should not be changed.
0x5A	Numerator Of The Servo-Loop Input Factor	Input factor numerator for the servo loop. Decimal number
0x5B	Denominator Of The Servo-Loop Input Factor	Input factor denominator for the servo loop. Decimal number
0x33	Motor Offset Positive	Drive offset for the positive direction of motion. 0 to 32767
0x34	Motor Offset Negative	Drive offset for the negative direction of motion. 0 to 32767
0x48	Motor Drive Offset	Velocity-dependent drive offset. Used if the commanded velocity does not equal zero (i.e., if the end of the dynamics profile has not been reached). 0 to 32767
0x5	Kvff	Feed-forward control of the commanded velocity. Used for minimizing positioning error. 0 to 65535
0x94	Notch Filter Frequency 1 (Hz)	Frequency of the first notch filter. The appropriate frequency component is reduced in the control value to compensate for undesired resonances in the mechanics. 40 to 20,000 [Hz]

0x95	Notch Filter Edge 1	Edge steepness of the first notch filter. The greater the value of this parameter, the narrower the notch filter bandwidth. 0.1 to 10 (dimensionless)
------	---------------------	---

4.10.8 Triggering Motion

Motion is triggered in **closed-loop** operation either by commands or an HID, e.g., a joystick. Following a trajectory and HID control are not possible in **open-loop** operation. Triggering motion with motion commands can be configured via the **Inhibit Motion Commands** parameter (0x130).

The motion status of the axes connected to the C-884.xDC can be queried with the [#5](#) command.

Motion triggered by commands can be **stopped** with the following commands:

- [#24, STP](#): Abrupt stop
- [HLT](#): Gentle stop

In both cases, the error code 10 is set for information.

HLT also triggers an abrupt stop when trajectories are being followed.

Triggering motion in closed-loop operation

Motion is triggered in closed-loop operation either by commands or an HID, e.g., a joystick. Motion commands and following trajectories are not permitted when HID control is activated for the axis.



Trigger of the motion	Commands	Description
Commands for point-to-point motion sent from the command line or via the PC software	<a href="#">MOV</a> , <a href="#">MVR</a> , <a href="#">MVE</a>	Motion to absolute or relative target position
	<a href="#">GOH</a>	Motion to zero position
	<a href="#">STE</a>	Jumps a specified distance and records the response
	<a href="#">FRF</a>	Starts reference moves
	<a href="#">FED</a>	Starts moves to signal edges
Controller macros with motion commands	<a href="#">MAC</a>	<p>Calls a macro function. Permits recording, deleting, and running macros on the controller.</p> <p>Any commands can be sent from the command line while a macro is running on the controller. The macro content and motion commands received from the command line can overwrite each other.</p>
HID control	<a href="#">HIN</a>	Activates or deactivates control of the C-884.xDC's axes via the HID's axes.
	<a href="#">HIA</a>	<p>Configures HID control for the axes of the C-884.xDC. The following motion parameters of the C-884.xDC's axes can be controlled via the HID's axes:</p> <ul style="list-style-type: none"> <li>■ Velocity of the axis</li> <li>■ Maximum velocity of the axis</li> </ul>
Trajectory execution	<a href="#">TGS</a>	Starts the execution of an externally calculated trajectory

### Information

Absolute target positions can only be commanded for positioners with incremental position sensor when the axis has been previously referenced, refer to [Referencing \(p. 42\)](#).

### Triggering motion in open-loop operation

Following trajectories and HID control are not possible in open-loop operation.

Motion is triggered by the following commands:

Commands	Description
<a href="#">SMO</a>	Directly sets the control value for the drive electronics in the C-884.xDC.

### Commands

#		Page
#24	Stop All Axes	126
#5	Request Motion Status	125
F		Page
FED	Find Edge	141

FRF	Fast Reference Move To Reference Switch	142
<b>G</b>		<b>Page</b>
GOH	Go To Home Position	143
<b>H</b>		<b>Page</b>
HIA	Configure Control Done By HID Axis	145
HIN	Set Activation State For HID Control	148
HLT	Halt Motion Smoothly	152
<b>M</b>		<b>Page</b>
MAC	Call Macro Function	159
MOV	Set Target Position	164
MVE	Set Target Position For Vectorial Move	165
MVR	Set Target Relative To Current Position	166
<b>S</b>		<b>Page</b>
SMO	Set Open-Loop Control Value	172
STE	Start Step And Response Measurement	175
STP	Stop All Axes	175
<b>T</b>		<b>Page</b>
TGS	Start Trajectory	178

#### Parameters

0x130	Inhibit Motion Commands	Configures the triggering of motion via motion commands. The setting is evaluated for closed-loop and open-loop operation. The value of the parameter has no influence on HID control of the C-884.xDC's axis (axes). Decimal number, bit-mapped. The values can be added.
-------	-------------------------	---

#### 4.10.9 On-Target State

In closed-loop operation, the on-target state can be used to check whether the target position has been reached:

- On-target state = true (1): The target position is considered as reached
- On-target state = false (0): The target position is considered as not reached

The C-884.xDC determines the on-target state on the basis of the following criteria:

- Settling window around the target position (**Settling Window (Encoder Counts)** parameter (0x36))
- Delay time for setting the on-target state (**Settling Time (s)** parameter (0x3F))

The on-target state has the value true in the following cases:

- The current position is inside the settling window and stays there at least for the duration of the delay time.
- If the value for the delay time is set to 0: The current position is in the settling window.

The on-target state can be read with the [ONT?](#), [#4](#) and [SRG?](#) commands.

## Commands

#		Page
#4	Request Status Register	124
O		Page
ONT?	Get On-Target State	166
S		Page
SRG?	Query Status Register Value	174

## Parameters

0x36	Settling Window (Encoder Counts)	Settling window around the target position. Presets the window limits (half of the window width). If the current position enters the settling window, the target position is considered as reached. If the current position exits the settling window, the target position is no longer considered as reached. Can be changed only if servo mode is switched off. 0 to maximum value. The maximum value depends on the controller but may not exceed 231 encoder counts.
0x3F	Settling Time (s)	Delay time for setting the on-target state. Recommended values: 0.000 to 1.000 seconds

### 4.10.10 Reference Switch Detection

The C-884.xDC receives signals from the reference switch at its [axis connector \(p. 232\)](#). The [TRS?](#) command can be used to determine whether an axis has a reference switch with direction sensing.

The signal from the reference switch of the positioner can be used for reference moves. In the case of a positioner with incremental position sensor, the controller knows the absolute axis position after a reference move ([Referencing \(p. 42\)](#)).

The following parameters can be used to configure how the C-884.xDC detects the reference switch:

- [Invert Reference?](#) (0x31)
- [Has Reference?](#) (0x14)
- [Reference Signal Type](#) (0x70)

## Commands

T		Page
TRS?	Indicate Reference Switch	181

## Parameters

0x31	Invert Reference?	Should the reference signal be inverted? Inverts the signal of the reference switch or a digital input that is used instead of the reference switch. 0 Reference signal not inverted 1 Reference signal inverted
0x14	Has Reference?	Do the mechanics have a reference switch? Activates respectively deactivates reference moves to the installed reference switch. 0 No reference switch 1 Reference switch available
0x70	Reference Signal Type	Reference signal type.

#### 4.10.11 Detecting Limit Switches

The C-884.xDC receives signals from limit switches at its [axis connector \(p. 232\)](#). The [LIM?](#) command can be used to determine whether an axis has limit switches.

The signals from the limit switches (also end-of-travel sensors) of a linear stage are used to stop motion in front of the hard stop at both ends of the travel range. In addition, [soft limits \(p. 41\)](#) can be set via parameters of the C-884.xDC.

The limit switch signals can also be used for reference moves. In the case of a positioner with incremental position sensor, the controller knows the absolute axis position after a reference move ([Referencing \(p. 42\)](#)).

The following parameters can be used to configure how the C-884.xDC detects the limit switches:

- [Limit Mode \(0x18\)](#)
- [Has No Limit Switches? \(0x32\)](#)
- [Use Limit Switches Only For Reference Moves? \(0x77\)](#)

##### Commands

L	Page
<a href="#">LIM?</a> Indicate Limit Switches	158

##### Parameters

0x18	Limit Mode	Signal logic of the limit switches. 0 pos-HI, neg-HI 1 pos-LO, neg-HI 2 pos-HI, neg-LO 3 pos-LO, neg-LO
0x32	Has No Limit Switches?	Do the mechanics not have limit switches? Activates motion stop at the installed limit switches. 0 Mechanics have limit switches 1 Mechanics do not have limit switches
0x77	Use Limit Switches Only For Reference Moves?	Should the limit switches only be used for reference moves? Is intended for use with rotation stages. Only evaluated when parameter 0x32 has the value 0.

## 4.10.12 Travel Range and Soft Limits

The physical limits of the travel range can be represented by the following items of a positioner:

- Limit switches
- If the positioner does not have integrated limit switches: Hard stops

### Settings for the travel range

The following parameters of the C-884.xDC reflect the physical travel range of the positioner:

- **Value At Reference Position (Phys. Unit) (0x16)**
- **Distance From Negative Limit To Reference Position (Phys. Unit) (0x17)**
- **Distance From Reference Position To Positive Limit (Phys. Unit) (0x2F)**

### Settings for Soft Limits

The C-884.xDC supports two parameter pairs for establishing soft limits. They are intended for different applications:

- **Maximum Travel In Positive Direction (Phys. Unit) (0x15)** and **Maximum Travel In Negative Direction (Phys. Unit) (0x30)**:
  - The limits establish the permissible travel range in closed-loop operation.
  - Motion commands are executed only if the commanded position is within these soft limits.
  - The limits always refer to the current zero position.
  - Appropriate values are loaded when the positioner is selected from the positioner database.
- **Range Limit Min (0x70000000)** und **Range Limit Max (0x70000001)**:
  - Using these limits is recommended only if open-loop motion is required. For logical reasons, the values are outside the soft limits specified via 0x15 and 0x30.
  - The limits apply both in closed-loop and open-loop operation.
  - Motion is stopped abruptly once the current position reaches a limit.
  - The limits are independent of the current zero position.
  - The values are not loaded from the positioner database and are set in the default settings so that the limits are deactivated.

### Commands

### Parameters

<b>0x16</b>	Value At Reference Position (Phys. Unit)	Position value at the reference switch. The current position is set to this value if the axis has performed a reference move to the reference switch. The parameter value is also used for calculating the position values set after reference moves to the limit switches; this also applies when the mechanics do not have a reference switch.
<b>0x17</b>	Distance From Negative Limit To Reference Position (Phys. Unit)	Gap between reference switch and negative travel range limit. The current position is set to the difference between the values of parameters 0x16 and 0x17 if the axis has done a reference move to the negative travel range limit.

<b>0x2F</b>	Distance From Reference Position To Positive Limit (Phys. Unit)	Gap between the reference switch and the positive travel range limit. If the axis has done a reference move to the positive travel range limit, the current position is set to the sum of the values of parameters 0x16 and 0x2F.
<b>0x15</b>	Maximum Travel In Positive Direction (Phys. Unit)	Soft limit in positive direction, in relation to the zero position. The positive travel range limit cannot be used for reference moves if this value is smaller than the position value for the positive travel range limit (that results from the sum of parameters 0x16 and 0x2F). The value can be negative.
<b>0x30</b>	Maximum Travel In Negative Direction (Phys. Unit)	Soft limit in a negative direction, in relation to the zero position. The negative travel range limit cannot be used for reference moves if this value is greater than the position value for the negative travel range limit (that results from the difference between parameters 0x16 and 0x17). The value can be negative.
<b>0x7000000</b>	Range Limit Min	Additional soft limit for the negative direction of motion (physical unit). If the current position reaches this value in either closed-loop or open-loop operation, the control value is set to zero and the motion is stopped as a result. The axis can move again as soon as the value for the soft limit has been decreased. The values are not loaded from the positioner database and are set in the default settings so that the limits are deactivated.
<b>0x7000001</b>	Range Limit Max	Additional soft limit for the positive direction of motion (physical unit). If the current position reaches this value in either closed-loop or open-loop operation, the control value is set to zero and the motion is stopped as a result. The axis can move again as soon as the value for the soft limit has been increased. The values are not loaded from the positioner database and are set in the default settings so that the limits are deactivated.

#### 4.10.13 Referencing

Whether referencing is necessary for the axis depends on the signal type of the position sensor:

- Absolute measuring position sensor: Referencing is not necessary.
- Incremental position sensor: Referencing is necessary.

Incremental sensors only supply relative motion information. When the positioner is equipped with an incremental position sensor, the controller does not know the absolute position of the axis when it is switched on or rebooted. Referencing must be done beforehand so that absolute target positions can be commanded and reached.

### Referencing options

Referencing can be done in different ways:

- **Reference move (default):** A reference move moves the axis to a specifically defined point, e.g., to the reference switch, to a limit switch or to a hard stop. At this point, the current position is set to a defined value. The controller now knows the absolute axis position. The reference move is started with the [FRF](#) command.
- **Setting the absolute position manually:** If referencing method was activated by the [RON](#) command, you can set the current position of the axis to an arbitrary value at an arbitrary point using the [POS](#) command. The axis is not moved here. The controller knows the absolute axis position afterwards.

[FRF?](#) can be used to query whether an axis is referenced. [RON?](#) can be used to query the referencing method for an axis.

When PIMikroMove is used for starting, a reference move is done by default. Knowledge of the commands and parameters described here is not needed for referencing using PIMikroMove.

### Reference Move Procedure

To achieve maximum repeatability when referencing, each reference move comprises the following steps:

1. First move to the selected switch.  
The maximum velocity is specified by the [Closed-Loop Velocity \(Phys. Unit/s\)](#) parameter (0x49) and the default direction for the reference move by the [Reference Travel Direction](#) parameter (0x47).
2. Stop when the switch edge has been reached.  
The higher the velocity on approach, the farther the axis overruns the edge of the switch (overshooting).
3. Move in the opposite direction to compensate for overshoot.
4. Second move to the selected switch.  
The maximum velocity is specified by the [Velocity For Reference Moves \(Phys. Unit/s\)](#) parameter (0x50).
5. Stop on reaching the switch edge.
6. Move in the opposite direction to compensate for overshoot.
7. Set the current position to a defined value.  
Referencing is finished.

The lower the velocity is when approaching the switch, the less the overshoot will be, and the higher the repeatability. Therefore, the maximum value of parameter 0x50 should be as large as the value of parameter 0x49, though ideally substantially less.

The actual velocities during the reference move are calculated from the values of the following parameters and can be lower than the maximum values:

- [Closed-Loop Velocity \(Phys. Unit/s\)](#) parameter (0x49) or [Velocity For Reference Moves \(Phys. Unit/s\)](#) parameter (0x50)
- [Distance Between Limit And Hard Stop \(Phys. Unit\)](#) parameter (0x63)
- [Closed-Loop Deceleration \(Phys. Unit/s<sup>2</sup>\)](#) parameter (0xC)

The [Distance From Limit To Start Of Ref. Search \(Phys. Unit\)](#) parameter (0x78) and [Distance For Reference Search \(Phys. Unit\)](#) parameter (0x79) are used for reference moves when both of the following conditions are fulfilled:

- The reference move is started with [FRF](#).
- The [Reference Signal Type](#) parameter (0x70) has the value 2 or 3.

Reference move sequence:

1. The axis moves to the corresponding limit switch.
2. The axis moves away from the limit switch by the distance specified in parameter 0x78

3. The axis moves to the index pulse and covers the maximum distance specified by parameter 0x79.

### Commands

F		Page
FRF	Fast Reference Move To Reference Switch	142
FRF?	Get Referencing Result	142
P		Page
POS	Set Real Position	167
R		Page
RON	Set Reference Mode	168
RON?	Get Reference Mode	169

### Parameters

0x49	Closed-Loop Velocity (Phys. Unit/s)	Velocity in closed-loop operation with dynamics profile. Limited by parameter 0xA. 0 to value of 0xA [phys. units / s]
0x47	Reference Travel Direction	Default direction for the reference move. 0 Automatic detection 1 Negative direction 2 Positive direction
0x50	Velocity For Reference Moves (Phys. Unit/s)	Maximum velocity for reference moves. 0 to value of value 0x49 [phys. units / s]
0x63	Distance Between Limit And Hard Stop (Phys. Unit)	Gap between internal limit switch and hard stop. Determines the maximum stopping distance during reference moves. The actual velocities during a reference move are calculated on the basis of this value, the set deceleration (0xC) and set velocities (0x49 and 0x50).
0xC	Closed-Loop Deceleration (Phys. Unit/s <sup>2</sup> )	Deceleration with dynamics profile in closed-loop operation. Limited by parameter 0x4B. 0 to value of 0x4B [phys. units. / s <sup>2</sup> ]
0x78	Distance From Limit To Start Of Ref Search (Phys. Unit)	Distance between the limit switch and the starting position for the reference move to the index pulse. Used for FRF when parameter 0x70 has the value 2.
0x79	Distance For Reference Search (Phys. Unit)	Maximum distance for motion to the index pulse.
0x70	Reference Signal Type	Reference signal type.



## 5 Unpacking / Transportation

### 5.1 Unpacking

#### Unpacking the C-884.xDC

1. Unpack the C-884.xDC with care.
2. If the C-884.xDC was supplied with ESD protective caps on the connectors: Do **not** remove the ESD protective caps.
3. Compare the contents with the scope of delivery according to the contract and the delivery note.
4. Inspect the contents for signs of damage. If any parts are damaged or missing, contact our [customer service department \(p. 10\)](#) immediately.
5. Keep all packaging materials in case the product needs to be returned.

### 5.2 Transportation

Pay attention to the [ambient conditions and classifications \(p. 230\)](#) when transporting the C-884.xDC.

1. Pack the C-884.xDC in the original packaging.
2. If the C-884.xDC is to be sent, use a stable outer box.

## 6 Installation

### 6.1 Mounting the C-884.xDC

The C-884.xDC can be used as a benchtop device or mounted on a surface in any orientation or installed in a control cabinet.

The housings of the C-884.4DC and .6DC models are identical.

#### Overview

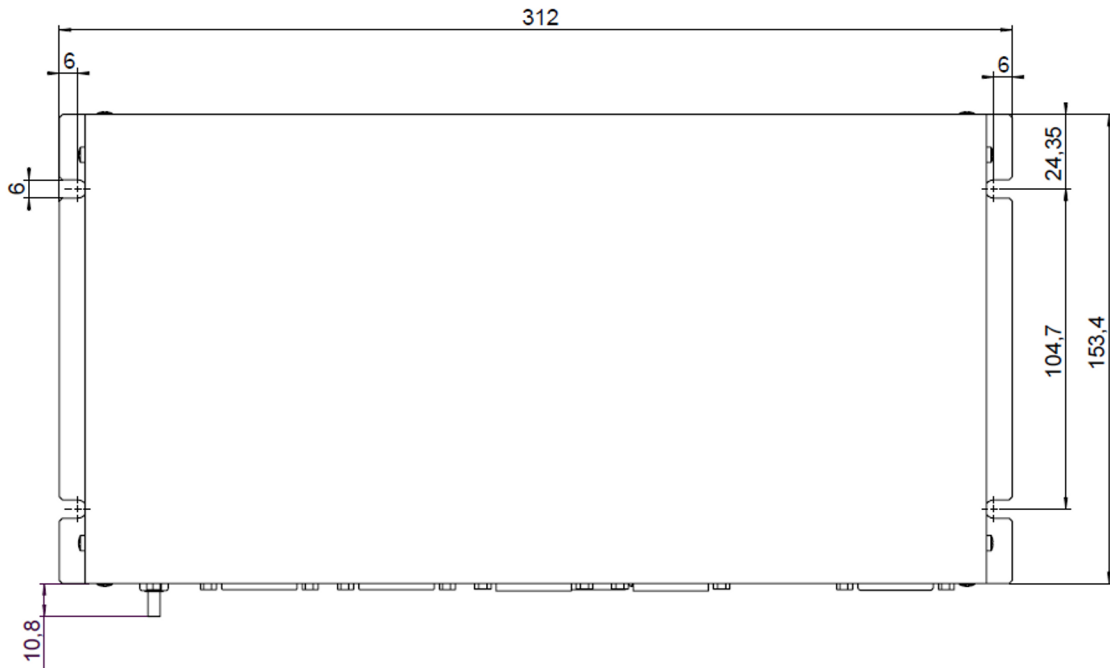


Figure 9: C-884.xDC: Layout of the mounting holes

#### Tools and Accessories

- Suitable screws
- Suitable screwdriver

#### Requirements

- ✓ You have read and understood the [general safety instructions \(p. 12\)](#).

#### NOTICE



#### Heating up of the C-884.xDC during operation!

High temperatures can overheat the C-884.xDC.

- ▶ Install the C-884.xDC with a gap of at least 10 cm to the top and rear panels and at least 5 cm to its sides. If this is not possible, make sure that the surroundings are cooled sufficiently.
- ▶ Ensure sufficient ventilation at the place of installation.
- ▶ Keep the ambient temperature at a noncritical level.

#### Mounting the C-884.xDC on a Surface

1. Drill the holes required into the surface.  
The arrangement of the recesses in the mounting rails of the C-884.xDC can be seen in the figure.

2. Tighten the C-884.xDC to the recesses provided with suitable screws.

## 6.2 Connecting the C-884.xDC to the Protective Earth Conductor

The C-884.xDC is not grounded via the voltage connection and must therefore be connected to the protective earth conductor. The protective earth connector is on the C-884.xDC's front panel.

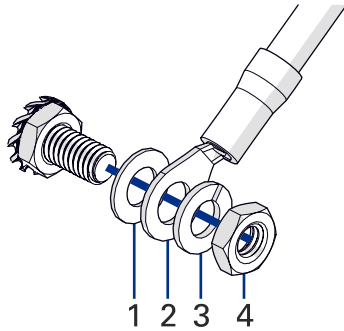


Figure 10: Connecting the protective earth conductor to the threaded bolt

1. Flat washer
2. Cable lug of the protective earth conductor
3. Spring washer
4. Nut

### Tools and Accessories

- Suitable protective earth conductor with cable lug:
  - Cable cross section  $\geq 0.75 \text{ mm}^2$
  - Contact resistance  $< 0.1 \text{ ohm}$  at 25 A at all connection points relevant for attaching the protective earth conductor
- Suitable wrench

### Requirements

- ✓ You have read and understood the [general safety instructions \(p. 12\)](#).
- ✓ The C-884.xDC is **not** connected to the power supply.

### Information

- Pay attention to the applicable standards for connecting the protective earth conductor.

### Connecting the C-884.xDC to the Protective Earth Conductor

1. If necessary, attach a suitable cable lug to the protective earth conductor.
2. Remove the outer nut and the spring washer from the threaded bolt of the protective earth connector.
3. Push the cable lug of the protective earth conductor and the spring washer onto the threaded bolt.
4. Screw the nut onto the threaded bolt again.
  - *The cable lug of the protective earth conductor is clamped between the flat washer and the spring washer.*
5. Tighten the nut with at least three turns and a torque of 1.2 Nm to 1.5 Nm.

## 6.3 Connecting the Power Supply to the C-884.xDC

### Tools and Accessories

- Included power supply (alternative: Sufficiently rated power supply)
- Adapter included for power supply connection (alternative: Sufficiently rated adapter)
- Included power cord (alternative: Sufficiently rated power cord)

### Requirements

- ✓ The power supply is **not** connected to the power socket via the power cord.
- ✓ The C-884.xDC is installed near the power supply so that the power plug can be quickly and easily disconnected from the mains.

### NOTICE



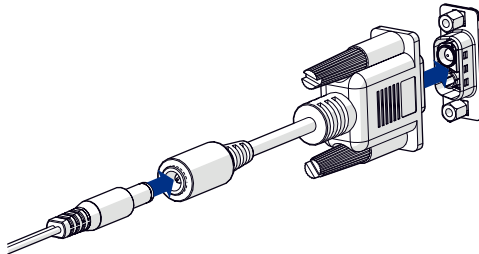
#### Motor damage due to excessively high operating voltage!

The output voltage on the C-884.xDC's [axis connectors \(p. 232\)](#) can be as high as the voltage supplied by the power adapter. Positioners without PWM amplifier are connected to the voltage supply and could be damaged by excessive voltage.

- ▶ Connect a power adapter whose output voltage does not exceed the permissible operating voltage of the positioner.
- ▶ Adjust the C-884.xDC's maximum output voltage to the positioner's permissible operating voltage with the **Maximum Motor Output** parameter (0x9).

### Connect the Power Supply to the C-884.xDC

1. Connect the Sub-D 2W2C of the adapter to the voltage connection of the C-884.xDC.
2. Secure the adapter against unintentional removal of the C-884.xDC.
3. Connect the barrel connector of the adapter to the barrel connector socket of the power supply.



4. Connect the power cord to the power supply.

## 6.4 Connecting the Positioner to the C-884.xDC

### Tools and Accessories

- Compatible positioner with DC motor or voice coil drive
- If necessary: Suitable adapter cable from PI, available as optional accessory for the positioner
- If the distance between the C-884.xDC and the positioner is too long: Suitable drive or extension cable from PI, available as optional accessory for the positioner

### Requirements

- ✓ The power adapter is **not** connected to the power socket via the power cord or the C-884.xDC is switched off.
- ✓ You have read and understood the user manual for the positioner to be connected.

- ✓ You have installed the positioner to be connected according to the instructions in its respective user manual.

### NOTICE



#### Damage if a wrong drive type is connected!

Connecting a positioner with incompatible drive type can cause irreparable damage.

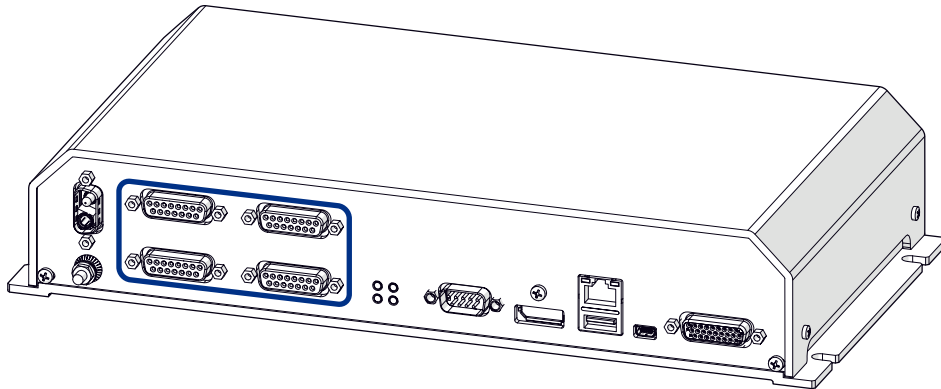
- Connect the C-884.xDC only to positioners with DC motor or voice coil drive.

### Information

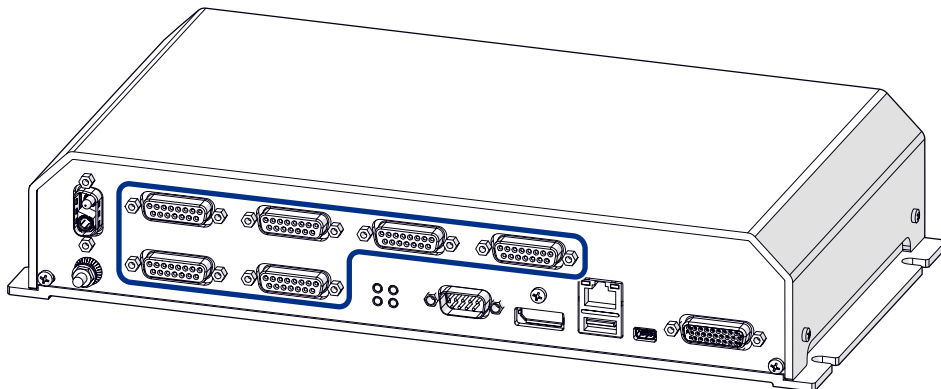
The C-884.xDC supports both positioners with PWM amplifier and positioners without PWM amplifier. Separate lines at the C-884.xDC's **DC Motor #** sockets are available for both positioner variants; refer to [Connector Pin Assignments \(p. 232\)](#). Make sure that suitable lines are selected via the positioner's connector. Proper connection is ensured for positioners from PI.

### Connecting the Positioner to the C-884.xDC

1. Connect the positioner to one of the following sockets on the C-884.xDC:
  - C-884.4DC: **DC Motor 1** to **DC Motor 4**



- C-884.6DC: **DC Motor 1** to **DC Motor 6**



2. Use the integrated screws to secure the connections against accidental disconnection.

## 6.5 Installing the PC Software

### 6.5.1 Installing the PC Software for the First Time

#### Tools and Accessories

- PC with [Windows or Linux \(p. 18\)](#) operating system and at least 30 MB free storage space

- Software from PI: On the storage device supplied or available for download from our website [www.pi.de](http://www.pi.de).
- Optional for custom positioners: Data storage device or archive file with the following content:
  - **Import PI CustomStage** program
  - Custom positioner database with the parameter set for the positioner

#### Installing the PC software in Windows

1. Insert the data storage device into the PC or go to the directory where you saved the downloaded software to.
2. Start the install wizard by double-clicking **PISoftwareSuite.exe**.  
→ *The **InstallShield Wizard** window opens for installing the PC software from PI.*
3. Follow the instructions on the screen.  
The PI software suite includes the following components:
  - Driver for use with NI LabVIEW software
  - Dynamic program library for GCS
  - PIMikroMove
  - PC software for updating the firmware of the C-884.xDC
  - PIUpdateFinder for updating the PC software
  - USB driver

#### Installing a Custom Positioner Database in Windows

If you have a **custom positioner database**, this must also be installed on the PC.

1. Insert the data storage device into the PC or go to the directory where you saved the downloaded software to.
2. Run the installation assistant for the custom positioner database by clicking **Import\_PI\_CustomStage.exe**.  
→ *The **Import PI Custom Stage** program is run and the parameter set is imported from the custom positioner database into PIStages3.*
3. If a message appears that installation of the custom positioner database failed:
  - a) [Update the PIStages3 positioner database on your PC \(p. 51\)](#).
  - b) Repeat the installation of the custom positioner database.

#### Installing the PC Software in Linux

1. Insert the data storage device into the PC or go to the directory where you saved the downloaded software to.
2. Unzip the tar archive from the **/linux** directory into a directory on your PC.
3. Open a terminal and go to the directory where you unpacked the tar archive.
4. Log in as superuser (root privileges).
5. Enter **./INSTALL** to start the installation.  
Pay attention to lower and upper case when entering commands.
6. Follow the instructions on the screen.
7. If you have received a **custom positioner database**: Copy the positioner database file into the following directory: **/usr/local/PI/pi\_gcs\_translator/**

### 6.5.2 Updating the PC Software

PI is constantly improving the PC software. Always install the latest version of the PC software and the positioner database.

PIUpdateFinder is a program which helps you find updates for your PI software on the PI server. It identifies the PI software installed on your computer and compares it with the software available on the PI server. This comparison is made using the Internet. If newer software versions are available on the PI server, you can download the software via a corresponding link.

Confidential customer data is not transmitted when comparing the software versions. The following information is transmitted:

- Software component and version
- Internet browser
- Operating system
- IP address

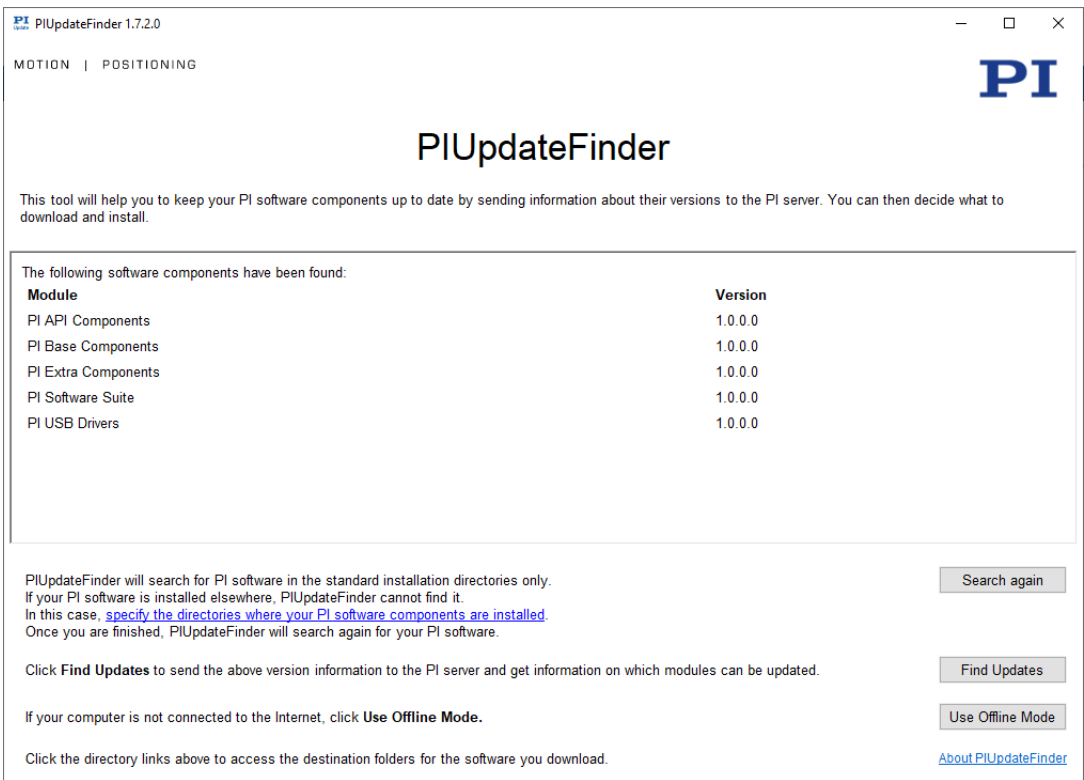
## Updating the PC Software and Positioner Database in Windows

### Requirements

- ✓ You have installed the PIUpdateFinder on the PC to be updated.
- ✓ In addition when updating the software on a PC without Internet connection:
  - PC with Internet connection
  - Portable data storage device, e.g., USB stick, for the data exchange between the computers

### Updating the Software on a PC with Internet Connection

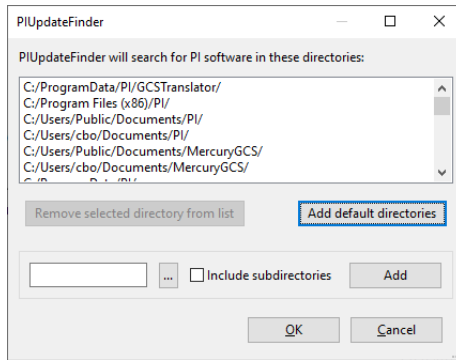
1. Start PIUpdateFinder on the PC to be updated.
- A table is displayed with information on the PI software installed on your computer.



If the table is empty or incomplete, proceed as follows:

- a) Click ***specify the directories where your PI software components are installed***.
 

→ A dialog window is opened, in which all directories are listed that PIUpdateFinder finds while searching for and comparing the software versions.



- b) Click the **...** button and select the directory on your hard disk, where the PI software is installed.  
 → *The directory is displayed in the input field at the bottom left of the window. You can also manually enter directories there.*
  - c) Activate the **Include subdirectories** checkbox to include subdirectories as well.
  - d) Click the **Add** button.  
 → *The directory, and any subdirectories, appear at the end of the list.*
  - e) Click **OK** to finalize input of the installation directory.  
 → *If PI software is found in the specified directories, it will be displayed in the table on the initial screen of the PIUpdateFinder.*
2. Click the **Find Updates** button.  
 → *A browser window opens and a table is displayed with the software information. Updates are available from the column **Download Link** via the **PI server** link.*

### PI Update Finder



PI Software Found on Your System

Name	Installed Version	Version on PI Server	Release Notes	Download Link
PI API Components	V1.0.0.0	V1.1.0.0	<a href="#">Release note</a>	<a href="#">PI server</a>
PI Base Components	V1.0.0.0	V1.1.0.0	<a href="#">Release note</a>	<a href="#">PI server</a>
PI Extra Components	V1.0.0.0	V1.1.0.0	<a href="#">Release note</a>	<a href="#">PI server</a>
PI Software Suite	V1.0.0.0	V1.1.0.0	<a href="#">Release note</a>	<a href="#">PI server</a>
PI USB Drivers	V1.0.0.0	V1.1.0.0	<a href="#">Release note</a>	<a href="#">PI server</a>

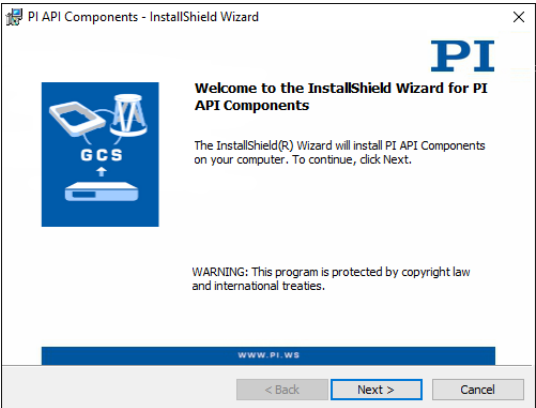
**Important note:**  
Before downloading and installing your update, read the corresponding release note to check if there are any known compatibility issues.

If you have any questions or problems please [contact us](#).  
 © 2011-2018 by [Physik Instrumente \(PI\) GmbH & Co. KG](#)

3. To download an update, click **PI server** in the **Download Link** column.  
 → *You are then asked to specify whether the file is to be downloaded or run directly. Depending on the browser settings, it is possible that the update file is downloaded directly. It is then in the download directory of your computer.*
4. Activate the option in the browser query to save the update file to the your computer's hard disk.  
 → *The file is saved to the download directory of your computer. Depending on the browser settings, it is also possible to run the file directly or open it. In this case, continue with step 6.*
5. Install the update:
  - Executable setup files (.exe): Run the downloaded update file.
  - Data archive (.zip): Unpack the archive to a directory on your PC. Open this directory and run the desired setup file (.exe).



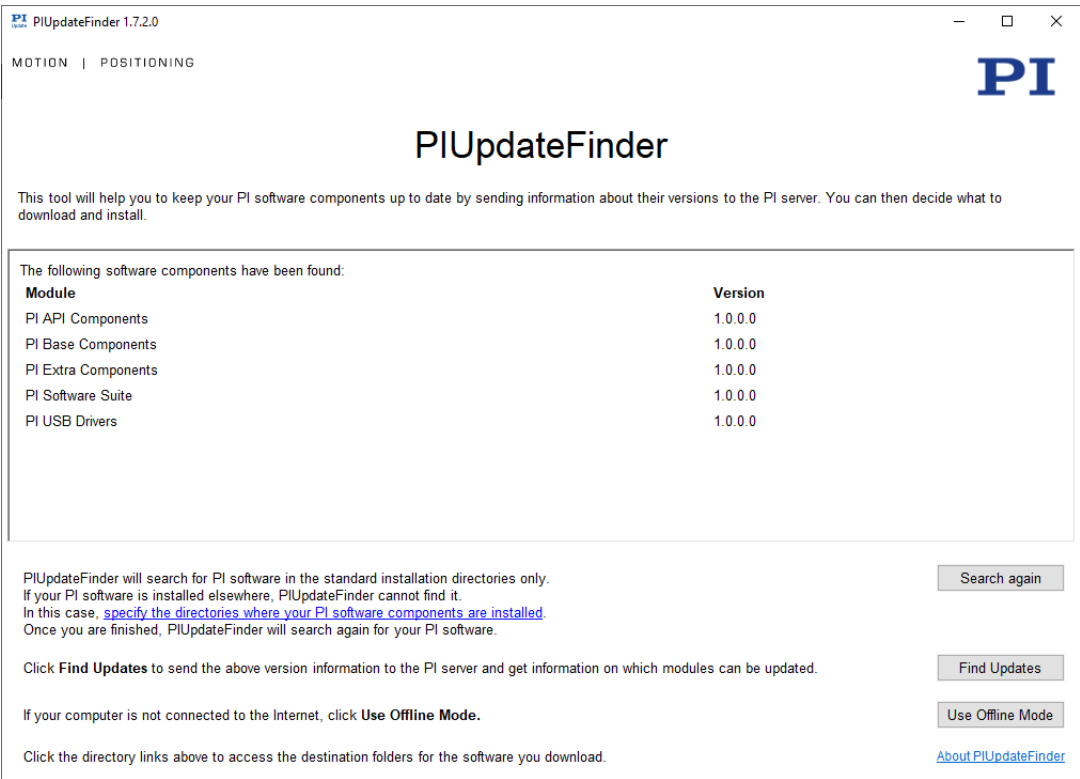
You may have to confirm running the file.  
→ *This opens the InstallShield Wizard, e.g., for PI API Components:*



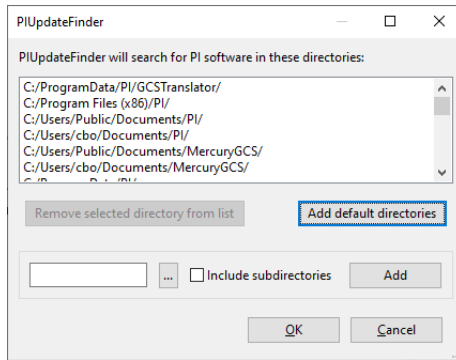
6. Follow the instructions in the InstallShield Wizard.

**Updating the Software on a PC without Internet Connection**

- 1. Connect the portable data storage device, e.g., an USB stick, with the PC to be updated.
  - 2. Start PIUpdateFinder on the PC to be updated.
- *A table is displayed with information on the PI software installed on your computer.*



If the table is empty or incomplete, proceed as follows:  
a) Click ***specify the directories where your PI software components are installed.***  
→ *A dialog window is opened, in which all directories are listed that PIUpdateFinder finds while searching for and comparing the software versions.*



b) Click the ... button and select the directory on your hard disk, where the PI software is installed.

→ *The directory is displayed in the input field at the bottom left of the window. You can also manually enter directories there.*

c) Activate the **Include subdirectories** checkbox to include subdirectories as well.

d) Click the **Add** button.

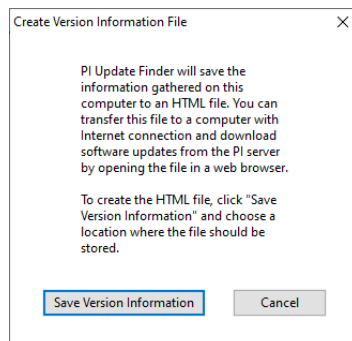
→ *The directory, and any subdirectories, appear at the end of the list.*

e) Click **OK** to finalize input of the installation directory.

→ *If PI software is found in the specified directories, it will be displayed in the table on the initial screen of the PIUpdateFinder.*

3. In the main window, click the **Use Offline Mode** button.

→ *The window **Create Version Information File** will open.*



4. Click the **Save Version Information** button to save the software version information to a HTML file on your portable data storage device.

5. Transfer the HTML file to a computer connected to the Internet.

6. Open the HTML file in a web browser on the computer connected to the Internet.

→ *A browser window opens and a table is displayed with the software information. Updates are available from the column **Download Link** via the **PI server** link.*

7. Download your updates and save them to your portable data storage device.

8. Transfer the updates to the PC without Internet connection.

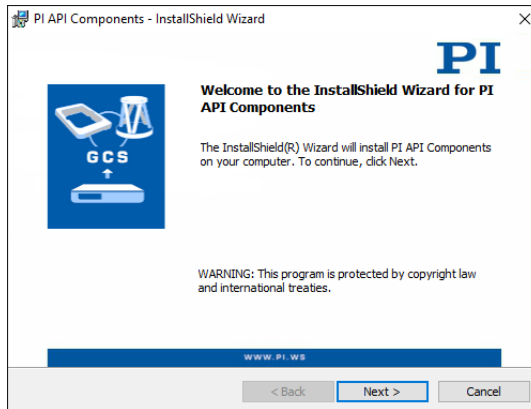
9. Install the updates.

■ Executable setup files (.exe): Run the downloaded update file.

■ Data archive (.zip): Unpack the archive to a directory on your PC. Open this directory and run the desired setup file (.exe).

You may have to confirm running the file.

→ *This opens the InstallShield Wizard, e.g., for PI API Components:*



10. Follow the instructions in the InstallShield Wizard.

## Updating the PC Software and Positioner Database in Linux

### Requirements

- ✓ Active connection to the Internet.

### Updating the PC software in Linux

1. Open the website [www.physikinstrumente.com/en/products/software-suite](http://www.physikinstrumente.com/en/products/software-suite)
2. Scroll down to **Downloads**.
3. Click **ADD TO LIST** for the **PI Software Suite C-990.CD1**.
4. Click **REQUEST**.
5. Fill out the download request form and send the request.  
→ *The download link will then be sent to the e-mail address entered.*
6. Via the link download the archive file with the **PI Software Suite**.
7. Unpack the archive file on your PC to a separate directory.
8. Go to the **linux** subdirectory in the directory with the unpacked files.  
You will find an additional archive file there.
9. Unpack the archive file in the **linux** directory by entering the command `tar -xvpf <name of the archive file>` on the console.
10. Log into the PC as superuser (root privileges).
11. Install the update.
12. If you also received an updated PISTAGES3.DB database from our customer service department: Install that update onto the PC.

## 6.6 Connecting the PC

Communication between the C-884.xDC and a PC is required to configure the C-884.xDC and to command motion using the GCS commands.

### Information

The C-884.xDC's communication interfaces are active at the same time. Commands are executed in the order in which the complete command lines arrive. However, simultaneous use of several communication interfaces can cause problems with the PC software.

- Always only use one interface of the C-884.xDC.

### 6.6.1 Connecting the C-884.xDC to a PC

#### Tools and Accessories

- Suitable cable for the selected communication interface (in the [scope of delivery \(p. 16\)](#) or available as [optional accessory \(p. 16\)](#))

#### Requirements

- ✓ The PC has a connection available for the selected communication interface.
- ✓ If necessary: The [interface parameters of the PC have been adapted for the C-884.xDC \(p. 17\)](#).

#### Connecting the C-884.xDC to the PC

1. Connect the cable to the selected communication interface of the C-884.xDC.
2. Connect the cable to an unused port on the PC.

### 6.6.2 Integrating the C-884.xDC into a Network

#### Tools and Accessories

- Suitable network cable (in the [scope of delivery \(p. 16\)](#) or available as [optional accessory \(p. 16\)](#))

#### Requirements

- ✓ A network access point is available for the C-884.xDC.
- ✓ The [default settings of the TCP/IP interface \(p. 18\)](#) do not collide with the settings of other devices in the network.

#### Integrating the C-884.xDC into a TCP/IP Network

1. Connect the RJ45 socket of the C-884.xDC to the network cable.
2. Connect the network cable to the network access point.

## 7 Startup and Operation

### 7.1 Switching on the C-884.xDC

#### Requirements

- ✓ You have read and understood the [general safety instructions \(p. 12\)](#).
- ✓ The C-884.xDC has been installed [properly \(p. 46\)](#).

#### Switching the C-884.xDC On

1. Switch the C-884.xDC on by connecting the power cord of the power adapter to the power socket.  
→ *The C-884.xDC performs the following steps:*
  - The C-884.xDC boots the firmware. During the booting process, the **PWR** and **STA** LEDs on the C-884.xDC flash alternately.
  - The C-884.xDC loads information from the nonvolatile memory into the volatile memory.
2. Wait until the booting process has completed.  
→ *The LEDs indicate the status of the C-884.xDC:*
  - LED **PWR** lights up continuously: The firmware has booted.
  - LED **STA** lights up continuously green: Information has loaded and the C-884.xDC is ready for normal operation.
3. If the **PWR** and **STA** LEDs do not light up after switching on, contact our [customer service department \(p. 10\)](#).

### 7.2 Establishing Communication with the PC

The procedure for PIMikroMove is described in the following.

The figures show the procedure for any electronics; the procedure for the C-884.xDC corresponds.

#### 7.2.1 Establishing Communication via RS-232

To establish communication successfully the PC's and [C-884.xDC's baud rates \(p. 18\)](#) must match.

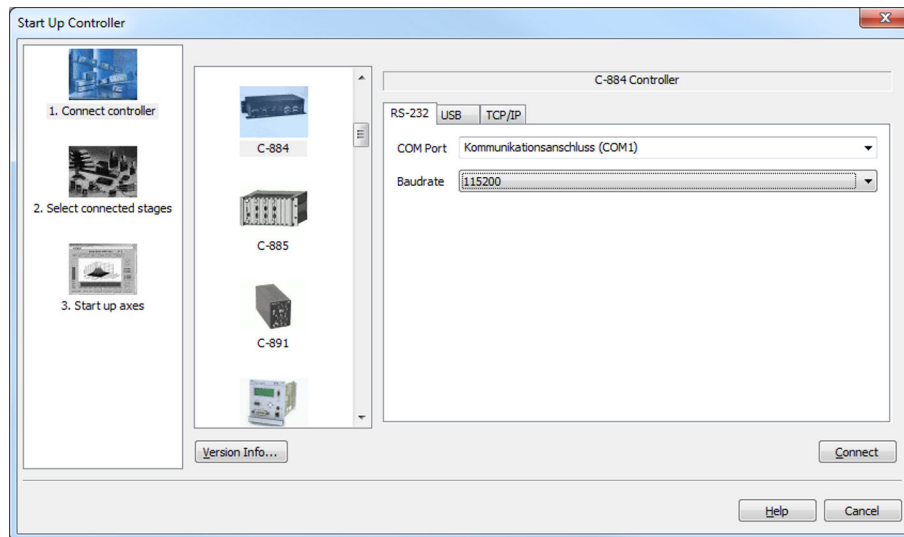
#### Requirements

- ✓ The C-884.xDC is connected to the [RS-232 interface of the PC \(p. 56\)](#).
- ✓ The C-884.xDC is [switched on \(p. 57\)](#).
- ✓ The PC is switched on.
- ✓ The required software and drivers are [installed \(p. 49\)](#) on the PC.
- ✓ You have read and understood the manual for the PC software. The links to the software manuals are in the A000T0081 file on the PI software data carrier.

#### Establishing Communication via RS-232

1. Start PIMikroMove.  
→ *The **Start up controller** window opens with the **Connect controller** step.*
2. If the **Start up controller** window does not automatically open, select the **Connections > New...** menu item in the main window.
3. Select **C-884** in the field for controller selection.

4. Select the **RS-232** tab on the right-hand side of the window.
  5. Make the following settings in the **RS-232** tab:
    - a) In the **COM Port** field, select the COM port of the PC to which you have connected the C-884.xDC.
    - b) In the **Baudrate** field, set the value that is set for the C-884.xDC.
- This adapts the baud rate of the PC to the baud rate of the C-884.xDC.



6. Click **Connect** to establish communication.
  - If communication could not be established, look for a solution to the problem in the "[Troubleshooting \(p. 226\)](#)" chapter.
  - If communication was established successfully, PIMikroMove guides you through the configuration of the C-884.xDC for the connected positioner, see "[Starting Motion \(p. 62\)](#)".

### 7.2.2 Establishing Communication via USB

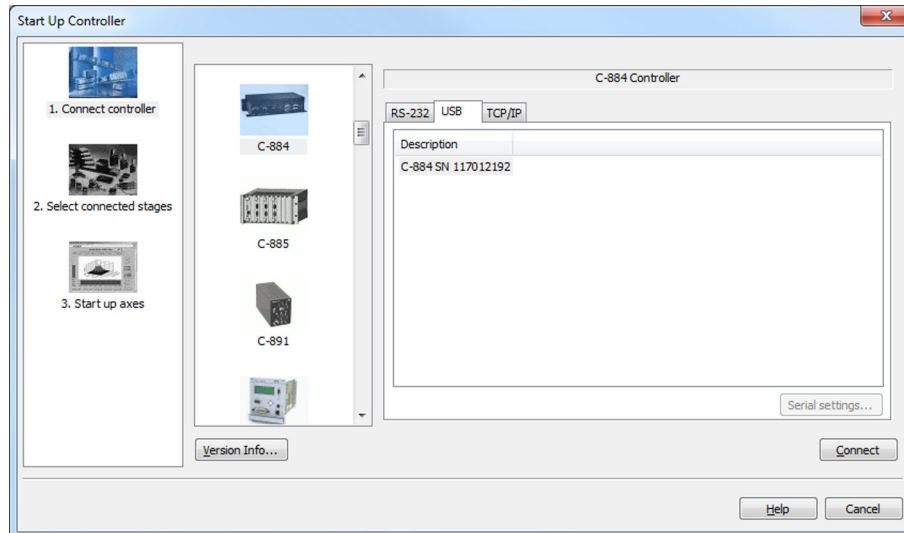
If the controller is connected via the USB connection and switched on, the USB interface in the PC software is also shown as a COM port.

#### Requirements

- ✓ The C-884.xDC is connected to the USB interface of the [PC \(p. 56\)](#).
- ✓ The C-884.xDC is [switched on \(p. 57\)](#).
- ✓ The PC is switched on.
- ✓ The required software and drivers are [installed \(p. 49\)](#) on the PC.
- ✓ You have read and understood the manual for the PC software. The links to the software manuals are in the A000T0081 file on the PI software data carrier.

#### Establishing Communication via USB

1. Start PIMikroMove.
  - The **Start up controller** window opens with the **Connect controller** step.
2. If the **Start up controller** window does not open automatically, select the **Connections > New...** menu item in the main window.
3. Select **C-884** in the controller selection field.
4. Select the **USB** tab on the right-hand side of the window.



5. Select the connected C-884.xDC in the **USB** tab.
6. Click **Connect** to establish communication.
  - If communication could not be established, look for a solution to the problem in the ["Troubleshooting \(p. 226\)"](#) chapter.
  - If communication was established successfully, PIMikroMove guides you through the configuration of the C-884.xDC for the connected positioner, refer to ["Starting Motion \(p. 62\)"](#).

### 7.2.3 Establishing Communication via TCP/IP

Before communication is established, it can be necessary to adapt the interface parameters once, depending on the type of networking:

- Network with DHCP server: No adjustment of the factory settings of the C-884.xDC interface parameters is required. You can [begin with setup of the communication \(p. 61\)](#).
- Network without DHCP server or direct connection of the C-884.xDC to the PC's Ethernet socket: it is necessary to [adapt the C-884.xDC's interface parameters \(p. 59\)](#). Make the necessary adaptations before establishing communication.

#### Requirements

- ✓ The C-884.xDC is [connected \(p. 56\)](#) to the network or directly to the PC via the RJ45 Ethernet socket.
- ✓ If several C-884.xDCs are connected to the same network via their TCP/IP interfaces: You have the serial number of the C-884.xDC ready with which the communication is to be established. The serial number can be found on the [type plate of the C-884.xDC \(p. 15\)](#).
- ✓ The C-884.xDC is [switched on \(p. 57\)](#).
- ✓ The PC is switched on.
- ✓ The required software and drivers are [installed \(p. 49\)](#) on the PC.
- ✓ You have read and understood the manual for the PC software. The links to the software manuals are in the A000T0081 file on the PI software data carrier.

#### Adapting the TCP/IP Interface Parameters of the C-884.xDC

If you need to adapt the interface parameters of the C-884.xDC to use the C-884.xDC in a network, proceed as follows.

1. Establish communication between the C-884.xDC and the PC via a different interface (e.g., [USB \(p. 58\)](#)).

2. Select the **C-884.xDC > Configure interface** menu item in the main window of PIMikroMove.  
→ *The **Configure Interface** window opens.*
3. Select the **TCP/IP** tab in the **Stored Settings** area in the **Configure Interface** window.

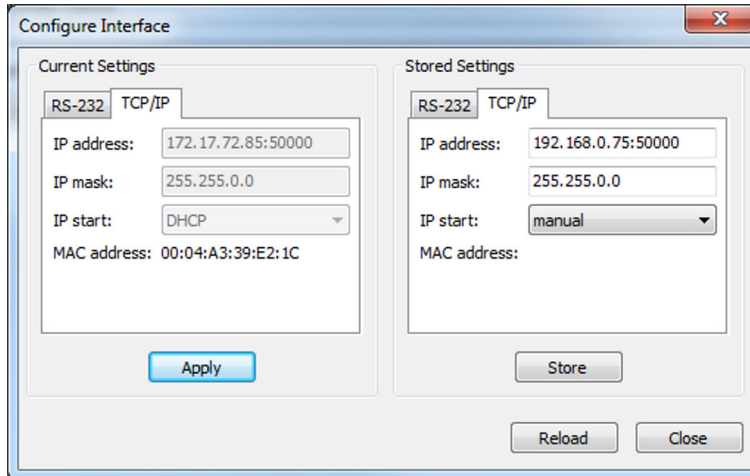


Figure 11: "Configure Interface" window with examples of settings

4. Make the necessary adaptations in the **TCP/IP** tab in the **Stored Settings** area:
  - a) **IP address** field: C-884.xDC's IP address in format xxx.xxx.xxx.xxx:50000
  - b) **IP mask** mask: Network's subnet mask
  - c) **IP start** field: C-884.xDC's startup behavior  
***manual***: Manually specified, static IP address is used  
***DHCP***: IP address is assigned automatically by a DHCP server.
5. Save the changed settings to the nonvolatile memory of the C-884.xDC by clicking **Store**.  
→ *The **Store interface settings** dialog opens.*
6. Click **Store settings** in the **Store interface settings** dialog.  
→ *The dialog closes. The settings were stored in the nonvolatile memory of the C-884.xDC .*
7. Close the **Configure Interface** window.
8. Close the connection with the C-884.xDC by selecting the **Connections > Close > C-884.xDC** menu item in the main window of PIMikroMove.
9. Switch the C-884.xDC of and on again by disconnecting it from and reconnecting to the power supply.



### CAUTION



#### Risk of crushing from unexpected motion!

When the communication between the C-884.xDC and the PC is established via TCP/IP, the PC software offers all electronics for selection that are available in the same network. After selecting a C-884.xDC for the connection, all commands are transmitted to this device. If the wrong device is selected, unexpected motion could be commanded and result in bruising injuries to the operating and maintenance staff of the positioner connected.

- If several C-884.xDC entries are displayed in the PC software, make sure that you select the right C-884.xDC.



### Information

Communication via TCP/IP can fail if the network cable was connected to the Ethernet socket of the C-884.xDC while the C-884.xDC was switched on.

- If communication cannot be established, switch the C-884.xDC off, connect the network cable, and switch the C-884.xDC on again.

### Information

For communication via TCP/IP, the C-884.xDC only has one unchangeable port (50000) available that cannot be used for more than one connection at a time.

### Establishing Communication via TCP/IP

1. Start PIMikroMove.

→ The **Start up controller** window opens with the **Connect controller** step.

2. If the **Start up controller** window does not open automatically, select the **Connections > New...** menu item in the main window.

3. Select **C-884** in the controller selection field.

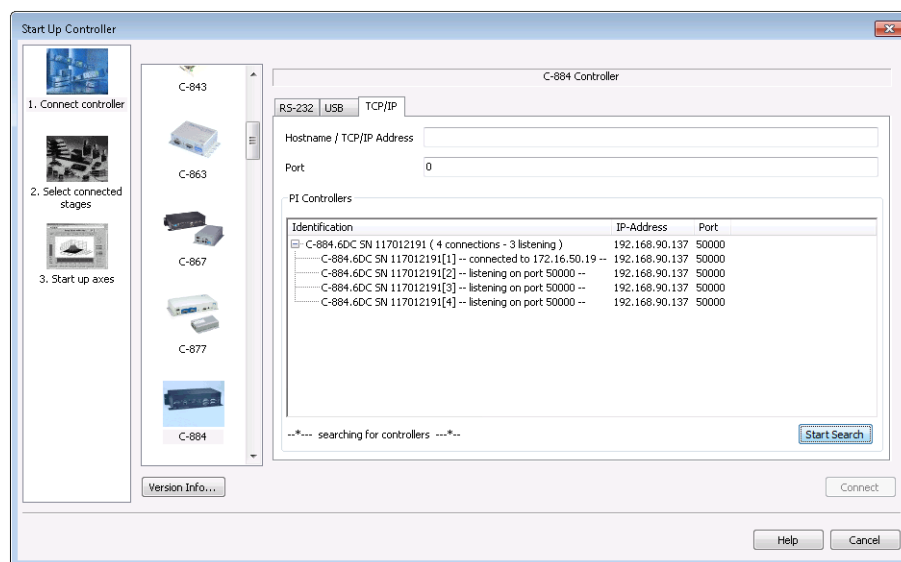
4. Select the **TCP/IP** tab on the right-hand side of the window.

→ The software now searches the network for all controllers of the C-884 type.

Click **Start Search** if the search for C-884 type controllers does not start automatically.

→ Searching the network for the C-884 type of controller has started.

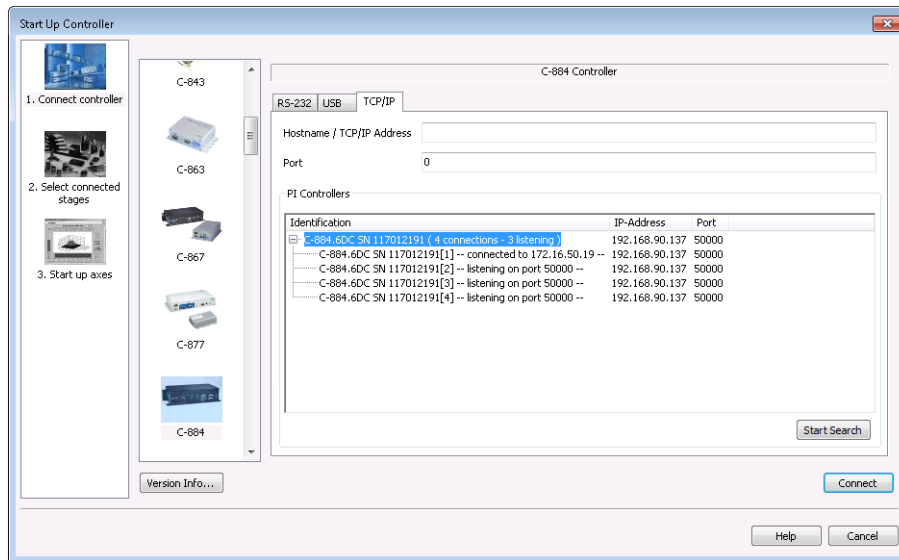
As long as the search is running, the **Connect** button is deactivated. If the search was successful, all C-884 controllers in the same network are displayed in the **PI Controllers** field.



5. Click the entry for your C-884.xDC found in the list of controllers. This must show the status "listening on port 50000".

- Do not select a controller that is already connected via TCP/IP (status "connected to ..."). Otherwise, an error message will be displayed as soon as you try to establish communication with this controller.
- If several entries with the same name are shown, identify your C-884.xDC on the basis of its nine-digit serial number.
- If the C-884.xDC is not displayed in the list of the controllers found, check the network settings. Consult your network administrator if necessary.

→ After a controller is selected in the list, its data is shown in the **Hostname / TCP/IP Address** and **Port** fields.



6. Click **Connect** to establish communication.

→ If communication could not be established, look for a solution to the problem in the ["Troubleshooting \(p. 226\)"](#) chapter.

→ If communication was established successfully, PIMikroMove guides you through the configuration of the C-884.xDC for the connected positioner, refer to ["Starting Motion \(p. 62\)"](#).

## 7.3 Starting Motion

The procedure for PIMikroMove is described in the following.

After communication has been established between the C-884.xDC and the PC, PIMikroMove guides you through the configuration of the C-884.xDC for the positioner. It is then possible to run the first motion tests of the positioner.

Selection of the configuration steps offered by PIMikroMove is based on evaluation of the following parameter values in the volatile memory of the C-884.xDC:

- **Stage Name** (0x3C): The value is used by PIMikroMove as criterion for finding a suitable parameter set in the positioner databases.
- **Stage Type** (0x0F000100): The value was loaded from the ID chip of the connected positioner when the C-884.xDC was switched on.

Possible configuration steps:

- If the values of the parameters 0x3C and 0x0F000100 are not identical, the **Stage Type Configuration** window opens. A corresponding message is displayed when a suitable parameter set is not in the positioner database.
- If the value of parameter 0x0F000100 is empty, e.g., because the positioner does not have an ID chip, the **Start up controller** window switches to the **Select connected stages** step.
- When the values of the parameters 0x3C and 0x0F000100 are identical, PIMikroMove assumes that all parameters of the C-884.xDC have already been adapted to the connected positioner. The **Start up controller** window goes directly to the **Start up axes** step.

The figures show the procedure for any electronics; the procedure for the C-884.xDC corresponds.

### Requirements

- ✓ PIMikroMove is [installed on the PC \(p. 49\)](#).
- ✓ You have [installed the latest version of the PISTAGES3.DB database onto your PC \(p. 49\)](#).
- ✓ If PI provided a custom positioner database for your positioner, the [dataset was imported into PIStages3 \(p. 50\)](#).
- ✓ You have installed the positioner in the same way as it will be used in your application (corresponding load, orientation, and mounting).
- ✓ You have connected the [positioner to the C-884.xDC \(p. 48\)](#).
- ✓ You have [established communication between the C-884.xDC and the PC with PIMikroMove \(p. 57\)](#).

### NOTICE



#### Selection of an incorrect positioner type!

Selection of an incorrect positioner type in the PC software can cause damage to the positioner.

- ▶ Make sure that the positioner type selected in the PC software matches the positioner connected.

### NOTICE



#### Damage due to disabled limit switch evaluation!

The collision of a moving part at the end of the travel range, or with an obstacle, as well as high accelerations, can cause damage to or considerable wear on the stage.

- ▶ Avoid motion in open-loop operation.
- ▶ If motion is necessary in open-loop operation: Stop the positioner in time. For this purpose, use the #24, STP or HLT command, or set the control value to zero with the SMO command.
- ▶ Do **not** choose parameter settings that would deactivate limit switch evaluation by the electronics.
- ▶ Check the function of the limit switches at about 10 % to 20 % of the maximum velocity.
- ▶ In the event of a malfunction of the limit switches, stop the motion immediately.

### NOTICE



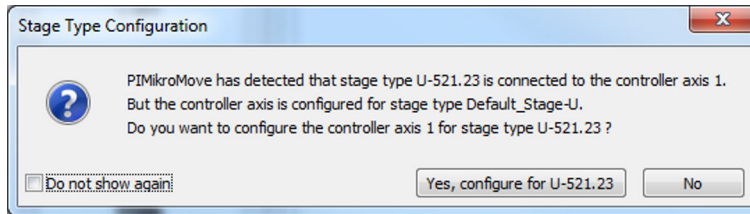
#### Oscillation!

Unsuitable settings of the notch filter and the C-884.xDC's servo control parameters can cause the positioner to oscillate. Oscillation can damage the positioner and/or the load fixed to it.

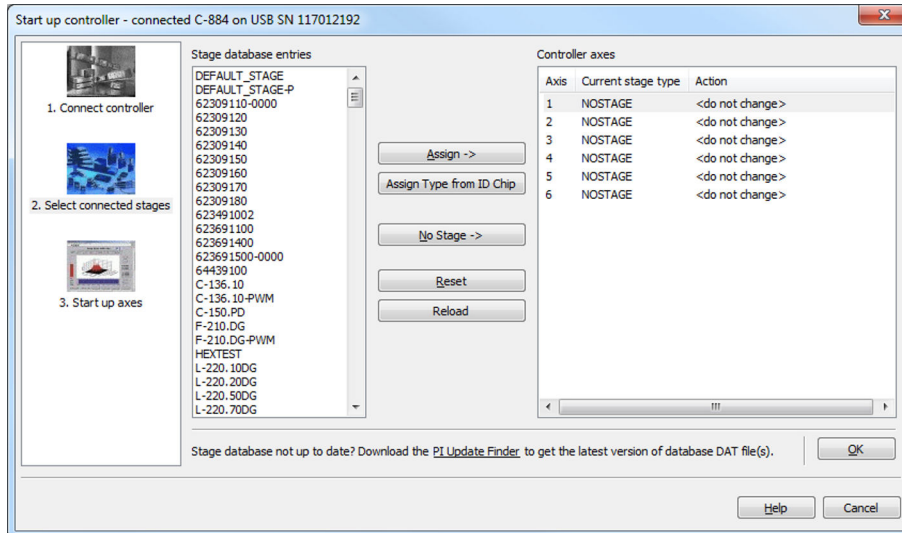
- ▶ Secure the positioner and all loads adequately.
- ▶ If the stage is oscillating (unusual operating noise), switch off servo mode or the C-884.xDC immediately.
- ▶ Switch servo mode back on only after you have changed the C-884.xDC's notch filter settings and servo control parameters; see "[Adjusting the Notch Filter \(p. 66\)](#)" and "[Optimizing the Servo Control Parameters \(p. 70\)](#)".
- ▶ If a very high load causes oscillation during the reference move, follow the instructions on the reference move under "[Troubleshooting](#)" (p. 226).

### Starting motion with PIMikroMove

1. Load the parameter set from the positioner database into the C-884.xDC:
  - When the **Stage Type Configuration** dialog has opened: Click the **Yes, configure for ...** button to load the matching parameter set from a positioner database into the C-884.xDC.

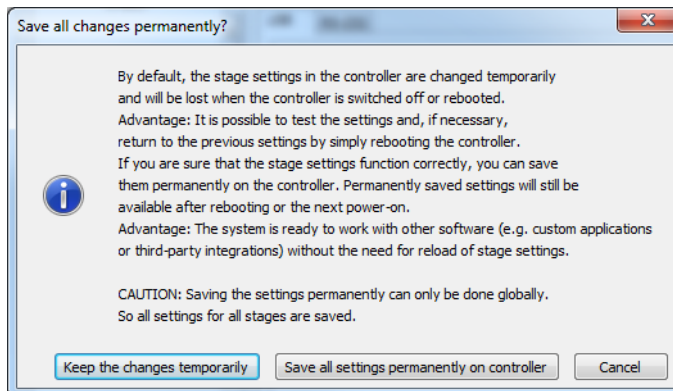


- If the **Select connected stages** step is displayed in the **Start up controller** window:



- Select the appropriate positioner type: Click either **Assign Type from ID Chip** or mark the matching positioner type in the **Stage database entries** list and click **Assign**.
- Confirm selection with **OK** to load the parameter settings for the selected positioner type from the positioner database into the C-884.xDC.

→ The **Save all changes permanently?** dialog is opened.



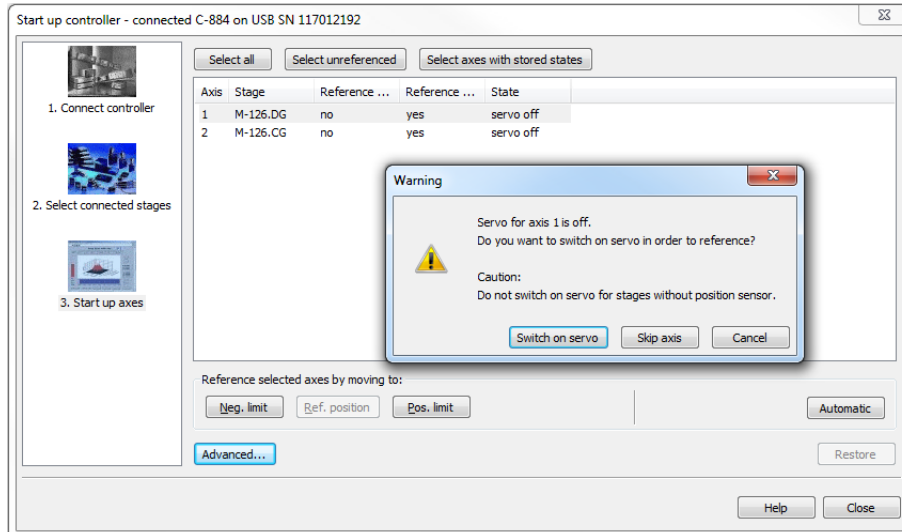
- Specify how you want to load the parameter settings into the C-884.xDC in the **Save all changes permanently?** dialog box:
  - Temporary load: Click **Keep the changes temporarily** to load the parameter settings into the volatile memory of the C-884.xDC. The settings are lost when the C-884.xDC is switched off or rebooted.
  - Load as default values: Click **Save all settings permanently on controller** to load the parameter settings into the nonvolatile memory of the C-884.xDC. The settings are available immediately after switching on or rebooting the C-884.xDC and do not need to be reloaded.

→ The **Start up controller** window with the **Start up axes** step is displayed.

- For axes connected with incremental sensor: During the **Start up axes** step, do a reference move for the axis so that the controller knows the absolute axis position: You

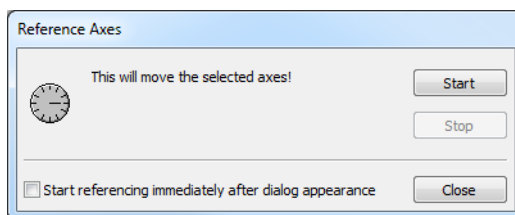
have the following options (options not supported by the positioner/controller either do not exist or cannot be activated):

- Start the reference move to the reference switch: Click **Ref. position**.
  - Start the reference move to the negative limit switch: Click **Neg. limit**.
  - Start the reference move to the positive limit switch: Click **Pos. limit**.
- a) If a warning message appears indicating that servo mode is switched off:



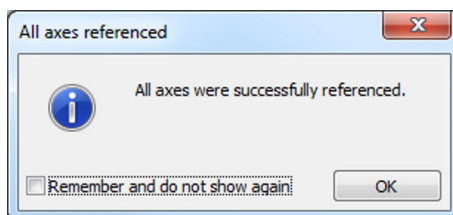
Switch on the servo mode by clicking on the **Switch on servo** button in the warning message.

→ The **Reference Axes** dialog appears after switching servo mode on:



b) Click the **Start** button in the **Reference Axes** dialog.

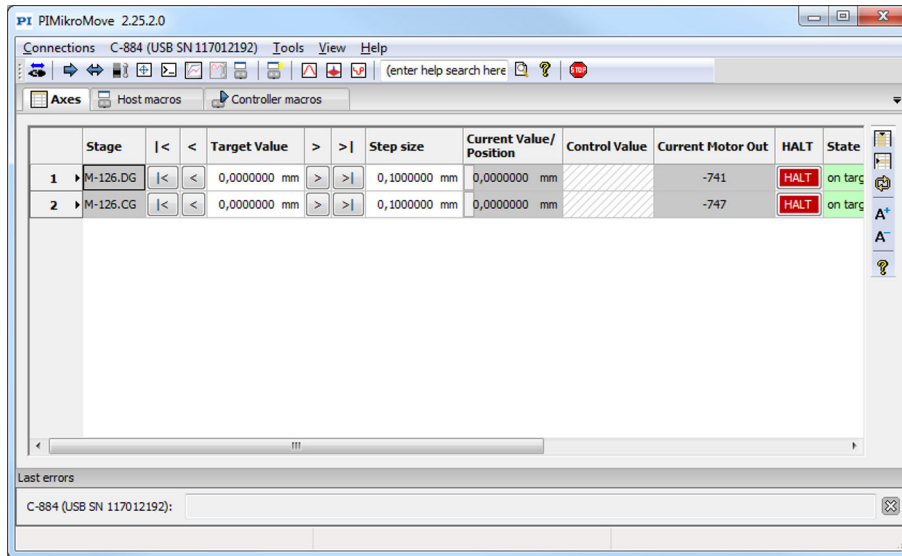
→ The axis performs the reference move. The **All axes referenced** message is displayed after a successful reference move:



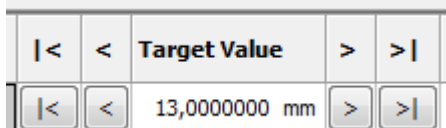
c) Close the **All axes referenced** message with **OK**.

4. Click **Close** to close the **Start up controller** window.

→ The main window of PIMikroMove opens.



5. Start testing axis motion: You can start motion by clicking the arrow keys for the corresponding axis in the PIMikroMove main window:



- |< or >|: Motion to the negative or positive limit of the travel range
- < or >: Motion along a certain path (specification in the **Step size** column)

## 7.4 Adapting the Dynamic Characteristics of the System

### 7.4.1 Adjusting the Notch Filter

The [notch filter \(p. 34\)](#) corrects the control value for the drive of the positioner connected to the C-884.xDC. The frequency component in the control value that would cause natural oscillation of the mechanics is reduced by the notch filter. Adjusting the notch filter frequency can be useful, particularly in the case of high loads.

To set the notch filter, a step response is recorded in open-loop operation. Adapting the notch filter is done via the following parameters:

- **Notch Filter Frequency 1 (0x94)**: Frequency component to be filtered out
- **Notch Filter Edge 1 (0x95)**: Bandwidth of the notch filter

The procedure for PIMikroMove is described in the following.

#### Requirements

- ✓ You have installed the positioner in the same way as it will be used in your application (corresponding load, orientation, and fastening).
- ✓ Initial motion [was started \(p. 62\)](#) in PIMikroMove.
- ✓ All devices are still ready for operation.

#### Information

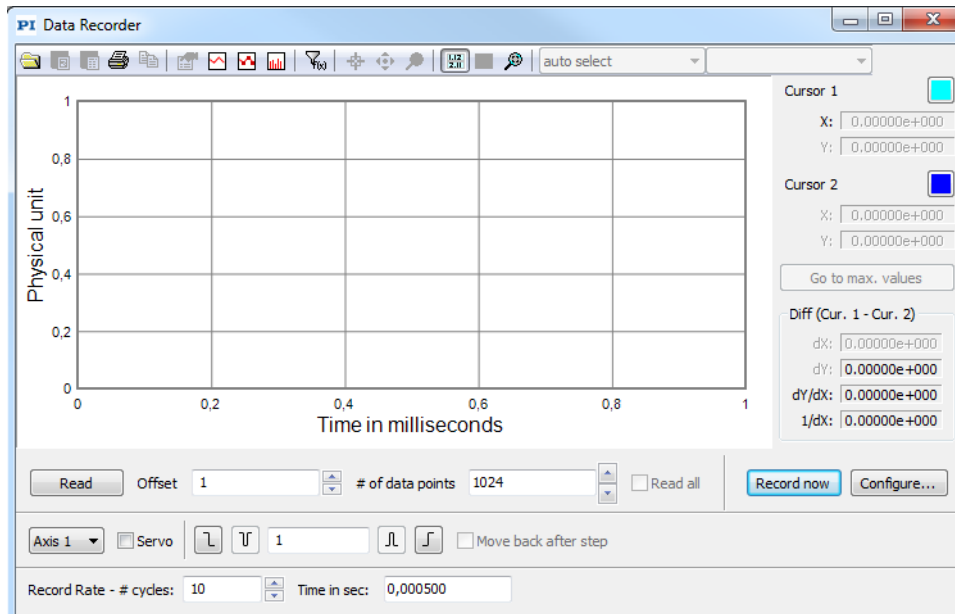
The settling behavior of the axis in closed-loop operation is influenced by the notch filter settings.

- Set the notch filter before you optimize the servo-control parameters.

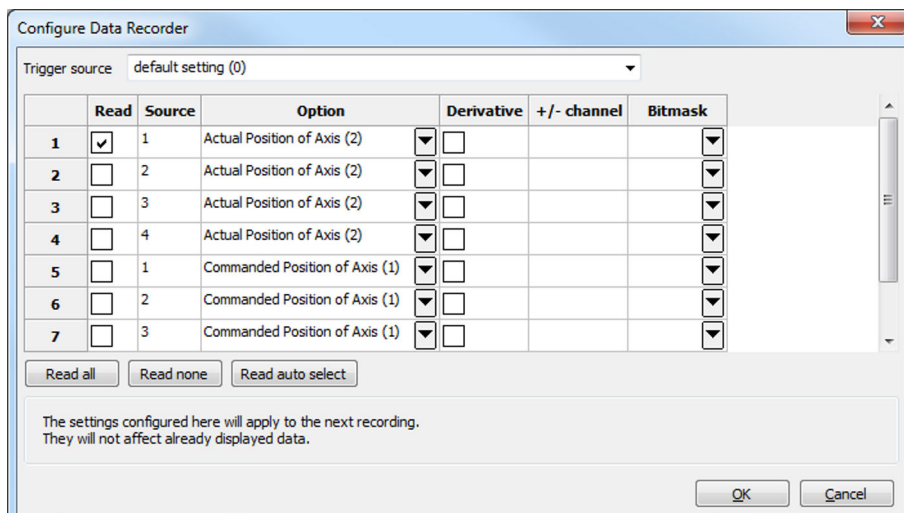
### Recording the step response

1. Open the **Data Recorder** window in PIMikroMove via the **C-884.xDC > Show data recorder** menu item.

→ *The Data Recorder window is opened.*



2. Make the following settings in the **Data Recorder** window:
  - a) Select the axis to be checked by clicking the **Axis #** button and use the **Servo** checkbox to make sure that servo mode is switched off for the axis: If the **Servo** checkbox is checked, uncheck to switch servo mode off.
  - b) Set the number of data points to be read for the graphical display (max. 8192) in the **# of data points** field.
  - c) Set the amplitude of the step to be performed to a value that is typical for your application (specified as physical units).
  - d) Set the number of servo cycles to be recorded in the **Record Rate - # cycles** field.
3. Click the **Configure...** button to open the window for configuring the data recorder.  
→ *The Configure Data Recorder window is opened.*
4. In the **Configure Data Recorder** window, make sure that the "Actual Position of Axis" configured for the axis is selected as the variable to be recorded, and close the window with **OK**.




	Read	Source	Option	Derivative	+/- channel	Bitmask
1	<input checked="" type="checkbox"/>	1	Actual Position of Axis (2)	▼	<input type="checkbox"/>	▼
2	<input type="checkbox"/>	2	Actual Position of Axis (2)	▼	<input type="checkbox"/>	▼
3	<input type="checkbox"/>	3	Actual Position of Axis (2)	▼	<input type="checkbox"/>	▼
4	<input type="checkbox"/>	4	Actual Position of Axis (2)	▼	<input type="checkbox"/>	▼
5	<input type="checkbox"/>	1	Commanded Position of Axis (1)	▼	<input type="checkbox"/>	▼
6	<input type="checkbox"/>	2	Commanded Position of Axis (1)	▼	<input type="checkbox"/>	▼
7	<input type="checkbox"/>	3	Commanded Position of Axis (1)	▼	<input type="checkbox"/>	▼

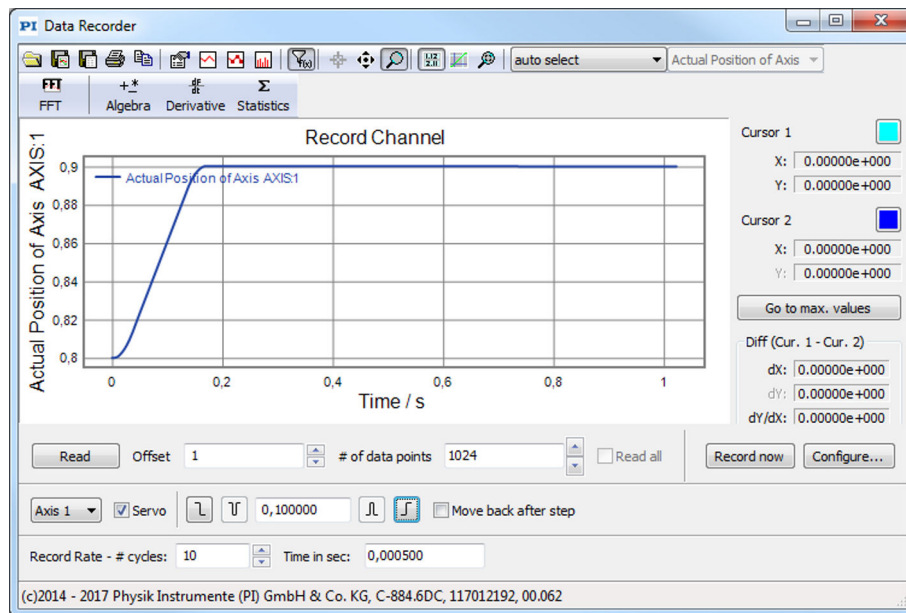
Read all Read none Read auto select



The settings configured here will apply to the next recording.  
They will not affect already displayed data.

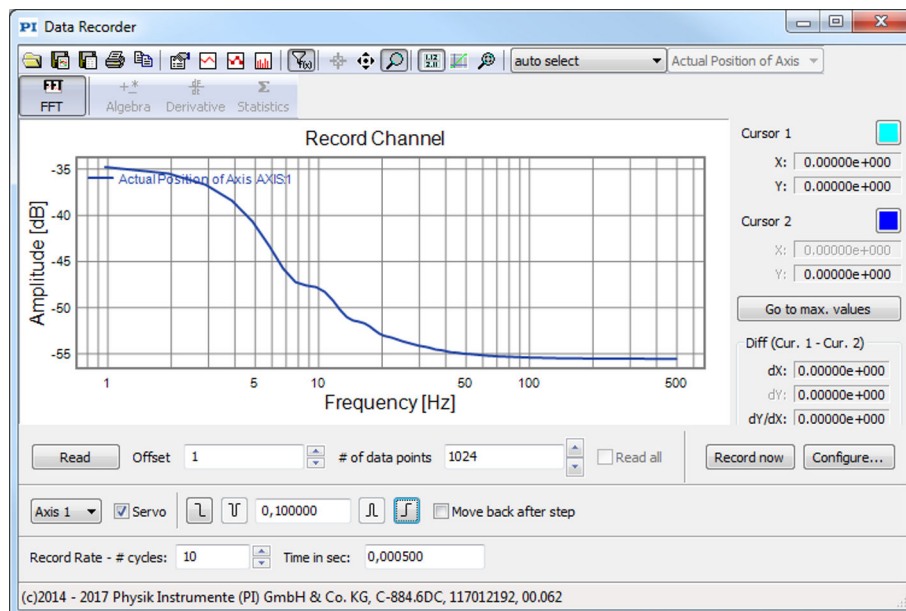
OK Cancel






5. Start the jump in the positive direction as well as the recording by clicking the  button in the **Data Recorder** window.
- The axis performs the step and the step response is recorded and displayed graphically.



6. Calculate the FFT (Fast Fourier Transformation) of the step response:
- Display the Data Toolbar via the  button.
  - Click the  button to calculate the FFT.
- The FFT of the step response is displayed graphically.



7. Determine the resonant frequency of the axis from the step response:
- If necessary, enlarge the view: Click the  button, press and hold down the left mouse button, and drag the magnifying glass symbol over an area in the graphical display (clicking with the right mouse button in the graphic field returns the view to its original size).
- Display the cursors in the graphical display by clicking the  button.
  - Activate the cursor movement with the mouse by clicking the  button.



- c) Click the **Go to max. values** button to position cursors 1 and 2 over the resonant frequencies.

→ The resonant frequencies are recognizable at the distinct maximum in the FFT diagram and are displayed in the respective **X:** field of the **Cursor 1** and **Cursor 2** areas on the right next to the graphical display.

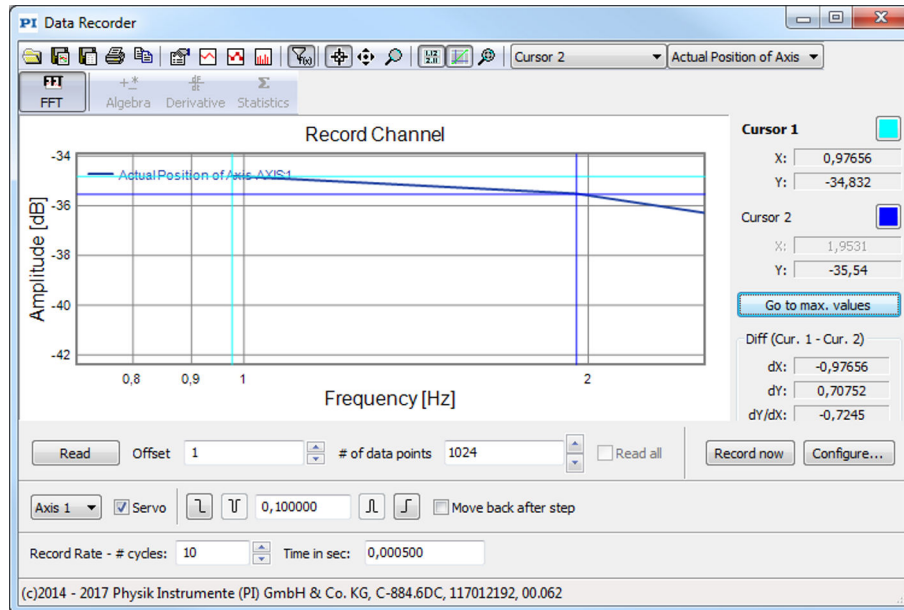
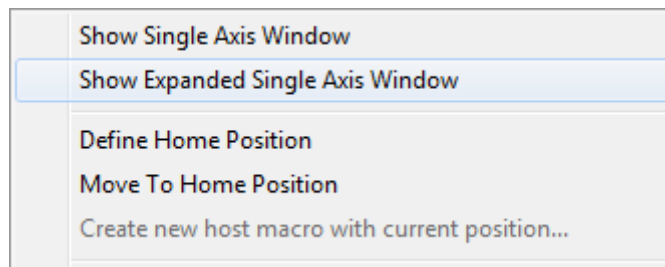


Figure 12: The example shows only one resonance frequency; it is about 1 Hz.

### Adjusting the Notch Filter

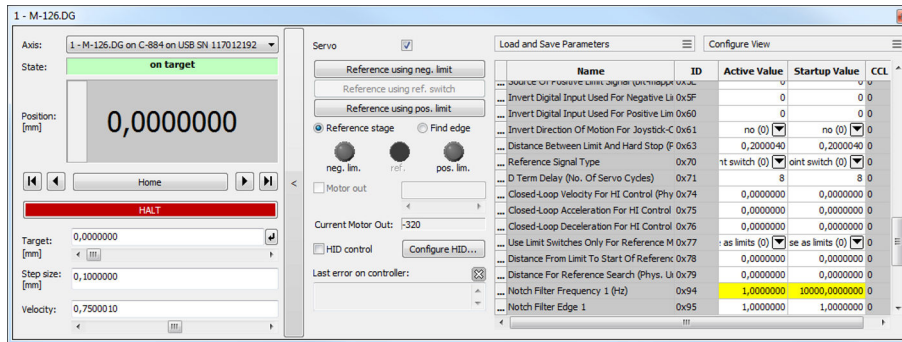
- Open the expanded single axis window of the connected positioner in the main window of PIMikroMove: Click the corresponding line of the **Axes** tab with the right mouse button and select the **Show Expanded Single Axis Window** in the context menu.



- Adapt the parameter values for the notch filter in the expanded single axis window:
  - If the **Notch Filter Frequency 1** and **Notch Filter Edge 1** parameters are not in the list on the right-hand side of the window, click **Configure View -> Select parameters...** and add them to the list.
  - Type the new parameter value into the corresponding input field in the **Active Value** column of the list.

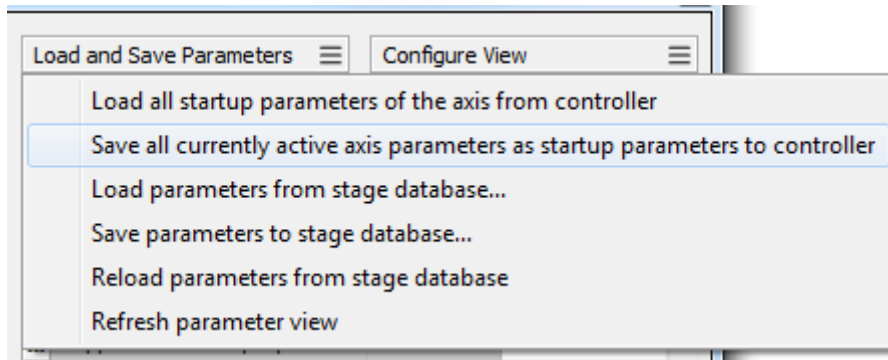
The value of the **Notch Filter Frequency 1** parameter must be set to the previously determined resonant frequency. If only one resonant frequency is filtered out, it is not necessary to adapt the parameter value for **Notch Filter Edge 1**.

- Press the **Enter** key on the PC keyboard or click outside the input field with the mouse to transfer the parameter value to the volatile memory of the controller. Note: If a parameter value in the volatile memory (**Active Value** column) is different to the parameter value in the nonvolatile memory (**Startup Value** column), the line in the list is highlighted in color.



3. Save the new settings. You have the following options:

- Save a parameter set to the positioner database on the PC, see "[Saving the Parameter Set to the Positioner Database \(p. 73\)](#)".
- Transfer the current values of the listed parameters from the volatile memory to the nonvolatile memory of the C-884.xDC by clicking **Load and Save Parameters -> Save all currently active axis parameters as startup parameters to controller**.



## 7.4.2 Optimizing the Servo Control Parameters

Adjusting the [servo controller \(p. 34\)](#) optimizes the system's dynamic properties (overshoot and settling time). The optimal servo controller settings depends on your application and your requirements.

Typically, optimization is determined empirically, i.e., the behavior of the positioner is monitored with different values in closed-loop operation.

The following describes the procedure for optimizing the servo control parameters in PIMikroMove.

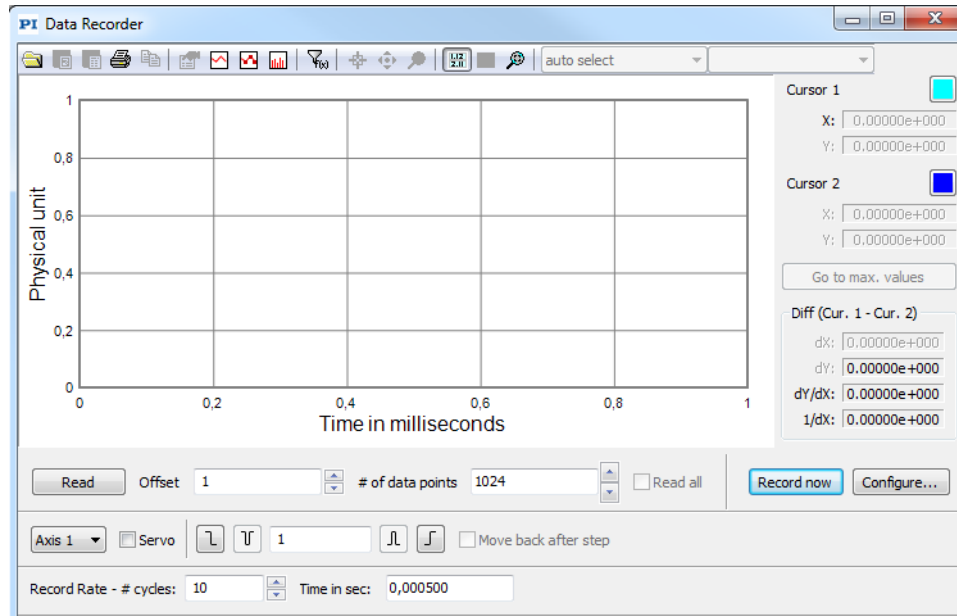
### Requirements

- ✓ You have installed the positioner in the same way as it will be used in your application (corresponding load, orientation, and mounting).
- ✓ You have started initial [motion \(p. 62\)](#) with PIMikroMove.
- ✓ If necessary, you have set the [notch filter \(p. 66\)](#).
- ✓ All devices are still ready for operation.

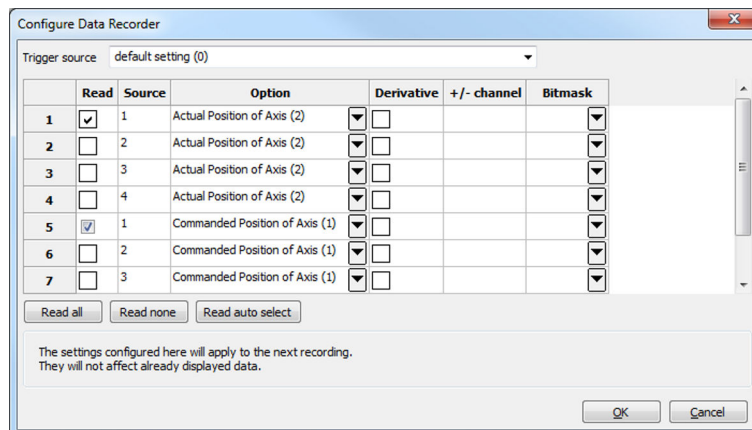
### Checking the Servo Control Parameters: Recording the Step Response


1. Open the **Data Recorder** window in PIMikroMove via the **C-884.xDC > Show data recorder** menu item.

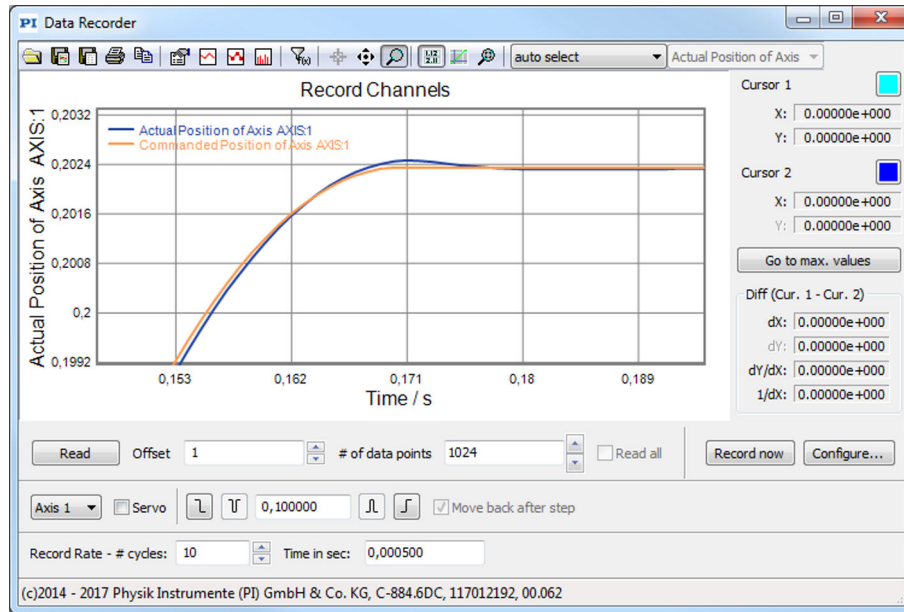
→ The **Data Recorder** window is opened.




2. Make the following settings in the **Data Recorder** window:
  - a) If necessary, select the axis to be checked by clicking the **Axis #** button and switch servo mode on by clicking the **Servo** checkbox (tick in the checkbox).
  - b) Set the number of data points to be read for the graphical display (max. 8192) in the **# of data points** field.
  - c) Set the amplitude of the step to be performed to a value that is typical for your application (specified as physical units).
  - d) Set the number of servo cycles to be recorded in the **Record Rate - # cycles** field.
3. Click the **Configure...** button to open the window for configuring the data recorder.  
 → *The **Configure Data Recorder** window is opened.*
4. Make sure that "Commanded Position of Axis" and "Actual Position of Axis" configured for the axis are selected as variables to be recorded in the **Configure Data Recorder** window and click **OK** to close the window.



5. Start the jump in the positive direction as well as the recording by clicking the  button in the **Data Recorder** window.  
 → *The axis performs the step and the step response is recorded and displayed graphically.*



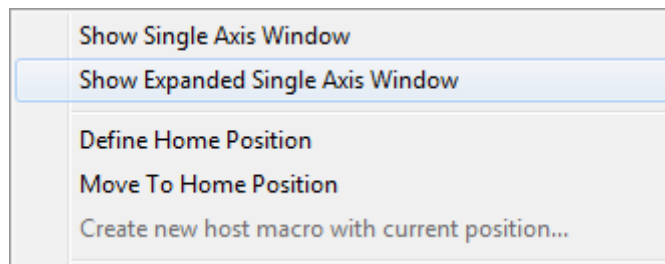
## 6. Check the displayed step response.

If necessary, enlarge the view: Click the  button, press and hold down the left mouse button, and drag the magnifying glass symbol over an area in the graphical display (clicking with the right mouse button in the graphic field returns the view to its original size).

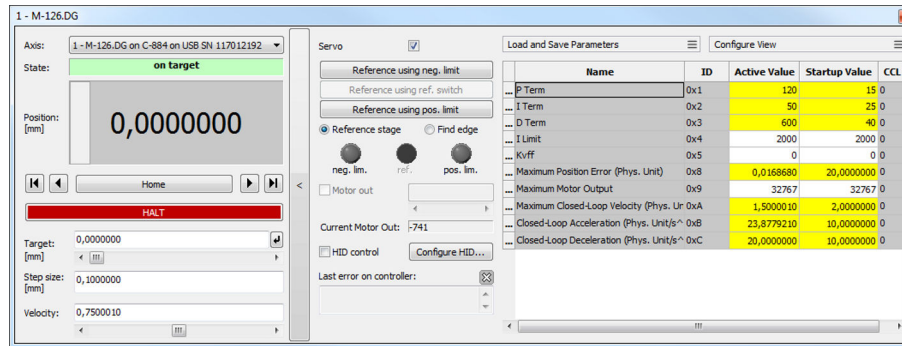
- If the result is satisfactory (i.e., minimum overshoot, settling time not too long):  
You already have optimum parameter settings and do not have to do anything further.
- If the result is not satisfactory:  
Optimize the servo control parameters, see below.

## Setting the Servo Control Parameters

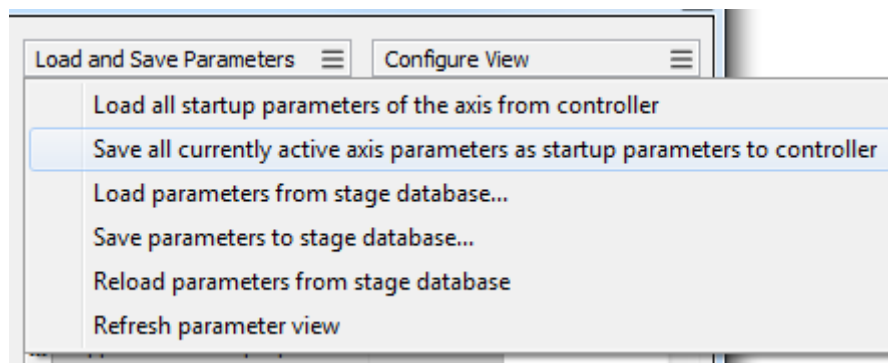
1. Open the expanded single axis window of the connected positioner in the main window of PIMikroMove: Click on the corresponding line of the **Axes** tab with the right mouse button and select the **Show Expanded Single Axis Window** in the context menu.



2. Enter new values for the parameters to be adapted into the expanded single axis window:
  - a) If the parameters to be changed are not included in the list on the right side of the window, click on **Configure View -> Select parameters...** and add them to the list.
  - b) Type the new parameter value into the corresponding input field in the **Active Value** column of the list.
  - c) Press the **Enter** key on the PC keyboard or click outside the input field with the mouse to transfer the parameter value to the volatile memory of the controller. Note: If a parameter value in the volatile memory (**Active Value** column) is different to the parameter value in the nonvolatile memory (**Startup Value** column), the line in the list is highlighted in color.



3. In the **Data Recorder** window, record the step response of the positioner again.  
If the result is not satisfactory, enter different values for the servo control parameters and record the step response again.
4. If you are satisfied with the result and want to keep the changed parameter settings, save the new settings. You have the following options:
  - Save a parameter set to the positioner database on the PC, see "[Saving the Parameter Set to the Positioner Database \(p. 73\)](#)".
  - Transfer the current values of the listed parameters from the volatile memory to the nonvolatile memory of the C-884.xDC by clicking **Load and Save Parameters -> Save all currently active axis parameters as startup parameters to controller**.



## 7.5 Making Data Backups

### Saving the Parameter Set to the Positioner Database

1. Click **Load and Save Parameters -> Save parameters to stage database...** in the expanded single axis window in PIMikroMove.  
→ *The **Save Parameters as User Stage Type** dialog opens.*
2. Save the changed parameter values as new positioner type in the **Save Parameters as User Stage Type** dialog:
  - a) Leave the entry in the **Parameters of axis** field unchanged.
  - b) Enter the name for the new positioner type into the **Save as** field.
  - c) Click **OK**.
 → *The new positioner type was saved to the PISTAGES3.DB database. The display of the connected positioner type was updated in the single axis window and in the main window of PIMikroMove.*

### 7.5.1 Saving Parameter Values

The C-884.xDC is configured via parameters, e.g., for adapting to the connected positioner. The parameter values can be saved to a text file so that they can be restored at a later time.

### Information

Changing parameter values can cause undesirable results.

- ▶ Create a backup copy on the PC before changing the parameter settings of the C-884.xDC. You can then restore the original settings at any time.
- ▶ Create an additional backup copy with a new file name each time after optimizing the parameter values or adapting the C-884.xDC to a particular positioner.

Parameter values saved in a text file on the PC can be loaded back to the C-884.xDC in PIMikroMove or PITerminal. The **Send file...** button is available for this purpose in the send command window. Before loading into the C-884.xDC, the individual lines of the text files must be converted into command lines that contain the corresponding SPA or SEP commands.

### Requirements

- ✓ You have established [communication between the C-884.xDC and the PC with PIMikroMove \(p. 57\)](#) or PITerminal.

### Saving Parameter Values in a Text File



1. If you are using PIMikroMove, open the window for transmitting commands: Select **Tools > Command entry** in the main window or press **F4** on the keyboard.  
→ *After communication has been established, the main window is opened in PITerminal automatically and commands can then be sent.*
2. Query the parameter values that you want backup.
  - If you want to save the parameter values from the volatile memory of the C-884.xDC: Send the **SPA?** command.
  - If you want to save the parameter values from the nonvolatile memory of the C-884.xDC: Send the **SEP?** command.
3. Click the **Save...** button.  
→ *The **Save content of terminal as textfile** window opens.*
4. Save the queried parameter values to a text file on your PC in the **Save content of terminal as textfile** window.

## 7.5.2 Saving Controller Macros

For example, making backups of controller macros on the PC can be useful before updating the firmware.

The procedure for PIMikroMove is described in the following.

### Saving Controller Macros on the PC

1. Select the **Controller macros** tab in PIMikroMove's main window.
2. Select the macros in the **Macros on controller** list that you want to back up to the PC:
  - Click the desired entry in the list to select a macro.
  - To select several macros, hold down the Shift button and click the desired entries in the list.
  - To deselect, click an open area in the list.
 → *By selecting one or more macros, the  (**Save selected macros to PC**) button becomes active.*
3. Save the selected macros on the PC:
  - a) Click the  button to open a directory selection window.
  - b) Select the directory on the PC where you want to save the macros.


c) Click **Save**.

→ *The macros are saved as text files (<macro name>.txt) in the directory selected of the PC.*

#### **Loading Controller Macros from the PC to the C-884.xDC**

1. Select the **Controller macros** tab in PIMikroMove's main window.

2. Load macros from the PC to the C-884.xDC:

a) Click the  button to open a file selection window.

b) Select the text files (<macro name>.txt) in the file selection window whose contents you want to load as a macro from the PC to the C-884.xDC.

c) Click **Open**.

→ *For each selected text file (<macro name>.txt), the content is loaded as a macro <macro name> into the C-884.xDC.*



## 8 C-884.xDC Functions

### 8.1 Protective Functions of the C-884.xDC

The C-884.xDC has functions that are intended to protect it against damage.

#### 8.1.1 Behavior with Motion Errors

Motion errors can be caused for example, by malfunctions of the drive or the position sensor of the positioner.

There is a motion error when the position error (i.e., the difference between the current and commanded position) exceeds the specified maximum value in closed-loop operation. The maximum value for the deviation is specified by the **Maximum Position Error (Phys. Unit)** parameter (0x8).

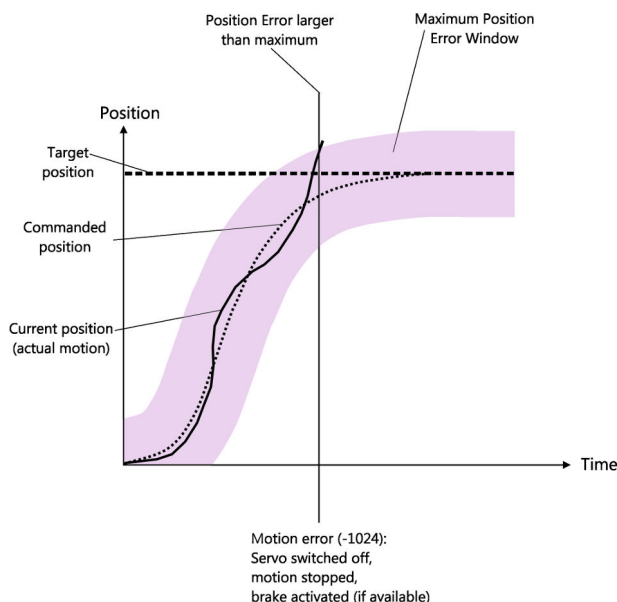


Figure 13: Determining motion errors

Motion errors can have the following causes, for example:

- Malfunction of the drive
- Malfunction of the position sensor
- Fault in the mechanics

If motion error occurs, the C-884.xDC reacts as follows to protect the system against damage:

- Servo mode is switched off for the axis.
- If applicable, the brake is activated for the axis in question.
- All motion is stopped.
- Error code -1024 is output.

Then restore the [operational readiness \(p. 77\)](#) for the C-884.xDC.



### 8.1.2 Restoring Operational Readiness

#### NOTICE



#### Unintentional motion after brake deactivated!

If servo mode is switched off, e.g., after a motion error occurs, the brake of the positioner can be deactivated by command. Deactivating the brake can cause the positioner to move unintentionally.

- Secure the positioner against moving unintentionally before you deactivate the brake by command.

#### Restoring the C-884.xDC's Operational Readiness

1. Send the [ERR?](#) command to read out the error code.  
→ *ERR?* resets the error code to zero during the query.
2. Check your system and make sure that all axes can be moved safely.
3. Switch servo mode on for the axis with the [SV0](#) command.  
→ *When servo mode is switched on, the target position is set to the current axis position and the brake is deactivated if applicable. Now the axis can move again and you can command a new target position.*

### 8.1.3 Commands

E		Page
ERR?	Get Error Number	140
S		Page
SV0	Set Servo Mode	175

### 8.1.4 Parameters

0x8	Maximum Position Error (Phys. Unit)	Maximum position error. A motion error occurs when the difference between the current position and the commanded position exceeds this value and the movement of the axis is stopped.
-----	-------------------------------------	--

## 8.2 Trajectories for Motion Paths

In closed-loop operation, the C-884.xDC can process externally calculated one- or two-dimensional motion paths (e.g., circles, sine curves) as trajectories.

### 8.2.1 Operating Principle of the Trajectory Buffer

The individual target positions of the motion path must be loaded into the trajectory buffer of the C-884.xDC as trajectory points. The buffer outputs the points with a fixed chronological interval when the trajectory is being followed. The points are output in the order they were loaded to the trajectory buffer (FIFO principle: **First In First Out**).

The content of a trajectory buffer is only present in the volatile memory of the C-884.xDC and cannot be saved to the nonvolatile memory.

**NOTICE****Following trajectories!**

The C-884.xDC does not calculate a dynamics profile while a trajectory is being followed. After the last trajectory point has been reached, the motion of the axis is abruptly stopped. This applies both to completing trajectories properly as well as cancelling them (e.g., by a stop command or error). Acceleration / deceleration, velocity, and steadiness of the motion therefore depend on the following factors during trajectory following:

- Values of the trajectory points
- Timing for the trajectories
- Sufficiently fast refilling of the trajectory buffer

Following an unsuitable trajectory can cause the positioner to oscillate or stop motion abruptly. Oscillation or stopping abruptly can damage the positioner and/or the load fixed to it.

Therefore, pay attention to the following when working with trajectories:

- ▶ The path that is specified by the trajectory points must be continuously differentiable at least twice.
- ▶ The maximum permissible velocity and acceleration of the axis may not be exceeded when the trajectory is being followed.
- ▶ When following the trajectory, an abrupt stop may not damage the load on the positioner.
- ▶ To generate the trajectory points and continuously transfer them to the C-884.xDC during the trajectory following, it is recommended to use a suitable program.

**Information**

When working with trajectories, it is recommended to use the *Trajectory Assistant* in PIMikroMove (call via the menu of the C-884.xDC). This allows you to define and follow trajectories conveniently.

**8.2.2 Working with Trajectories**

The trajectories are permanently assigned to the C-884.xDC's axes: Trajectory 1 to axis 1, trajectory 2 to axis 2 etc.

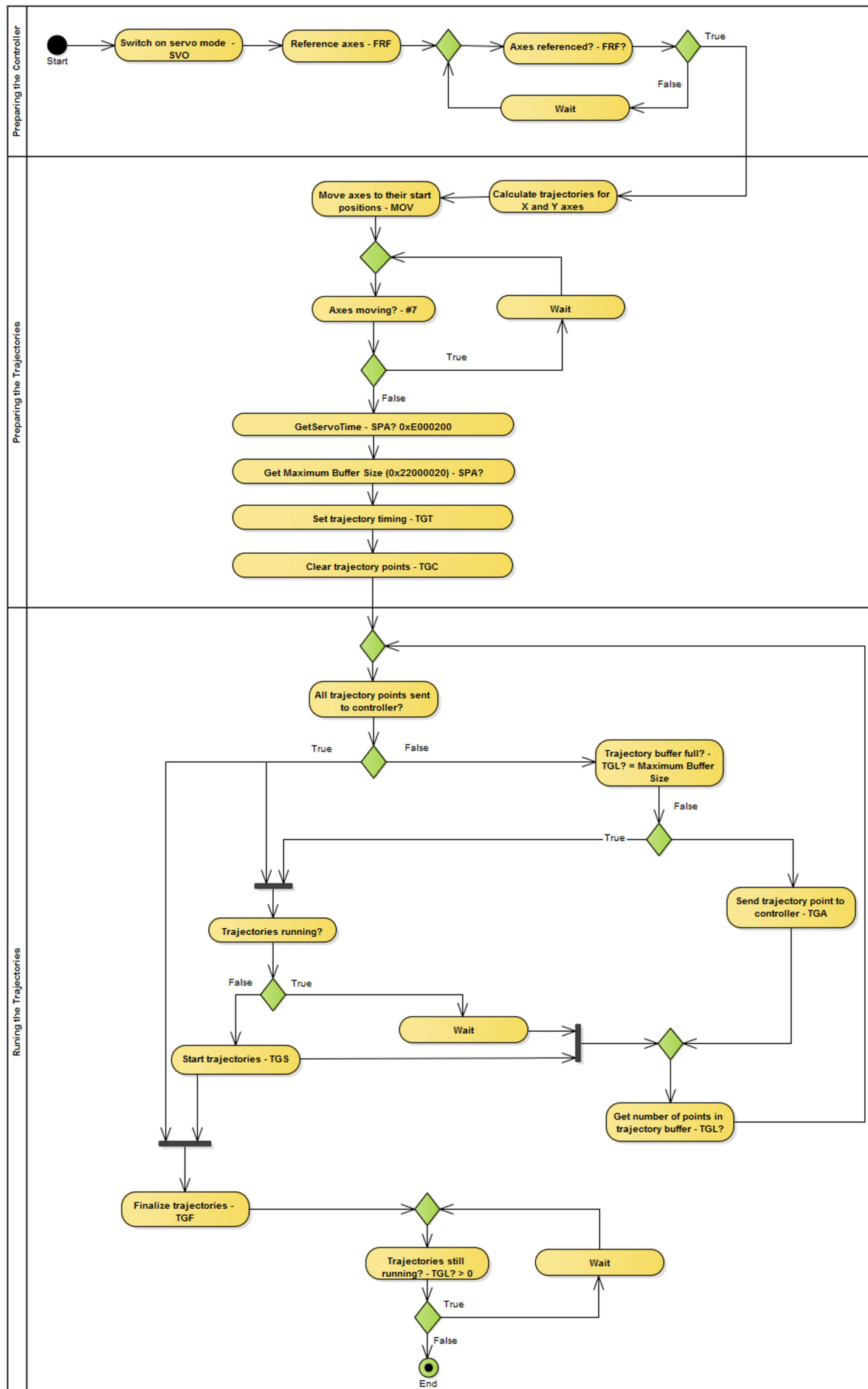
The timing for trajectories is set with the [TGT](#) command.

At least 4 points must be loaded to the trajectory buffer with the [TGA](#) command before a trajectory can be followed. The maximum number of points in the trajectory buffer is determined by the **Maximum Buffer Size** parameter (0x22000020).

The [TGS](#) command starts trajectory following. The buffer must be refilled fast enough when a trajectory is being followed. The [TGF](#) command completes trajectory following properly.

If trajectory following is cancelled after an error or stopped with [STP](#), [#24](#), or [HLT](#), the trajectory points that have not been processed by this time remain in the buffer. Therefore, before loading or executing a new trajectory, make sure that there are no invalid trajectory points in the buffer (query with [TGL?](#), deletion with [TGC](#)).

The following diagram shows an example of the trajectory following sequence. Corresponding example programs are found on the PC after the PC software for the C-884.xDC has been installed.



### 8.2.3 Commands

T		Page
TGA	Append Value To Trajectory	177
TGC	Clear All Values In Trajectory	177
TGF	Finalize Trajectory	178
TGL?	Get Number Of Values In Trajectory	178
TGS	Start Trajectory	178
TGT	Set Trajectory Timing	179

### 8.2.4 Parameters

0x22000020	Maximum Buffer Size	Maximum number of trajectory points in memory. Trajectory points must be reloaded for processing trajectories that have more than this number of points.
------------	---------------------	---

## 8.3 Data Recorder

The C-884.xDC contains a real-time data recorder. The data recorder can record different variables, e.g., the current position of an axis.

The recorded data is stored temporarily in data recorder tables. Each data recorder table contains the data of one data source.

The C-884.xDC has 8 data recorder tables with 8192 data points each.

You can configure the data recorder for example, by defining the data type to be recorded and the data sources, and by specifying how the recording is to be started.

### 8.3.1 Setting up the Data Recorder

#### Reading general information from the data recorder

1. Send the [HDR? command](#).

→ The options available for recording and triggering are displayed together with the information on additional parameters and commands for data recording.

#### Determining the data to be recorded

1. Configure the data recorder with the [DRC command](#).

→ This assigns the data sources and the recording options to the data recorder tables. The current configuration can be read out with the [DRC? command](#).

Data recorder tables with the record option 0 are deactivated, i.e., nothing is recorded. By default, the data recorder tables of the C-884.xDC record the following:

- Table 1: Current position of axis 1
- Table 2: Current position of axis 2
- Table 3: Current position of axis 3
- Table 4: Current position of axis 4
- Table 5: Commanded position of axis 1

- Table 6: Commanded position of axis 2
- Table 7: Commanded position of axis 3
- Table 8: Commanded position of axis 4

Setting the trigger for recording

1. Set the trigger option with the [DRT](#) command.  
The trigger specifies how the recording is to be triggered. The current trigger option can be queried with the [DRT?](#) command.  
→ The trigger option applies to all data recorder tables whose record option is not set to 0.

Setting the record table rate

1. Send the [RTR](#) command to set the recording rate of the data recorder.  
The parameter indicates the number of servo cycles required for recording each data point. The recording rate can be read out with the [RTR?](#) command.  
As the record table rate increases, the maximum duration of the data recording is increased.

Configuring data processing

1. Configure recorded data processing with the following parameters:
  - [Recorded Points Per Trigger](#) (0x16000001)
  - [Clearing Of RecTable On Trigger](#) (0x16000002)
  - [Data Recorder Buffer Mode](#) (0x16000003)
  - [Data Recorder Buffer Overflow](#) (0x16000004)

8.3.2 Starting Data Recording

Starting the Recording

1. Start the recording with the trigger option set with [DRT](#).  
Irrespective of the set trigger option, data recording is always triggered when step response measuring is started with [STE](#).  
The data recording always takes place for all data recorder tables whose record option is not set to 0. It ends when the data recorder tables are full.

8.3.3 Reading Out Recorded Data

Information

Reading the recorded data can take some time, depending on the number of data points. The data can also be read out while data is being recorded.

Reading data out

1. Read the last recorded data out with the [DRR?](#) command.  
→ The data is output in GCS array format.
2. Query the number of points contained in the last recording with the [DRL? command](#).

8.3.4 Commands

D		Page
DRC	Set Data Recorder Configuration	137

DRC?	Get Data Recorder Configuration	138
DRL?	Get Number Of Recorded Points	138
DRR?	Get Recorded Data Values	138
DRT	Set Data Recorder Trigger Source	139
DRT?	Get Data Recorder Trigger Source	140
<b>H</b>		<b>Page</b>
HDR?	Get All Data Recorder Options	143
<b>R</b>		<b>Page</b>
RTR	Set Record Table Rate	170
RTR?	Get Record Table Rate	170
<b>S</b>		<b>Page</b>
STE	Start Step And Response Measurement	175

### 8.3.5 Parameters

0x16000001	Recorded Points Per Trigger	Number of data points recorded per trigger impulse. 0 Unlimited (default) n>0 n data points are recorded Maximum value for n at C-884.xDC: 8192
0x16000002	Clearing Of RecTable On Trigger	Write mode during the recording. Determines how the data points are written to the data recorder tables when the recording is started by a trigger. 0 Recorded points are added to the contents of the data recorder table (default setting) 1 Trigger deletes the content of the data recorder tables, i.e., recording always starts at the first point
0x16000003	Data Recorder Buffer Mode	Behavior with full data recorder tables. 0 Recording ends (default setting) 1 Recording is continued with the first point in the data recorder table and overwrites existing content The value of parameter 0x16000004 is increased by 1 with each overflow.
0x16000004	Data Recorder Buffer Overflow	Buffer overflow counter of the data recorder. Counts how often recording starts with the first point in the data recorder table when parameter 0x16000003 has the value 1. Reading out with DRR? resets the parameter to zero. (write-protected)

## 8.4 Digital Inputs and Outputs

The C-884.xDC's digital inputs and outputs are available at the I/O socket. The number of available input and output lines to the C-884.xDC can be queried with the [TI0?](#) command.

### Overview



Connector of the C-884.xDC for digital inputs and outputs

### 8.4.1 Connecting the Digital Outputs

#### Tools and Accessories

- Suitable D-sub 26-pole (m) connector, with cable
- Device to be triggered with digital input for TTL signals

#### Information

Digital output signals are available on pins 8, 9, 18 and 26 of the *I/O* socket.

#### Connecting a Device to be Triggered

1. Connect an appropriate device to one of pins 8, 9, 18, and 26 of the *I/O* socket of the C-884.xDC.

### 8.4.2 Connecting the Digital Inputs

#### Tools and Accessories

- Suitable signal source: If the digital inputs are to be used as the source for the reference and limit switch signals of an axis, the signal level may only change once across the entire travel range.
- Suitable D-sub 26-pole (m) connector, with cable

#### Information

Digital TTL input signals can be fed to the C-884.xDC via pins 7, 16, 17, and 25 of the *I/O* socket.

The digital inputs can be used as follows:

- Use in macros
- Source for the reference switch and axis limit switch signals

#### Connecting a Digital Signal Source

1. Connect an appropriate signal source to one of pins 7, 16, 17 or 25 of the *I/O* socket of the C-884.xDC.

### 8.4.3 Digital Output Signals

The number of output lines available on the C-884.xDC can be queried with the [TI0?](#) command.

External devices can be triggered via the digital outputs of the C-884.xDC. Potential applications:

- Linking the trigger output to the motion of the axis ([CT0](#), [TR0](#); query with [CT0?](#), [TR0?](#))

- Direct switching of output lines ([DIO](#)), e.g., in macros

### Setting "Position Distance" Trigger Mode

The **Position Distance** trigger mode is suitable for scanning applications. A trigger pulse is output as soon as the axis has covered the distance set with CTO parameter ID = 1 (TriggerStep). The pulse width is one servo cycle. The distance unit (TriggerStep) depends on the settings of the Numerator Of The Counts-Per-Physical-Unit Factor parameter (0xE) and the Denominator Of The Counts-Per-Physical-Unit Factor parameter (0xF). Default is mm.

1. Configure the digital output line <TrigOutID> to be used as trigger output:
  - a) Send `CTO <TrigOutID> 2 A`, where A indicates the axis to be moved.
  - b) Send `CTO <TrigOutID> 3 0`, where 0 determines the Position Distance trigger mode.
  - c) Send `CTO <TrigOutID> 1 S`, where S indicates the distance.
2. If you want to enable the conditions for trigger output, send `TR0 <TrigOutID> 1`.

Example: A pulse on digital output line 1 is output each time axis 1 of the mechanics covers a distance of 0.1 µm.

Send:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
TR0 1 1
```

### "Position Distance" trigger mode with start and stop values for positive motion direction of the axis

Optionally, you can determine the start and stop values for triggering for limiting the range and specifying the motion direction of the axis (positive or negative).

### Information

If start and stop values have the same value, they are ignored.

If the direction of motion is reversed before the axis position has reached the stop value, trigger pulses continue to be output.

The following commands must be sent to configure the <TrigOutID> digital output line to be used as trigger output:

```
CTO <TrigOutID> 2 A, where A indicates the axis to be moved
CTO <TrigOutID> 3 0, where 0 determines the Position Distance trigger mode
CTO <TrigOutID> 1 S, where S indicates the distance
CTO <TrigOutID> 8 Start, where Start indicates the start value
CTO <TrigOutID> 9 Stop, where Stop indicates the stop value
```

Example for positive direction of axis motion: A pulse on digital output line 1 is output every time axis 1 of the mechanics covers a distance of 0.1 µm, as long as axis 1 is moving in a positive direction of motion within the range of 0.2 µm to 0.55 µm (start value < stop value).

Send:

```
CTO 1 2 1
CTO 1 3 0
CTO 1 1 0.0001
CTO 1 8 0.0002
CTO 1 9 0.00055
TR0 1 1
```



Example for negative direction of axis motion: A pulse on digital output line 1 is output every time axis 1 of the mechanics covers a distance of 0.1  $\mu\text{m}$ , as long as axis 1 is moving in a negative direction of motion within the range of 0.55  $\mu\text{m}$  to 0.2  $\mu\text{m}$  (stop value < start value).

Send:

```
CT0 1 2 1
CT0 1 3 0
CT0 1 1 0.0001
CT0 1 8 0.00055
CT0 1 9 0.0002
TR0 1 1
```

### Setting Up "On Target" Trigger Mode

In the **On Target** trigger mode the on-target state of the axis selected is output at the selected trigger output.

1. Configure the digital output line <TrigOutID> to be used as trigger output:
  - a) Send `CT0 <TrigOutID> 2 A` where A indicates the axis to be moved.
  - b) Send `CT0 <TrigOutID> 3 2`, where 2 specifies the On Target trigger mode.
2. If you want to enable the conditions for trigger output, send `TR0 <TrigOutID> 1`.

Example: The on-target state of axis 1 is to be output on digital output line 1.

Send:

```
CT0 1 2 1
CT0 1 3 2
TR0 1 1
```

### Setting Up "Motion Error" Trigger Mode

The **Motion Error** trigger mode is suitable for monitoring motion. The selected digital output line becomes active when a motion error occurs on the connected axis. The line remains active until the error code is reset to 0 (by a query with ERR?).

## Information

A motion error occurs when the difference between the current and commanded position exceeds the maximum specified value in closed-loop operation. The maximum value for the deviation is specified by the **Maximum Position Error (Phys. Unit)** parameter (0x8).

1. Configure the digital output line <TrigOutID> to be used as trigger output:
  - a) Send `CT0 <TrigOutID> 3 5`, where 5 specifies the Motion Error trigger mode.
2. If you want to enable the conditions for trigger output, send `TR0 <TrigOutID> 1`.

### Setting "In Motion" Trigger Mode

The motion state of the selected axis in the **In Motion** trigger mode is output at the selected trigger output. The line is active as long as the selected axis is in motion.

1. Configure the digital output line <TrigOutID> to be used as trigger output:
  - a) Send `CT0 <TrigOutID> 2 A`, where A indicates the axis to be moved.
  - b) Send `CT0 <TrigOutID> 3 6`, where 6 specifies the In Motion trigger mode.
2. If you want to enable the conditions for trigger output, send `TR0 <TrigOutID> 1`.

Example: Digital output 1 is to be active when axis 1 of the mechanics is in motion.

Send:

```
CT0 1 2 1
CT0 1 3 6
```

## TR0 1 1

### Setting "Position+Offset" Trigger Mode

The **Position+Offset** trigger mode is suitable for scanning applications. The first trigger pulse is output when the axis has reached a specified position (TriggerPosition). The next trigger pulses are output respectively when the axis position equals the sum of the last valid trigger position and a specified distance (TriggerStep). The trigger output is stopped when a stop value has been reached. The polarity sign of the distance value determines the direction of motion in which trigger pulses are to be output.

The pulse width is one servo cycle. The unit for TriggerPosition, TriggerStep, and the stop value depends on the settings for the Numerator Of The Counts-Per-Physical-Unit Factor parameter (0xE) and the Denominator Of The Counts-Per-Physical-Unit Factor parameter (0xF). Default is mm.

### Information

The velocity setting of the axis must be appropriate for the distance setting (TriggerStep) commanded by the CT0 command. Recommended value:

Maximum velocity = track \* n kHz / 2

where n is the frequency of the C-884.xDC's servo cycle.

- Configure the digital output line <TrigOutID> to be used as trigger output:
    - Send CT0 <TrigOutID> 2 A, where A indicates the axis to be moved.
    - Send CT0 <TrigOutID> 3 7, where 7 determines the Position+Offset trigger mode.
    - Send CT0 <TrigOutID> 1 S, where S indicates the distance.
    - Send CT0 <TrigOutID> 10 TriPos, where TriPos indicates the position for the output of the first trigger pulse.
    - Send CT0 <TrigOutID> 9 Stop, where Stop indicates the stop value.
  - If you want to enable the conditions for trigger output, send TR0 <TrigOutID> 1.
- Example 1: The first trigger pulse is to be output on digital output line 1 if the absolute position of axis 1 is 1.5 mm. A pulse should then be output on this line every time axis 1 has covered a distance of 0.1 µm in the positive direction. The last trigger pulse is to be output if the absolute axis position is 2.5 mm.

Send:

```
CT0 1 2 1
CT0 1 3 7
CT0 1 1 0.0001
CT0 1 10 1.5
CT0 1 9 2.5
```

TR0 1 1

Example 2: The first trigger pulse is to be output on digital output line 2 if the absolute position of axis B is 0.4 mm. A pulse should then be output on this line every time axis B has covered a distance of 1 µm in the negative direction. The last trigger pulse is to be output if the absolute axis position is 0.1 mm.

Send:

```
CT0 2 2 B
CT0 2 3 7
CT0 2 1 -0.001
CT0 2 10 0.4
CT0 2 9 0.1
TR0 1 1
```

### Setting "Single Position" Trigger Mode

The selected digital output line is active in **Single Position** trigger mode, when the axis position has reached or exceeded a specified position (TriggerPosition).

The unit for TriggerPosition depends on the settings for the Numerator Of The Counts-Per-Physical-Unit Factor parameter (0xE) and the Denominator Of The Counts-Per-Physical-Unit Factor (0xF) parameter. Default is mm.

1. Configure the digital output line <TrigOutID> to be used as trigger output:
  - a) Send `CTO <TrigOutID> 2 A`, where A indicates the axis to be moved.
  - b) Send `CTO <TrigOutID> 3 8`, where 8 determines the Single Position trigger mode.
  - c) Send `CTO <TrigOutID> 10 TriPos`, where TriPos indicates the position at which the output line is to become active.

2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example: Digital output line 1 is to become active when the absolute position of axis 1 is at least 1.5 mm.

Send:

```
CTO 1 2 1
CTO 1 3 8
CTO 1 10 1.5
TRO 1 1
```

### Setting "Hardware Trigger" Mode

The **HardwareTrigger** mode basically corresponds to the "Position+Offset" trigger mode but is run by the C-884.xDC's FPGA circuit (shorter processing time).

The first trigger pulse is output when the axis has reached a specified position (TriggerPosition). The next trigger pulses are output respectively when the axis position equals the sum of the last valid trigger position and a specified distance (TriggerStep). The trigger output is stopped when a stop value is reached. The polarity sign of the distance value determines the direction of motion in which trigger pulses are to be output. A specified factor n (PulseWidth) determines the pulse width as follows:  $\text{Pulse width} = n * 33.3 \text{ ns}$ .

### Information

The velocity setting of the axis must be appropriate for the distance setting (TriggerStep) commanded by the CTO command. Recommended value:

Maximum velocity = track \* n kHz / 2

where n is the frequency of the C-884.xDC's servo cycle.

Fixed assignment of the axes to the digital output lines applies to the HardwareTrigger mode: Axis 1 to line 1, axis 2 line 2 etc.

1. Configure the digital output line <TrigOutID> to be used as trigger output:
  - a) Send `CTO <TrigOutID> 2 A`, where A indicates the axis to be moved.
  - b) Send `CTO <TrigOutID> 3 9`, where 9 indicates the HardwareTrigger trigger mode.
  - c) Send `CTO <TrigOutID> 1 S`, where S indicates the distance.
  - d) Send `CTO <TrigOutID> 10 TriPos`, where TriPos indicates the position for the output of the first trigger pulse.
  - e) Send `CTO <TrigOutID> 9 Stop`, where Stop indicates the stop value.
  - f) Send `CTO <TrigOutID> 11 n`, where n indicates the factor for calculating the pulse width.

2. If you want to enable the conditions for trigger output, send `TRO <TrigOutID> 1`.

Example 1: The first trigger pulse is to be output on digital output line 1 if the absolute position of axis 1 is 1.5 mm. A pulse should then be output on this line every time axis 1 has covered a distance of 0.5 µm in the positive direction. The last trigger pulse is to be output if the absolute axis position is 2.5 mm. The pulse width should be approximately 0.8 µs.

Send:

```
CTO 1 2 1
CTO 1 3 9
CTO 1 1 0.0005
CTO 1 10 1.5
CTO 1 9 2.5
CTO 1 11 24
TRO 1 1
```

Example 2: The first trigger pulse is to be output on digital output line 1 if the absolute position of axis B is 0.4 mm. A pulse should then be output on this line every time axis B has covered a distance of 1  $\mu$ m in the negative direction. The last trigger pulse is to be output if the absolute axis position is 0.1 mm. The pulse width should be approximately 0.166  $\mu$ s.

Send:

```
CTO 1 2 B
CTO 1 3 9
CTO 1 1 -0.001
CTO 1 10 0.4
CTO 1 9 0.1
CTO 1 11 5
TRO 1 1
```

#### Setting Signal Polarity

The polarity of the signal at the digital output used for triggering can be selected with the Polarity CTO parameter. The polarity can have the following values:

- active high = 1 (default setting)
  - active low = 0
1. Configure the digital output line <TrigOutID> to be used as trigger output:
    - a) Send `CTO <TrigOutID> 7 P`, where P indicates the polarity.

Example: The signal polarity for digital output line 1 is to be set to active low.

Send:

```
CTO 1 7 0
```

### 8.4.4 Digital Input Signals

The number of input lines available on the C-884.xDC can be queried with the `TI0?` command. The status of the digital input lines can be queried with the `DIO?` command.

Application possibilities of the digital input signals:

- Use in macros (`CPY`, `JRC`, `MEX`, `WAC`)
- Use as switch signals (`FED`, `FRF`)

#### Using Digital Input Signals as Switch Signals

The digital inputs of the `I/O` socket can be used as source of the reference point and limit switch signals (e.g., for reference moves) for an axis.

#### Information

The level of the digital input signal that you use instead of the reference switch or a built-in limit switch may only change once across the entire travel range.

- Use suitable signal sources.
- If necessary, invert the signal logic of the digital input lines by setting the corresponding parameters.

### Using Digital Input as Reference Signal

#### Information

The **Has Reference?** parameter (0x14) has no influence on the use of a digital input line as the source of the reference signal.

1. Select the source of the reference signal for the axis by changing the **Source Of Reference Signal** parameter (0x5C).
2. If necessary, invert the signal logic of the digital input line by setting the **Invert Reference?** parameter (0x31).

### Using Digital Inputs as Source of the Limit Switch Signals

#### Information

Several digital inputs can be selected as the source for a limit switch signal. If a limit switch signal is used for reference moves, only **one** digital input line may be selected as the source of the limit switch signal.

The **Has No Limit Switches?** parameter (0x32) determines whether the C-884.xDC evaluates the signals of the limit switch built into the connected mechanics. This parameter has no influence on the use of digital input lines as the source of the limit switch signal.

1. Select the source(s) of the negative limit switch signal for an axis by changing the **Source Of Negative Limit Signal** parameter (0x5D).
2. Select the source(s) of the positive limit switch signal for an axis by changing the **Source Of Positive Limit Signal** parameter (0x5E).
3. If necessary, invert the signal logic of the digital input lines by setting the **Invert Digital Input Used For Negative Limit** parameter (0x5F) and **Invert Digital Input Used For Positive Limit** parameter (0x60).

Example: Digital input lines 1, 3, and 4 are to be used for axis 1 as the sources of the positive limit switch signal. In addition, the signal polarity of lines 1 and 3 is to be inverted for axis 1.

Send:

SPA 1 0x5E 13 to select lines 1, 3, and 4

SPA 1 0x60 5 to invert the signal polarity of lines 1 and 3

## 8.4.5 Commands

C		Page
CPY	Copy Into Variable	129
CTO	Set Configuration Of Trigger Output	131
CTO?	Get Configuration Of Trigger Output	133
D		Page
DIO	Set Digital Output Lines	136
DIO?	Get Digital Input Lines	136
F		Page
FED	Find Edge	141
FRF	Fast Reference Move To Reference Switch	142

J		Page
JRC	Jump Relatively Depending On Condition	158
M		Page
MEX	Stop Macro Execution Due To Condition	163
T		Page
TIO?	Tell Digital I/O Lines	179
TRO	Set Trigger Output State	181
TRO?	Get Trigger Output State	181
W		Page
WAC	Wait For Condition	184

### 8.4.6 Parameters

0x14	Has Reference?	Do the mechanics have a reference switch? Activates respectively deactivates reference moves to the installed reference switch. 0 No reference switch 1 Reference switch available
0x5C	Source Of Reference Signal	Reference signal source for axis motion to the reference switch.
0x31	Invert Reference?	Should the reference signal be inverted? Inverts the signal of the reference switch or a digital input that is used instead of the reference switch. 0 Reference signal not inverted 1 Reference signal inverted
0x32	Has No Limit Switches?	Do the mechanics not have limit switches? Activates motion stop at the installed limit switches. 0 Mechanics have limit switches 1 Mechanics do not have limit switches
0x5D	Source Of Negative Limit Signal	Reference signal source for axis motion to the negative travel range limit.
0x5E	Source Of Positive Limit Signal	Reference signal source for axis motion to the positive travel range limit.
0x5F	Invert Digital Input Used For Negative Limit	Inverts the polarity of the digital inputs that are used as the source of the negative limit switch signal.
0x60	Invert Digital Input Used For Positive Limit	Inverts the polarity of the digital inputs that are used as the source of the positive limit switch signal.

## 8.5 Analog Input Signals

The analog inputs of the C-884.xDC are available on the *I/O* socket. The number of input lines available on the C-884.xDC can be queried with the [TAC?](#) command. The voltage at the

analog inputs can be queried with the [TAV?](#) command. The analog input signals can be recorded with the data recorder ([DRC](#) command with recording option 81).

Potential applications:

- Use in macros ([CPY](#), [JRC](#), [MEX](#), [WAC](#))
- Scan applications with PIMikroMove

### 8.5.1 Connecting Analog Signal Sources

#### Overview



Analog input signals (-10 V to 10 V) can be fed to the C-884.xDC via pins 1, 10, 11, and 19 of the *I/O* socket .

#### Tools and Accessories

- Suitable signal source
- Suitable D-sub 26-pole (m) connector, with cable

#### Connecting an Analog Signal Source

1. Connect an appropriate signal source to one of pins 1, 10, 11 or 19 of the *I/O* socket of the C-884.xDC.

### 8.5.2 Commands

C		Page
CPY	Copy Into Variable	129
D		Page
DRC	Set Data Recorder Configuration	137
J		Page
JRC	Jump Relatively Depending On Condition	158
M		Page
MEX	Stop Macro Execution Due To Condition	163
T		Page
TAC?	Tell Analog Channels	176
TAV?	Get Analog Input Voltage	176
W		Page
WAC	Wait For Condition	184

### 8.5.3 Parameters

## 8.6 Controlling with HID

HID (Human Interface Device) denotes an input or output device connected to the C-884.xDC and is intended to be operated manually. Joysticks and gamepads are typical HID.

HID control means controlling motion variables of a positioner's axis connected to the C-884.xDC by displacing an axis of the HID.

### 8.6.1 How it Works

An HID's axis can control the following motion variables of a positioner's axis connected to the C-884.xDC:

- Velocity
- Maximum velocity

The relationship between the displacement of the axis of the HID and the motion variable of the positioner axis is created by the C-884.xDC using a lookup table. The 256 values in the lookup table are factors that are applied to the motion variable to be controlled during HID control (further information: [HIA \(p. 145\)](#) command description). The values range from -1.0000 to 1.0000.

During HID control, the target position of the controlled axis of the C-884.xDC is set to the soft limit that is specified by the **Maximum Travel In Positive Direction (Phys. Unit)** parameter (0x15) respectively **Maximum Travel In Negative Direction (Phys. Unit)** parameter (0x30). Details on the parameters can be found in "travel range and Soft Limits". When HID control is deactivated, the target position is set to the current position of the controlled axis.

#### Information

The **maximum velocity** motion variable can only be assigned to an HID axis when the **velocity** motion variable has been assigned to a different HID axis.

#### Information

Motion commands are not permitted when HID control is enabled for the axis.  
HID control is not possible in open-loop operation (servo mode Off).

### 8.6.2 HID Control Configuration

Control of the axis/axes connected to the C-884.xDC by the HID's axes is configured via the [HIA](#) command. The current HID control configuration can be queried with the [HIA?](#) command.

Assigning a lookup table to an HID axis is done with the [HDT](#) command. The current lookup table assignments to HID axes can be queried with the [HDT?](#) command. The values in the lookup table are factors that are applied to the motion variable to be controlled during HID control. The C-884.xDC's firmware gives a choice of two predefined lookup table types (linear and parabolic) and allows four customer-specific lookup tables to be filled with individual values. Lookup tables can be filled with values with the [HIT](#) command. [HIT?](#) queries the values of the points in the lookup tables.

Use the [HIN](#) command to activate/deactivate HID control of the axes connected to the C-884.xDC. The [HIN?](#) command queries the HID control activation status. When HID control is deactivated, the target position is set to the current position of the controlled axis.

The [HIS?](#) command queries the properties of the HID's operating elements and the properties can be set with the [HIS](#) command. The current status of the HID buttons can be queried with [HIB?](#) and the current displacement of HID axes with [HIE?](#). The status of HID output units (LEDs) can be queried with [HIL?](#) and set with [HIL](#).



The direction of motion of HID-controlled axes can be inverted via the **Invert Direction Of Motion For Joystick-Controlled Axis?** parameter (0x61). The maximum values for velocity, acceleration, and deceleration during HID control are specified with the following parameters:

- **Closed-Loop Velocity For HI Control (Phys. Unit/s)** (0x74)
- **Closed-Loop Acceleration For HI Control (Phys. Unit/s<sup>2</sup>)** (0x75)
- **Closed-Loop Deceleration For HI Control (Phys. Unit/s<sup>2</sup>)** (0x76)

### 8.6.3 Programming HID Control

HID output units (e.g., buttons and LEDs) can be used for example, in controller macros to program HID control.

### 8.6.4 Connecting an HID

#### Overview



C-884.xDC's connector for a manually operated device such as a joystick or gamepad

#### Tools and Accessories

- Suitable HID with type A USB connector such as a joystick or gamepad
- If several HID's are to be connected: Suitable USB hub

#### Information

The C-884.xDC assigns identifiers to connected HID's as follows:

- When an HID is connected directly to the C-884.xDC, it always receives the identifier 1.
- When human interface devices are connected to a USB hub connected to the C-884.xDC when the C-884.xDC is **switched off**, the C-884.xDC assigns the identifiers of the human interface devices according to the numbering of the physical slots when it is switched on.
- When HID's are connected to a USB hub connected to the C-884.xDC when the C-884.xDC is **switched on**, the C-884.xDC assigns the identifiers in the order of connection. The identifiers of the HID's can therefore change after the C-884.xDC is rebooted.
- For a consistent assignment of the identifiers, always connect the HID's in the same order and in the same slots on the hub when the C-884.xDC is **switched on**.

#### Connecting an HID to the C-884.xDC

1. Connect the HID to the USB connection type A on the C-884.xDC.

First of all, we recommend testing the HID's operating elements after connecting it to the C-884.xDC. If the response behavior of the HID's axes does not meet your requirements, it is possible to calibrate its axes.

HID control can be set up and activated after testing and optional calibration of the axes.

The procedure for PIMikroMove is described in the following.

### 8.6.5 Testing the HID and Calibrating the Axes

The positioner does not need to be connected to the C-884.xDC for testing the HID and calibrating its axes.

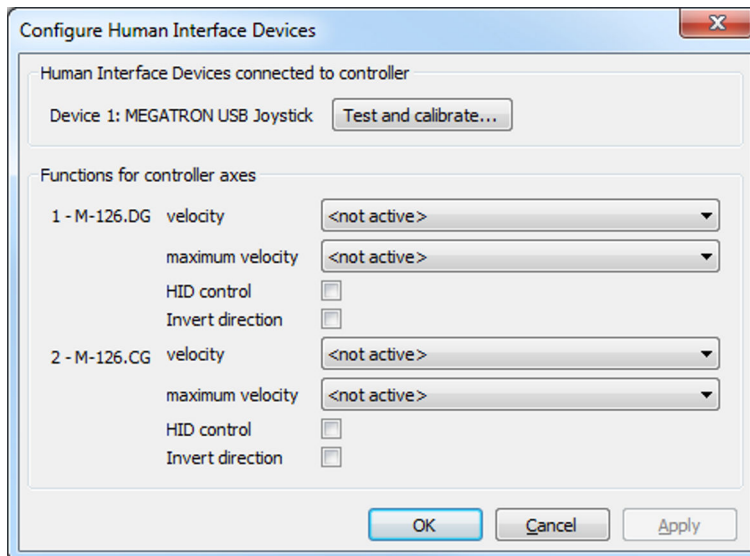
When calibrating the HID's axes in PIMikroMove, the appropriate lookup table must be selected. The parabolic lookup table allows for greater sensitivity when moving slowly. Lookup tables named "User Table" are intended to be filled with individual values.

## Requirements

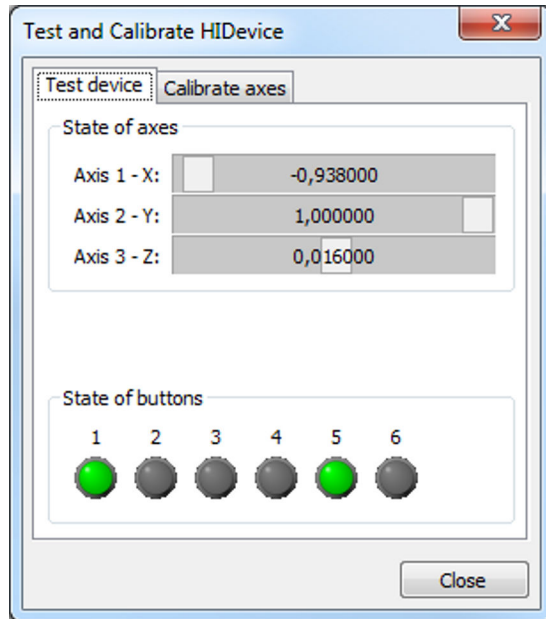
- ✓ PIMikroMove is [installed on the PC \(p. 49\)](#).
- ✓ PIMikroMove has established communication between the C-884.xDC and the [PC \(p. 57\)](#).
- ✓ The C-884.xDC has been connected to the [HID \(p. 93\)](#).

## Testing the HID

1. Open the window for configuring HID control via the **C-884.xDC > Configure controller HIDevice(s)...** menu item in PIMikroMove's main window.  
→ *The **Configure Human Interface Devices** window opens.*



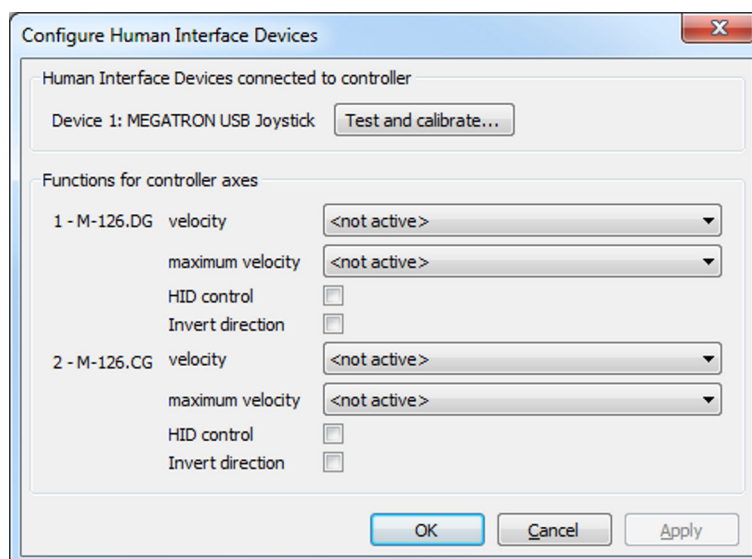
2. Open the window for testing and calibrating the HID by clicking **Test and calibrate....**  
→ *The **Test and Calibrate HIDevice** window opens.*
3. Select the **Test device** tab in the **Test and Calibrate HIDevice** window and test the HID's operating elements:
  - a) Move the HID's axes and at the same time, watch the status indicators in the **State of axes** area.
  - b) Press the HID's buttons and at the same time, watch the status indicators in the **State of buttons** area.
  - c) Enter various values into the fields in the **State of LEDs** area (if any) and at the same time, watch the behavior of the corresponding operating elements on the HID.
 → *In the example shown, a joystick with 3 axes and 6 buttons is connected to a C-884.xDC. Current status in the figure: The X axis of the joystick is displaced in the negative direction, the Y axis is displaced in the positive direction, the Z axis is displaced in the positive direction, and buttons 1 and 5 are pressed.*



4. Depending on what is next, do the following:
  - If you want to calibrate the HID's axes directly afterwards, proceed as [described below \(p. 95\)](#).
  - If you want to set up and activate HID control for the C-884.xDC directly afterwards, close the **Test and Calibrate HIDevice** window with **Close** and continue as described in "[Setting up and Activating HID Control \(p. 97\)](#)".
  - If you do not want to make any further settings at this point, close **Test and Calibrate HIDevice** with **Close** and the **Configure Human Interface Devices** window with **OK**.

### Calibrating HID Axes

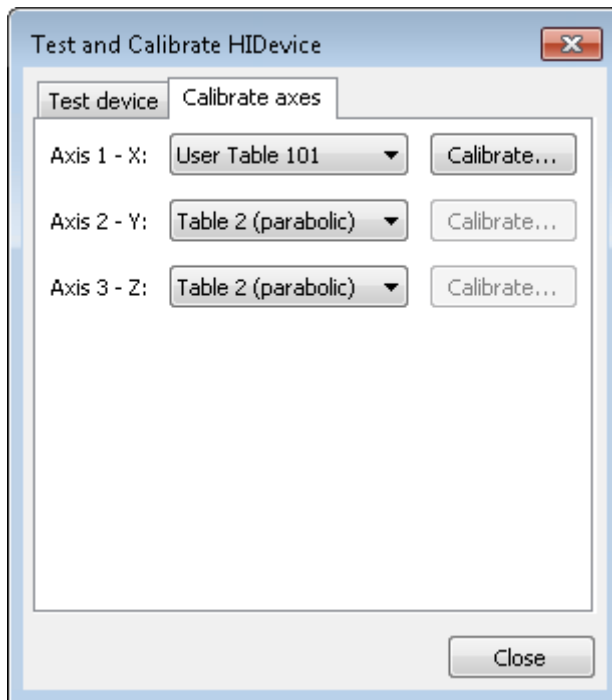
1. If necessary, open the window for configuring HID control via the **C-884.xDC > Configure controller HIDevice(s)...** menu item in PIMikroMove's main window.  
 → *The **Configure Human Interface Devices** window opens.*



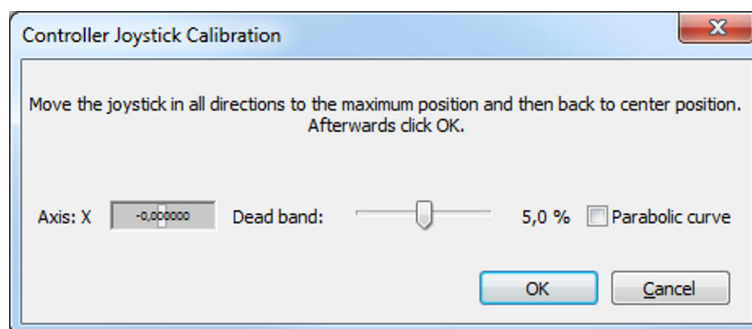
2. If necessary, open the window for calibrating the HID and click the **Test and calibrate...** button.  
 → *The **Test and Calibrate HIDevice** window opens.*
3. Select the **Calibrate axes** tab in the **Test and Calibrate HIDevice** window.

4. Select the respective lookup tables to be used for the HID's axes via the selection fields in the **Calibrate axes** tab.

→ The example shows that a user-defined lookup table was selected for axis 1. The respective predefined parabolic lookup table was retained for axes 2 and 3.



5. If you have selected a user-defined lookup table and want to fill the table with values:
  - a) Click the corresponding **Calibrate...** button to open the **Controller Joystick Calibration** window.



- b) Move the HID's axes to all extreme positions. The custom lookup table values are determined in this way.
    - c) Let go of the axis.
    - d) If you want to change the neutral area of the axis (i.e., the area around the center position of the axis where no change in the controlled motion variable is triggered), set the **Dead band** slider accordingly.
    - e) If the values in the user-defined lookup table are to describe a parabolic waveform, click the **Parabolic curve** checkbox.
    - f) Click **OK** in the **Controller Joystick Calibration** window to copy the appropriate values from the lookup table.
- The lookup table values are written to the C-884.xDC's volatile memory. The writing progress is indicated in a separate window. The window for the writing process and the Controller Joystick Calibration window automatically close after the writing process has finished.
6. Close the **Test and Calibrate HIDevice** window with **Close**.

7. Depending on what is next, do the following:

- If you want to set up and activate HID control for the C-884.xDC directly afterwards, continue as described in "[Setting up and Activating HID Control \(p. 97\)](#)".
- If you want to save the lookup table assignments to the HID's axes and the content of user-defined lookup table to the C-884.xDC's nonvolatile memory directly afterwards, close the **Configure Human Interface Devices** window with **OK** and continue as described in "[Saving the Configuration of HID Control Permanently \(p. 98\)](#)".
- If you do not want to make any further settings at this point, close the **Configure Human Interface Devices** window with **OK**.

### 8.6.6 Setting and Activating HID Control

The following motion variables of the C-884.xDC's axes can be controlled via HID:

- Velocity - Velocity for motion of the axis
- Maximum Velocity - Maximum velocity for motion of the axis

Before activating HID control, the following steps are recommended:

- Testing the HID
- Calibrating the HID axes

Refer to "[Testing the HID and Calibrating the Axes \(p. 93\)](#)" for a description of these steps.

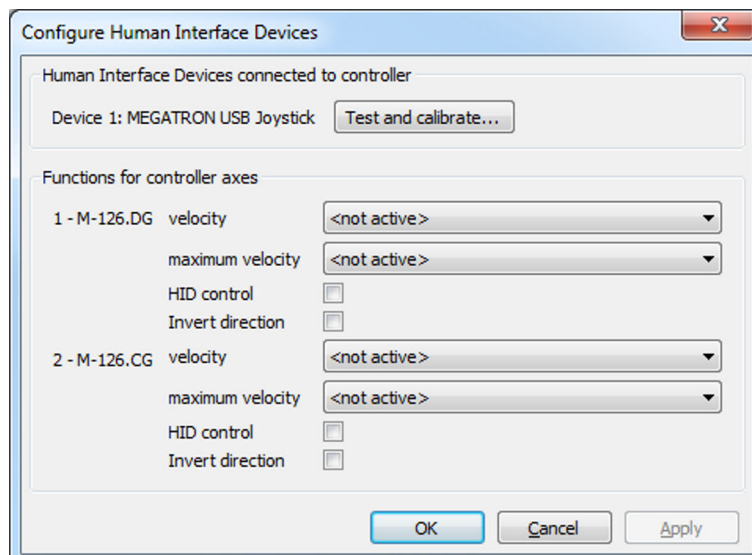
#### Requirements

- ✓ If necessary, you have done a successful respective [reference move \(p. 62\)](#) for the axes of the C-884.xDC.
- ✓ You have [connected the HID to the C-884.xDC \(p. 93\)](#).
- ✓ All devices are still ready for operation.

#### Setting Up and Activating HID Control

1. If necessary, open the window for configuring HID control via the **C-884.xDC > Configure controller HID device(s)...** menu item in PIMikroMove's main window.

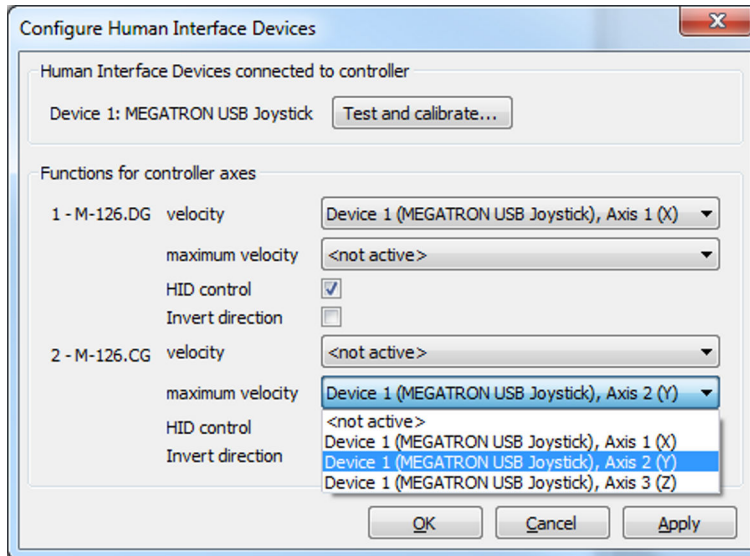
→ *The **Configure Human Interface Devices** window opens.*



2. Make the following respective settings for the C-884.xDC's axes displayed in the **Functions for controller axes** area:
  - a) Select the HID's axis in the corresponding field that is to be used for the motion variable to be controlled.
  - b) Activate HID control by clicking the **HID control** checkbox.

c) If the direction of motion is to be inverted during HID control, click the **Invert direction** checkbox.

→ In the example shown, axis 1 of the controller is set to the X-axis and axis 2 of the controller is set to the Y-axis of HID 1 (USB joystick), and HID control is activated.



3. Click the **Apply** button in the **Configure Human Interface Devices** window to activate the settings.
4. Send the settings for setting up HID control to the C-884.xDC by clicking the **OK** button.  
→ The **Configure Human Interface Devices** window closes.
5. Make sure that servo mode for the C-884.xDC's axes is switched on in PIMikroMove (e.g., by clicking the **Servo** checkbox in the **Axes** tab in PIMikroMove's main window).  
→ The C-884.xDC's axes can now be controlled by the HID according to the settings made.
6. If you want to save the new settings for HID control to the C-884.xDC's nonvolatile memory, continue as described in "[Saving the Configuration of HID Control Permanently \(p. 98\)](#)".

### Saving the Configuration of the HID Control Permanently

Select the **C-884.xDC > Save parameters to non-volatile memory** menu item in PIMikroMove's main window.

→ The **Save Parameters to Non-Volatile Memory** dialog opens.

1. Enter either the password HID in the selection field of the **Save Parameters to Non-Volatile Memory** dialog, or select the **Settings of HDT, HIA, HIT (HID)** entry.
2. Click **OK** to save and to close the dialog.

## 8.6.7 Commands

H		Page
HDT	Set HID Default Lookup Table	144
HDT?	Get HID Default Lookup Table	144
HIA	Configure Control Done By HID Axis	145
HIA?	Get Configuration Of Control Done By HID Axis	146
HIB?	Get State Of HID Button	146

HIE?	Get Deflection Of HID Axis	147
HIL	Set State of HID LED	147
HIL?	Get State Of HID LED	147
HIN	Set Activation State For HID Control	148
HIN?	Get Activation State Of HID Control	148
HIS	Configure HI Device	149
HIS?	Get Configuration Of HI Device	150
HIT	Fill HID Lookup Table	151
HIT?	Get HID Lookup Table Values	151

### 8.6.8 Parameters

0x15	Maximum Travel In Positive Direction (Phys. Unit)	Soft limit in positive direction, in relation to the zero position. The positive travel range limit cannot be used for reference moves if this value is smaller than the position value for the positive travel range limit (that results from the sum of parameters 0x16 and 0x2F). The value can be negative.
0x30	Maximum Travel In Negative Direction (Phys. Unit)	Soft limit in a negative direction, in relation to the zero position. The negative travel range limit cannot be used for reference moves if this value is greater than the position value for the negative travel range limit (that results from the difference between parameters 0x16 and 0x17). The value can be negative.
0x61	Invert Direction Of Motion For Joystick-Controlled Axis?	Inverts the direction of motion for HID-controlled axes. 0 Direction of motion not inverted (default) 1 Direction of motion inverted
0x74	Closed-Loop Velocity For HI Control (Phys. Unit/s)	Maximum velocity during HID control.
0x75	Closed-Loop Acceleration for HI Control (Phys. Unit/s <sup>2</sup> )	Maximum acceleration during HID control.
0x76	Closed-Loop Deceleration for HI Control (Phys. Unit/s <sup>2</sup> )	Maximum deceleration during HID control.

## 8.7 Controller Macros

The C-884.xDC can save and process command sequences as macros.

The following functionalities make macros an important tool in many application areas:

- Several macros can be stored at the same time.



- Any macro can be defined as the startup macro. The startup macro runs each time the C-884.xDC is switched on or rebooted.
- Processing a macro and stopping a macro can be linked to conditions. In this way, loops can be realized as well.
- Macros can call up themselves or other macros at several nesting levels.
- Variables can be set for the macro and in the macro itself and used in different operations.
- Input signals can be evaluated for conditions and variables.

#### Working with Macros

- The C-884.xDC can save up to 32 macros simultaneously.
- Up to 10 nesting levels are possible in macros.
- Local and global variables can be used in macros.
- A macro is overwritten if a macro with the same name is rerecorded.
- For working with controller macros, it is recommended to use the **Controller macros** tab in PIMikroMove. There you can conveniently record, start, and manage controller macros.
- The PITerminal or the PIMikroMove's **Command entry** window can be used for entering commands, e.g., for starting macros.

#### GCS commands in macros

Basically all GCS commands can be included in a macro. Exceptions:

- **RBT** for rebooting the C-884.xDC
- **MAC BEG** and **MAC END** for macro recording
- **MAC DEL** for deleting a macro

Query commands can be used in macros in conjunction with the **CPY**, **JRC**, **MEX**, and **WAC** commands. Otherwise they have no effect because macros do not send responses to interfaces.

### 8.7.1 Recording Macros

The **MAC BEG** and **MAC END** commands may not be specified when macros are recorded in the **Controller macros** tab in PIMikroMove.

A macro is overwritten if a macro with the same name is rerecorded.

If you record a macro on a controller whose address differs from 1, pay attention to the following when entering commands that need to be an integral part of the macro:

- If you are working with PITerminal and have established communication via the **Connect...** button, the target address must typed into every command line.
- If you are working with PIMikroMove or have established PITerminal communication via the **GCS DLL...** button, the target address is sent automatically and may not be typed in.

Recording macros for PITerminal and PIMikroMove is described in the following.

1. Start macro recording.
  - If you are working with PITerminal or in the **Command entry** in the PIMikroMove's window: Send the **MAC BEG** macro name command where "macro name" is the name of the macro.
  - If you are working in PIMikroMove in the **Controller macros** tab: Click the **Create new empty macro** icon to create a tab for entering a new macro.
2. Enter the commands to be included in the "macro name macro" line-by-line using the normal command syntax.
 

Macros can call up themselves or other macros at several nesting levels.
3. End the macro recording.
  - If you are working with PITerminal or in PIMikroMove's **Command entry** window: Send the **MAC END** command.
  - If you are working in PIMikroMove in the **Controller macros** tab: Click the **Send macro to controller** symbol and enter the macro name into a separate dialog window.



→ The macro has been stored in the nonvolatile memory of the C-884.xDC.

4. If you want to check in PITerminal or in the PIMikroMove's **Command entry** window whether the macro was recorded correctly:
  - a) Query which macros are saved in the C-884.xDC by sending the `MAC?` command.
  - b) Query the content of "macro name" with the `MAC? macro name` command.
5. If you want to check PIMikroMove's **Controller macros** tab to see whether the macro was recorded correctly:
  - a) Click the **Read list of macros from controller** icon.
  - b) Mark the macro to be checked in the list on the left-hand side and click the **Load selected macro from controller** icon.

#### Example Macro: Moving the Axis Back and Forth

The axis <axis> is to move back and forth. 3 macros are recorded for this purpose. Macro 1 starts the motion in the positive direction and waits until the axis has reached the target position. Macro 2 performs this task for the negative direction of motion. Macro 3 calls up macro 1 and 2.

Record the macros by sending:

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

### 8.7.2 Running the Macros

Any commands can be sent from the command line while a macro is running on the controller. The macro content and motion commands received from the command line can overwrite each other.

It is not possible to run several macros simultaneously. Only one macro can be run at a time.

You can link the conditions for running the macro with the `JRC` and `WAC` commands. The commands must be included in the macro.

A delay time for running the macro can be specified with `DEL`.

Variables can be used in macros. Setting is done via the `VAR` command and querying variable values via `VAR?`. Responses to query commands can be copied into variables with `CPY` and values can be added and saved to variables with `ADD`.

In the following, PITerminal or PIMikroMove's **Command entry** window is used to enter commands.

1. If the macro should continue running despite an error: Set the **Ignore Macro Error?** parameter (0x72) accordingly: Send the `SPA 1 0x72 Status` command where "Status" can take the value 0 (stop macro on error [standard]) or 1 (ignore macro error).
2. Start the macro:
  - If the macro is to be run once, send `MAC START macro name string` where "macro name" is the name of the macro.

- If the macro is to be run *n* times, send the `MAC NSTART` macro name *n* string command where "macro name" is the name of the macro and "n" indicates the number of times to be run.

Specifying "string" is optional and stands for the values of local variables. The values only need to be specified when the macro contains corresponding local variables. The sequence of the values in the input must correspond to the numbering of the appropriate local variables, starting with the value of the local variable 1. The individual values must be separated from each other by spaces.

3. If you want to check that the macro is running:

- Query whether a macro is running on the controller by sending the `#8` command.
- Query the name of the macro currently running on the controller by sending the `RMC?` command.

#### Example Macro: Moving an Axis with Variable Travel Range Back and Forth

The axis <axis> is to move back and forth. The travel to the left and to the right is to be variably adjustable without having to change the macros used. Local and global variables are therefore used.

Create the global variables LEFT and RIGHT by sending:

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

LEFT therefore has the value 5, and RIGHT has the value 15. These values can be changed at any time, e.g., by sending the `VAR` command again.

Create the global variables again each time that the C-884.xDC is switched on or rebooted, since they are only written to the volatile memory of the C-884.xDC.

Record the MOVLR macro by sending:

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

MOVLR successively starts the MOVWAI macro (which is still to be recorded) for both directions of motion. The values of the global variables LEFT and RIGHT are used when MOVWAI is started, to set the value of the local variable 1 contained in MOVWAI (dollar signs and braces are necessary for the local variable 1 in the macro to actually be replaced by the value of the global variable and not by its name).

Record the MOVWAI macro by sending:

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

MOVWAI moves axis <axis> to the target position which is specified by the value of the local variable 1 and waits until the axis has reached the target position.

Start the execution of the MOVLR macro by sending:

```
MAC NSTART movlr 5
```

The MOVLR macro is executed five times in succession, i.e., axis <axis> alternately moves to the positions 5 and 15 five times. You can also select any other value for the number of executions.

#### Example macro: Realizing multiple calls of a macro in a loop

The TESTDION macro checks the status of the digital input lines on the I/O socket. It uses a local variable to identify the digital input line (1 to 4). A further macro is recorded with a loop so that the TESTDION macro does not have to be called separately for each input line.

Record the LOOPDION macro by sending:

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

The COUNTER variable is created with the value 1. After this, the TESTDION macro is started for the input line whose identifier is specified via the COUNTER variable. Then the value of the COUNTER is increased by 1. As long as the value of the COUNTER is less than 5, the macro line pointer subsequently jumps 2 lines back, so that TESTDION can be started for the next digital input line.

### 8.7.3 Stopping the Macros

Stopping the macro can be linked to a condition with the [MEX](#) command. The command must be included in the macro.

1. Stop the macro with the [#24](#) or [STP](#) commands.
2. If you want to check whether an error occurred while the macro was running, send the [MAC ERR?](#) command. The response shows the last error that has occurred.

### 8.7.4 Configuring a Startup Macro

Any macro can be defined as the startup macro. The startup macro runs each time the C-884.xDC is switched on or rebooted.

Deleting a macro does **not** delete its selection as a startup macro.

1. Set a macro as the startup macro:
  - Send the [MAC DEF macro name](#) command to set a macro as startup macro where "macro name" is the name of the macro.
  - If you want to cancel the selection of the startup macro and do not want to define another macro as the startup macro, send [MAC DEF](#) only.
2. Query the name of the currently defined startup macro by sending the [MAC DEF?](#) command.

#### Example Macro: Preparing an Axis for Closed-Loop Operation via a Startup Macro

The STARTCL macro switches the HID control off and the servo mode on for axis 1 and starts a reference move to the negative physical limit of the travel range. As STARTCL is defined as the startup macro, axis 1 is ready for closed-loop operation immediately after switch-on.

Send:

```
MAC BEG startcl
HIN 1 0
SVO 1 1
DEL 1000
FRF 5
MAC END
MAC DEF startcl
```

When this macro is used, the C-884.xDC's parameter settings should be adapted to the connected positioner in the nonvolatile memory. Alternatively, the parameter settings can also be configured in the volatile memory via the startup macro.

### 8.7.5 Deleting Macros

A macro cannot be deleted while it is running.

Deleting a macro does not delete its selection as a startup macro.

1. Delete a macro with the `MAC DEL macro name` command where "macro name" is the name of the macro.

### 8.7.6 Saving and Loading Macros

For example, making backups of controller macros on the PC can be useful before updating the firmware.


The use of the **Controller macros** tab in PIMikroMove is recommended for backing up and loading controller macros.

Saving and loading controller macros with PIMikroMove is described in the following.


#### Saving Controller Macros on the PC

1. Select the **Controller macros** tab in PIMikroMove's main window.
  2. Select the macros in the **Macros on controller** list that you want to back up to the PC:
    - Click the desired entry in the list to select a macro.
    - To select more than one macro, press and hold down the **shift** key and click the desired entries in the list.
    - To deselect, click an open area in the list.
- The **Save selected macros to PC** button becomes active when selecting one or more macros.



3. Save the selected macros on the PC:
    - a) Click the  button to open a directory selection window.
    - b) Select the directory on the PC where you want to save the macros.
    - c) Click **Save**.
- The macros are saved as text files (<macro name>.txt) in the selected directory of the PC.

#### Loading Controller Macros from the PC to the C-884.xDC

1. Select the **Controller macros** tab in PIMikroMove's main window.
  2. Load macros from the PC to the C-884.xDC:
    - a) Click the  button to open a file selection window.
    - b) Select the text files (<macro name>.txt) in the file selection window whose contents you want to load as a macro from the PC to the C-884.xDC.
    - c) Click **Open**.
- For each selected text file (<macro name>.txt), the content is loaded as a macro <macro name> into the C-884.xDC.

### 8.7.7 Example Macros

#### Example Macro: Stopping Motion by Pushbutton

1. Connect digital input line 1 on the **I/O** socket to an appropriate signal source.  
The digital input signal can be used e. g. for a conditional jump of the macro execution pointer.
2. Record the HALT macro on the controller.  
`MAC BEG halt`

```

MVR 1 5
JRC 2 DIO? 1 = 1
JRC -1 ONT? 1 = 0
HLT 1
MAC END

```

→ The macro has the following tasks:

*Start relative motion of axis 1*

*Set condition: If digital input line 1 has the high state, the macro execution pointer jumps two lines forward. This stops the axis. Otherwise, the macro continues to the next line.*

*Set condition: As long as axis 1 has not yet reached the target position, the macro execution pointer jumps back one line. A loop is established as a result.*

3. Start the HALT macro on the controller.

```
MAC START halt
```

→ Axis 1 starts to move. It is stopped by switching digital input line 1 to the high state (e. g. by pushbutton). Regardless of whether the axis has reached the target position or was halted previously, the error code is set to 10 via the HLT command.

4. If error code 10 interferes: Record alternative HALTVAR macro which uses a variable.

```

MAC BEG haltvar
MVR 1 5
JRC 2 DIO? 1 = 1
JRC -1 ONT? 1 = 0
CPY TARGET POS? 1
MOV 1 ${TARGET}
VAR TARGET
MAC END

```

→ The macro has the same tasks as the HALT macro. However, axis 1 is not stopped by pushbutton via the HLT command; instead the result of the POS? 1 query is copied to the TARGET variable. Then this variable is used as the target position for the MOV command. As a result, the axis stays right where it was. To clean up, TARGET is defined as empty with the VAR command which deletes the variable.

5. Start the HALTVAR macro on the controller.

```
MAC START haltvar
```

→ Axis 1 starts to move. It is stopped by switching digital input line 1 to the high state (e. g. by pushbutton). Error code 10 is not set because no halt or stop command is used.

### Example Macro: HID Control Alternating with Relative Motion

In the following example, a gamepad is connected as a human interface device. This gamepad has a motor for the vibration function. The vibration frequency of the motor can be adjusted (available as "LED" in the corresponding commands). Furthermore, the gamepad as at least one axis and at least 5 buttons. The values can be completely different for other human interface devices.

#### Task

The identifier of axis 1 was changed in X with the SAI command. The velocity of axis X is to be controlled with axis1 of HID 1 (e.g., gamepad). HID control should only be active when button 5 of the HID is pressed at the same time. This state is to be signalized by the slow running of the motor (LED3) in the HID. Buttons 1 to 4 of the HID also allow up to four positions to be saved in the controller (long press of the button) or approached by the axis (short press of the button). Fast running (vibration) of the HID motor is intended to signal that saving has taken place.

### Approach

1. Initialization
2. Loop for querying the state of the buttons of the HID (counter variable required)
3. Querying of the state of button 5 of the HID (enabling / disabling HID control)
4. Querying buttons 1 to 4 of the HID; Depending on the button state, the current position is transmitted and saved in the STORE<n> variable or the saved position is approached.

### Procedure

1. Connect the HID to the C-884.xDC's type A USB connector.
2. Record the STARTUP macro on the controller:

```
MAC BEG startup
HIN X 0
SVO X 1
FRF X
HIA X 3 1 1
CPY STORE1 POS? X
CPY STORE2 POS? X
CPY STORE3 POS? X
CPY STORE4 POS? X
MAC START MAINLOOP
MAC END
```

The macro has the following tasks:

- Deactivate HID control for axis X first
- Switch on the servo mode for axis X
- Start a reference move for axis X
- Configure HID control for axis X: The velocity is to be controlled by axis 1 of HID 1
- Initialize global variables for storing the position of axis X
- Start MAINLOOP macro for the main loop

3. Record the MAINLOOP macro on the controller:

```
MAC BEG mainloop
MAC START TESTJOYB
VAR counter 1
MAC START TESTDION $ {COUNTER}
ADD COUNTER $ {COUNTER} 1
JRC -2 VAR? COUNTER < 5
MAC START MAINLOOP
MAC END
```

The macro has the following tasks:

- Start TESTJOYB macro for HID control
- Assign the value 1 to the COUNTER variable
- Start the TESTDION macro for all buttons of the HID in succession (loop)
- Increment loop counter
- Jump back 2 lines (to MAC START TESTDION) as long as the loop counter is < 5
- Call itself to set up the main loop

4. Record the TESTJOYB macro on the controller:

```
MAC BEG testjoyb
MEX HIB? 1 5 = 0
HIN X 1
HIL 1 3 100
```

```
JRC 6 HIB? 1 5 = 0
DEL 50
HIL 1 3 0
JRC 3 HIB? 1 5 = 0
DEL 200
JRC -6 HIB? 1 5 = 1
HIN X 0
HIL 1 3 0
MAC END
```

The macro has the following tasks:

- Stop macro if button 5 on HID 1 is no longer pressed
- Activate HID control for axis X
- Switch motor on (slow running): Set vibration frequency of LED 3 of HID 1 to 100
- Jump forward 6 lines (to HIN X 0) if button 5 on HID 1 is no longer pressed
- Wait 50 ms
- Switch motor off: Set vibration frequency of LED 3 of HID 1 to 0
- Jump forward 3 lines (to HIN X 0) if button 5 on HID 1 is no longer pressed
- Wait 200 ms
- Jump back 6 lines (to HIL 1 3 100) if button 5 on HID 1 is still pressed
- Deactivate HID control for axis X
- Switch motor off: Set vibration frequency of LED 3 of HID 1 to 0

5. Record the TESTDION macro on the controller:

```
MAC BEG testdion
MEX VAR? 0 != 1
MEX HIB? 1 $1 = 0
DEL 300
JRC 3 HIB? 1 $1 = 1
MAC START MVAX2ST 1
MEX HIB? 1 $1 = 0
HIL 1 3 255
WAC HIB? 1 $1 = 0
HIL 1 3 0
CPY STORE$1 POS? X
MAC END
```

The macro has the following tasks:

- Stop running the macro if the number of local variables specified is not 1 when starting TESTDION
- Stop running the macro if the button on HID 1 specified by the local variable 1 is no longer pressed
- Wait 300 ms
- Jump 3 lines forward (to DEL 400) if the button specified by variable 1 is still pressed
- Start the MVAX2ST macro because the button was only briefly pressed. The value of the local variable 1 is also used for local variable 1 in MVAX2ST. MVAX2ST moves axis X to the position stored for the button.
- Stop macro if button is no longer pressed
- Wait 400 ms
- Switch motor on (fast running): Set vibration frequency of LED 3 of HID 1 to 255
- The macro continues to run only if the button is no longer pressed
- Switch motor off: Set vibration frequency of LED 3 of HID 1 to 0
- Save the current position of axis X in the global variable designated via local variable 1

6. Record the MVAX2ST macro on the controller:

```
MAC BEG MVAX2ST
CPY 2 VAR? STORE$1
MOV X $2
MAC END
```

The macro has the following tasks:

- Query the global variable designated via local variable 1 and copy its value to local variable 2
- Move the axis to the target position specified by local variable 2

#### 7. Run the STARTUP macro on the controller:

```
MAC START startup
```

HID control is activated by pressing button 5. When HID control is activated, the motor of human interface device1 runs (slow running) and indicates that the buttons of the HID should not be pressed. After button 5 has been released, HID control is deactivated and the motor switches off. Buttons 1 to 4 can now be used for moving to stored positions or for saving the current position.

The corresponding button on the HID is pressed briefly to move the positioner's axis to a stored position.

To save the current position of the positioner's axis, a button on the HID is pressed until the motor starts to vibrate (fast running).

### 8.7.8 Commands

#		Page
#24	Stop All Axes	126
#8	Query If Macro Is Running	126
A		Page
ADD	Add And Save To Variable	127
C		Page
CPY	Copy Into Variable	129
D		Page
DEL	Delay The Command Interpreter	134
J		Page
JRC	Jump Relatively Depending On Condition	158
M		Page
MAC BEG	Call Macro Function: BEG	159
MAC DEF	Call Macro Function: DEF	159
MAC DEF?	Call Macro Function: DEF?	159
MAC DEL	Call Macro Function: DEL	160
MAC END	Call Macro Function: END	160
MAC ERR?	Call Macro Function: ERR?	160
MAC NSTART	Call Macro Function: NSTART	160
MAC START	Call Macro Function: START	161



<b>MAC?</b>	List Macros	161
<b>MEX</b>	Stop Macro Execution Due To Condition	163
<b>R</b>		<b>Page</b>
<b>RBT</b>	Reboot System	167
<b>RMC?</b>	List Running Macros	168
<b>S</b>		<b>Page</b>
<b>STP</b>	Stop All Axes	175
<b>V</b>		<b>Page</b>
<b>VAR</b>	Set Variable Value	182
<b>VAR?</b>	Get Variable Values	182
<b>W</b>		<b>Page</b>
<b>WAC</b>	Wait For Condition	184

### 8.7.9 Parameters

<b>0x72</b>	Ignore Macro Error?	Ignore macro error? 0 Stop macro on error (default) 1 Ignore error
-------------	---------------------	--

## 8.8 Working with GCS Commands

### 8.8.1 GCS Command Syntax

#### Notation

The following notation is used to define the GCS syntax and to describe the commands:

<...>	Angle brackets indicate an argument of a command, can be an element identifier or a command-specific parameter.
[...]	Square brackets indicate an optional entry
{...}	Braces indicate repeated specifications, i.e., it is possible to access more than one element (e.g., several axes) in one command line.
␣	Line Feed (ASCII character 10), default termination character (character at the end of a command line)
␣	Space (ASCII character 32), empty space
→	Horizontal tab (ASCII character 9)
#...	Single-character command, "..." indicates the ASCII character in decimal notation, e.g., #7 for ASCII character <sup>B</sup> 7.

#### Syntax

A GCS command consists of three letters, e.g., **CMD**, or three letters and a question mark, e.g., **CMD?**.

Exceptions:

- Single-character commands consist of only one ASCII character. In this manual, the ASCII character is written as combination of # and the character code in decimal format, e.g., as #24.
- \*IDN? (for GPIB compatibility).

The command mnemonic is not case sensitive. The command mnemonic and all arguments (e.g., axis and channel identifiers, parameters etc.) must be separated from each other by a space ( ). The command line ends with the termination character (␣).

- CMD[ { { } <Argument> } ] ␣
- CMD? [ { { } <Argument> } ] ␣

Exception:

- Single-character commands are not followed by a termination character. However, the response to a single-character command is followed by a termination character.

More than one command mnemonic per line is not allowed. Several groups of arguments following a command mnemonic are allowed.

When all arguments are optional and not specified, the command is executed for all possible argument values.

### Sending a Command

The axis identified with "1" is to be moved to position 10.0. The unit depends on the controller (e.g., μm or mm).

1. Send

```
MOV_1_10.0␣
```

→ *Axis 1 moves to position 10.0 (physical unit).*

### Sending the Command with Several Arguments

Two axes are to be moved that are connected to the same controller: The axis with axis identifier "1" is to be moved to position 17.0 and the axis with axis identifier "2" is to be moved to position 2.05. The unit depends on the controller (e.g., μm or mm).

### Information

When a part of a command line cannot be executed, the line is not executed at all.

1. Send

```
MOV_1_17.3_2_2.05␣
```

→ *Axis 1 moves to position 17.0 (physical unit), axis 2 moves to position 2.05 (physical unit).*

### Sending commands without arguments

The position of all axes is to be queried.

1. Send

```
POS?␣
```

→ *Outputs the position of all axes. The response syntax is as follows:*

```
{ { <Argument> [ { _ <Argument> } ] "=" ] <Wert> _ ␣ }  
[ <Argument> [ { _ <Argument> } ] "=" ] <Wert> ␣ (for the last line)
```

## 8.8.2 Variables

The electronics support variables for more flexible programming. While global variables are always available, local variables are only valid for a specified macro. Typically, variables are used when working with macros.

Variables are in volatile memory (RAM) only. The variable values are the STRING data type. The following conventions apply to variable names:

- Variable names may not contain special characters (especially not "\$").
- The maximum number of characters is 8.
- Names of global variables can consist of characters A to Z and 0 to 9. They must begin with a letter.
- Names of local variables must not contain alphabetic characters. Possible characters are 0 to 9.
- The variable name can also be specified via the value of another variable.

If the value of a variable is to be used, the notation must be as follows:

- The variable name must be preceded by the dollar sign (\$).
- Variable names consisting of multiple characters must be put in braces.

If the variable name consists of a single character, no braces are necessary.

Note that when braces are omitted for multicharacter variable names, the first character after the "\$" is interpreted as the variable name.

## 8.9 Adapting Settings

The properties of the C-884.xDC and the mechanics connected are stored in the C-884.xDC as parameter values.

The parameters can be divided into the following categories:

- Protected parameters whose default settings cannot be changed
- Parameters that can be set by the user to adapt to the application

Write permission for the parameters is determined by command levels.

Each parameter is in the C-884.xDC's volatile and nonvolatile memory. The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the C-884.xDC. The values in the volatile memory determine the current behavior of the system.

The designation "Active Values" is used for the parameter values in the volatile memory and "Startup Values" is used for the parameter values in the nonvolatile memory in the PC software from PI.

Use the [HPA?](#) command to query the parameters available in the C-884.xDC.

### NOTICE



#### Unsuitable parameter settings!

The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the C-884.xDC and take effect immediately. Unsuitable parameter settings can cause damage to the mechanics.

- ▶ Change parameter values only after careful consideration.
- ▶ Save the current parameter values to the PC before you make changes in the nonvolatile memory.

### 8.9.1 Parameter Commands

The following general commands are available for changing parameters:

Command	Function
<a href="#">CCL</a>	Change to another command level
<a href="#">CCL?</a>	Query active command level
<a href="#">HPA?</a>	Query available controller parameters
<a href="#">RPA</a>	Copy a parameter value from the nonvolatile to the volatile memory
<a href="#">SEP</a>	Modify parameter value in nonvolatile memory
<a href="#">SEP?</a>	Query parameter values from the nonvolatile memory
<a href="#">SPA</a>	Modify parameter value in volatile memory
<a href="#">SPA?</a>	Query parameter values from the volatile memory
<a href="#">WPA</a>	Copy a current parameter value from the volatile to the nonvolatile memory. Here it is used as a default value.

The following special commands only change the corresponding parameters in the volatile memory. When necessary, the changed values must be written to the nonvolatile memory with the [WPA](#) command.

Command	Adaptable parameters
ACC	Acceleration in closed-loop operation (0xB)
DEC	Deceleration in closed-loop operation (0xC)
VEL	Velocity in closed-loop operation (0x49)

## 8.9.2 Saving Parameter Values in a Text File

### Overview

The C-884.xDC is configured via parameters, e.g., for adapting to the mechanics. Changing parameter values can cause undesirable results.

- Create a backup copy on the PC before changing the parameter settings of the C-884.xDC. You can then restore the original settings at any time.
- Create an additional backup copy with a new file name each time after optimizing the parameter values or adapting the C-884.xDC to specific mechanics.

Parameter values saved in a text file on the PC can be loaded back to the C-884.xDC in PIMikroMove or PITerminal. The **Send file...** button is available for this purpose in the send command window. Before loading into the C-884.xDC, the individual lines of the text files must be converted into command lines that contain the corresponding SPA or SEP commands.

### Requirements

- ✓ You have [established communication between the C-884.xDC and the PC \(p. 57\)](#) with PIMikroMove or PITerminal.

### Saving Parameter Values in a Text File

1. If you are using PIMikroMove, open the window for transmitting commands: Select **Tools > Command entry** in the main window or press **F4** on the keyboard.  
After communication has been established, the main window is opened PITerminal automatically and commands can be sent.
2. Query the parameter values that you want to create a backup.

- If you want to save the parameter values from the volatile memory of the C-884.xDC: Send the [SPA?](#) command.
  - If you want to save the parameter values from the nonvolatile memory of the C-884.xDC: Send the [SEP?](#) command.
3. Click the **Save...** button.
  4. The **Save content of terminal as textfile** window opens.
  5. Save the queried parameter values to a text file on your PC in the **Save content of terminal as textfile** window.

### 8.9.3 Changing Parameter Values

#### Overview

The following procedure is generally recommended for changing parameter values:

1. Change the parameter values in the volatile memory.
2. Check whether the C-884.xDC works correctly with the changed parameter values.  
If so:
  - Write the changed parameter values into the nonvolatile memory.
 If not:
  - Change and check the parameter values in the volatile memory again.

#### Requirements

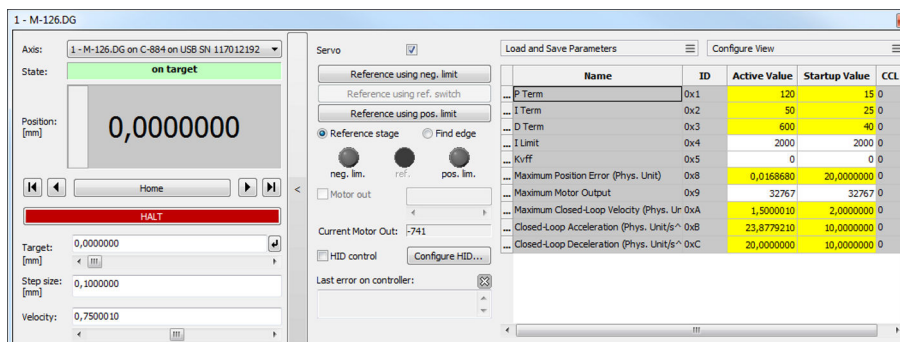
- ✓ If you want to change parameter values in the C-884.xDC's nonvolatile memory: You have saved the C-884.xDC's parameter values in a text file [on the PC \(p. 112\)](#).
- ✓ You have [established communication between the C-884.xDC and the PC \(p. 57\)](#) with PIMikroMove or PITerminal.

#### Information

Write access for the parameters of the C-884.xDC is defined by command levels. After the controller is switched on or rebooted, the active command level is always 0. On command levels > 1, write access is only available to PI service personnel.

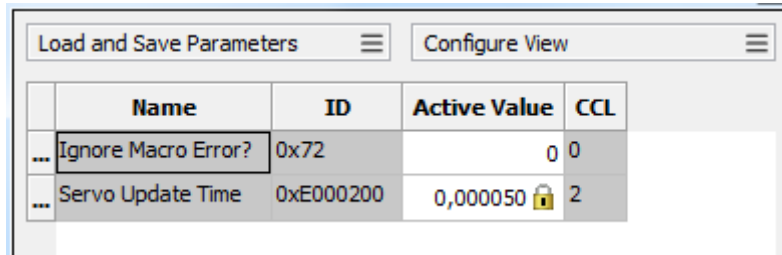
#### Changing Parameter Values

1. If you want to change the axis-related parameters of the C-884.xDC:
    - a) Open the expanded single axis window for the connected positioner in the main window of PIMikroMove by clicking the right mouse button on the corresponding line of the **Axes** tab and selecting **Show Expanded Single Axis Window** in the context menu.
    - b) If the parameter to be modified is not included in the list on the right-hand side of the window, click **Configure View > Select parameters...** and add it to the list. You can also display certain groups of parameters or all axis-related parameters.
- *The list of axis-related parameters is displayed.*



2. If you want to change the system-related parameters of the C-884.xDC:
  - a) Open the window for the system-related parameters of the C-884.xDC in the main window of PIMikroMove by selecting **C-884.xDC > Show system parameters** in the menu.

→ The list of system-related parameters is displayed.



	Name	ID	Active Value	CCL
...	Ignore Macro Error?	0x72	0 0	
...	Servo Update Time	0xE000200	0,000050	2

3. If you want to change the parameter values in the C-884.xDC's **volatile memory**, you have the following options:
  - a) Type the new parameter values into the corresponding input field in the **Active Value** column and press the enter key on the PC's keyboard or click the mouse button outside of the input field.
 

→ The modified parameter value is transferred to the C-884.xDC's volatile memory.
  - b) Click **Load and Save Parameters -> Load all startup parameters of the axis / system from controller**.
 

→ The values of all axis-related / system-related parameters are loaded from the C-884.xDC's nonvolatile memory.
  - c) Click **Load and Save Parameters > Load parameters from stage database...** in the expanded single axis window.

You can use **Load and Save Parameters > Reload parameters from stage database...** to reload the currently loaded parameter set.

→ A selected parameter set for the axis is loaded from the positioner database.

4. If you want to change parameter values in the C-884.xDC's **nonvolatile memory**, you have the following options:
  - a) Type the new parameter value into the corresponding input field in the list's **Startup Value** column and press the enter key on the PC's keyboard or click the mouse button outside of the input field.
 

→ The changed parameter value is transferred to the C-884.xDC's volatile memory.
  - b) Click **Load and Save Parameters -> Save all currently active axis / system parameters as startup parameters to controller**.

You can skip parameters that do not have write access on the current command level.

→ The values of all axis-related / system-related parameters are written from the C-884.xDC's volatile memory to the nonvolatile memory.

If a parameter value in the volatile memory (**Active Value** column) is different from the parameter value in the nonvolatile memory (**Startup Value** column), the line in the list is highlighted in color.

## 8.9.4 Creating or Changing Parameter Sets for Positioners

### Overview

You can create and edit new parameter records in the PIStages3 database. This can be required in the following cases, for example:

- You want to operate a positioner with different servo control parameter settings than the one from the default parameter set.
- You want to adapt the soft limits of the positioner to your application.

- You have a custom positioner.

Possibilities for creating and editing parameter sets in the PISTAGES3.DB database:

- You can create a new positioner type easily by changing an existing parameter set in PIMikroMove and saving it under a new name.
- You can open and edit the positioner database directly with the PISTages3Editor.

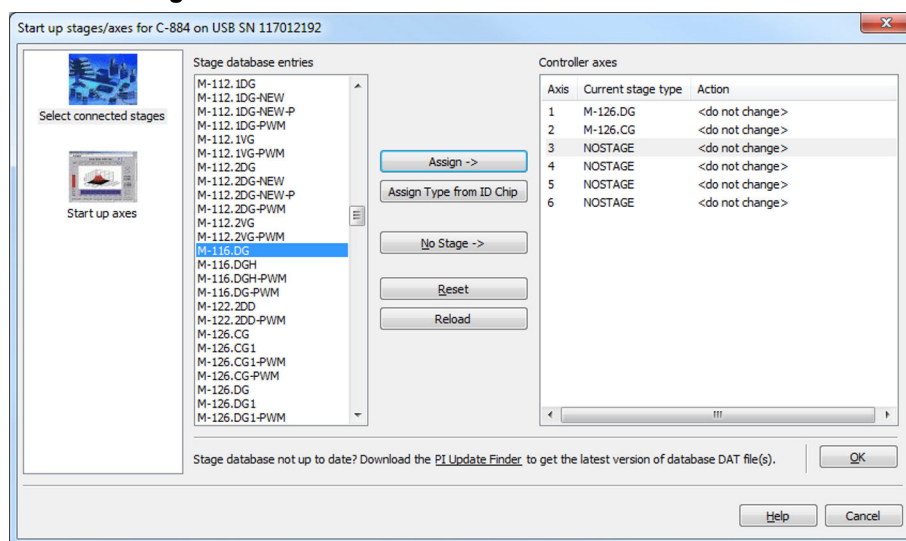
The following describes how to use PIMikroMove to create and change a parameter set for a positioner.

### Requirements

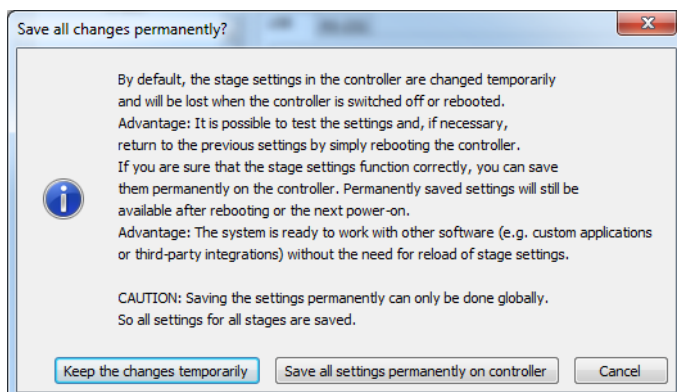
- ✓ You have installed the latest version of the PISTAGES3.DB database onto your PC.
- ✓ If PI provided a custom positioner database for your positioner, the dataset was imported into PISTages3.
- ✓ PIMikroMove has established communication between the C-884.xDC and the PC.

### Creating a Parameter Set for Positioners

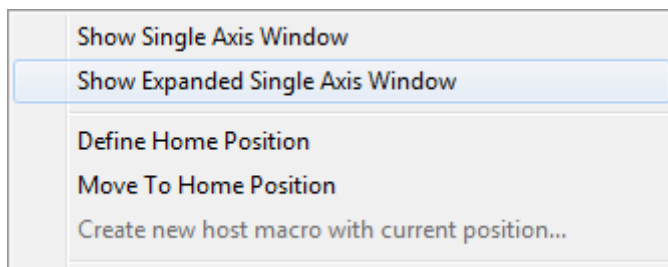
1. Select the **C-884.xDC > Select connected stages...** menu item in the main window of PIMikroMove.  
→ *The **Start up stages/axes for C-884.xDC** window opens and the **Select connected stages** step is active.*
2. Select an appropriate type of positioner during the **Select connected stages** step:
  - a) Click **Assign Type from ID Chip**.
  - or
  - a) Highlight the positioner in the **Stage database entries** list.
  - b) Click **Assign**.



- c) Confirm the selection with **OK**.
- *The **Save all changes permanently?** dialog is opened.*



3. Click **Keep the changes temporarily** in the **Save all changes permanently** dialog to load the parameter settings into the volatile memory of the C-884.xDC.  
 → *The **Start up stages/axes** window changes to the **Start up axes** step.*
4. Click **Close** to close the **Start up stages/axes** window.
5. Open the expanded single axis window for the selected positioner in the main window of PIMikroMove by clicking the right mouse button on the corresponding line of the **Axes** tab and selecting **Show Expanded Single Axis Window** in the context menu.



6. Enter new values for the parameters to be changed:

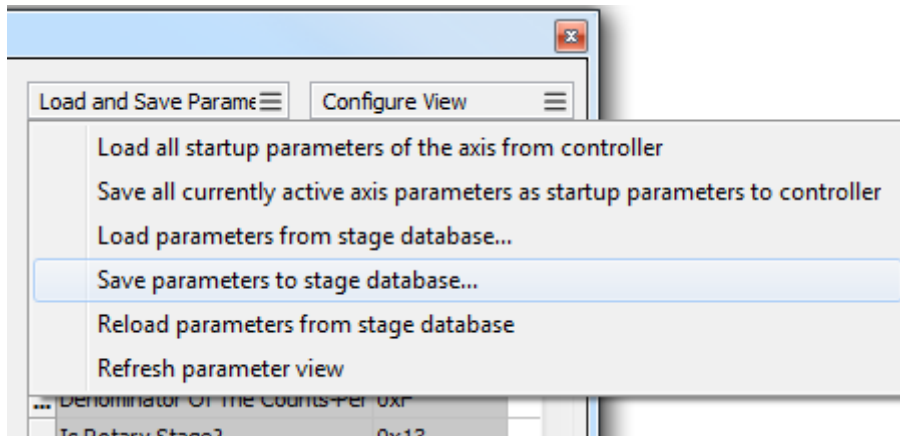
Load and Save Parameters		Configure View		
	Name	ID	Active Value	CCL
...	P-Term	0x1	5	0
...	I-Term	0x2	1000	0
...	D-Term	0x3	0	0
...	I-Limit	0x4	2000	0
...	Kvff	0x5	0	0
...	Maximum Position Error (Phys. Unit)	0x8	16,145833330	0
...	Maximum Motor Output	0x9	2000	0
...	Numerator Of The Counts-Per-Physical-Unit Factor	0xE	11520000	0
...	Denominator Of The Counts-Per-Physical-Unit Factor	0xF	1	0
...	Is Rotary Stage?	0x13	1	0

- a) If the parameter to be modified is not included in the list on the right-hand side of the window, click **Configure view > Select parameters...** and add it to the list. You can also display certain groups of parameters or all axes-related parameters.
- b) Type the new parameter value into the corresponding input field in the **Active Value** column of the list.
- c) Press the Enter key on the PC keyboard or click outside the input field with the mouse to transfer the parameter value to the volatile memory of the controller. Note: If a parameter value in the volatile memory (**Active Value** column) is different to the



parameter value in the nonvolatile memory (**Startup Value** column), the line in the list is highlighted in color.

7. Click **Load and Save Parameters > Save parameters to stage database....**

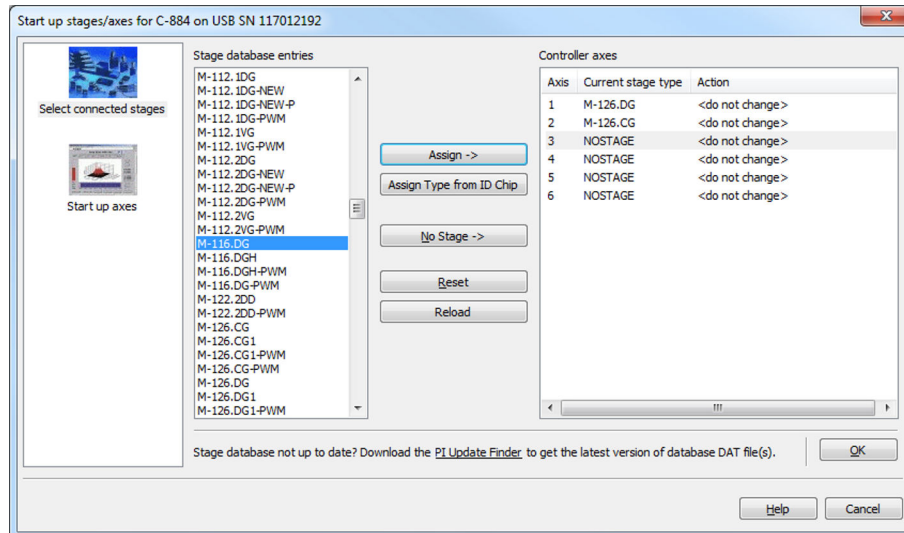


8. The **Save Parameters as User Stage Type** dialog opens.
  9. Save the changed parameter values as new positioner type in **Dialog Save Parameters as User Stage Type**:
    - a) Leave the entry in the **Parameters of axis** field unchanged.
    - b) Enter the name for the new positioner type into the **Save as** field.
    - c) Click **OK**.
- *The new positioner type was saved in the positioner database. The displayed positioner type was updated in PIMikroMove. The new positioner type is also available immediately for selection in the **Select connected stages** step.*

### Changing a Positioner's Parameter Set

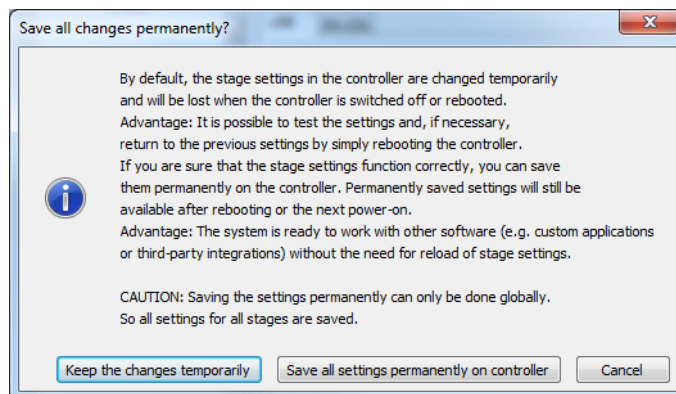
1. Select the **C-884.xDC > Select connected stages...** menu item in the main window of PIMikroMove.
 

→ *The **Start up stages/axes for C-884.xDC** window opens and the **Select connected stages** step is active.*
2. Select a positioner type that you created during the **Select connected stages** as described above:
  - a) Click **Assign Type from ID Chip**.
  - or
  - a) Highlight the positioner in the **Stage database entries** list.
  - b) Click **Assign**.



c) Confirm the selection with **OK**.

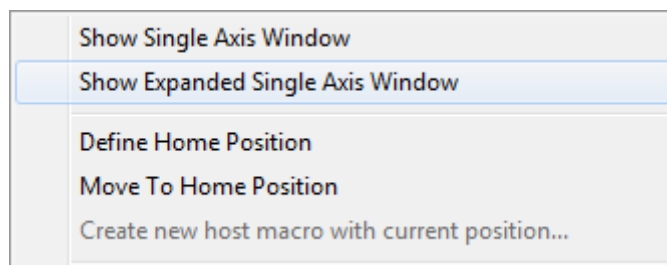
→ *The **Save all changes permanently?** dialog is opened.*



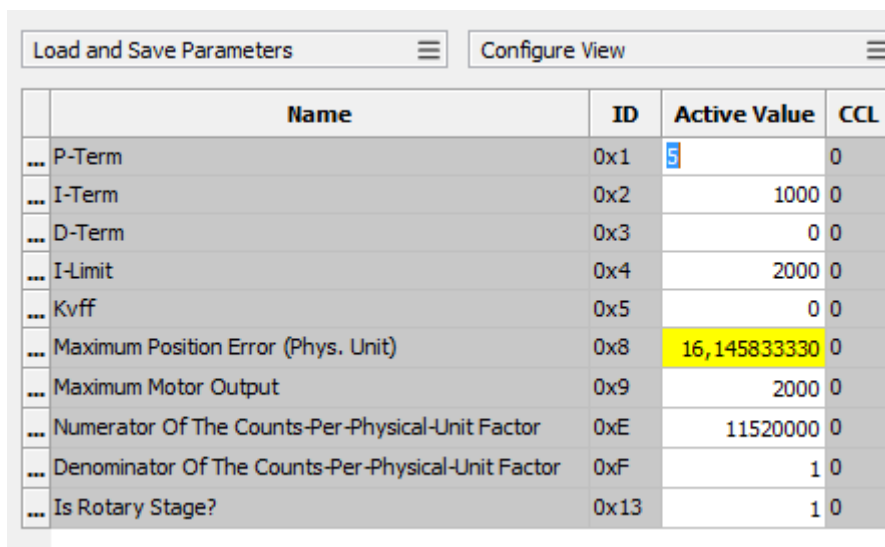
3. Click **Keep the changes temporarily** in the **Save all changes permanently** dialog to load the parameter settings into the volatile memory of the C-884.xDC.

→ *The **Start up stages/axes** window changes to the **Start up axes** step.*

4. Click **Close** to close the **Start up stages/axes** window.
5. Open the expanded single axis window for the selected positioner in the main window of PIMikroMove by clicking the right mouse button on the corresponding line of the **Axes** tab and selecting **Show Expanded Single Axis Window** in the context menu.

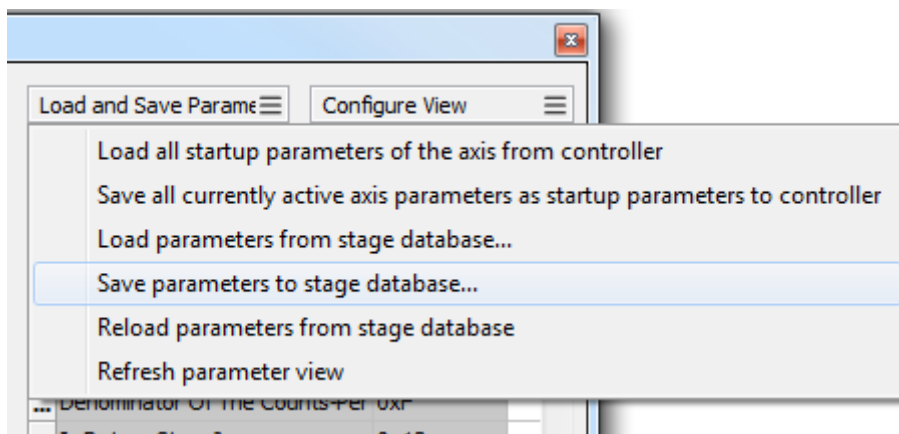


6. Enter new values for the parameters to be changed:



	Name	ID	Active Value	CCL
...	P-Term	0x1	5	0
...	I-Term	0x2	1000	0
...	D-Term	0x3	0	0
...	I-Limit	0x4	2000	0
...	Kvff	0x5	0	0
...	Maximum Position Error (Phys. Unit)	0x8	16,145833330	0
...	Maximum Motor Output	0x9	2000	0
...	Numerator Of The Counts-Per-Physical-Unit Factor	0xE	11520000	0
...	Denominator Of The Counts-Per-Physical-Unit Factor	0xF	1	0
...	Is Rotary Stage?	0x13	1	0

- If the parameter to be modified is not included in the list on the right-hand side of the window, click **Configure view > Select parameters...** and add it to the list. You can also display certain groups of parameters or all axes-related parameters.
  - Type the new parameter value into the corresponding input field in the **Active Value** column of the list.
  - Press the Enter key on the PC keyboard or click outside the input field with the mouse to transfer the parameter value to the volatile memory of the controller. Note: If a parameter value in the volatile memory (**Active Value** column) is different to the parameter value in the nonvolatile memory (**Startup Value** column), the line in the list is highlighted in color.
7. Click **Load and Save Parameters > Save parameters to stage database...**



- The **Save Parameters as User Stage Type** dialog opens.
  - Save the modified parameter values of the positioner type in the **Save Parameters as User Stage Type** dialog:
    - Leave the entry in the **Parameters of axis** field unchanged.
    - Leave the entry in the **Save as** field unchanged.
    - Click **OK**.
    - Click **Change settings** in the **Stage type already defined** dialog. The **Save Parameters as User Stage Type** dialog closes automatically after a short time.
- The parameter values of the positioner type were updated in the positioner database and in the main window of PIMikroMove.

## 9 Command Reference

#		Page
#24	Stop All Axes	126
#4	Request Status Register	124
#5	Request Motion Status	125
#7	Request Controller Ready Status	125
#8	Query If Macro Is Running	126
*		Page
*IDN?	Get Device Information	126
A		Page
ACC	Set Closed-Loop Acceleration	127
ACC?	Get Closed-Loop Acceleration	127
ADD	Add And Save To Variable	127
B		Page
BRA	Set Brake Activation State	128
BRA?	Get Brake Activation State	128
C		Page
CCL	Set Command Level	129
CCL?	Get Command Level	129
CPY	Copy Into Variable	129
CST?	Get Assignment Of Stages To Axes	130
CSV?	Get Current Syntax Version	130
CTO	Set Configuration Of Trigger Output	131
CTO?	Get Configuration Of Trigger Output	133
D		Page
DEC	Set Closed-Loop Deceleration	134
DEC?	Get Closed-Loop Deceleration	134
DEL	Delay The Command Interpreter	134
DFH	Define Home Position	135
DFH?	Get Home Position Definition	135
DIO	Set Digital Output Lines	136
DIO?	Get Digital Input Lines	136
DRC	Set Data Recorder Configuration	137
DRC?	Get Data Recorder Configuration	138
DRL?	Get Number Of Recorded Points	138
DRR?	Get Recorded Data Values	138
DRT	Set Data Recorder Trigger Source	139

<b>DRT?</b>	Get Data Recorder Trigger Source	140
<b>E</b>		<b>Page</b>
<b>ERR?</b>	Get Error Number	140
<b>F</b>		<b>Page</b>
<b>FED</b>	Find Edge	141
<b>FRF</b>	Fast Reference Move To Reference Switch	142
<b>FRF?</b>	Get Referencing Result	142
<b>G</b>		<b>Page</b>
<b>GOH</b>	Go To Home Position	143
<b>H</b>		<b>Page</b>
<b>HDR?</b>	Get All Data Recorder Options	143
<b>HDT</b>	Set HID Default Lookup Table	144
<b>HDT?</b>	Get HID Default Lookup Table	144
<b>HIA</b>	Configure Control Done By HID Axis	145
<b>HIA?</b>	Get Configuration Of Control Done By HID Axis	146
<b>HIB?</b>	Get State Of HID Button	146
<b>HIE?</b>	Get Deflection Of HID Axis	147
<b>HIL</b>	Set State of HID LED	147
<b>HIL?</b>	Get State Of HID LED	147
<b>HIN</b>	Set Activation State For HID Control	148
<b>HIN?</b>	Get Activation State Of HID Control	148
<b>HIS</b>	Configure HI Device	149
<b>HIS?</b>	Get Configuration Of HI Device	150
<b>HIT</b>	Fill HID Lookup Table	151
<b>HIT?</b>	Get HID Lookup Table Values	151
<b>HLP?</b>	Get List Of Available Commands	151
<b>HLT</b>	Halt Motion Smoothly	152
<b>HPA?</b>	Get List Of Available Parameters	152
<b>HPV?</b>	Get Parameter Value Description	153
<b>I</b>		<b>Page</b>
<b>IFC</b>	Set Interface Parameters Temporarily	154
<b>IFC?</b>	Get Current Interface Parameters	155
<b>IFS</b>	Set Interface Parameters As Default Values	156
<b>IFS?</b>	Get Interface Parameters As Default Values	157
<b>J</b>		<b>Page</b>
<b>JRC</b>	Jump Relatively Depending On Condition	158
<b>L</b>		<b>Page</b>
<b>LIM?</b>	Indicate Limit Switches	158

M		Page
MAC	Call Macro Function	159
MAC BEG	Call Macro Function: BEG	159
MAC DEF	Call Macro Function: DEF	159
MAC DEF?	Call Macro Function: DEF?	159
MAC DEL	Call Macro Function: DEL	160
MAC END	Call Macro Function: END	160
MAC ERR?	Call Macro Function: ERR?	160
MAC NSTART	Call Macro Function: NSTART	160
MAC START	Call Macro Function: START	161
MAC?	List Macros	161
MAN?	Get Help String For Command	161
MAT	Calculate And Save To Variable	162
MEX	Stop Macro Execution Due To Condition	163
MOV	Set Target Position	164
MOV?	Get Target Position	164
MVE	Set Target Position For Vectorial Move	165
MVR	Set Target Relative To Current Position	166
O		Page
ONT?	Get On-Target State	166
P		Page
POS	Set Real Position	167
POS?	Get Real Position	167
R		Page
RBT	Reboot System	167
RMC?	List Running Macros	168
RON	Set Reference Mode	168
RON?	Get Reference Mode	169
RPA	Reset Volatile Memory Parameters	169
RTR	Set Record Table Rate	170
RTR?	Get Record Table Rate	170
S		Page
SAI	Set Current Axis Identifiers	170
SAI?	Get List Of Current Axis Identifiers	171
SEP	Set Nonvolatile Memory Parameters	171
SEP?	Get Nonvolatile Memory Parameters	172
SMO	Set Open-Loop Control Value	172
SMO?	Get Control Value	173

SPA	Set Volatile Memory Parameters	173
SPA?	Get Volatile Memory Parameters	173
SRG?	Query Status Register Value	174
STE	Start Step And Response Measurement	175
STP	Stop All Axes	175
SVO	Set Servo Mode	175
SVO?	Get Servo Mode	176
<b>T</b>		<b>Page</b>
TAC?	Tell Analog Channels	176
TAV?	Get Analog Input Voltage	176
TCV?	Get Commanded Closed-Loop Velocity	177
TGA	Append Value To Trajectory	177
TGC	Clear All Values In Trajectory	177
TGF	Finalize Trajectory	178
TGL?	Get Number Of Values In Trajectory	178
TGS	Start Trajectory	178
TGT	Set Trajectory Timing	179
TGT?	Get Trajectory Timing	179
TIM	Set Timer Value	179
TIM?	Get Timer Value	179
TIO?	Tell Digital I/O Lines	179
TMN?	Get Minimum Commandable Position	180
TMX?	Get Maximum Commandable Position	180
TNR?	Get Number Of Record Tables	180
TRO	Set Trigger Output State	181
TRO?	Get Trigger Output State	181
TRS?	Indicate Reference Switch	181
TVI?	Tell Valid Character Set For Axis Identifiers	182
<b>V</b>		<b>Page</b>
VAR	Set Variable Value	182
VAR?	Get Variable Values	182
VEL	Set Closed-Loop Velocity	183
VEL?	Get Closed-Loop Velocity	183
VER?	Get Versions Of Firmware And Drivers	183
<b>W</b>		<b>Page</b>
WAC	Wait For Condition	184
WPA	Save Parameters To Nonvolatile Memory	185

#4 Request Status Register

Used in: (29), On-Target State (38)

Description: Queries system status information.  
This command is identical in function to [SRG? \(p. 174\)](#) except that only one character is sent via the interface. Therefore, #4 can also be used when the controller is doing time-consuming tasks.

Format: #4  
#4 corresponds to the EOT control character in ISO/IEC 6429.

Response: <StatusRegister>  
<StatusRegister>     System status information (HEX)

<StatusRegister> is bit-mapped. Includes per axis:

Bit	Description
15	On-target state
14	Referencing is being done
13	In motion
12	Servo mode activated
11	
10	Sensor signal valid
9	Reference edge found
8	Error
7	Digital input 4 active
6	Digital input 3 active
5	Digital input 2 active
4	Digital input 1 active
3	Sensor is referenced
2	Positive limit switch active
1	Reference switch active
0	Negative limit switch active

Deactivated axes are not included in the response.



#5 Request Motion Status

Used in: *Triggering Motion (36)*

Description: Queries the motion status of the axes.

Format: #5

#5 corresponds to the ENQ control character in ISO/IEC 6429.

Response: <MotionStatus>

<MotionStatus> motion status (HEX)

<MotionStatus> is bit-mapped. Each axis corresponds to one bit and the bit value corresponds to the axis number, e.g.,

Response	Description
1	Axis 1 is moving
2	Axis 2 is moving
5	Axis 3 and 1 are moving

#7 Request Controller Ready Status

Used in: *Commandable Items (25)*

Description: Queries the controller's ready state.

The controller is ready as soon as a new command can be executed.

Format: #7

#7 corresponds to the  $\text{P}_{\text{CL}}$  control character in ISO/IEC 6429.

Response: <ReadyStatus>

<ReadyStatus> ready state (HEX)

Possible answers:

Response from con- troller	Character in ISO/IEC 8859-1	Description
0xB1	±	Controller is ready
0xB0	°	Controller is not ready

Troubleshooting: The response characters may be displayed differently in other character sets.

#8 Query If Macro Is Running

Used in: (101)

Description: Tests if a macro is running on the controller.

Format: #8

#8 corresponds to the BS control character in ISO/IEC 6429.

Response: <MacroRunning>

<MacroRunning> Macro is running (UINT)

<MacroRunning>	Description
1	A macro is running.
0	No macro is running.

#24 Stop All Axes

Used in: (103), Triggering Motion (36)

Description: Stops all axes abruptly.

Stops all motion started by motion commands (e.g., [MOV \(p. 164\)](#), [MVR \(p. 166\)](#), [MVE \(p. 165\)](#), [STE \(p. 175\)](#), [SMO \(p. 172\)](#)), referencing commands (FNL, FPL, [FRF \(p. 142\)](#)), and macros.

Also stops the macro.

Sets the error code to 10.

After the axes are stopped, their target positions are set to their current positions.

This command is identical in function to [STP \(p. 175\)](#) except only one character is sent via the interface. Therefore, #24 can also be used when the controller is doing time-consuming tasks.

Format: #24

#24 corresponds to the CAN control character in ISO/IEC 6429.

\*IDN? Get Device Identification

Used in: Commandable Items (25)

Description: Queries the device identification string.

Format: \*IDN?

Response: <DeviceInformation> LF

<DeviceInformation> Device identification string with controller name, serial number, and firmware version. (STRING)

## ACC Set Closed-Loop Acceleration

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	Sets acceleration of specified axes.  Changes the value of parameter <a href="#">0xB (p. 196)</a> in the volatile memory. The maximum value that can be set with ACC and is specified by <a href="#">0x4A (p. 206)</a> .  The acceleration set with ACC is only taken into account when the respective axis is in closed-loop operation (servo mode on).	
Format:	ACC{_ <a href="#">AxisID</a> _ <a href="#">Acceleration</a> }	
Arguments:	<AxisID>	Axis ID
	<Acceleration>	Acceleration in physical units / s <sup>2</sup> (FLOAT)
Troubleshooting:	Illegal axis identifier	

## ACC? Get Closed-Loop Acceleration

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	Queries the acceleration value set with <a href="#">ACC (p. 127)</a> .	
Format:	ACC?[{_ <a href="#">AxisID</a> >}]	
Arguments:	<AxisID>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{<AxisID>=<Acceleration>_LF}	
	<AxisID>	Axis ID
	<Acceleration>	Acceleration in physical units / s <sup>2</sup> (FLOAT)
Troubleshooting:	Illegal axis identifier	

## ADD Add and Save To Variable

*Used in: Running the Macros (101)*

Description:	Adds two summands (as variable or specified directly) and saves the sum as variable.	
Format:	ADD_ <a href="#">Variable</a> _ <a href="#">Summand1</a> _ <a href="#">Summand2</a>	
Arguments:	<Variable>	Name of the variable where the sum is to be saved (STRING)
	<Summand1>	First summand (FLOAT)
	<Summand2>	Second summand (FLOAT)

**BRA Set Brake Activation State**

Description:	Activates or deactivates the brake for the specified axes. The brake can only be activated when parameter <a href="#">0x1A (p. 201)</a> has the value "1" and servo mode is switched off.	
Format:	<b>BRA_{&lt;AxisID&gt;_&lt;BrakeState&gt;}</b>	
Arguments:	<AxisID>	Axis ID
	<BrakeState>	Brake status (BOOL)
Troubleshooting:	Illegal axis identifier Servo mode active ( <a href="#">SVO? (p. 176)</a> responds with the value "1") Selected axis does not have a brake (parameter <a href="#">0x1A (p. 201)</a> has the value "0")	

**BRA? Get Brake Activation State**

Description:	Queries the brake activation status for the specified axes. The value for all axes will be queried if no arguments are specified.	
Format:	<b>BRA?[{_&lt;AxisID&gt;}]</b>	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<BrakeState>_LF}	
	<BrakeState>	Brake activation status (BOOL)
Troubleshooting:	Illegal axis identifier	

CCL Set Command Level

Used in: (24), Parameter Commands (111)

Description:	Changes the active command level.  The command level determines the availability of commands and write access to system parameters.  Command level: 0 - standard; 1 - expert mode; > 1 - PI service mode Command levels > 1 cannot be activated; they are intended for PI service personnel. Contact the customer service department if there are problems with the parameters for command level 2 or <a href="#">higher (p. 10)</a> .  The active command level is always 0 when the electronics are switched on or rebooted.  The <a href="#">HPA? (p. 152)</a> command lists the parameters and includes information on which command level allows write access to them.	
Format:	CCL_<Level>[_<PSWD>]	
Arguments:	<Level>	Controller command level (UINT)
	<PSWD>	Password for changing to the corresponding command level  Password for changing to command level 1: advanced
Troubleshooting:	Wrong password	

CCL? Get Command Level

Used in: (24), Parameter Commands (111)

Description:	Queries the active command level.	
Format:	CCL?	
Response:	<Level> LF	
	<Level>	Command level (UINT)

CPY Copy Into Variable

Used in: Analog Input Signals (90), Digital Input Signals (88), Running the Macros (101)

Description:	Copies the response to a command into a variable.	
Format:	CPY_<Variable>_<CMD?>	
Arguments:	<Variable>	Name of the variable to be copied to
	<CMD?>	Query command that responds with a single value

CST? Get Assignment Of Stages To Axes

Used in: Commandable Items (25)

Description:	Queries the name of the positioner type that is configured for the specified axis.  The positioner name is read from parameter <a href="#">0x3C (p. 204)</a> . If the parameter has the value "NOSTAGE", the axis is deactivated.	
Format:	CST? [{<AxisID>}]	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<String> <sub>LF</sub> }	
	<AxisID>	Axis ID
	<String>	Name of the positioner type (STRING)

CSV? Get Current Syntax Version

Used in: (24)

Description:	Get current GCS syntax version used in the firmware.	
Format:	CSV?	
Response:	<SyntaxVersion>	
	<SyntaxVersion>	GCS syntax version (STRING)

<SyntaxVersion>	Description
1.0	GCS syntax version 1.0
2.0	GCS syntax version 2.0

## CTO Set Configuration Of Trigger Output

*Used in: Digital Output Signals (83), Setting "Hardware Trigger" Mode (87), Setting "In Motion" Trigger Mode (85), Setting "Position Distance" Trigger Mode (84), Setting "Position+Offset" Trigger Mode (86), Setting "Single Position" Trigger Mode (87), Setting Signal Polarity (88), Setting Up "Motion Error" Trigger Mode (85), Setting Up "On Target" Trigger Mode (85)*

**Description:** Configures the trigger output conditions for the specified digital output line.

Available CTO parameters:

<CTO - Pam>	Parameters	Possible values for <Value>	Description
1	TriggerStep	0	Distance
2	Axis	(Axis identifier)	Selects the axis that the trigger output is configured for.
3	TriggerMode	0	PositionDistance A trigger pulse is output each time the axis has traveled the distance set with <b>TriggerStep</b> (<CTOPam> 1).
		2	OnTarget The on-target status of the selected axis is transferred to the selected digital output line.
		5	MotionError The selected digital output line becomes active when a motion error occurs. The line remains active until the error code is reset to 0 (by a query with <a href="#">ERR?</a> (p. 140)).
		6	InMotion The selected digital output line is active as long as the selected axis is in motion.
		7	Position+Offset The first trigger pulse is output when the axis has reached the position specified with <b>TriggerPosition</b> (<CTOPam> 10). The next trigger pulses are output respectively when the axis position equals the sum of the last valid trigger position and the distance specified by <b>TriggerStep</b> (<CTOPam> 1). Trigger output ends when the axis position exceeds the value specified by <b>StopThreshold</b> (<CTOPam> 9).

<CTO - Pam>	Parameters	Possible values for <Value>	Description
			The sign of the <b>TriggerStep</b> value determines which motion direction trigger pulses are to be output for.
		8	<b>SinglePosition</b> The selected digital output line is active when the axis position has reached or exceeded the position specified by <b>TriggerPosition</b> (<CTOPam> 10).
		9	<b>HardwareTrigger</b> Basically corresponds to the <b>Position+Offset</b> trigger mode but is done by the electronics (shorter processing time). Assignment of the axes to the digital output lines can be fixed.
7	Polarity	0	Sets the signal polarity to "active low"
		1	Sets the signal polarity to "active high" (default value)
8	StartThreshold	(Position - value)	Trigger output start position for the following trigger modes: <ul style="list-style-type: none"> <li>■ <b>PositionDistance</b> (Trigger-Mode 0)</li> </ul>
9	StopThreshold	(Position - value)	Trigger output stop position for the following trigger modes: <ul style="list-style-type: none"> <li>■ <b>PositionDistance</b> (Trigger-Mode 0)</li> <li>■ <b>Position+Offset</b> (Trigger-Mode 7)</li> <li>■ <b>HardwareTrigger</b> (Trigger-Mode 9)</li> </ul>
10	TriggerPosition	(Position - value)	Position of the (first) trigger output for the following trigger modes: <ul style="list-style-type: none"> <li>■ <b>Position+Offset</b> (Trigger-Mode 7)</li> <li>■ <b>SinglePosition</b> (Trigger-Mode 8)</li> <li>■ <b>HardwareTrigger</b> (Trigger-Mode 9)</li> </ul>



<CTO - Pam>	Parameters	Possible values for <Value>	Description
11	PulseWidth	(Integer factor)	Factor that determines the pulse width for the Hardware-Trigger mode (TriggerMode 9). Pulse width = 33.3 ns x Pulse-Width

Format:	CTO{_ <b>&lt;TrigOutID&gt;</b> _ <b>&lt;CTOPam&gt;</b> _ <b>&lt;Value&gt;</b> }		
Arguments:	<b>&lt;TrigOutID&gt;</b>	Digital output of the electronics	
	<b>&lt;CTOPam&gt;</b>	CTO parameter ID	
	<b>&lt;Value&gt;</b>	Value that the CTO parameter is set to	

CTO? Get Configuration Of Trigger Output

Used in: Digital Output Signals (83)

Description:	Queries the value that is configured for the specified trigger output line and the specified CTO parameter.		
Format:	CTO?[{<_<TrigOutID>_<CTOPam>}]		
Arguments:	<TrigOutID>	Digital output of the electronics	
	<CTOPam>	CTO parameter ID	
	If no arguments are specified, the response contains the values for all parameters and all output lines.		
Response:	{<TrigOutID>_<CTOPam>=<Value> LF }		
	<TrigOutID>	Digital output of the electronics	
	<CTOPam>	CTO parameter ID	
	<Value>	Value of <CTOPam> that was configured for <TrigOutID>	

## DEC Set Closed-Loop Deceleration

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	Sets the deceleration for the specified axis.  Changes the value of parameter <a href="#">0xC (p. 196)</a> in the volatile memory. The maximum value that can be set with DEC, is specified by parameter <a href="#">0x4B (p. 206)</a> .  The deceleration set with DEC is only taken into account when the respective axis is in closed-loop operation (servo mode on).	
Format:	<b>DEC{_<a href="#">AxisID</a>_<a href="#">Deceleration</a>}</b>	
Arguments:	<a href="#">AxisID</a>	Axis ID
	<a href="#">Deceleration</a>	Deceleration in physical units / s <sup>2</sup> (FLOAT)
Troubleshooting:	Illegal axis identifier	

## DEC? Get Closed-Loop Acceleration

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	Queries the deceleration value set with <a href="#">DEC (p. 134)</a> .	
Format:	<b>DEC? [{<a href="#">AxisID</a>}]</b>	
Arguments:	<a href="#">AxisID</a>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{ <a href="#">AxisID</a> == <a href="#">Acceleration</a> >LF}	
	<a href="#">AxisID</a>	Axis ID
	<a href="#">Acceleration</a>	Acceleration in physical units / s (FLOAT)
Troubleshooting:	Illegal axis identifier	

## DEL Delay The Command Interpreter

*Used in: Running the Macros (101)*

Description:	Delays running the macro for a fixed period of time.	
	DEL can only be used in macros.	
	Note: Do not mix up DEL (delayed) with <a href="#">MAC DEL (p. 160)</a> (deletes macros).	
Format:	<b>DEL_<a href="#">uint</a></b>	
Arguments:	<a href="#">uint</a>	Delay time span in milliseconds [UINT]

# DFH Define Home Position

Description:	Redefines the zero position of the specified axis by setting the position value to zero at the current position.	
	<p>Sets the current position of the axis to zero and saves the position value which was valid when the command was called as offset in the volatile memory. By adding this offset to the response, the output values of the following commands are adapted to the new zero position:</p> <ul style="list-style-type: none"> <li>■ <a href="#">POS? (p. 167)</a></li> <li>■ <a href="#">TMN? (p. 180)</a></li> <li>■ <a href="#">TMX? (p. 180)</a></li> </ul> <p>Does <b>not</b> change the parameter values for the definition of travel range and soft limits.</p> <p>The offset is reset to zero in the following cases:</p> <ul style="list-style-type: none"> <li>■ When switching on and rebooting the C-884.xDC: For all axes</li> <li>■ During referencing: For the affected axis</li> </ul>	
Format:	DFH[[_<AxisID>]]	
Arguments:	<AxisID>	Axis ID
	The zero positions for all axes will be newly defined if no arguments are specified.	
Troubleshooting:	Illegal axis identifier	

# DFH? Get Home Position Definition

Description:	Queries the position value that is configured for the specified axis as offset for shifting the zero position.	
	<p>The offset for shifting the zero position is in the volatile memory. The offset is reset to zero in the following cases:</p> <ul style="list-style-type: none"> <li>■ When switching on or rebooting the Electronics: For all axes</li> <li>■ During referencing: For the affected axis</li> </ul>	
Format:	DFH?[_<AxisID>]]	
Arguments:	<AxisID>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{<AxisID>=<PositionOffset> LF }	
	<AxisID>	Axis ID
	<PositionOffset>	Axis position that was valid when executing <a href="#">DFH (p. 135)</a>
Troubleshooting:	Illegal axis identifier	

DIO Set Digital Output Lines

Used in: Digital Output Signals (83)

Description:	Switches a digital output line to the specified state.  All digital output lines with a bit-mapped hexadecimal number for <OutputOn> can be switched with <DIOID> = 0. Use <a href="#">TIO? (p. 179)</a> to query the number of available digital I/O lines.		
Format:	<b>DIO{&lt;_&gt;&lt;DIOID&gt;&lt;_&gt;&lt;OutputOn&gt;}</b>		
Arguments:	<DIOID>	Digital output of the electronics	
	<OutputOn>	Status of the digital output line (BOOL)	

DIO? Get Digital Input Lines

Used in: Digital Input Signals (88)

Description:	Queries the status of a digital input line.  Use <a href="#">TIO? (p. 179)</a> to query the number of available digital I/O lines.		
Format:	<b>DIO?[{_&lt;DIOID&gt;}]</b>		
Arguments:	<DIOID>	Digital input line of the electronics	
	The state of all digital input lines is queried and output as bit-mapped hexadecimal number if no arguments are specified.		
Response:	{<DIOID>=<InputOn>_LF }		
	<DIOID>	Digital input line of the electronics	
	<InputOn>	Digital input line state (HEX)	

DRC Set Data Recorder Configuration

Used in: Analog Input Signals (90), Commandable Items (25), Determining the data to be recorded (80)

Description:	<p>Determines the data source to be used and the data type to be recorded for a data recorder table.</p> <p>The data type to be recorded is specified via a record option. The record options depend on the controller and can be queried with <a href="#">HDR? (p. 143)</a>.</p>	
Format:	<b>DRC{_ &lt;RecTableID&gt;_ &lt;Source&gt;_ &lt;RecOption&gt;}</b>	
Arguments:	<RecTableID>	Data recorder table
	<Source>	Data source ID
	<RecOption>	Record option, data type to be recorded
Record options	Data source: all	
	0	Nothing is Recorded
	Data source: axis	
	1	Commanded Position of Axis
	2	Actual Position of Axis
	3	Position Error of Axis
	44	Timestamp ( <a href="#">TIM? (p. 179)</a> )
	70	Commanded Velocity of Axis
	71	Commanded Acceleration of Axis
	73	Motor Output of Axis
	74	Kp of Axis
	75	Ki of Axis
	76	Kd of Axis
	80	Signal Status Register of Axis
	86	Number of Trajectory Points in Buffer
	87	Interpolation Data
	91	Motor Current
	Data source: input channel (1 - 4)	
	81	Analog Input

## DRC? Get Data Recorder Configuration

*Used in: Determining the data to be recorded (80)*

Description:	Queries the settings for the data to be recorded.	
Format:	<b>DRC?[{&lt;RecTableID&gt;}]</b>	
Arguments:	<RecTableID>	Data recorder table of the controller
	If no arguments are specified, the settings for all data recorder tables will be queried.	
Response:	{<RecTableID>=<Source>_<RecOption>_LF}	
	<RecTableID>	Data recorder table
	<Source>	Data source ID, e.g., controller axis or channel
	<RecOption>	Data type to be recorded (record option, see <a href="#">DRC (p. 137)</a> )

## DRL? Get Number Of Recorded Points

*Used in: Reading data out (81)*

Description:	Queries the number of points stored by the last recording.	
Format:	<b>DRL?[{&lt;RecTableID&gt;}]</b>	
Arguments:	<RecTableID>	Data recorder table
Response:	{<RecTableID>=<uint>_LF}	
	<RecTableID>	Data recorder table
	<uint>	Number of saved points (UINT)

## DRR? Get Recorded Data Values

*Used in: Reading data out (81)*

Description:	Gets the last recorded data.	
	Can be used during recording.	
	Querying can take some time depending on the number of points to be read.	
Format:	<b>DRR?[_&lt;StartPoint&gt; &lt;NumberOfPoints&gt; [{&lt;RecTableID&gt;}]]</b>	
Arguments:	<StartPoint>	First point to be read (UINT >0)
	<NumberOfPoints>	Number of points to be read per table
	<RecTableID>	Data recorder table
	If <RecTableID> is not specified, all data from all tables is read whose recording option is different to zero.	
	If no arguments are specified, the data is read from all recorder tables with recording option not equal to zero.	
Response:	(Data in GCS array format)	

DRT Set Data Recorder Trigger Source

Used in: *Setting the trigger for recording (81), Starting the Recording (81)*

Description:	Defines a trigger option for the specified data recorder table.  Irrespective of the trigger option set, data recording is always triggered when step response measuring is done with <a href="#">STE (p. 175)</a> . Trigger options depend on the controller and can be queried with <a href="#">HDR? (p. 143)</a> .		
Format:	DRT_<RecTableID>_<TriggerSource>_<Value>		
Arguments:	<RecTableID>	Data recorder table ID	
		<RecTableID> = 0 defines the trigger option for all available data recorder tables.	
	<TriggerSource>	ID of the trigger option	
	<Value>	Depends on the trigger option	

Trigger options

0	<b>Default Setting</b> Data recording is triggered with <a href="#">STE (p. 175)</a> . <Value> is a dummy.
1	<b>Any Command Changing Position</b> Data recording is triggered by commands that influence the target position (e.g., <a href="#">MVR (p. 166)</a> , <a href="#">MOV (p. 164)</a> , <a href="#">MVE (p. 165)</a> ). <Value> is a dummy.
2	<b>Next Command</b> Resets the trigger afterwards. <Value> is a dummy.
3	<b>External Trigger</b> Data recording is triggered with a digital output line. <Value> is the ID of the digital output line.
6	<b>Any Command Changing Position, Reset Trigger After Execution</b> Corresponds to <TriggerSource> = 1, resets the trigger afterwards. <Value> is a dummy.



DRT? Get Data Recorder Trigger Source

*Used in: Setting the trigger for recording (81)*

Description:	Queries the trigger option for the data recorder tables.		
Format:	DRT?[{<RecTableID>}]		
Arguments:	<RecTableID> data recorder table		
	If no arguments are specified, the trigger option for all data recorder tables will be queried.		
Response:	{<RecTableID>=<TriggerSource>,<Value> LF }		
	<RecTableID>	Data recorder table ID <RecTableID> = 0 queries the trigger option for all available data recorder tables.	
	<TriggerSource>	ID of the trigger option (see <a href="#">DRT (p. 139)</a> )	
	<Value>	Depends on the trigger option	

ERR? Get Error Number

*Used in: Restoring the C-884.xDC's Operational Readiness (77)*

Description:	Queries the error code of the last error that occurred and resets the error code to 0.		
	Only the last error is buffered; therefore, in the case of a problem, ERR? should be called after each command.		
Format:	ERR?		
Response:	<ErrorNumber>		
	<ErrorNumber>	Code for the last error (INT)	
Troubleshooting:	Communication breakdown		



# FED Find Edge

*Used in: Digital Input Signals (88), Triggering motion in closed-loop operation (36)*

Description:	<p>Moves an axis to a specified signal edge.</p> <p>FED does not set a specific position value at the selected edge (in contrast to FNL, FPL, and <a href="#">FRF (p. 142)</a>); i.e., the axis is not referenced after using FED.</p> <p>Servo mode must be switched on for the specified axis (axes) (closed-loop operation). HID control may not be activated for the corresponding axis.</p> <p>If multiple axes are specified in the command, they are moved synchronously.</p> <p>The following settings influence what type of motion is possible with FED:</p> <ul style="list-style-type: none"> <li>■ Is there a reference switch (parameter <a href="#">0x14 (p. 198)</a>)?</li> <li>■ Are there limit switches (parameter <a href="#">0x32 (p. 203)</a>)?</li> <li>■ If the reference switch outputs an index pulse: How is the move to the index pulse to be done (parameters <a href="#">0x70 (p. 212)</a>, <a href="#">0x78 (p. 214)</a>, <a href="#">0x79 (p. 215)</a>)?</li> </ul> <p>You can use the digital input lines instead of the switches as source of the switch signals for FED.</p> <p>FED can be used to measure the physical travel range of a new mechanics and therefore determine the values for the corresponding parameters:</p> <ul style="list-style-type: none"> <li>■ Distance from the negative to the positive limit switch</li> <li>■ Gap between the negative limit switch and the reference switch (parameter ID <a href="#">0x17 (p. 200)</a>)</li> <li>■ Gap between the reference switch and the positive limit switch (parameter ID <a href="#">0x2F (p. 201)</a>).</li> </ul>						
Format:	FED{_ <a href="#">AxisID</a> >_ <a href="#">EdgeID</a> >_ <a href="#">Param</a> >}						
Arguments:	<table> <tr> <td>&lt;AxisID&gt;</td><td>Axis ID</td></tr> <tr> <td>&lt;EdgeID&gt;</td><td>Edge type, which the axis is to move to</td></tr> <tr> <td>&lt;Param&gt;</td><td>Depends on the selected edge and determines it more precisely</td></tr> </table> <p>If multiple axes are specified in the command, they are moved synchronously.</p>	<AxisID>	Axis ID	<EdgeID>	Edge type, which the axis is to move to	<Param>	Depends on the selected edge and determines it more precisely
<AxisID>	Axis ID						
<EdgeID>	Edge type, which the axis is to move to						
<Param>	Depends on the selected edge and determines it more precisely						
Troubleshooting:	<p>Illegal axis identifier</p> <p>Limit and/or reference switches deactivated</p> <p>Servo mode not active (<a href="#">SVO? (p. 176)</a> responds with the value "0")</p>						

FRF Fast Reference Move To Reference Switch

Used in: Digital Input Signals (88), Referencing options (43), Triggering motion in closed-loop operation (36)

Description:	<p>Performs a reference move.</p> <p>The <b>Reference Signal Type</b> parameter (<a href="#">0x70 (p. 212)</a>) controls how the reference move is done and to which switch.</p> <p>FRF sets the current position to the value of the <b>Value at Reference Position</b> (<a href="#">0x16 (p. 199)</a>) parameter. If the reference move was successful, absolute motion will then be possible in closed-loop operation.</p> <p>The motion can be stopped by <a href="#">#24 (p. 126)</a>, <a href="#">STP (p. 175)</a>, and <a href="#">HLT (p. 152)</a>.</p> <p>Servo mode must be switched on for the specified axis (axes) (closed-loop operation).</p>
Format:	FRF[{_<AxisID>}]
Arguments:	<p>&lt;AxisID&gt;                      Axis ID</p> <p>If multiple axes are specified in the command, they are moved synchronously.</p> <p>If no arguments are specified, all axes are moved synchronously.</p>
Troubleshooting:	<p>Illegal axis identifier</p> <p>Servo mode not active (<a href="#">SVO? (p. 176)</a> responds with the value "0")</p>

FRF? Get Referencing Result

Used in: Referencing options (43)

Description:	Queries whether the specified axis is referenced.						
Format:	FRF?[{_<AxisID>}]						
Arguments:	<p>&lt;AxisID&gt;                      Axis ID</p> <p>The value for all axes will be queried if no arguments are specified.</p>						
Response:	<p>{AxisID}&lt;=&lt;uint&gt;<sub>LF</sub></p> <p>&lt;uint&gt; Reference state of the axis (BOOL)</p> <table><tr><th>&lt;uint&gt;</th><th>Description</th></tr><tr><td>1</td><td>Axis referenced successfully</td></tr><tr><td>0</td><td>Axis not referenced</td></tr></table>	<uint>	Description	1	Axis referenced successfully	0	Axis not referenced
<uint>	Description						
1	Axis referenced successfully						
0	Axis not referenced						
Troubleshooting:	Illegal axis identifier						

## GOH Go To Home Position

*Used in: Triggering motion in closed-loop operation (36)*

Description:	<p>Moves the specified axis to the zero position.</p> <p>GOH{&lt;AxisID&gt;} identical to <a href="#">MOV{&lt;AxisID&gt;.0}</a> (p. 164).</p> <p>The motion can be stopped by <a href="#">#24</a> (p. 126), <a href="#">STP</a> (p. 175), and <a href="#">HLT</a> (p. 152).</p> <p>Servo mode must be switched on for the specified axis (axes) (closed-loop operation). HID control may not be activated for the corresponding axis.</p>	
Format:	<b>GOH[ {&lt;AxisID&gt;} ]</b>	
Arguments:	<AxisID>	Axis ID
	<p>If multiple axes are specified in the command, they are moved synchronously.</p> <p>If no arguments are specified, all axes are moved synchronously.</p>	
Troubleshooting:	<p>Illegal axis identifier</p> <p>Servo mode not active (<a href="#">SVO?</a> (p. 176) responds with the value "0")</p>	

## HDR? Get All Data Recorder Options

*Used in: Reading general information from the data recorder (80)*

Description:	<p>Shows a help text that contains all available information on data recording.</p> <p>Shows information on:</p> <ul style="list-style-type: none"> <li>■ Recording options</li> <li>■ Trigger options</li> <li>■ Additional parameters and commands for data recording</li> </ul>	
Format:	<b>HDR?</b>	
Response:	<pre>#RecordOptions_LF {&lt;RecOption&gt;=&lt;DescriptionString&gt;[_of_&lt;Channel&gt;]_LF} #TriggerOptions_LF [{&lt;TriggerOption&gt;=&lt;DescriptionString&gt;_LF}] #Parameters_to_be_set_with_SPA_LF [{&lt;ParameterID&gt;=&lt;DescriptionString&gt;_LF}] #Additional_information_LF [{&lt;CommandDescription&gt;(&lt;Command&gt;)_LF}] #Sources_for_Record_Options_LF [{&lt;RecOption&gt;=&lt;Source&gt;_LF}] end_of_help</pre>	

### HDT Set HID Default Lookup Table

*Used in: HID Control Configuration (92)*

Description:	Assigns a lookup table to the specified axis of the specified HID.	
Format:	<b>HDT{_ &lt;HIDDeviceID&gt;_ &lt;HIDDeviceAxis&gt;_ &lt;HIDTableID&gt;}</b>	
Arguments:	<HIDDeviceID>	Identifier of an HID connected to the electronics
	<HIDDeviceAxis>	HID axis
	<HIDTableID>	Lookup table identifier

### HDT? Get HID Default Lookup Table

*Used in: HID Control Configuration (92)*

Description:	Queries the currently assigned lookup table for the specified axis of an HID.	
Format:	<b>HDT?[{_ &lt;HIDDeviceID&gt;_ &lt;HIDDeviceAxis&gt;}]</b>	
Arguments:	<HIDDeviceID>	Identifier of an HID connected to the electronics
	<HIDDeviceAxis>	HID axis
Response:	{<HIDDeviceID>_ <HIDDeviceAxis>=<HIDTableID> LF}	
	<HIDDeviceID>	Identifier of an HID connected to the electronics
	<HIDDeviceAxis>	HID axis
	<HIDTableID>	Lookup table identifier

# HIA Configure Control Done By HID Axis

*Used in: HID Control Configuration (92), Triggering motion in closed-loop operation (36)*

**Description:** Configures control of electronics' axes via HID axes (HID control).  
 Assigns an HID axis to the specified motion variable.  
 The configuration of the HID control is saved only to the volatile memory (RAM) of the electronics.  
 HID control may not be activated for the corresponding axis.  
 Motion variables that can be controlled by HIDs (depending on the controller; use the [HIA? \(p. 146\)](#) command to query the current configuration of the HID controlled motion variables):

<MotionParam>	Description
0	Deletes the current configuration of the HID control. Can be sent from <HIDDeviceID> and <HIDDeviceAxis> without specification.
1	Absolute target position The lookup table value corresponding to the current displacement of the human interface device axis is mapped to the travel range of the C-884.xDC axis to be controlled. The travel range limits are specified by the values of parameters <a href="#">0x30 (p. 202)</a> and <a href="#">0x15 (p. 199)</a> can be queried with <a href="#">TMN? (p. 180)</a> and <a href="#">TMX? (p. 180)</a> .
2	Relative target position Each pulse received (if applicable: Each mechanical detent) triggers relative motion over the distance set with the SST command. No lookup tables are used to control the relative target position.
3	Velocity of the axis Product of the lookup table value corresponding to the current displacement of the HID axis and the currently valid maximum velocity of the controller axis.
4	Maximum velocity of the axis Product of the lookup table value corresponding to the current displacement of the HID axis and the currently valid maximum velocity of the controller axis. Motion variable 4 can only be assigned to an HID axis when motion variable 3 has been assigned to a different HID axis.

**Format:** HIA{<AxisID>\_<MotionParam>\_<HIDDeviceID>\_<HIDDeviceAxis>}

**Arguments:**

<AxisID>	Axis ID
<MotionParam>	Axis motion variable
<HIDDeviceID>	Identifier of an HID connected to the electronics
<HIDDeviceAxis>	HID axis

Troubleshooting:      <MotionParam> has the value zero, i.e., function to be controlled has not been selected for the axis  
                             <HIDeviceID> has the value zero, i.e., HID has not been selected  
                             <HIDeviceAxis> has the value zero, i.e., HID's axis has not been selected for HID control

**HIA? Get Configuration Of Control Done By HID Axis**

*Used in: HID Control Configuration (92)*

Description:            Queries an HID's axis assigned to the specified motion variable of the specified motion variable.

Format:                 **HIA?[{\_<AxisID>\_<MotionParam>}]**

Arguments:            <AxisID>                    Axis ID

                             <MotionParam>            Axis motion variable

Response:              {<AxisID>\_<MotionParam>=<HIDeviceID>\_<HIDeviceAxis>\_LF}

                             <AxisID>                    Axis ID

                             <MotionParam>            Axis motion variable

                             <HIDeviceID>            Identifier of an HID connected to the electronics

                             <HIDeviceAxis>            HID axis

**HIB? Get State Of HID Button**

*Used in: HID Control Configuration (92)*

Description:            Queries the current state of the specified button of the specified HID.

Format:                 **HIB?[{\_<HIDeviceID>\_<HIDeviceButton>}]**

Arguments:            <HIDeviceID>            Identifier of an HID connected to the electronics

                             <HIDeviceButton>        HID button

Response:              {<HIDeviceID>\_<HIDeviceButton>=<HIDButtonState>\_LF}

                             <HIDeviceID>            Identifier of an HID connected to the electronics

                             <HIDeviceButton>        HID button

                             <HIDButtonState>        Button status (INT)

Any possible values of <HIDButtonState> depend on the button type. The value range can be queried with [HIS? \(p. 150\)](#).

## HIE? Get Deflection Of HID Axis

*Used in: HID Control Configuration (92)*

Description:	Queries the current displacement of the specified axis of the specified HID.	
Format:	<b>HIE?[{_&lt;HIDeviceID&gt;_&lt;HIDeviceAxis&gt;}]</b>	
Arguments:	<HIDeviceID>	Identifier of an HID connected to the electronics
	<HIDeviceAxis>	HID axis
Response:	{<HIDeviceID>_<HIDeviceAxis>=<HIDDeflection>_LF}	
	<HIDeviceID>	Identifier of an HID connected to the electronics
	<HIDeviceAxis>	HID axis
	<HIDDeflection>	Displacement of the HID's axis (FLOAT)
	<HIDDeflection> is a value between -1.0 and 1.0. A value almost equal to 0.0 corresponds to the axis' center position, -1.0 respectively 1.0 of the maximum displacement in a negative respectively positive direction.	

## HIL Set State of HID LED

*Used in: HID Control Configuration (92)*

Description:	Sets the specified output unit's state (e.g., LED) of the specified HID.  The meaning of <HIDeviceLED> and the possible values for <HIDLEDState> depend on the HID connected and can be queried with the <a href="#">HIS? (p. 150)</a> command.	
Format:	<b>HIL[{_&lt;DeviceID&gt;_&lt;HIDeviceLED&gt;_&lt;HIDLEDState&gt;}]</b>	
Arguments:	<HIDeviceID>	Identifier of an HID connected to the electronics
	<HIDeviceLED>	HID output unit
	<HIDLEDState>	Output unit state

## HIL? Get State Of HID LED

*Used in: HID Control Configuration (92)*

Description:	Queries the state of the specified output unit (e.g., LED) of the specified HID.	
Format:	<b>HIL?[{_&lt;HIDeviceID&gt;_&lt;HIDeviceLED&gt;}]</b>	
Arguments:	<HIDeviceID>	Identifier of an HID connected to the electronics
	<HIDeviceLED>	HID output unit
Response:	{<HIDeviceID>_<HIDeviceLED>=<HIDLEDState>_LF}	
	<HIDeviceID>	Identifier of an HID connected to the electronics
	<HIDeviceLED>	HID output unit
	<HIDLEDState>	Output unit state

# HIN Set Activation State For HID Control

*Used in: HID Control Configuration (92), Triggering motion in closed-loop operation (36)*

Description:	Sets the status of HID control for the specified axis.  Servo mode must be switched on for the specified axis (axes) (closed-loop operation).	
Format:	<b>HIN{&lt;_&lt;AxisID&gt;_&lt;HIDControlState&gt;}</b>	
Arguments:	<AxisID>	Axis ID
	<HIDControlState>	HID control's activation state (BOOL)
Troubleshooting:	Illegal axis identifier Servo mode not active ( <a href="#">SVO? (p. 176)</a> responds with the value "0") <a href="#">HID control is not suitably configured (p. 145)</a>	

# HIN? Get Activation State Of HID Control

*Used in: HID Control Configuration (92)*

Description:	Queries status of HID control for the specified axis.	
Format:	<b>HIN?[{&lt;_&lt;AxisID&gt;}]</b>	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<HIDControlState>_LF}	
	<AxisID>	Axis ID
	<HIDControlState>	HID control's activation state (BOOL)





# HIS Configure HI Device

Used in: *HID Control Configuration (92)*

**Description:** Configures the specified HID.

All supported operating elements of the HID are numbered consecutively for <HIDItemID> beginning with 1. The possible values of <HIDItemID> for connected HIDs can be determined with [HIS? \(p. 150\)](#). <HIDItemID> may only take values that correspond to output units; axes or buttons of HIDs cannot be configured with HIS.

Possible values for <HIDPropID>:

<HIDPropID>	Description
2	Value for the current state of the output unit

Any possible values for <HIDPropValue> depend on the connected HID.

**Format:** HIS{\_**<HIDDeviceID>**\_**<HIDItemID>**\_**<HIDPropID>**\_**<HIDPropValue>**}

**Arguments:**

<HIDDeviceID>	Identifier of an HID connected to the electronics
<HIDItemID>	HID operating element
<HIDPropID>	Property of the operating element
<HIDPropValue>	The property of the operating element is set to this value (STRING)

HIS? Get Configuration Of HI Device

Used in: Commandable Items (25), HID Control Configuration (92)

Description: Queries the specified property for the specified operating element of an HID.

Format: HIS?[{<HIDDeviceID>\_<HIDItemID>\_<HIDPropID>}]

Arguments: <HIDDeviceID> Identifier of an HID connected to the electronics  
<HIDItemID> HID operating element  
<HIDPropID> Property of the operating element

If no arguments are specified, information is queried on the supported operating elements of all HIDs.

Response: {<HIDDeviceID>\_<HIDItemID>\_<HIDPropID>=<HIDPropValue>\_LF}  
<HIDDeviceID> Identifier of an HID connected to the electronics  
<HIDItemID> HID operating element  
<HIDPropID> Property of the operating element  
<HIDPropValue> The property of the operating element is set to this value (STRING)

Possible values:

<HIDPropID>	Description
	Axis_<x>: HID axis, e.g., joystick axis or continuous slider, <x> indicates the identifier
	Button_<x>: HID button, <x> indicates the identifier
	Led_<x>: Output unit, e.g., LED or vibration motor on/off time, <x> indicates the identifier
2	Operating element status, e.g., axis displacement or LED activation status (FLOAT)
3	Name of the operating element (STRING)
4	HID Name (STRING)
5	Smallest possible value for the status of a "Button"- or "Led"-type operating element (INT)
6	Largest possible value for the status of a "Button"- or "Led"-type operating element (INT)

HIT Fill HID Lookup Table

Used in: *HID Control Configuration (92)*

Description:	Fills the specified lookup table with values.  HIT can only be used to fill user-defined tables. Tables with identifier $\leq 100$ are predefined and write-protected.  The first point of a lookup table corresponds to the maximum axis displacement of the HID in the negative direction; the 256th point corresponds to the maximum displacement in the positive direction. The values for points 1 to maximally 127 have a negative sign by default, while the remaining values have a positive sign. The sign of the values determines the direction of motion of the HID-controlled axis. Parameter <a href="#">0x61 (p. 210)</a> can be used to reverse the assigned direction that is specified by the values in the lookup table for an HID-controlled axis.  The <a href="#">HDT (p. 144)</a> command assigns the lookup tables to HID axes.	
Format:	HIT{<HIDTableID>,<HIDTableAddr>,<HIDTableValue>}	
Arguments:	<HIDTableID>	Lookup table identifier
	<HIDTableAddr>	Index of a point in the lookup table
	<HIDTableValue>	Value of the point with the index <HIDTableAddr> (FLOAT, -1.0...1.0)

HIT? Get HID Lookup Table Values

Used in: *HID Control Configuration (92)*

Description:	Queries the values of the specified points in the specified lookup table.	
Format:	HIT? [<StartPoint>,<NumberOfPoints>,{<HIDTableID>}]	
Arguments:	<StartPoint>	Index of the first point that is queried
	<NumberOfPoints>	Number of points to be queried per lookup table
	<HIDTableID>	Lookup table identifier
Response:	(Data in GCS array format)	

HLP? Get List Of Available Commands

Used in: *(24)*

Description:	Lists a help string which contains all commands available.
Format:	HLP?
Response:	(List of available commands)



HLT Halt Motion Smoothly

Used in: *Triggering Motion (36)*

Description:	Stops motion for the specified axis while considering the maximum set deceleration.	
	Sets the error code to 10.	
	Does not apply to trajectories: HLT also triggers an abrupt stop of motion when a trajectory is being followed.	
Format:	HLT[{_<AxisID>}]	
Arguments:	<AxisID>	Axis ID
Troubleshooting:	Illegal axis identifier	

HPA? Get List Of Available Parameters

Used in: *(24), Adapting Settings (111), Parameter Commands (111)*

Description:	Shows a help text that contains all available parameters with a short description.	
Format:	HPA?	
Response:	List of available parameters in the format: <PamID>=TAB<CmdLevel>TAB<MaxItem>TAB<Data-Type>TAB<FunctionGroupDescription>TAB<ParameterDescription>TAB[{<PossibleValue>=<ValueDescription>}]	
	<PamID>	Parameter ID
	<CmdLevel>	Command level for write access to the parameter
	<MaxItem>	Maximum number of elements of the same type that are affected by the parameter
	<DataType>	Data type of the parameter value
	<FunctionGroupDescription>	Name of the function group which the parameter belongs to
	<ParameterDescription>	Name of the parameter
	<PossibleValue>	Possible value
	<ValueDescription>	Value description



HPV? Get Parameter Value Description

Description:	Displays a help text that contains possible parameter values.		
Format:	HPV?		
Response:	<div>#Possible_parameter_values_are: LF {&lt;PamID&gt;_&lt;ItemID&gt;=&lt;ListType&gt;[{→&lt;PossibleValue&gt;=&lt;ValueDescription&gt;}] LF} #CCL_levels_are: LF {&lt;PamID&gt;_&lt;ItemID&gt;=&lt;CmdLevel&gt; LF} end of help  &lt;PamID&gt;                   Parameter ID &lt;ItemID&gt;                  Element of the electronics                              &lt;ItemID&gt; = 0: Description applies to all elements  &lt;ListType&gt;                Value list type                              &lt;ListType&gt; = 0: Parameter does not apply to this element                             &lt;ListType&gt; = 1: List of possible values                             &lt;ListType&gt; = 2: Minimum and maximum value  &lt;PossibleValue&gt;       Possible value &lt;ValueDescription&gt;   Value description &lt;CmdLevel&gt;            Command level for write access to the parameter</div>		

IFC Set Interface Parameters Temporarily

Used in: C-884.xDC Interfaces (17)

**Description:** Configures the interface parameters in the volatile memory.

The changed interface parameters are active immediately. The PC's interface configuration may also have to be changed and the connection to the electronics re-established.

The configuration of the interface parameters is saved only to the volatile memory (RAM) of the electronics. Use [IFS \(p. 156\)](#) to change interface parameters in the nonvolatile memory.

**Format:** IFC{\_**<InterfacePam>**\_**<PamValue>**}

**Arguments:** **<InterfacePam>** Interface parameters (STRING)  
**<PamValue>** Parameter value

Possible values

<InterfacePam>	<PamValue>	Description
RSBAUD	9600, 19200, 38400, 57600 or 115200	Baud rate for the RS-232 interface
IPADR	<UINT>.<UINT>.<UINT>.<UINT>:50000	IP address and port for the TCP/IP interface Port 50000 cannot be changed. IPADR is only used when IPSTART = 0.
IPSTART	0, 1	<PamValue> = 0: The IP address defined by IPADR is used <PamValue> = 1: DHCP is used (default)
IPMASK	<UINT>.<UINT>.<UINT>.<UINT>	TCP/IP interface subnet mask
IPGTWAY	<UINT>.<UINT>.<UINT>.<UINT>	Default gateway for TCP/IP communication



IFC? Get Current Interface Parameters

Used in: C-884.xDC Interfaces (17)

Description: Queries the interface parameter values in the volatile memory.

Format: IFC?[{\_<InterfacePam>}]

Arguments: <InterfacePam> Interface parameters (STRING)

Response: {<InterfacePam>=<PamValue>\_LF}  
<InterfacePam> Interface parameters (STRING)  
<PamValue> Parameter value

in the volatile memory

Possible values for <InterfacePam>:

<Interfa- cePam>	Description
RSBAUD	Baud rate for the RS-232 interface
IPADR	IP address and port for the TCP/IP interface
IPSTART	Startup behavior (DHCP status) of the TCP/IP interface
IPMASK	TCP/IP interface subnet mask
MACADR	Mac address (unique network hardware address)
IPGTWAY	Default gateway for TCP/IP communication



IFS Set Interface Parameters As Default Values

Used in: C-884.xDC Interfaces (17)

**Description:** Configures the interface parameters in the nonvolatile memory.  
The changed interface parameters are active after the next reboot.  
The PC's interface configuration may also have to be changed.  
Notice: Note that the number of write cycles in the nonvolatile memory is limited. Therefore, save to the nonvolatile memory only when necessary.  
Use [IFC \(p. 154\)](#) to change the interface parameters in the volatile memory (RAM).

**Format:** IFS <Pswd>[\_<InterfacePam>\_<PamValue>]

**Arguments:** <Pswd> Password for writing to the nonvolatile memory  
Default value is "100".  
<InterfacePam> Interface parameters (STRING)  
<PamValue> Parameter value

Possible values

<InterfacePam>	<PamValue>	Description
RSBAUD	9600, 19200, 38400, 57600 or 115200	Baud rate for the RS-232 interface
IPADR	<UINT>.<UINT>.<UINT>.<UINT>:50000	IP address and port for the TCP/IP interface Port 50000 cannot be changed. IPADR is only used when IPSTART = 0.
IPSTART	0, 1	<PamValue> = 0: The IP address defined by IPADR is used <PamValue> = 1: DHCP is used (default)
IPMASK	<UINT>.<UINT>.<UINT>.<UINT>	TCP/IP interface subnet mask
IPGTWAY	<UINT>.<UINT>.<UINT>.<UINT>	Default gateway for TCP/IP communication





IFS? Get Interface Parameters As Default Values

Used in: C-884.xDC Interfaces (17)

Description: Queries the values of the interface parameters in the nonvolatile memory.

Format: IFS?[{\_<InterfacePam>}]

Arguments: <InterfacePam> Interface parameters (STRING)

Response: {<InterfacePam>=<PamValue>\_LF}  
<InterfacePam> Interface parameters (STRING)  
<PamValue> Parameter value

in the nonvolatile memory

Possible values for <InterfacePam>:

<Interfa- cePam>	Description
RSBAUD	Baud rate for the RS-232 interface
IPADR	IP address and port for the TCP/IP interface
IPSTART	Startup behavior (DHCP status) of the TCP/IP interface
IPMASK	TCP/IP interface subnet mask
MACADR	Mac address (unique network hardware address)
IPGTWAY	Default gateway for TCP/IP communication

## JRC Jump Relatively Depending On Condition

Used in: *Analog Input Signals (90), Digital Input Signals (88), Running the Macros (101)*

**Description:** Jumps relative to a specified number of program lines within a macro.

Jumps irrespective of a specified condition.

Can only be used in macros.

Possible relational operators:

<OP>	Description
=	Equal
!=	Not equal
<=	Smaller than or equal
<	Smaller than
>=	Larger than or equal
>	Larger than

**Format:** JRC\_<Jump>\_<CMD?>\_<OP>\_<Value>

**Arguments:**

<Jump>	Size of relative jump
<CMD?>	Query command that responds with a single value
<OP>	Relational operator
<Value>	Relational value with <CMD?>

**Troubleshooting:**

- Wrong jump target (<Jump>) specified
- Wrong relational operator (<OP>) specified

## LIM? Indicate Limit Switches

Used in: *Detecting Limit Switches (40)*

**Description:** Queries whether the specified axis has limit switches.

Limit switch is detected by [0x32 \(p. 203\)](#).

**Format:** LIM?[{\_<AxisID>}]

**Arguments:**

<AxisID>	Axis ID
----------	---------

The value for all axes will be queried if no arguments are specified.

**Response:**

{<AxisID>=<uint>_LF}	
<AxisID>	Axis ID
<uint>	Limit switches available? (BOOL)

**Troubleshooting:** Illegal axis identifier

## MAC Call Macro Function

*Used in: Triggering motion in closed-loop operation (36)*

Description: Calls a macro function.

Possible macro functions are described separately:

- [MAC BEG \(p. 159\)](#)
- [MAC DEF \(p. 159\)](#)
- [MAC DEF? \(p. 159\)](#)
- [MAC DEL \(p. 160\)](#)
- [MAC END \(p. 160\)](#)
- [MAC ERR? \(p. 160\)](#)
- [MAC NSTART \(p. 160\)](#)
- [MAC START \(p. 161\)](#)

Format: **MAC\_<Keyword>{[\_<Parameter>]}**

Arguments:      <Keyword>              Macro function called  
                          <Parameter>              Function-dependent parameters

Troubleshooting:      Macro recording is active

## MAC BEG Call Macro Function: BEG

*Used in: (100)*

Description: Start recording a macro.

Recording is stopped by [MAC END \(p. 160\)](#).

Format: **MAC\_BEG\_<MacroName>**

Arguments:      <MacroName>              Macro name

## MAC DEF Call Macro Function: DEF

*Used in: (103)*

Description: Specifies a macro as startup macro.

Format: **MAC\_DEF\_<MacroName>**

Arguments:      <MacroName>              Name of the macro that is specified as startup macro

A startup macro is not used when no arguments are specified.

## MAC DEF? Call Macro Function: DEF?

*Used in: (103)*

Description: Queries the name of the startup macro.

Format: **MAC\_DEF?**

Response:      [<MacroName>] LF  
                          <MacroName>              Startup macro name

## MAC DEL Call Macro Function: DEL

*Used in: (104)*

Description:	Deletes the specified macro.	
Format:	<b>MAC_DEL_&lt;MacroName&gt;</b>	
Arguments:	<MacroName>	Macro name to be deleted

## MAC END Call Macro Function: END

*Used in: (100)*

Description:	Ends macro recording.
Format:	<b>MAC_END</b>

## MAC ERR? Call Macro Function: ERR?

*Used in: (103)*

Description:	Reports the last error that occurred while the macro was running.	
Format:	<b>MAC_ERR?</b>	
Response:	<MacroName>_<uint1>=<uint2>"<CMD>">	
	<MacroName>	Name of the macro
	<uint1>	Line in the macro where the error occurred
	<uint2>	Error code
	<CMD>	Bad command

## MAC NSTART Call Macro Function: NSTART

*Used in: (101)*

Description:	Runs the specified macro several times.  The macro is restarted when the last macro recording has completed, until <uint> has been reached.	
Format:	<b>MAC_NSTART_&lt;Macro-Name&gt;_&lt;uint&gt;[_&lt;String1&gt;[_&lt;String2&gt;[_&lt;String3&gt;[_&lt;String4&gt;]]]</b>	
Arguments:	<MacroName>	Macro name
	<uint>	Number of runs
	<String1...4>	Local variables 1 to 4
Troubleshooting:	No local variables specified although local variables are used in the macro	

MAC START Call Macro Function: START

*Used in: (101)*

Description:	Runs the specified macro.		
Format:	<b>MAC_START_&lt;Macro-Name&gt;[_&lt;String1&gt;[_&lt;String2&gt;[_&lt;String3&gt;[_&lt;String4&gt;]]]]</b>		
Arguments:	<MacroName>	Macro name	
	<String1...4>	Local variables 1 to 4	
Troubleshooting:	No local variables specified although local variables are used in the macro		

MAC? List Macros

*Used in: (100)*

Description:	Lists macros or content of a specified macro.		
Format:	MAC?[_<MacroName>]		
Arguments:	<MacroName>	Name of a macro	
	If no arguments are specified, the names of all saved macros are listed.		
Response:	<String>		
	<String>	Content of the macro or list of the names of all saved macros	
Troubleshooting:	Wrong macro name (<MacroName>)		

MAN? Get Help String For Command

Description:	Shows a help text for a command.		
Format:	MAN?_<CMD>		
Arguments:	<CMD>	Command that the help text is to be shown for	
Response:	<String>		
	<String>	Help text	



**MAT Calculate And Save To Variable**

Description: Performs a mathematical or bit operation and saves the result as a variable.

The variable is available in volatile memory (RAM) only.

Possible operators:

<OP>	Operation
+	Addition (mathematical)
-	Subtraction (mathematical)
*	Multiplication (mathematical)
AND	AND (logical)
OR	OR (logical)
XOR	EXCLUSIVE OR (logical)

Format: MAT\_<Variable>=<Float1>\_<OP>\_<Float2>

- Arguments:
- <Variable> Name of the variable where the result is to be saved
  - <Float1>, <Float2> Values for calculating the result
  - <OP> Operator to be used

MEX Stop Macro Execution Due To Condition

Used in: Analog Input Signals (90), Digital Input Signals (88), Stopping the Macros (103)

Description: Stops the macro due to a specified condition.

If the parser encounters this command, the condition is checked. If the condition is fulfilled at a later time, it is ignored by the parser.

Can only be used in macros.

Possible relational operators:

<OP>	Description
=	Equal
!=	Not equal
<=	Smaller than or equal
<	Smaller than
>=	Larger than or equal
>	Larger than

Format: MEX\_<CMD?>\_<OP>\_<Value>

Arguments:

<CMD?>	Query command that responds with a single value
<OP>	Relational operator
<Value>	Relational value with <CMD?>

# MOV Set Target Position

*Used in: Triggering motion in closed-loop operation (36), Variables for Generating a Dynamics Profile (30)*

Description:	<p>Sets the absolute target position for the specified axis.</p> <p>MOV overwrites the last target position received by the electronics. Motion commands from macros and command lines can overwrite each other. The last target position received may then not be reached.</p> <p>Servo mode must be switched on for the specified axis (axes) (closed-loop operation). HID control may not be activated for the corresponding axis.</p> <p>The target position must be inside the soft limits.</p> <p>The motion can be stopped by <a href="#">#24 (p. 126)</a>, <a href="#">STP (p. 175)</a>, and <a href="#">HLT (p. 152)</a>.</p>	
Format:	<b>MOV{&lt;AxisID&gt;,&lt;Position&gt;}</b>	
Arguments:	<AxisID>	Axis ID
	<Position>	Absolute target position in physical units
Troubleshooting:	<p>Illegal axis identifier</p> <p>Servo mode not active (<a href="#">SVO? (p. 176)</a> responds with the value "0")</p> <p>Executing motion commands deactivated by parameter <a href="#">0x130 (p. 216)</a></p> <p>Target position outside of the soft limits</p> <p>HID control is enabled</p>	

# MOV? Get Target Position

Description:	Queries last valid commanded target position.	
Format:	<b>MOV?[{&lt;AxisID&gt;}]</b>	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<float> LF}	
	<AxisID>	Axis ID
	<float>	Last valid target position in physical units (FLOAT)
Troubleshooting:	Illegal axis identifier	



## MVE Set Target Position For Vectorial Move

*Used in: Triggering motion in closed-loop operation (36), Variables for Generating a Dynamics Profile (30)*

Description:	Sets the absolute target positions for the specified axes for vectorial motion.  MVE is intended for simultaneous motion of coupled axes that results in a straight line.  Servo mode must be switched on for the specified axis (axes) (closed-loop operation). HID control may not be activated for the corresponding axis.  The target position must be inside the soft limits.  The motion can be stopped by <a href="#">#24 (p. 126)</a> , <a href="#">STP (p. 175)</a> , and <a href="#">HLT (p. 152)</a> .	
Format:	<b>MVE{_<a href="#">AxisID</a>}_&lt;Position&gt;}</b>	
Arguments:	<AxisID>	Axis ID
	<Position>	Absolute target position in physical units
Troubleshooting:	Illegal axis identifier Servo mode not active ( <a href="#">SVO? (p. 176)</a> responds with the value "0") Executing motion commands deactivated by parameter <a href="#">0x130 (p. 216)</a> Target position outside of the soft limits HID control is enabled	

## MVR Set Target Relative To Current Position

*Used in: Triggering motion in closed-loop operation (36), Variables for Generating a Dynamics Profile (30)*

Description:	<p>Moves the specified axis relative to the last commanded target position.</p> <p>The new target position is calculated from the sum of the last commanded target position and &lt;Distance&gt;.</p> <p>Servo mode must be switched on for the specified axis (axes) (closed-loop operation). HID control may not be activated for the corresponding axis.</p> <p>The target position must be inside the soft limits.</p> <p>The motion can be stopped by <a href="#">#24 (p. 126)</a>, <a href="#">STP (p. 175)</a>, and <a href="#">HLT (p. 152)</a>.</p>	
Format:	<b>MVR{&lt;_AxisID&gt;_&lt;Distance&gt;}</b>	
Arguments:	<AxisID>	Axis ID
	<Distance>	Distance that the axis has to move, specification in physical units
Troubleshooting:	<p>Illegal axis identifier</p> <p>Servo mode not active (<a href="#">SVO? (p. 176)</a> responds with the value "0")</p> <p>Executing motion commands deactivated by parameter <a href="#">0x130 (p. 216)</a></p> <p>Target position outside of the soft limits</p> <p>HID control is enabled</p>	

## ONT? Get On-Target State

*Used in: On-Target State (38)*

Description:	<p>Queries the on-target status of the specified axis.</p> <p>The on-target state is influenced by the <a href="#">0x36 (p. 204)</a> and <a href="#">0x3F (p. 205)</a> settings.</p> <p>Servo mode must be switched on for the specified axis (axes) (closed-loop operation).</p>	
Format:	<b>ONT?[{&lt;_AxisID&gt;}]</b>	
Arguments:	<AxisID>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{<AxisID>=<uint>_LF}	
	<AxisID>	Axis ID
	<uint>	On-Target-Status (BOOL)
Troubleshooting:	<p>Illegal axis identifier</p> <p>Servo mode not active (<a href="#">SVO? (p. 176)</a> responds with the value "0")</p>	

## POS Set Real Position

*Used in: Referencing options (43)*

Description:	<p>Sets the current position of the axis (does not trigger motion).</p> <p>POS can only be used when <a href="#">the referencing method is set to "0"</a> (p. 168).</p> <p>The minimum (<a href="#">TMN? (p. 180)</a>) and maximum (<a href="#">TMX? (p. 180)</a>) commandable positions are not influenced by POS. This means that target positions are possible that are permitted by the electronics but cannot be reached by the mechanics, or could be reached by the mechanics but are not permitted by the electronics.</p> <p>■</p> <p>■</p>	
Format:	POS{<_AxisID>_<Position>}	
Arguments:	<AxisID>	Axis ID
	<Position>	New current position [phys. unit]
Troubleshooting:	Illegal axis identifier	

## POS? Get Real Position

Description:	Queries the current axis position.	
Format:	POS?[{<_AxisID>}]	
Arguments:	<AxisID>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{<AxisID>=<float> LF}	
	<AxisID>	Axis ID
	<float>	Current axis position in physical units
Troubleshooting:	Illegal axis identifier	

## RBT Reboot System

*Used in: Controller Macros (99)*

Description:	<p>Restarts the electronics.</p> <p>The electronics behave in the same way after restarting as they do after switching on.</p> <p>RBT cannot be used in macros.</p>	
Format:	RBT	

RMC? List Running Macros

Used in: (101)

Description: Queries all macros currently running.

Format: RMC?

Response: {[<MacroName>] LF}  
<MacroName> Macro name

RON Set Reference Mode

Used in: Referencing options (43)

Description: Sets the referencing method for the specified axis.

Format: RON{\_**<AxisID>**\_**<ReferenceOn>**}

Arguments: **<AxisID>** Axis ID

**<ReferenceOn>** Referencing method

Possible referencing modes:

- 0 Absolute position value is assigned with [POS \(p. 167\)](#) or a reference move is done with [FRF \(p. 142\)](#) (also FNL or FPL if supported by the controller). Relative motion is possible with [MVR \(p. 166\)](#), even when the axis has not been referenced.
- 1 A reference move must be done with [FRF \(p. 142\)](#) (also FNL or FPL if supported by the controller) for referencing, [POS \(p. 167\)](#) is not permitted. Motion in closed-loop operation is only possible after the axis has been referenced. (Default)

Troubleshooting: Illegal axis identifier

# RON? Get Reference Mode

*Used in: Referencing options (43)*

Description:	Queries referencing method of the specified axis.
Format:	<b>RON?[{_&lt;AxisID&gt;}]</b>
Arguments:	<AxisID> Axis identifier
Response:	<p>{&lt;AxisID&gt;=&lt;ReferenceOn&gt; LF}</p> <p>&lt;AxisID&gt; Axis identifier</p> <p>&lt;ReferenceOn&gt; Referencing mode</p> <p>Possible referencing modes:</p> <ol style="list-style-type: none"> <li>0 Absolute position value is assigned with <a href="#">POS (p. 167)</a> or a reference move is done with <a href="#">FRF (p. 142)</a> (also FNL or FPL if supported by the controller). Relative motion is possible with <a href="#">MVR (p. 166)</a>, even when the axis has not been referenced.</li> <li>1 A reference move must be done with <a href="#">FRF (p. 142)</a> (also FNL or FPL if supported by the controller) for referencing, <a href="#">POS (p. 167)</a> is not permitted. Motion in closed-loop operation is only possible after the axis has been referenced. (Default)</li> </ol>
Troubleshooting:	Illegal axis identifier

# RPA Reset Volatile Memory Parameters

*Used in: Parameter Commands (111)*

Description:	Resets the specified parameter.
	<p>The parameter value in the volatile memory is overwritten by the value in the nonvolatile memory.</p> <p>RPA resets the parameter for the setting of hardware-specific parameters. Wrong values can lead to faulty operation or damage to the hardware.</p>
Format:	<b>RPA[{_&lt;ItemID&gt;_&lt;PamID&gt;}]</b>
Arguments:	<p>&lt;ItemID&gt; Element of the electronics</p> <p>For example, axis identifier Parameter &lt;PamID&gt; is reset for this element.</p> <p>&lt;PamID&gt; Parameter ID</p>
Troubleshooting:	<p>Impermissible element identifier</p> <p>Wrong parameter ID</p>

RTR Set Record Table Rate

Used in: *Setting the record table rate (81)*

Description:	Sets the record table rate of the data recorder.  The recording rate multiplied by the cycle time of the electronics results in the recording interval of the data recorder. Settings greater than 1 allow recording over longer periods of time.
Format:	RTR <RecordTableRate>
Arguments:	<RecordTableRate> Data recording rate of the data recorder in number of cycles (UINT >0)

RTR? Get Record Table Rate

Used in: *Setting the record table rate (81)*

Description:	Queries the current recording rate of the data recorder.  The recording rate multiplied by the cycle time of the electronics results in the recording interval of the data recorder. Settings greater than 1 allow recording over longer periods of time.
Format:	RTR?
Response:	<RecordTableRate> LF  <RecordTableRate> Data recording rate of the data recorder in number of cycles (UINT >0)

SAI Set Current Axis Identifiers

Used in: *Commandable Items (25)*

Description:	Sets the axis identifier for the specified axis.  The new axis identifier is stored in the nonvolatile memory of the electronics.  Use <a href="#">TVI? (p. 182)</a> to query the valid characters for the axis identifier.
Format:	SAI{_ <b>&lt;AxisID&gt;</b> _ <b>&lt;NewIdentifier&gt;</b> }
Arguments:	<AxisID> Axis ID  <NewIdentifier> New axis ID



SAI? Get List Of Current Axis Identifiers

*Used in: Commandable Items (25)*

Description:	Queries the axis identifiers.  ALL ensures that the response also includes axes that are deactivated for electronics that permit axes to be deactivated.	
Format:	SAI?[_ALL]	
Arguments:	[_ALL]	Ensures that electronics that allow axes to be deactivated (parameter <a href="#">0x3C (p. 204)</a> = "NOSTAGE") also allow deactivated axes to be queried.
Response:	{<AxisID>_LF}	
	<AxisID>	Axis ID

SEP Set Non-Volatile Memory Parameters

*Used in: Parameter Commands (111), Variables for Generating a Dynamics Profile (30)*

Description:	Sets a parameter in the nonvolatile memory to a specific value.  Up to four parameters can be set per command.  Wrong values can lead to faulty operation or damage to the hardware.	
Format:	SEP_<Pswd>{_<ItemID>_<PamID>_<PamValue>}	
Arguments:	<Pswd>	Password for writing to the nonvolatile memory
	<ItemID>	Element of the electronics
	<PamID>	Parameter ID
	<PamValue>	Parameter value
Troubleshooting:	Impermissible element identifier Wrong parameter ID Wrong password	

# SEP? Get Non-Volatile Memory Parameters

*Used in: Parameter Commands (111), Saving Parameter Values in a Text File (112)*

Description:	<p>Queries the value of a parameter in the nonvolatile memory.</p> <p>Up to four parameters can be queried per command.</p>	
Format:	<b>SEP?[_&lt;ItemID&gt;_&lt;PamID&gt;]</b>	
Arguments:	<ItemID>	Element of the electronics
	<PamID>	Parameter ID
Response:	{<ItemID>_<PamID>=<PamValue>_LF }	
	<ItemID>	Element of the electronics
	<PamID>	Parameter ID
	<PamValue>	Parameter value
Troubleshooting:	Impermissible element identifier	
	Wrong parameter ID	

# SMO Set Open-Loop Control Value

*Used in: Triggering motion in open-loop operation (37)*

Description:	Sets the control value of an axis.	
	Profile generator, sensor feedback, and servo algorithm are not taken into consideration.	
	Servo mode must be switched off for the specified axis (axes) (open-loop operation).	
	Note: In the case of large control values, the mechanics can collide with the end of the mechanical travel range despite limit switch function!	
Format:	<b>SMO{[_&lt;AxisID&gt;_&lt;ControlValue&gt;]}</b>	
Arguments:	<AxisID>	Axis ID
	<ControlValue>	Control value, dimensionless
Troubleshooting:	Illegal axis identifier	
	Servo mode active ( <a href="#">SVO? (p. 176)</a> responds with the value "1")	
	Executing motion commands deactivated by parameter <a href="#">0x130 (p. 216)</a>	



## SMO? Get Control Value

Description:	Returns last valid control value of the specified axis.	
Format:	<b>SMO?[{_&lt;AxisID&gt;}]</b>	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<float>_LF}	
	<AxisID>	Axis ID
	<float>	Control value, dimensionless
Troubleshooting:	Illegal axis identifier	

## SPA Set Volatile Memory Parameters

*Used in: Parameter Commands (111), Variables for Generating a Dynamics Profile (30)*

Description:	Sets a parameter in the volatile memory to a specific value.	
	Up to four parameters can be set per command.	
	Wrong values can lead to faulty operation or damage to the hardware.	
Format:	<b>SPA{[_&lt;ItemID&gt;_&lt;PamID&gt;_&lt;PamValue&gt;}]</b>	
Arguments:	<ItemID>	Element of the electronics
	<PamID>	Parameter ID
	<PamValue>	Parameter value
Troubleshooting:	Impermissible element identifier	
	Wrong parameter ID	

## SPA? Get Volatile Memory Parameters

*Used in: Parameter Commands (111), Saving Parameter Values in a Text File (112)*

Description:	Queries the value of a parameter in the volatile memory.	
	Up to four parameters can be queried per command.	
Format:	<b>SPA?[{[_&lt;ItemID&gt;_&lt;PamID&gt;}]</b>	
Arguments:	<ItemID>	Element of the electronics
	<PamID>	Parameter ID
Response:	{<ItemID>_<PamID>=<PamValue>_LF}	
	<ItemID>	Element of the electronics
	<PamID>	Parameter ID
	<PamValue>	Parameter value
Troubleshooting:	Impermissible element identifier	
	Wrong parameter ID	

SRG? Query Status Register Value

Used in: (29), On-Target State (38)

Description: Queries system status.

Format: SRG?{\_<ItemID>\_<RegisterID>}

Arguments: <ItemID> Element of the electronics  
<RegisterID> Register ID

Response: {<ItemID>\_<RegisterID>=<Value>\_LF}

<ItemID> Element of the electronics

<RegisterID> Register ID

<Value> Register value (HEX)

<StatusRegister> is bit-mapped. It comprises:

Bit	Description
15	On-target state
14	Referencing is being done
13	In motion
12	Servo mode activated
11	
10	Sensor signal valid
9	Reference edge found
8	Error
7	Digital input 4 active
6	Digital input 3 active
5	Digital input 2 active
4	Digital input 1 active
3	Sensor is referenced
2	Positive limit switch active
1	Reference switch active
0	Negative limit switch active

# STE Start Step And Response Measurement

*Used in: Starting the Recording (81), Triggering motion in closed-loop operation (36), Variables for Generating a Dynamics Profile (30)*

Description:	<p>Starts a step for the specified axis and records the step response.</p> <p>Servo mode must be switched on for the specified axis (axes) (closed-loop operation). HID control may not be activated for the corresponding axis.</p> <p>The target position must be inside the soft limits.</p>	
Format:	<b>STE_&lt;AxisID&gt;_&lt;Amplitude&gt;</b>	
Arguments:	<AxisID>	Axis ID
	<Amplitude>	Jump size in physical units
Troubleshooting:	Illegal axis identifier	
	Servo mode not active ( <a href="#">SVO? (p. 176)</a> responds with the value "0")	
	Executing motion commands deactivated by parameter <a href="#">0x130 (p. 216)</a>	
	Target position outside of the soft limits	
	HID control is enabled	

# STP Stop All Axes

*Used in: (103), Triggering Motion (36)*

Description:	<p>Stops all axes abruptly.</p> <p>Stops all motion started by motion commands (e.g., <a href="#">MOV (p. 164)</a>, <a href="#">MVR (p. 166)</a>, <a href="#">MVE (p. 165)</a>, <a href="#">STE (p. 175)</a>, <a href="#">SMO (p. 172)</a>), referencing commands (FNL, FPL, <a href="#">FRF (p. 142)</a>), and macros.</p> <p>Also stops the macro.</p> <p>Sets the error code to 10.</p> <p>After the axes are stopped, their target positions are set to their current positions.</p>	
Format:	<b>STP</b>	

# SVO Set Servo Mode

*Used in: (29), Restoring the C-884.xDC's Operational Readiness (77)*

Description:	<p>Sets the servo mode for the specified axis.</p> <p>The target position is set to the current position when switching to closed-loop operation. SVO deactivates the brake when it is activated.</p>	
Format:	<b>SV0{_&lt;AxisID&gt;_&lt;ServoState&gt;}</b>	
Arguments:	<AxisID>	Axis ID
	<ServoState>	Servo mode (UINT)

# SVO? Get Servo Mode

*Used in: (29)*

Description:	Queries the servo mode for the specified axis.	
Format:	<b>SVO?[{_&lt;AxisID&gt;}]</b>	
Arguments:	<AxisID>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{<AxisID>=<ServoState>_LF}	
	<AxisID>	Axis ID
	<ServoState>	Axis servo mode (UINT)
Troubleshooting:	Illegal axis identifier	

# TAC? Tell Analog Channels

*Used in: Analog Input Signals (90), Commandable Items (25)*

Description:	Queries the number of available analog lines.	
Format:	<b>TAC?</b>	
Response:	<uint>	
	<uint> Number of analog lines (inputs and outputs)	

# TAV? Get Analog Input Voltage

*Used in: Analog Input Signals (90), Commandable Items (25)*

Description:	Queries the voltage at the analog input.	
Format:	<b>TAV?[{_&lt;AnalogInputID&gt;}]</b>	
Arguments:	<AnalogInputID> is the identifier of an analog input channel	
Response:	{<AnalogInputID>=<float>_LF}	
	<AnalogInputID> is the identifier of an analog input channel	
	<float> Current voltage at the analog input (FLOAT)	
	All analog inputs are queried if no arguments are specified.	

## TCV? Get Commanded Closed-Loop Velocity

Description:	Queries the current velocity value that was calculated by the profile generator.	
Format:	TCV?[{_<AxisID>}]	
Arguments:	<AxisID>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{AxisID>=<float> <sub>LF</sub> }	
	<AxisID>	Axis ID
	<float>	Velocity in physical units / s (FLOAT)

## TGA Append Value To Trajectory

*Used in: Working with Trajectories (78)*

Description:	Adds a trajectory point to a specified trajectory.  The trajectory buffer can only accommodate the maximum number of trajectory points that were set with parameter <a href="#">0x22000020 (p. 223)</a> .	
Format:	TGA{<Trajectory>_<Point>}	
Arguments:	<Trajectory>	Trajectory ID
	<Point>	Absolute position in physical units (FLOAT)
Troubleshooting:	Wrong trajectory ID Trajectory buffer full (parameter <a href="#">0x22000020 (p. 223)</a> )	

## TGC Clear All Values In Trajectory

*Used in: Working with Trajectories (78)*

Description:	Deletes all trajectory points of a specified trajectory.  If trajectory processing is stopped because of an error or with <a href="#">STP (p. 175)</a> respectively <a href="#">#24 (p. 126)</a> , the trajectory points not yet processed at this point will remain in the trajectory buffer. They must be deleted explicitly with TGC.	
Format:	TGC[{_<Trajectory>}]	
Arguments:	<Trajectory>	Trajectory ID
	All trajectories are emptied if no arguments are specified.	
Troubleshooting:	Wrong trajectory ID	

## TGF Finalize Trajectory

*Used in: Working with Trajectories (78)*

Description:	Completes trajectory processing.	
	TGF signals to the firmware of the electronics that no further points will be sent for the specified trajectory. This means that the remaining trajectory points will be processed without an error occurring when the minimum number of points is not reached.	
Format:	TGF[[_<Trajectory>]]	
Arguments:	<Trajectory>	Trajectory ID
	All trajectories are completed if no arguments are specified.	
Troubleshooting:	Wrong trajectory ID	
	The specified trajectory is currently not being processed	

## TGL? Get Number Of Values In Trajectory

*Used in: Working with Trajectories (78)*

Description:	Queries the number of remain points of a trajectory.	
Format:	TGL?[_<Trajectory>]	
Arguments:	<Trajectory>	Trajectory ID
	All trajectories are queried if no arguments are specified.	
Response:	{<Trajectory>=<uint> LF }	
	<Trajectory>	Trajectory ID
	<uint> Number of remaining points	

## TGS Start Trajectory

*Used in: Triggering motion in closed-loop operation (36), Working with Trajectories (78)*

Description:	Process the specified trajectory.	
Format:	TGS[[_<Trajectory>]]	
Arguments:	<Trajectory>	Trajectory ID
	All trajectories are processed if no arguments are specified.	
Troubleshooting:	Wrong trajectory ID	
	Trajectory does not contain enough points	

## TGT Set Trajectory Timing

*Used in: Working with Trajectories (78)*

**Description:** Sets the time interval between the points in a trajectory.  
The specified value applies to all trajectories.

**Format:** TGT <NoOfServoCycles>

**Arguments:** <NoOfServoCycles> Time interval between points of a trajectory in number of servo cycles (UINT)

**Description:** Queries the time interval between the points of a trajectory.  
The returned value applies to all trajectories.

**Format:** TGT?

**Response:** <NoOfServoCycles> LF  
<NoOfServoCycles> Time interval between points of a trajectory in number of servo cycles (UINT)

**Description:** Sets the timer to the specified value.  
The timer is incremented each servo cycle and can be used for time measurement. The increment starts at zero every time the electronics are switched on or rebooted.  
The servo cycle time is stored in parameter [0xE000200 \(p. 219\)](#).

**Format:** TIM[\_<Float>]

**Arguments:** <Float> Value that the timer set to, in milliseconds (UINT)  
If no arguments are specified, the timer is reset to zero.

**Description:** Queries the current value of the timer.

**Format:** TIM?

**Response:** <Float> LF  
<Float> Current value of the timer in milliseconds

## TIO? Tell Digital I/O Lines

*Used in: Digital Input Signals (88), Digital Inputs and Outputs (83), Digital Output Signals (83)*

**Description:** Queries the number of available digital I/O lines.

**Format:** TIO?

**Response:** I=<uint1>\_ LF  
O=<uint2> LF  
<uint1> Number of digital input lines (UINT)  
<uint2> Number of digital output lines (UINT)

TMN? Get Minimum Commandable Position

Description:	Queries the minimum commandable position. The minimum commandable position is determined by parameter <a href="#">0x30 (p. 202)</a> .	
Format:	TMN? [{_<AxisID>}]	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<float> LF }	
	<AxisID>	Axis ID
	<float>	Minimum commandable position in physical units (FLOAT)

TMX? Get Maximum Commandable Position

Description:	Queries the maximum commandable position. The maximum commandable position is defined by parameter <a href="#">0x15 (p. 199)</a> .	
Format:	TMX? [{_<AxisID>}]	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<float> LF }	
	<AxisID>	Axis ID
	<float>	Maximum commandable position in physical units (FLOAT)

TNR? Get Number Of Record Tables

*Used in: Commandable Items (25)*

Description:	Queries the number of available data recorder tables.	
Format:	TNR?	
Response:	<uint>	
	<uint>	Number of available data recorder tables (UINT)



## TRO Set Trigger Output State

*Used in: Digital Output Signals (83), Setting "Hardware Trigger" Mode (87), Setting "In Motion" Trigger Mode (85), Setting "Position Distance" Trigger Mode (84), Setting "Position+Offset" Trigger Mode (86), Setting "Single Position" Trigger Mode (87), Setting Up "On Target" Trigger Mode (85)*

Description:	Activates or deactivates the trigger output for a digital output line.	
Format:	TRO{<_<TrigOutID>>_<TrigMode>>}	
Arguments:	<TrigOutID>	Digital output of the electronics
	<TrigMode>	Trigger output state (UINT)
Troubleshooting:	Impermissible element identifier	

## TRO? Get Trigger Output State

*Used in: Digital Output Signals (83)*

Description:	Queries the trigger output state for a a digital output line.	
Format:	TRO?[{<_<TrigOutID>>}]	
Arguments:	<TrigOutID>	Digital output of the electronics
	If no arguments are specified, the state of all digital output lines is queried.	
Response:	{<TrigOutID>=<TrigMode>_LF}	
	<TrigMode>	Trigger output state (UINT)
Troubleshooting:	Impermissible element identifier	

## TRS? Indicate Reference Switch

*Used in: Reference Switch Detection (39)*

Description:	Queries whether the specified axis has a reference switch with direction sensing.	
	TRS? queries the value of parameter <a href="#">0x14 (p. 198)</a> .	
Format:	TRS?[{<_<AxisID>>}]	
Arguments:	<AxisID>	Axis ID
Response:	{<AxisID>=<uint>_LF}	
	<AxisID>	Axis ID
	<uint>	Direction sensing reference switch available? (BOOL)
Troubleshooting:	Illegal axis identifier	

## TVI? Tell Valid Character Set For Axis Identifiers

Description:	Queries permissible characters for axis identifiers.
Format:	TVI?
Response:	<String> <String> Characters that are permitted for use in axis identifiers

## VAR Set Variable Value

*Used in: Running the Macros (101)*

Description:	Sets a variable to a specific value. Local variables can only be set in macros. The variable is only in volatile memory (RAM).
Format:	VAR_<Variable>_<String>
Arguments:	<Variable> Name of the variable whose value is set <String> Value, that the variable is set to Can be specified directly or via the value of a variable.

## VAR? Get Variable Values

*Used in: Running the Macros (101)*

Description:	Queries the value of a variable. Local variables can only be queried when a macro is running that contains local variables.
Format:	VAR?[{_<Variable>}]
Arguments:	<Variable> Name of the variable being queried All variables are queried if no arguments are specified.
Response:	{<Variable>=<String>_LF} <Variable> Name of the variable <String> Value of the variable

# VEL Set Closed-Loop Velocity

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	<p>Sets the velocity for the specified axis.</p> <p>Changes the value of the parameter <a href="#">0x49 (p. 206)</a> in the volatile memory. The maximum value that can be set with VEL, is specified by parameter <a href="#">0xA (p. 196)</a>.</p> <p>The setting only takes effect when the specified axis is in closed-loop operation (servo mode ON).</p>	
Format:	<b>VEL{&lt;_AxisID&gt;,&lt;Velocity&gt;}</b>	
Arguments:	<AxisID>	Axis ID
	<Velocity>	Velocity [physical units / s]]
Troubleshooting:	Illegal axis identifier	

# VEL? Get Closed-Loop Velocity

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	Gets the commanded velocity.	
Format:	<b>VEL?[{&lt;_AxisID&gt;}]</b>	
Arguments:	<AxisID>	Axis ID
	The value for all axes will be queried if no arguments are specified.	
Response:	{<AxisID>=<float> LF }	
	<AxisID>	Axis ID
	<float>	Velocity in physical units / s (FLOAT)
Troubleshooting:	Illegal axis identifier	

# VER? Get Versions Of Firmware And Drivers

*Used in: Important Firmware Components (23)*

Description:	<p>Queries the version numbers of the firmware.</p> <p>VER? also queries the version numbers of further components such as drivers and libraries.</p>	
Format:	<b>VER?</b>	
Response:	{<string1>:<string2> LF }	
	<string1>	Name of the component
	<string2>	Version information and optional specifications

WAC Wait For Condition

Used in: Analog Input Signals (90), Digital Input Signals (88), Running the Macros (101)

Description:                      Waits until a condition is met.

WAC compares a specified value with a queried value according to a specified rule.

Can only be used in macros.

Possible relational operators:

<OP>	Description
=	Equal
!=	Not equal
<=	Smaller than or equal
<	Smaller than
>=	Larger than or equal
>	Larger than

Format:                              WAC\_<CMD?>\_<OP>\_<Value>

Arguments:                      <CMD?>                      Query command that responds with a single value

   <OP>                      Relational operator

   <Value>                      Relational value with <CMD?>

WPA Save Parameters To Non-Volatile Memory

Used in: Parameter Commands (111)

Description:	<p>Writes the value of a parameter from the volatile memory (RAM) to the nonvolatile memory.</p> <p>WPA can also save parameter-independent settings. The used password determines what is saved with WPA:</p> <table><tr><th>&lt;Pswd&gt;</th><th>Description</th></tr><tr><td>100</td><td>Saves current valid values of all parameters and the current valid settings for <a href="#">HDT (p. 144)</a>, <a href="#">HIA (p. 145)</a>, and <a href="#">HIT (p. 151)</a>.</td></tr><tr><td>101</td><td>Saves the currently valid values of all parameters.</td></tr><tr><td>HID</td><td>Saves the current valid settings for <a href="#">HDT (p. 144)</a>, <a href="#">HIA (p. 145)</a>, and <a href="#">HIT (p. 151)</a>.</td></tr></table> <p>Wrong values can lead to faulty operation or damage to the hardware.</p> <p>Notice: Note that the number of write cycles in the nonvolatile memory is limited. Therefore, save to the nonvolatile memory only when necessary.</p>	<Pswd>	Description	100	Saves current valid values of all parameters and the current valid settings for <a href="#">HDT (p. 144)</a> , <a href="#">HIA (p. 145)</a> , and <a href="#">HIT (p. 151)</a> .	101	Saves the currently valid values of all parameters.	HID	Saves the current valid settings for <a href="#">HDT (p. 144)</a> , <a href="#">HIA (p. 145)</a> , and <a href="#">HIT (p. 151)</a> .
<Pswd>	Description								
100	Saves current valid values of all parameters and the current valid settings for <a href="#">HDT (p. 144)</a> , <a href="#">HIA (p. 145)</a> , and <a href="#">HIT (p. 151)</a> .								
101	Saves the currently valid values of all parameters.								
HID	Saves the current valid settings for <a href="#">HDT (p. 144)</a> , <a href="#">HIA (p. 145)</a> , and <a href="#">HIT (p. 151)</a> .								
Format:	<b>WPA_&lt;Pswd&gt;[{_&lt;ItemID&gt;_&lt;PamID&gt;}]</b>								
Arguments:	<table><tr><td>&lt;Pswd&gt;</td><td>Password for writing to the nonvolatile memory</td></tr><tr><td>&lt;ItemID&gt;</td><td>Element of the electronics</td></tr><tr><td>&lt;PamID&gt;</td><td>Parameter ID</td></tr></table> <p>The &lt;ItemID&gt; and &lt;PamID&gt; specifications are not supported by all electronics.</p>	<Pswd>	Password for writing to the nonvolatile memory	<ItemID>	Element of the electronics	<PamID>	Parameter ID		
<Pswd>	Password for writing to the nonvolatile memory								
<ItemID>	Element of the electronics								
<PamID>	Parameter ID								
Troubleshooting:	<p>Impermissible element identifier</p> <p>Wrong parameter ID</p> <p>Wrong password</p>								

## 10 Parameter Reference

0x1	P term	Proportional constant of the PID servo algorithm. Is used for fast correction of the position error. 0 to 65535
0x2	I term	Integration constant of the PID servo algorithm. Used for reducing static position error. 0 to 65535
0x3	D term	Differential constant of the PID servo algorithm. Used for damping rapid control oscillation. The D term can be calculated as a floating average over several servo cycles. The parameter 0x71 (D-Term Delay) specifies how many values (i.e., servo cycles) are to be used for calculating the average. 0 to 65535 The preset value of this parameter should not be changed.
0x4	I limit	Limit of the integration constant. 0 to 1048575
0x5	Kvff	Feed-forward control of the commanded velocity. Used for minimizing positioning error. 0 to 65535
0x8	Maximum Position Error (Phys. Unit)	Maximum position error. A motion error occurs when the difference between the current position and the commanded position exceeds this value and the movement of the axis is stopped.
0x9	Maximum Motor Output	Maximum control value for driving an axis respectively a channel. See the response to HPA? for possible values
0xA	Maximum Closed-Loop Velocity (Phys. Unit/s)	Maximum velocity with dynamics profile in closed-loop operation. Specifies the maximum value for parameter 0x49.
0xB	Closed-Loop Acceleration (Phys. Unit/s <sup>2</sup> )	Acceleration with dynamics profile in closed-loop operation. Limited by parameter 0x4A. 0 to value of 0x4A [phys. units. / s <sup>2</sup> ]
0xC	Closed-Loop Deceleration (Phys. Unit/s <sup>2</sup> )	Deceleration with dynamics profile in closed-loop operation. Limited by parameter 0x4B. 0 to value of 0x4B [phys. units. / s <sup>2</sup> ]

<b>0xE</b>	Numerator Of The Counts-Per-Physical-Unit Factor	<p>Numerator of the factor for converting impulses to physical units.</p> <p>The factor for the counts per physical unit of length specifies the unit of length for position queries and motion commands in closed-loop operation. The values of every parameter, whose unit is either the physical unit of length itself or a unit of measurement based on it, are automatically adapted to the set factor. The factor for the counts per physical unit of length has no impact on the stability of the servo loop but is used for the input and output scaling of position values.</p> <p>Decimal number</p>
<b>0xF</b>	Denominator Of The Counts-Per-Physical-Unit Factor	<p>Denominator of the factor for converting impulses to physical units.</p> <p>The factor for the counts per physical unit of length specifies the unit of length for position queries and motion commands in closed-loop operation. The values of every parameter, whose unit is either the physical unit of length itself or a unit of measurement based on it, are automatically adapted to the set factor. The factor for the counts per physical unit of length has no impact on the stability of the servo loop but is used for the input and output scaling of position values.</p> <p>Decimal number</p>
<b>0x10</b>	Output Mode	Present for compatibility reasons only.
<b>0x13</b>	Is Rotary Stage?	<p>Is this a rotation stage?</p> <p>Is not evaluated by the electronics but instead by the PC software.</p> <p>0 Not a rotation stage 1 Rotation stage</p>
<b>0x14</b>	Has Reference?	<p>Do the mechanics have a reference switch?</p> <p>Activates respectively deactivates reference moves to the installed reference switch.</p> <p>0 No reference switch 1 Reference switch available</p>
<b>0x15</b>	Maximum Travel In Positive Direction (Phys. Unit)	<p>Soft limit in positive direction, in relation to the zero position.</p> <p>The positive travel range limit cannot be used for reference moves if this value is smaller than the position value for the positive travel range limit (that results from the sum of parameters 0x16 and 0x2F).</p> <p>The value can be negative.</p>

<b>0x16</b>	Value At Reference Position (Phys. Unit)	<p>Position value at the reference switch. The current position is set to this value if the axis has performed a reference move to the reference switch.</p> <p>The parameter value is also used for calculating the position values set after reference moves to the limit switches; this also applies when the mechanics do not have a reference switch.</p>
<b>0x17</b>	Distance From Negative Limit To Reference Position (Phys. Unit)	<p>Gap between reference switch and negative travel range limit.</p> <p>The current position is set to the difference between the values of parameters 0x16 and 0x17 if the axis has done a reference move to the negative travel range limit.</p>
<b>0x18</b>	Limit Mode	<p>Signal logic of the limit switches.</p> <p>0 pos-HI, neg-HI 1 pos-LO, neg-HI 2 pos-HI, neg-LO 3 pos-LO, neg-LO</p>
<b>0x1A</b>	Has Brake?	<p>Do the mechanics have a brake?</p> <p>If there is a brake, servo mode and brake activation are coupled to each other.</p> <p>0 Brake not available 1 Brake available</p>
<b>0x2F</b>	Distance From Reference Position To Positive Limit (Phys. Unit)	<p>Gap between the reference switch and the positive travel range limit.</p> <p>If the axis has done a reference move to the positive travel range limit, the current position is set to the sum of the values of parameters 0x16 and 0x2F.</p>
<b>0x30</b>	Maximum Travel In Negative Direction (Phys. Unit)	<p>Soft limit in a negative direction, in relation to the zero position.</p> <p>The negative travel range limit cannot be used for reference moves if this value is greater than the position value for the negative travel range limit (that results from the difference between parameters 0x16 and 0x17).</p> <p>The value can be negative.</p>
<b>0x31</b>	Invert Reference?	<p>Should the reference signal be inverted?</p> <p>Inverts the signal of the reference switch or a digital input that is used instead of the reference switch.</p> <p>0 Reference signal not inverted 1 Reference signal inverted</p>
<b>0x32</b>	Has No Limit Switches?	<p>Do the mechanics not have limit switches?</p> <p>Activates motion stop at the installed limit switches.</p> <p>0 Mechanics have limit switches 1 Mechanics do not have limit switches</p>
<b>0x33</b>	Motor Offset Positive	<p>Drive offset for the positive direction of motion.</p> <p>0 to 32767</p>



0x34	Motor Offset Negative	Drive offset for the negative direction of motion. 0 to 32767
0x36	Settling Window (Encoder Counts)	Settling window around the target position. Presets the window limits (half of the window width). If the current position enters the settling window, the target position is considered as reached. If the current position exits the settling window, the target position is no longer considered as reached. Can be changed only if servo mode is switched off. 0 to maximum value. The maximum value depends on the controller but may not exceed 231 encoder counts.
0x3C	Stage Name	Positioner name. Default value: NOSTAGE String up to 20 characters
0x3F	Settling Time (s)	Delay time for setting the on-target state. Recommended values: 0.000 to 1.000 seconds
0x47	Reference Travel Direction	Default direction for the reference move. 0 Automatic detection 1 Negative direction 2 Positive direction
0x48	Motor Drive Offset	Velocity-dependent drive offset. Used if the commanded velocity does not equal zero (i.e., if the end of the dynamics profile has not been reached). 0 to 32767
0x49	Closed-Loop Velocity (Phys. Unit/s)	Velocity in closed-loop operation with dynamics profile. Limited by parameter 0xA. 0 to value of 0xA [phys. units / s]
0x4A	Maximum Closed- Loop Acceleration (Phys. Unit/s <sup>2</sup> )	Maximum acceleration in closed-loop operation with dynamics profile. Specifies the maximum value for parameter 0xB.
0x4B	Maximum Closed- Loop Deceleration (Phys. Unit/s <sup>2</sup> )	Maximum deceleration in closed-loop operation with dynamics profile. Specifies the maximum value for parameter 0xC.
0x50	Velocity For Reference Moves (Phys. Unit/s)	Maximum velocity for reference moves. 0 to value of value 0x49 [phys. units / s]
0x5A	Numerator Of The Servo-Loop Input Factor	Input factor numerator for the servo loop. Decimal number
0x5B	Denominator Of The Servo-Loop Input Factor	Input factor denominator for the servo loop. Decimal number
0x5C	Source Of Reference Signal	Reference signal source for axis motion to the reference switch.
0x5D	Source Of Negative Limit Signal	Reference signal source for axis motion to the negative travel range limit.

0x5E	Source Of Positive Limit Signal	Reference signal source for axis motion to the positive travel range limit.
0x5F	Invert Digital Input Used For Negative Limit	Inverts the polarity of the digital inputs that are used as the source of the negative limit switch signal.
0x60	Invert Digital Input Used For Positive Limit	Inverts the polarity of the digital inputs that are used as the source of the positive limit switch signal.
0x61	Invert Direction Of Motion For Joystick-Controlled Axis?	Inverts the direction of motion for HID-controlled axes. 0 Direction of motion not inverted (default) 1 Direction of motion inverted
0x63	Distance Between Limit And Hard Stop (Phys. Unit)	Gap between internal limit switch and hard stop. Determines the maximum stopping distance during reference moves. The actual velocities during a reference move are calculated on the basis of this value, the set deceleration (0xC) and set velocities (0x49 and 0x50).
0x70	Reference Signal Type	Reference signal type.
0x71	D-Term Delay (No. Of Servo Cycles)	D term delay. Determines how many values (i.e., servo cycles) are used for calculating the mean value of the D term. 0 to 15. The preset value of this parameter should not be changed.
0x72	Ignore Macro Error?	Ignore macro error? 0 Stop macro on error (default) 1 Ignore error
0x74	Closed-Loop Velocity For HI Control (Phys. Unit/s)	Maximum velocity during HID control.
0x75	Closed-Loop Acceleration for HI Control (Phys. Unit/s <sup>2</sup> )	Maximum acceleration during HID control.
0x76	Closed-Loop Deceleration for HI Control (Phys. Unit/s <sup>2</sup> )	Maximum deceleration during HID control.
0x77	Use Limit Switches Only For Reference Moves?	Should the limit switches only be used for reference moves? Is intended for use with rotation stages. Only evaluated when parameter 0x32 has the value 0.
0x78	Distance From Limit To Start Of Ref Search (Phys. Unit)	Distance between the limit switch and the starting position for the reference move to the index pulse. Used for FRF when parameter 0x70 has the value 2.
0x79	Distance For Reference Search (Phys. Unit)	Maximum distance for motion to the index pulse.

0x7C	Maximum Motor Output (V)	Maximum permissible voltage to be output to the motor Depends on the electronics
0x94	Notch Filter Frequency 1 (Hz)	Frequency of the first notch filter. The appropriate frequency component is reduced in the control value to compensate for undesired resonances in the mechanics. 40 to 20,000 [Hz]
0x95	Notch Filter Edge 1	Edge steepness of the first notch filter. The greater the value of this parameter, the narrower the notch filter bandwidth. 0.1 to 10 (dimensionless)
0x130	Inhibit Motion Commands	Configures the triggering of motion via motion commands. The setting is evaluated for closed-loop and open-loop operation. The value of the parameter has no influence on HID control of the C-884.xDC's axis (axes). Decimal number, bit-mapped. The values can be added.
0x3003320	Sensor Signal Type	Signal type output by the position sensor.
0x3003330	Abs Sensoroffset	Offset for correcting the signals of an absolute-measuring sensor.
0x7000000	Range Limit Min	Additional soft limit for the negative direction of motion (physical unit). If the current position reaches this value in either closed-loop or open-loop operation, the control value is set to zero and the motion is stopped as a result. The axis can move again as soon as the value for the soft limit has been decreased. The values are not loaded from the positioner database and are set in the default settings so that the limits are deactivated.
0x7000001	Range Limit Max	Additional soft limit for the positive direction of motion (physical unit). If the current position reaches this value in either closed-loop or open-loop operation, the control value is set to zero and the motion is stopped as a result. The axis can move again as soon as the value for the soft limit has been increased. The values are not loaded from the positioner database and are set in the default settings so that the limits are deactivated.
0x7000601	Axis Unit	Unit symbol. For example, the unit symbol is "mm", if the factor for the counts per physical unit of length is set with parameters 0xE and 0xF so that the encoder counts are converted into millimeters. The unit for rotation stages is normally "deg". String up to 20 characters
0xE000200	Servo Update Time	Servo cycle time.

<b>0xF000100</b>	Stage Type	Mechanics type. x-xxx default positioners x-xxxKxxx customized positioners
<b>0xF000200</b>	Stage Serial Number	Serial number of the mechanics. 9-digit number
<b>0xF000300</b>	Stage Assembly Date	Manufacturing date of the mechanics. Date in DDMMYY format
<b>0xF000400</b>	Stage HW Version	Version number of the mechanics hardware.
<b>0x16000001</b>	Recorded Points Per Trigger	Number of data points recorded per trigger impulse. 0 Unlimited (default) n>0 n data points are recorded Maximum value for n at C-884.xDC: 8192
<b>0x16000002</b>	Clearing Of RecTable On Trigger	Write mode during the recording. Determines how the data points are written to the data recorder tables when the recording is started by a trigger. 0 Recorded points are added to the contents of the data recorder table (default setting) 1 Trigger deletes the content of the data recorder tables, i.e., recording always starts at the first point
<b>0x16000003</b>	Data Recorder Buffer Mode	Behavior with full data recorder tables. 0 Recording ends (default setting) 1 Recording is continued with the first point in the data recorder table and overwrites existing content The value of parameter 0x16000004 is increased by 1 with each overflow.
<b>0x16000004</b>	Data Recorder Buffer Overflow	Buffer overflow counter of the data recorder. Counts how often recording starts with the first point in the data recorder table when parameter 0x16000003 has the value 1. Reading out with DRR? resets the parameter to zero. (write-protected)
<b>0x16000200</b>	Recorder Max Points	Maximum number of all points of the data recorder tables PC software commands (SPA, SEP) or operating elements Depends on the electronics
<b>0x22000020</b>	Maximum Buffer Size	Maximum number of trajectory points in memory. Trajectory points must be reloaded for processing trajectories that have more than this number of points.

Ox1 P Term

*Used in: Settings for the servo algorithm (34)*

Description:	Proportional constant of the PID servo algorithm. Is used for fast correction of the position error.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 65535

Ox2 I Term

*Used in: Settings for the servo algorithm (34)*

Description:	Integration constant of the PID servo algorithm. Used for reducing static position error.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 65535

## Ox3 D Term

*Used in: Settings for the servo algorithm (34)*

Description:	<p>Differential constant of the PID servo algorithm.</p> <p>Used for damping rapid control oscillation.</p> <p>The D term can be calculated as a floating average over several servo cycles. The parameter <a href="#">0x71 (p. 213)</a> (<b>D-Term Delay</b>) specifies how many values (i.e., servo cycles) are to be used for calculating the average.</p>
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	<p>0 to 65535</p> <p>The preset value of this parameter should not be changed.</p>

## Ox4 I Limit

*Used in: Settings for the servo algorithm (34)*

Description:	Limit of the integration constant.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 1048575

## Ox5 Kvff

*Used in: Settings for the servo algorithm (34)*

Description:	Feed-forward control of the commanded velocity. Used for minimizing positioning error.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 65535

## Ox8 Maximum Position Error (Phys. Unit)

*Used in: Behavior with Motion Errors (76)*

Description:	Maximum position error. A <a href="#">motion error (p. 76)</a> occurs when the difference between the current position and the commanded position exceeds this value and the movement of the axis is stopped.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

## Ox9 Maximum Motor Output

Description:	Maximum control value for driving an axis respectively a channel.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	See the response to HPA? for possible values

## OxA Maximum Closed-Loop Velocity (Phys. Unit/s)

*Used in: Variables for Generating a Dynamics Profile (30)*

**Description:** Maximum velocity with dynamics profile in closed-loop operation.  
Specifies the maximum value for parameter [0x49 \(p. 206\)](#).

**Data type** FLOAT

**Command level** 0

**Item type** Axis

**Source of data** Positioner database

## OxB Closed-Loop Acceleration (Phys. Unit/s<sup>2</sup>)

*Used in: Variables for Generating a Dynamics Profile (30)*

**Description:** Acceleration with dynamics profile in closed-loop operation.  
Limited by parameter [0x4A \(p. 206\)](#).

**Data type** FLOAT

**Command level** 0

**Item type** Axis

**Source of data** Positioner database

**Possible values** 0 to value of 0x4A [phys. units. / s<sup>2</sup>]

## OxC Closed-Loop Deceleration (Phys. Unit/s<sup>2</sup>)

*Used in: Reference Move Procedure (43), Variables for Generating a Dynamics Profile (30)*

**Description:** Deceleration with dynamics profile in closed-loop operation.  
Limited by parameter [0x4B \(p. 206\)](#).

**Data type** FLOAT

**Command level** 0

**Item type** Axis

**Source of data** Positioner database

**Possible values** 0 to value of 0x4B [phys. units. / s<sup>2</sup>]



OxE Numerator Of The Counts-Per-Physical-Unit Factor

Used in: Physical Units (27)

Description:	Numerator of the factor for converting impulses to physical units. The factor for the counts per physical unit of length specifies the unit of length for position queries and motion commands in closed-loop operation. The values of every parameter, whose unit is either the physical unit of length itself or a unit of measurement based on it, are automatically adapted to the set factor. The factor for the counts per physical unit of length has no impact on the stability of the servo loop but is used for the input and output scaling of position values.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	Decimal number

OxF Denominator Of The Counts-Per-Physical-Unit Factor

Used in: Physical Units (27)

Description:	Denominator of the factor for converting impulses to physical units. The factor for the counts per physical unit of length specifies the unit of length for position queries and motion commands in closed-loop operation. The values of every parameter, whose unit is either the physical unit of length itself or a unit of measurement based on it, are automatically adapted to the set factor. The factor for the counts per physical unit of length has no impact on the stability of the servo loop but is used for the input and output scaling of position values.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	Decimal number

## Ox10 Output Mode

Description:	Present for compatibility reasons only.
Data type	INT
Command level	0
Item type	Axis
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements

## Ox13 Is Rotary Stage?

Description:	Is this a rotation stage? Is not evaluated by the electronics but instead by the PC software.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Not a rotation stage 1 Rotation stage

## Ox14 Has Reference?

*Used in: Reference Switch Detection (39), Using Digital Input as Reference Signal (89)*

Description:	Do the mechanics have a reference switch? Activates respectively deactivates reference moves to the installed reference switch.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 No reference switch 1 Reference switch available

**Ox15 Maximum Travel In Positive Direction (Phys. Unit)**

*Used in: How it Works (92), Settings for Soft Limits (41)*

Description:	Soft limit in positive direction, in relation to the zero position. The positive travel range limit cannot be used for reference moves if this value is smaller than the position value for the positive travel range limit (that results from the sum of parameters <a href="#">Ox16 (p. 199)</a> and <a href="#">Ox2F (p. 201)</a> ).
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	The value can be negative.

**Ox16 Value At Reference Position (Phys. Unit)**

*Used in: Settings for the travel range (41)*

Description:	Position value at the reference switch. The current position is set to this value if the axis has performed a reference move to thereference switch. The parameter value is also used for calculating the position values set after reference moves to the limit switches; this also applies when the mechanics do not have a reference switch.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

**0x17 Distance From Negative Limit To Reference Position (Phys. Unit)**

*Used in: Settings for the travel range (41)*

Description:	Gap between reference switch and negative travel range limit. The current position is set to the difference between the values of parameters <a href="#">0x16 (p. 199)</a> and <a href="#">0x17 (p. 200)</a> if the axis has done a reference move to the negative travel range limit.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

**0x18 Limit Mode**

*Used in: Detecting Limit Switches (40)*

Description:	Signal logic of the limit switches.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 pos-HI, neg-HI 1 pos-LO, neg-HI 2 pos-HI, neg-LO 3 pos-LO, neg-LO

## Ox1A Has Brake?

Description:	Do the mechanics have a brake? If there is a brake, servo mode and brake activation are coupled to each other.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Brake not available 1 Brake available

## Ox2F Distance From Reference Position To Positive Limit (Phys. Unit)

*Used in: Settings for the travel range (41)*

Description:	Gap between the reference switch and the positive travel range limit. If the axis has done a reference move to the positive travel range limit, the current position is set to the sum of the values of parameters <a href="#">0x16 (p. 199)</a> and <a href="#">0x2F (p. 201)</a> .
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

## Ox30 Maximum Travel In Negative Direction (Phys. Unit)

*Used in: How it Works (92), Settings for Soft Limits (41)*

Description:	Soft limit in a negative direction, in relation to the zero position. The negative travel range limit cannot be used for reference moves if this value is greater than the position value for the negative travel range limit (that results from the difference between parameters <a href="#">0x16 (p. 199)</a> and <a href="#">0x17 (p. 200)</a> ).
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	The value can be negative.

## Ox31 Invert Reference?

*Used in: Reference Switch Detection (39), Using Digital Input as Reference Signal (89)*

Description:	Should the reference signal be inverted? Inverts the signal of the reference switch or a digital input that is used instead of the reference switch.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Reference signal not inverted 1 Reference signal inverted

## Ox32 Has No Limit Switches?

*Used in: Detecting Limit Switches (40), Using Digital Inputs as Source of the Limit Switch Signals (89)*

Description:	Do the mechanics not have limit switches? Activates motion stop at the installed limit switches.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Mechanics have limit switches 1 Mechanics do not have limit switches

## Ox33 Motor Offset Positive

*Used in: Settings for the servo algorithm (34)*

Description:	Drive offset for the positive direction of motion.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 32767

## Ox34 Motor Offset Negative

*Used in: Settings for the servo algorithm (34)*

Description:	Drive offset for the negative direction of motion.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 32767

Ox36 Settling Window (Encoder Counts)

Used in: On-Target State (38)

Description:	Settling window around the target position. Presets the window limits (half of the window width). If the current position enters the settling window, the target position is considered as reached. If the current position exits the settling window, the target position is no longer considered as reached. Can be changed only if servo mode is switched off.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to maximum value. The maximum value depends on the controller but may not exceed 2 <sup>31</sup> encoder counts.

Ox3C Stage Name

Used in: Commandable Items (25)

Description:	Positioner name. Default value: NOSTAGE
Data type	CHAR
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	String up to 20 characters



## Ox3F Settling Time (s)

*Used in: On-Target State (38)*

Description:	Delay time for setting the on-target state.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	Recommended values: 0.000 to 1.000 seconds

## Ox47 Reference Travel Direction

*Used in: Reference Move Procedure (43)*

Description:	Default direction for the reference move.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Automatic detection 1 Negative direction 2 Positive direction

## Ox48 Motor Drive Offset

*Used in: Settings for the servo algorithm (34)*

Description:	Velocity-dependent drive offset. Used if the commanded velocity does not equal zero (i.e., if the end of the dynamics profile has not been reached).
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 32767

## 0x49 Closed-Loop Velocity (Phys. Unit/s)

*Used in: Reference Move Procedure (43), Variables for Generating a Dynamics Profile (30)*

Description:	Velocity in closed-loop operation with dynamics profile. Limited by parameter <a href="#">0xA (p. 196)</a> .
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to value of <a href="#">0xA (p. 196)</a> [phys. units / s]

## 0x4A Maximum Closed-Loop Acceleration (Phys. Unit/s<sup>2</sup>)

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	Maximum acceleration in closed-loop operation with dynamics profile. Specifies the maximum value for parameter <a href="#">0xB (p. 196)</a> .
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

## 0x4B Maximum Closed-Loop Deceleration (Phys. Unit/s<sup>2</sup>)

*Used in: Variables for Generating a Dynamics Profile (30)*

Description:	Maximum deceleration in closed-loop operation with dynamics profile. Specifies the maximum value for parameter <a href="#">0xC (p. 196)</a> .
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

## Ox50 Velocity For Reference Moves (Phys. Unit/s)

*Used in: Reference Move Procedure (43)*

Description:	Maximum velocity for reference moves.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to value of value <a href="#">0x49 (p. 206)</a> [phys. units / s]

## Ox5A Numerator Of The Servo-Loop Input Factor

*Used in: Settings for the servo algorithm (34)*

Description:	Input factor numerator for the servo loop.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	Decimal number

## Ox5B Denominator Of The Servo-Loop Input Factor

*Used in: Settings for the servo algorithm (34)*

Description:	Input factor denominator for the servo loop.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	Decimal number

**Ox5C Source Of Reference Signal**

*Used in: Using Digital Input as Reference Signal (89)*

Description:	Reference signal source for axis motion to the reference switch.					
Data type	INT					
Command level	0					
Item type	Axis					
Source of data	Positioner database					
Possible values	<table><tr><td>0</td><td>Reference switch</td></tr><tr><td>&gt; 0: &lt;n&gt;</td><td>&lt;n&gt; = ID of the digital input line, bit-mapped</td></tr></table>		0	Reference switch	> 0: <n>	<n> = ID of the digital input line, bit-mapped
0	Reference switch					
> 0: <n>	<n> = ID of the digital input line, bit-mapped					

**Ox5D Source Of Negative Limit Signal**

*Used in: Using Digital Inputs as Source of the Limit Switch Signals (89)*

Description:	Reference signal source for axis motion to the negative travel range limit.					
Data type	INT					
Command level	0					
Item type	Axis					
Source of data	Positioner database					
Possible values	<table><tr><td>0</td><td>Negative limit switch</td></tr><tr><td>&gt; 0: &lt;n&gt;</td><td>&lt;n&gt; = ID of the digital input line, bit-mapped</td></tr></table>		0	Negative limit switch	> 0: <n>	<n> = ID of the digital input line, bit-mapped
0	Negative limit switch					
> 0: <n>	<n> = ID of the digital input line, bit-mapped					

**Ox5E Source Of Positive Limit Signal**

*Used in: Using Digital Inputs as Source of the Limit Switch Signals (89)*

Description:	Reference signal source for axis motion to the positive travel range limit.					
Data type	INT					
Command level	0					
Item type	Axis					
Source of data	Positioner database					
Possible values	<table><tr><td>0</td><td>Positive limit switch</td></tr><tr><td>&gt; 0: &lt;n&gt;</td><td>&lt;n&gt; = ID of the digital input line, bit-mapped</td></tr></table>		0	Positive limit switch	> 0: <n>	<n> = ID of the digital input line, bit-mapped
0	Positive limit switch					
> 0: <n>	<n> = ID of the digital input line, bit-mapped					

**Ox5F Invert Digital Input Used For Negative Limit**

*Used in: Using Digital Inputs as Source of the Limit Switch Signals (89)*

Description:	Inverts the polarity of the digital inputs that are used as the source of the negative limit switch signal.					
Data type	INT					
Command level	0					
Item type	Axis					
Source of data	Positioner database					
Possible values	<table><tr><td>0</td><td>Digital input not inverted</td></tr><tr><td>&gt; 0: &lt;n&gt;</td><td>&lt;n&gt; = ID of the digital input line, bit-mapped</td></tr></table>		0	Digital input not inverted	> 0: <n>	<n> = ID of the digital input line, bit-mapped
0	Digital input not inverted					
> 0: <n>	<n> = ID of the digital input line, bit-mapped					

**Ox60 Invert Digital Input Used For Positive Limit**

*Used in: Using Digital Inputs as Source of the Limit Switch Signals (89)*

Description:	Inverts the polarity of the digital inputs that are used as the source of the positive limit switch signal.					
Data type	INT					
Command level	0					
Item type	Axis					
Source of data	Positioner database					
Possible values	<table><tr><td>0</td><td>Digital input not inverted</td></tr><tr><td>&gt; 0: &lt;n&gt;</td><td>&lt;n&gt; = ID of the digital input line, bit-mapped</td></tr></table>		0	Digital input not inverted	> 0: <n>	<n> = ID of the digital input line, bit-mapped
0	Digital input not inverted					
> 0: <n>	<n> = ID of the digital input line, bit-mapped					

**Ox61 Invert Direction Of Motion For Joystick-Controlled Axis?**

*Used in: HID Control Configuration (92)*

Description:	Inverts the direction of motion for HID-controlled axes.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Direction of motion not inverted (default) 1 Direction of motion inverted



**Ox63 Distance Between Limit And Hard Stop (Phys. Unit)**

*Used in: Reference Move Procedure (43)*

Description:	Gap between internal limit switch and hard stop. Determines the maximum stopping distance during reference moves. The actual velocities during a reference move are calculated on the basis of this value, the set deceleration ( <a href="#">0xC (p. 196)</a> ) and set velocities ( <a href="#">0x49 (p. 206)</a> and <a href="#">0x50 (p. 207)</a> ).
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

Ox70 Reference Signal Type

Used in: Reference Move Procedure (43), Reference Switch Detection (39)

Description: Reference signal type.

Data type INT

Command level 0

Item type Axis

Source of data Positioner database

Possible values	
0	Direction-sensing reference switch The signal level changes when passing the reference switch.
1	Pulse signal (pulse width of several nanoseconds) Parameter <a href="#">0x47 (p. 205)</a> must be correctly set.
2	Index pulse The reference switch is approached via the negative travel range limit.
3	Index pulse The reference switch is approached via the positive travel range limit.
4	No reference signal Is only evaluated by the electronics that do not support the FNL and FPL commands.
5	The reference move is made to the negative limit switch or hard stop; this is set as reference position. Is only evaluated by the electronics that do not support the FNL and FPL commands.
6	The reference move is made to the positive limit switch or hard stop; this is set as reference position. Is only evaluated by the electronics that do not support the FNL and FPL commands.



## Ox71 D-Term Delay (No. Of Servo Cycles)

*Used in: Settings for the servo algorithm (34)*

Description:	D term delay. Determines how many values (i.e., servo cycles) are used for calculating the mean value of the D term.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 to 15. The preset value of this parameter should not be changed.

## Ox72 Ignore Macro Error?

*Used in: (101)*

Description:	Ignore macro error?
Data type	INT
Command level	0
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements
Possible values	0 Stop macro on error (default) 1 Ignore error

## Ox74 Closed-Loop Velocity For HI Control (Phys. Unit/s)

*Used in: HID Control Configuration (92), Variables for Generating a Dynamics Profile (30)*

Description:	Maximum velocity during HID control.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

## Ox75 Closed-Loop Acceleration For HI Control (Phys. Unit/s<sup>2</sup>)

*Used in: HID Control Configuration (92), Variables for Generating a Dynamics Profile (30)*

## Ox76 Closed-Loop Deceleration For HI Control (Phys. Unit/s<sup>2</sup>)

*Used in: HID Control Configuration (92), Variables for Generating a Dynamics Profile (30)*

## Ox77 Use Limit Switches Only For Reference Moves?

*Used in: Detecting Limit Switches (40)*

**Description:** Should the limit switches only be used for reference moves?  
Is intended for use with rotation stages.  
Only evaluated when parameter [0x32 \(p. 203\)](#) has the value 0.

**Data type** INT

**Command level** 0

**Item type** Axis

**Source of data** Positioner database

**Possible values**

- |   |  |
|---|--|
| 0 | Use limit switches for stopping at the end of the travel range and for reference moves (default) |
| 1 | Use limit switches for reference moves only  |

## Ox78 Distance From Limit To Start Of Ref Search (Phys. Unit)

*Used in: Reference Move Procedure (43)*

**Description:** Distance between the limit switch and the starting position for the reference move to the index pulse.  
Used for [FRF \(p. 142\)](#) when parameter [0x70 \(p. 212\)](#) has the value 2.

**Data type** FLOAT

**Command level** 0

**Item type** Axis

**Source of data** Positioner database

## Ox79 Distance For Reference Search (Phys. Unit)

*Used in: Reference Move Procedure (43)*

Description:	Maximum distance for motion to the index pulse.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database

## Ox7C Maximum Motor Output (V)

Description:	Maximum permissible voltage to be output to the motor
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	Depends on the electronics

## Ox94 Notch Filter Frequency 1 (Hz)

*Used in: Settings for the Notch Filter (34)*

Description:	Frequency of the first notch filter. The appropriate frequency component is reduced in the control value to compensate for undesired resonances in the mechanics.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	40 to 20,000 [Hz]

Ox95 Notch Filter Edge 1

*Used in: Settings for the Notch Filter (34)*

Description:	Edge steepness of the first notch filter. The greater the value of this parameter, the narrower the notch filter bandwidth.
Data type	FLOAT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0.1 to 10 (dimensionless)

Ox130 Inhibit Motion Commands

*Used in: Triggering Motion (36)*

Description:	Configures the triggering of motion via motion commands. The setting is evaluated for closed-loop and open-loop operation. The value of the parameter has no influence on HID control of the C-884.xDC's axis (axes).
Data type	INT
Command level	0
Item type	Axis
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements
Possible values	Decimal number, bit-mapped. The values can be added.

0	All motion commands are executed (default setting).
1	Motion commands from controller macros are not permitted.
2	Motion commands sent via RS-232 are not permitted.
4	Motion commands sent via USB are not permitted.
8	Motion commands sent via TCP/IP are not permitted.
16	Motion commands sent via SPI are not permitted.

## Ox3003320 Sensor Signal Type

Description: Signal type output by the position sensor.

Data type INT

Command level 1

Item type Axis

Source of data ID chip of the mechanics

Possible values

0	No sensor
1	Incremental sensor, AB quadrature signals
2	Incremental sensor, sin/cos signals
3	Incremental sensor, BiSS (32 bit)
4	Absolute-measuring sensor, BiSS (32 bit)
5	Incremental sensor, BiSS (24 bit) and absolute-measuring sensor, BiSS (32 bit)
6	Absolute-measuring sensor, BiSS (32 bit) and incremental sensor, BiSS (24 bit)
7	Incremental sensor, BiSS (24 bit)
8	Absolute-measuring sensor, BiSS (24 bit)

## Ox3003330 Abs Sensoroffset

Description: Offset for correcting the signals of an absolute-measuring sensor.

Data type FLOAT

Command level 1

Item type Axis

Source of data ID chip of the mechanics



**0x7000000 Range Limit Min**

*Used in: Settings for Soft Limits (41)*

Description:	Additional soft limit for the negative direction of motion (physical unit).  If the current position reaches this value in either closed-loop or open-loop operation, the control value is set to zero and the motion is stopped as a result. The axis can move again as soon as the value for the soft limit has been decreased.
Data type	FLOAT
Command level	0
Item type	Axis
Possible values	The values are not loaded from the positioner database and are set in the default settings so that the limits are deactivated.

**0x7000001 Range Limit Max**

*Used in: Settings for Soft Limits (41)*

Description:	Additional soft limit for the positive direction of motion (physical unit).  If the current position reaches this value in either closed-loop or open-loop operation, the control value is set to zero and the motion is stopped as a result. The axis can move again as soon as the value for the soft limit has been increased.
Data type	FLOAT
Command level	0
Item type	Axis
Possible values	The values are not loaded from the positioner database and are set in the default settings so that the limits are deactivated.

## Ox7000601 Axis Unit

*Used in: Physical Units (27)*

Description:	Unit symbol. For example, the unit symbol is "mm", if the factor for the counts per physical unit of length is set with parameters <a href="#">0xE (p. 197)</a> and <a href="#">0xF (p. 197)</a> so that the encoder counts are converted into millimeters. The unit for rotation stages is normally "deg".
Data type	CHAR
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	String up to 20 characters

## OxE000200 Servo Update Time

Description:	Servo cycle time.
Data type	FLOAT
Command level	2
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements

## OxF000100 Stage Type

Description:	Mechanics type.
Data type	CHAR
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics
Possible values	x-xxx default positioners x-xxxKxxx customized positioners

**OxF000200 Stage Serial Number**

Description:	Serial number of the mechanics.
Data type	CHAR
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics
Possible values	9-digit number

**OxF000300 Stage Assembly Date**

Description:	Manufacturing date of the mechanics.
Data type	CHAR
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics
Possible values	Date in DDMMYY format

**OxF000400 Stage HW Version**

Description:	Version number of the mechanics hardware.
Data type	INT
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics



**0x16000001 Recorded Points Per Trigger**

*Used in: Configuring data processing (81)*

Description:	Number of data points recorded per trigger impulse.
Data type	INT
Command level	0
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements
Possible values	0 Unlimited (default) n>0 n data points are recorded Maximum value for n at C-884.xDC: 8192

**0x16000002 Clearing Of RecTable On Trigger**

*Used in: Configuring data processing (81)*

Description:	Write mode during the recording. Determines how the data points are written to the data recorder tables when the recording is started by a trigger.
Data type	INT
Command level	0
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements
Possible values	0 Recorded points are added to the contents of the data recorder table (default setting) 1 Trigger deletes the content of the data recorder tables, i.e., recording always starts at the first point

**0x16000003 Data Recorder Buffer Mode**

*Used in: Configuring data processing (81)*

Description:	Behavior with full data recorder tables.
Data type	INT
Command level	0
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements
Possible values	0 Recording ends (default setting) 1 Recording is continued with the first point in the data recorder table and overwrites existing content The value of parameter <a href="#">0x16000004 (p. 222)</a> is increased by 1 with each overflow.

**0x16000004 Data Recorder Buffer Overflow**

*Used in: Configuring data processing (81)*

Description:	Buffer overflow counter of the data recorder. Counts how often recording starts with the first point in the data recorder table when parameter <a href="#">0x16000003 (p. 220)</a> has the value 1. Reading out with <a href="#">DRR? (p. 138)</a> resets the parameter to zero.
Data type	INT
Command level	0
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements
Possible values	(write-protected)



**0x16000200 Recorder Max Points**

Description:	Maximum number of all points of the data recorder tables
Data type	INT
Command level	1
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements
Possible values	Depends on the electronics

**0x22000020 Maximum Buffer Size**

*Used in: Working with Trajectories (78)*

Description:	Maximum number of trajectory points in memory. Trajectory points must be reloaded for processing trajectories that have more than this number of points.
Data type	INT
Command level	2
Item type	System
Source of data	PC software commands ( <a href="#">SPA (p. 173)</a> , <a href="#">SEP (p. 171)</a> ) or operating elements

# 11 Maintenance

## 11.1 Cleaning

### Requirements

- ✓ You have disconnected the C-884.xDC from the power supply.

### Auxiliary Materials Required

- Soft, lint-free cloth
- Mild cleaning agent or disinfectant

If you have any questions on the auxiliary materials recommended for the C-884.xDC, contact our [customer service department \(p. 10\)](#).

### NOTICE



#### Short circuits or flashovers!

The C-884.xDC contains electrostatically sensitive devices that can be damaged by short circuits or flashovers when cleaning fluids penetrate the housing.

- ▶ Before cleaning, disconnect the C-884.xDC from the power supply.
- ▶ Prevent cleaning fluid from penetrating the case.

### Cleaning the C-884.xDC

1. Dampen the cloth with the cleaning agent or disinfectant.
2. Carefully wipe the surfaces of the C-884.xDC.

## 11.2 Updating the Firmware

To update the firmware, the C-884.xDC and PC must communicate via the TCP/IP interface.

When updating the firmware, the macros saved on the C-884.xDC are retained.

The firmware of the C-884.xDC consists of several components that can be updated separately.

### Requirements

- ✓ The C-884.xDC is connected via TCP to the PC either [directly \(p. 56\)](#) or via the [network \(p. 56\)](#).
- ✓ "PIFirmwareManager" is [installed on the PC \(p. 49\)](#).
- ✓ You have copied the new firmware file, which you have received from our customer service department, to a directory on the PC.
- ✓ You have read and understood the documentation that you received from our customer service department together with the new firmware. You have learned from the documentation whether new parameters are introduced with the firmware update or the memory management of the C-884.xDC changes.
- ✓ You have saved the parameter values of the C-884.xDC [in a text file on the PC \(p. 73\)](#).
- ✓ You have saved the controller macros of the C-884.xDC [in files on the PC \(p. 74\)](#).
- ✓ You have [established communication between the C-884.xDC and the PC with PIMikroMove® or PITerminal \(p. 57\)](#).

**NOTICE****Malfunction due to faulty firmware update!**

A faulty or incomplete update of the C-884.xDC's firmware may mean that the C-884.xDC can only be made operational again by the PI customer service department.

- ▶ Update the C-884.xDC's firmware only after consulting our customer service department. If possible, ask our customer service department to do the firmware update for you.
- ▶ Before updating the firmware, make sure that you have received a suitable firmware from our customer service department and have stored it at a location that is accessible to the update program.

**Updating the Firmware of the C-884.xDC**

1. Start the **PIFirmwareManager** program on the PC and update the controller firmware. Proceed as described in the user manual [SM164E \(p. 8\)](#).

**Restarting the C-884.xDC**

1. Switch off the C-884.xDC.
2. Switch the C-884.xDC on again.
  - *If the firmware update was successful, the C-884.xDC exits the firmware update mode and indicates that it is in operating mode again.*
  - *If new parameters were added during updating of the firmware or the memory management of the C-884.xDC was changed: Initialize the C-884.xDC.*

**Initializing the C-884.xDC after a Firmware Update**

1. Make sure that the current parameter values and controller macros of the C-884.xDC have been saved on the PC.  
The initialization of the C-884.xDC resets all parameters to their factory settings and deletes all controller macros. Consequently, parameter values and controller macros that are not saved are lost during the initialization process.
2. On the PC, start PITerminal or PIMikroMove, connect to the C-884.xDC, and, if necessary, open the window to send commands.  
Initialize the C-884.xDC by sending the following commands successively:  

```
ZZZ 100 parameter
ZZZ 100 macros
```

  - *After successful initialization, the controller issues a corresponding message.*
3. Adapt the parameter values of the C-884.xDC:
  - Reset the parameters that were already present prior to the firmware update to the saved values from the text file.
  - Set the parameters that were introduced with the firmware update to the appropriate values.
4. If you have saved controller macros on the PC: Load the controller macros into the C-884.xDC, see "[Loading Controller Macros from the PC into the C-884.xDC \(p. 75\)](#)".

## 12 Troubleshooting

The positioner does not move	
Cable not connected correctly	► Check the cable connections.
Unsuitable positioner cable used. Interference with the signal transmission between the positioner and C-884.xDC can occur when an unsuitable cable is used.	► Only use genuine PI parts when connecting the positioner to the C-884.xDC. ► If you need extension cables, contact our <a href="#">customer service department (p. 10)</a> .
Positioner or drive cable is defective	► If available, replace the defective positioner with another one and test the new combination. ► If available, replace the defective drive cable with another one and test the new combination.
Positioner was connected to the switched-on C-884.xDC The sensor electronics in the positioner was not initialized and the sensor's ID chip was not read out.	► Switch the C-884.xDC off and on again, or reboot the C-884.xDC with the <b>RBT</b> command or with the corresponding functions of the PC software.
Limit switch signal logic wrongly set In order for the positioner to be able to move, the settings of the C-884.xDC must correspond to the limit switch logic level of the positioner.	► Adjust the <b>Limit Mode (p. 200)</b> parameter (0x18) accordingly.
Limit switch signals not compatible with the C-884.xDC It is possible that positioners from third-party suppliers use unsuitable limit switch signals.	► Contact the customer service department or the manufacturer of the positioner.
Motion platform has triggered the limit switch	► Switch the servo mode on for the affected axis again. ► Command the axis to move away from the limit switch.
Incorrect axis or channel commanded	► Make sure that the correct axis or channel identifier is used and that the positioner is connected correctly.
Incorrect configuration	► Check the parameter settings of the C-884.xDC with the <b>SPA?</b> (p. 173) (volatile memory) and the <b>SEP?</b> (p. 172) commands (nonvolatile memory), and make the necessary corrections, refer to <a href="#">adapting settings (p. 111)</a> .
Incorrect command or incorrect syntax	► Send the <b>ERR?</b> command and check the error code that is returned.
HID control active Motion commands are not permitted when HID control is activated for the axis or the channel.	► Deactivate <a href="#">HID control (p. 92)</a> .
When following a trajectory: Trajectory buffer empty	► Make sure that a sufficient number of trajectory points is always in the buffer, refer to <a href="#">Trajectories for Motion Paths (p. 77)</a> .

At least 4 points must be loaded into the trajectory buffer with TGA before a trajectory can be followed. The buffer must be refilled fast enough when a trajectory is being followed. Trajectory following must be completed with TGF.

#### Positioner performs unintentional motion

- |  |   |
|--|---|
| Control device is not connected, but HID control is activated in the C-884.xDC   | ▶ Activate <a href="#">HID control (p. 92)</a> only when a control device is actually connected to the C-884.xDC.   |
| HID axis is not calibrated   | ▶ Calibrate the axis of the <a href="#">control device (p. 95)</a> .  |
| Startup macro is run   | ▶ Check whether a macro is specified as the <a href="#">start-up macro (p. 103)</a> and cancel selection of the startup macro if necessary.   |
| Positioner brake is deactivated  | ▶ Switch off the servo mode and activate the brake with the <a href="#">BRA (p. 128)</a> command.<br>▶ Secure the positioner against moving unintentionally before you deactivate the brake by command! |
| Wrong trajectory<br>The trajectory IDs are permanently assigned to the C-884.xDC's axes: Trajectory 1 to axis 1, trajectory 2 to axis 2 etc. | ▶ Start the desired trajectory for the desired axis.  |

#### Stage is oscillating or positions inaccurately

- |  |   |
|--|---|
| The load was changed   | ▶ Reset the <a href="#">notch filter (p. 66)</a> and the <a href="#">servo control parameters (p. 70)</a> according to the change of load.  |
| When following a trajectory: Unsuitable trajectory design<br>Following an unsuitable trajectory can cause the positioner to oscillate or stop motion abruptly. Oscillation or stopping abruptly can damage the positioner and/or the load fixed to it. | ▶ Make sure that the path specified by the trajectory points is continuously differentiable at least twice.<br>▶ Design the trajectory (trajectory points and timing) so that the maximum permissible velocity and acceleration of the axis is not exceeded, refer to <a href="#">Trajectories for Motion Paths (p. 77)</a> . |

#### Stage is oscillating already during the reference move

- |                                  |  |
|----------------------------------|--|
| Very high load on the positioner | ▶ Stop the reference move immediately.<br>▶ Switch servo mode off.<br>▶ Enter suitable values for setting the <a href="#">notch filter (p. 66)</a> .<br>▶ Restart the reference move.<br>▶ If the positioner continues to oscillate, repeat the specified steps until the reference move has completed successfully without oscillation. |
|----------------------------------|--|

#### Communication between the C-884.xDC and the PC not functioning

- |                                |   |
|--------------------------------|---|
| Wrong communication cable used | ▶ Use a straight-through cable for TCP/IP connections to a network.<br>▶ Use a crossover network cable for TCP/IP connections direct to the PC.<br>▶ Use a null-modem cable for RS-232 connections. |
|--------------------------------|---|

	<ul style="list-style-type: none"> <li>▶ Use a standard-compliant USB cable (type A to type B or type A to type Mini-B) for USB connections</li> </ul>
Communication cable defective	<ul style="list-style-type: none"> <li>▶ Replace the communication cable.</li> </ul>
RS-232 interface not configured correctly	<ul style="list-style-type: none"> <li>▶ Check the port settings, the baud rate and the handshake setting of the PC.</li> </ul>
TCP/IP interface not configured correctly	<ul style="list-style-type: none"> <li>▶ Connect the controller to the network <b>before</b> you switch it on. Restart the C-884.xDC if necessary.</li> <li>▶ Check the network settings.</li> <li>▶ Check the firewall settings to ensure that UDP broadcast is permitted. If necessary, contact your network administrator. Alternatively, enter the IP address of the controller manually.</li> <li>▶ Make sure that the network is not blocked for unknown devices.</li> <li>▶ Make sure that you have selected the correct C-884.xDC when establishing communication.</li> <li>▶ If you cannot solve the problems, consult your network administrator if necessary.</li> </ul>
Another program is accessing the interface	<ul style="list-style-type: none"> <li>▶ Close the other program.</li> </ul>
The start procedure of the C-884.xDC's firmware has not finished yet	<ul style="list-style-type: none"> <li>▶ Wait until the corresponding LED indicates operational readiness after switching on or rebooting the C-884.xDC.</li> <li>▶ Try to establish communication.</li> </ul>
Problems with special software	<ul style="list-style-type: none"> <li>▶ Check whether the system works with other software, e.g., a terminal program, or a development environment; for this purpose, enter the <code>*IDN?</code> or <code>HLP?</code> command using that software. Make sure that you end commands with an LF (line feed); a command is only executed when an LF is received.</li> </ul>

#### C-884.xDC does not send an error code in the case of incorrect system behavior

The error code was already queried by a different instance

In the case of simultaneous access to the C-884.xDC by several instances, the error code is only returned to the first instance that sent the `ERR?` command. The error code is reset to 0 during the query.

- ▶ If possible, access the C-884.xDC with one instance only.
- ▶ Check whether the error code is queried regularly in the background by a macro or script or PC software (e.g., PIMikroMove).

#### LEDs do not light up even though the C-884.xDC is switched on

C-884.xDC not connected to the power supply or the power cord is defective.

- ▶ Switch off the C-884.xDC.
- ▶ Make sure that the C-884.xDC is connected to the power supply and the power cord is not defective.
- ▶ Switch on the C-884.xDC.

If the problem is not listed in the table or cannot be solved as described, contact our [customer service department \(p. 10\)](#).



## 13 Technical Data

Subject to change. You can find the latest product specifications on the product web page at [www.pi.ws](http://www.pi.ws).

### 13.1 Specifications

C-884.4DC / C-884.6DC	
Function	Position control for closed-loop DC motors
Processor	Dual-core architecture. Controller on a DSP core, with extendable command interpreter in an ARM core under Linux
Axes	4 / 6
Supported functions	Linear vector motion. Point-to-point motion. User-definable trajectories. Startup macro. PI Python. Data recorder for recording operating data such as motor voltage, velocity, position or position error. ID chip detection.
Motion and control	
Controller type	PID controller, parameters modifiable during operation
Servo cycle time	100 µs
Profile Generator	Trapezoidal velocity profile
Encoder input	A/B quadrature (TTL differential according to RS-422), 50 MHz; BiSS interface
Stall detection	Automatic motor stop when a programmable position error is exceeded
Input limit switch	2 × TTL per axis (programmable polarity)
Input reference switch	1 × TTL per axis
Motor brake	1 × TTL per axis, can be switched by software
Electrical properties	
Max. output voltage*	24 V
Max. output power	240 W
Current Limitation	2.5 A per axis

\* Depending on the power adapter used

Interfaces and operation		C-884.4DC / C-884.6DC
Communication interfaces	TCP/IP: RJ45/Ethernet; USB: Mini USB type B; RS-232: D-sub 9 (m); SPI: DisplayPort	
Motor connector	D-sub 15 (f)	
I/O lines	4 analog inputs (-10 to 10 V), resolution: 10-bit 4 digital inputs (5 V TTL) 4 digital outputs (5 V TTL)	
Command set	PI General Command Set (GCS)	
User software	PIMikroMove	
Application programming interfaces	C, C++, C#, MATLAB, NI LabVIEW, Python	
Manual control	USB interface for HID-compliant devices	
Miscellaneous		C-884.4DC / C-884.6DC
Operating voltage	External power adapter 24 V / 5 A (120 W) included in the scope of delivery	
Max. current consumption	11 A / 16 A	
Current consumption without load	500 mA	
Operating temperature range	0 to 50 °C	
Mass	1.77 kg / 1.97 kg	
Dimensions	312 mm × 153.4 mm × 59.2 mm (incl. mounting rails)	

## 13.2 Maximum Ratings

The C-884.xDC is designed for the following operating data:

Maximum operating voltage	Operating frequency	Maximum current consumption
24 V DC	==	C-884.4DC: 11 A / C-884.6DC: 16 A

## 13.3 Ambient Conditions and Classifications

The following ambient conditions and classifications for the C-884.xDC must be observed:

Area of application	For indoor use only
Maximum altitude	2000 m above msl
Air pressure	1100 hPa to 0.1 hPa
Relative humidity	Max. 80 % for temperatures to 31 °C, decreasing linearly to 50 % at 40 °C
Storage temperature	0 °C to 70 °C
Transport temperature	-25 °C to +85 °C
Overvoltage category	II
Protection class	I
Degree of pollution	2
Degree of protection according to IEC 60529	IP20

# 13.4 Dimensions

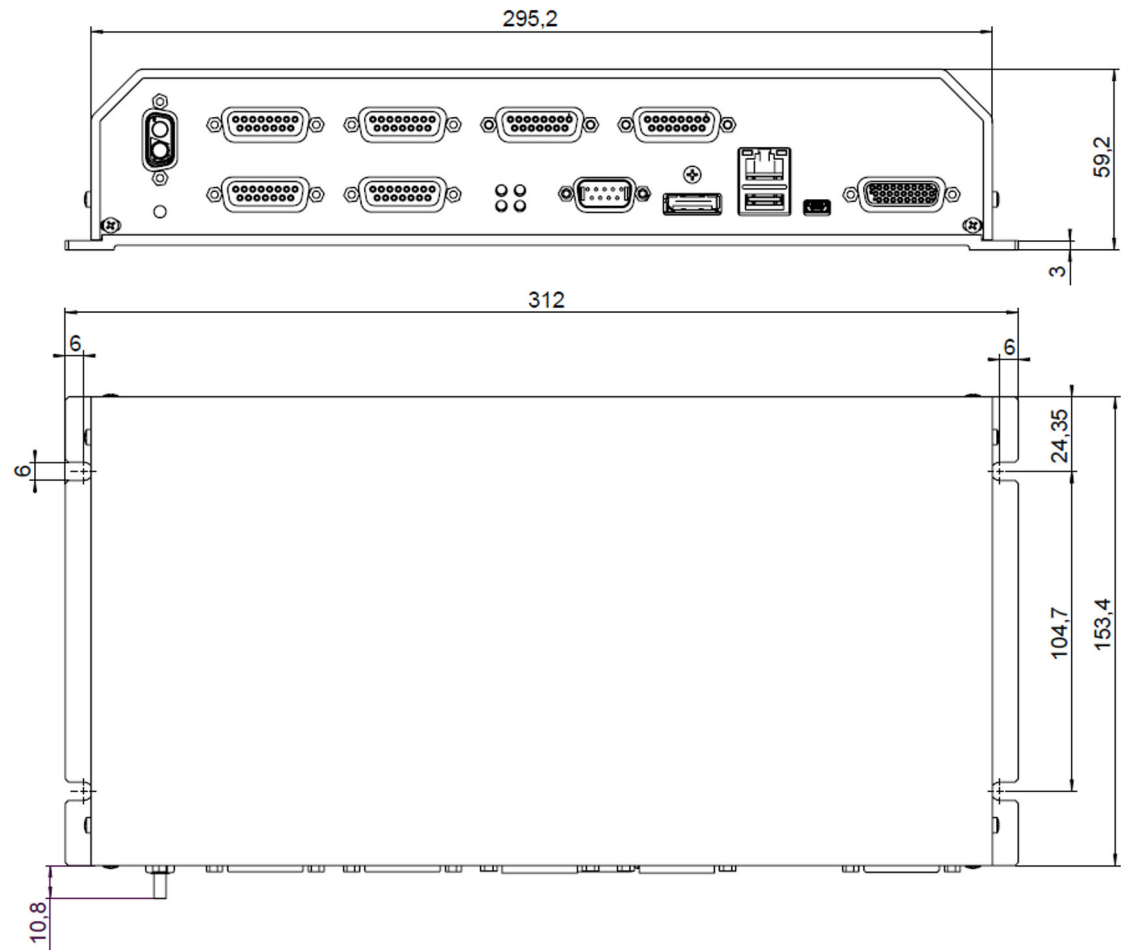


Figure 14: Dimensions of the C-884.xDC

Dimensions in mm. Note that the decimal places are separated by a comma in the drawings.

# 14 Appendix

## 14.1 Pin Assignment

### 14.1.1 Axis Connector

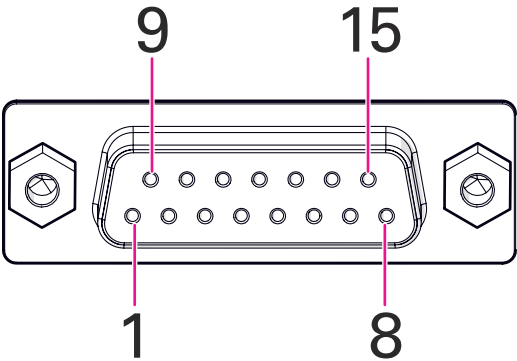


Figure 15: D-Sub 15 (f)

Pin	Signal	Function
1	Output	Motor brake (0 or + 5 V)
2	Output	Motor + (differential; power PWM); for positioners without PWM amplifier
3	Output	PWM magnitude (TTL); for positioners with PWM amplifier
4	Output	+5 V
5	Input	Positive limit switch
6	Bidirectional	Data line for ID chip
7	Input	Encoder: A/B: A inverted, RS-422; BiSS: MA-
8	Input	Encoder: A/B: B inverted, RS-422; BiSS: SL-
9	Output	Motor + (differential; power PWM); for positioners without PWM amplifier
10	GND	Ground
11	Output	PWM sign (TTL); for positioners with PWM amplifier
12	Input	Negative limit switch
13	Input	Reference switch
14	Input	Encoder: A/B: A, RS-422; BiSS: MA+
15	Input	Encoder: A/B: B, RS-422; BiSS: SL+

14.1.2 I/O Connector

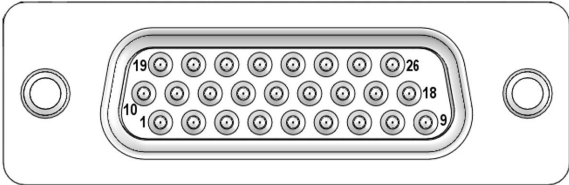


Figure 16: HD D-sub 26 (f)

Pin	Function
1	Analog input 2 (-10 V to 10 V)
2	GND (analog)
3	Reserved
4	Reserved
5	Reserved
6	Vcc (+5 V)
7	Digital input 3 (TTL)
8	Digital output 4 (TTL)
9	Digital output 1 (TTL)
10	Analog input 1 (-10 V to 10 V)
11	Analog input 4 (-10 V to 10 V)
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Digital input 4 (TTL)
17	Digital input 1 (TTL)
18	Digital output 2 (TTL)
19	Analog input 3 (-10 V to 10 V)
20	GND
21	Reserved
22	GND
23	Reserved
24	GND
25	Digital input 2 (TTL)
26	Digital output 3 (TTL)

14.1.3 RS-232 Connector

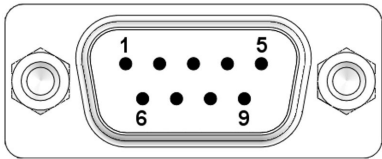


Figure 17: D-sub 9 (m)

Pin	Function
1	Not connected
2	RxD (PC to controller)
3	TxD (controller to PC)
4	Not connected
5	GND
6	Not connected
7	Not connected
8	Not connected
9	Not connected

14.1.4 Power Adapter Connector



Figure 18: D-sub 2W2C

Pin	Function
Socket contact (left)	GND (power)
Pin contact (right)	Input: 24 V DC

14.2 GCS Error Codes

The error codes listed here are those of the PI General Command Set. As such, some may be not relevant to your controller and will simply never occur.

Controller errors		
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

Controller errors		
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_LIMIT_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



Controller errors		
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_RECORD_TABLE	Data Record Table does not exist

Controller errors		
58	PI_CNTR_INVALID_RECORD- ER_SRC_OPT	Source does not exist; number too low or too high
59	PI_CNTR_INVALID_RECORD- ER_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_PARAME- TER	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_M EMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DE- LAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_AR- RAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN SIN- GLE_MODE_RESTART	Attempt to restart the generator while it is running in single step mode
72	PI_CNTR_ANALOG_TARGET_AC- TIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERA- TOR_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).

Controller errors		
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.

Controller errors		
94	PI_CNTR_OPEN_LOOP_JOY- STICK_ENABLED	Servo cannot be switched on when open-loop joy- stick control is activated.
95	PI_CNTR_INVALID_SER- VO_STATE_FOR_PARAMETER	This parameter cannot be changed in current ser- vo mode.
96	PI_CNTR_UN- KNOWN_STAGE_NAME	Unknown stage name
97	PI_CNTR_INVALID_VAL- UE_LENGTH	Invalid length of value (too much characters)
98	PI_CNTR_AUTOZERO_FAILED	AutoZero procedure was not successful
99	PI_CNTR_SENSOR_VOLT- AGE_OFF	Sensor voltage is off
100	PI_LABVIEW_ERROR	PI driver for use with NI LabVIEW reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PAR- AM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PAR- AM_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_INCOM- PLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTH- ING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLE- GAL_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

Controller errors		
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

Controller errors		
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

Controller errors		
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

Controller errors		
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



Controller errors		
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_PROTECT_TRIGGERED_BY_I2T	Overcurrent protection triggered by I2T-module
656	PI_CNTR_OVER_CURR_PROTECT_TRIGGERED_BY_AMP_MODULE	Overcurrent protection triggered by amplifier module
657	PI_CNTR_SAFETY_STOP_TRIGGERED	Safety stop triggered
658	PI_SENSOR_OFF	Sensor off?
659	PI_CNTR_PARAM_CONFLICT	Parameter could not be set. Conflict with another parameter.

Controller errors		
700	PI_CNTR_COMMAND_NOT_ALLOWED_IN_EXTERNAL_MODE	Command not allowed in external mode
710	PI_CNTR_EXTERNAL_MODE_ERROR	External mode communication error
715	PI_CNTR_INVALID_MODE_OF_OPERATION	Invalid mode of operation
716	PI_CNTR_FIRMWARE_STOPPED_BY_CMD	Firmware stopped by command (#27)
717	PI_CNTR_EXTERNAL_MODE_DRIVER_MISSING	External mode driver missing
718	PI_CNTR_CONFIGURATION_FAILURE_EXTERNAL_MODE	Missing or incorrect configuration of external mode
719	PI_CNTR_EXTERNAL_MODE_CYCLETIME_INVALID	External mode cycletime invalid
720	PI_CNTR_BRAKE_ACTIVATED	Brake is activated
725	PI_CNTR_DRIVE_STATE_TRANSITION_ERROR	Drive state transition error
731	PI_CNTR_SURFACEDETECTION_RUNNING	Command not allowed while surface detection is running
732	PI_CNTR_SURFACEDETECTION_FAILED	Last surface detection failed
733	PI_CNTR_FIELDBUS_IS_ACTIVE	Fieldbus is active and is blocking GCS control commands
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004

Controller errors		
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
1006	PI_CNTR_INVALID_IDENTIFIER	Invalid identifier (invalid special characters, ...)
1007	PI_CNTR_UNKNOWN_VARIABLE_OR_ARGUMENT	Variable or argument not defined
1008	PI_CNTR_RUNNING_MACRO	Controller is (already) running a macro
1009	PI_CNTR_MACRO_INVALID_OPERATOR	Invalid or missing operator for condition. Check necessary spaces around operator.
1010	PI_CNTR_MACRO_NO_ANSWER	No response was received while executing WAC/MEX/JRC/...
1011	PI_CMD_NOT_VALID_IN_MACRO_MODE	Command not valid during macro execution
1012	PI_CNTR_ERROR_IN_MACRO	Error occurred 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

Controller errors		
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 cannot be defined as PPC axis
5003	PI_CNTR_PCC_SCAN_OUT_OF_RANGE	Defined scan area is larger than the travel range
5004	PI_CNTR_PCC_TYPE_NOT_EXISTING	Given PicoCompensation type is not defined
5005	PI_CNTR_PCC_PAM_ERROR	PicoCompensation parameter error
5006	PI_CNTR_PCC_TABLE_ARRAY_TOO_LARGE	PicoCompensation table is larger than maximum table length
5100	PI_CNTR_NEXLINE_ERROR	Common error in NEXLINE® firmware module

Controller errors		
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

Interface errors		
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

Interface errors		
-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

Interface errors		
-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



Interface errors		
-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 errors		
-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

DLL errors		
-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

DLL errors		
-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

DLL errors		
-1057	PI_WAVE_EDITOR_WRONG_PARAMETERS	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_PARAMETER_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

DLL errors		
-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 <a href="http://www.pi.ws">www.pi.ws</a> .
-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

DLL errors		
-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.

DLL errors		
-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_CONTAINS_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_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.





DLL errors		
-1001 2	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.
-1001 3	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.
-1001 4	PI_PARAMETER_DB_FAILED_TO_SET_PARAMETERS_CORRECTLY	One or more parameters could not be set correctly on the controller.
-1001 5	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.

## Glossary

### **Daisy chain**

Wiring diagram by which one controller is connected to the next in sequence (series connection principle). The first controller is connected directly to the PC. The additional controllers are always connected to the ones that precede them so that a chain is formed. The signal to and from a controller goes to the PC via the previous controllers.

### **Dynamics profile**

The dynamic profile includes the target position, speed, and acceleration of the axis calculated by the profile generator of the electronics for each point in time of motion. The calculated values are called "commanded values".

### **GCS**

Abbreviation for "General Command Set", the command set for electronics from PI. Piezo and servo controllers can be operated together with minimal programming effort thanks to GCS.

### **HID**

Abbreviation for "Human Interface Device". HID refers to an input or output device that is connected to the electronics and intended for manual operation. Depending on the electronics, the connection can be made via USB, analog or digital interfaces. Joysticks and gamepads are typical HIDs.

### **Nonvolatile memory**

Electronics read-only memory. The default values for the parameters are loaded from the nonvolatile memory into the volatile memory when the electronics are started. The parameter values in the nonvolatile memory are also referred to as "Startup Values" in the PC software from PI.

### **Specifications**

The performance specifications are checked before dispatch. The performance specifications apply to room temperature ( $22 \pm 3$  °C), systems in closed-loop operation are calibrated at this temperature. It may be necessary to reset the operating parameters when operating at considerably lower or higher temperatures.

### **Volatile memory**

Electronics main memory. Parameters are stored in the volatile memory when the device is switched on. The parameter values in the volatile memory determine the current behavior of the system. The parameter values in the volatile memory are also referred to as "Active Values" in the PC software from PI.