MS204E
C-887 Hexapod Controller / Digital Motion Controller
User Manual

Version: 1.1.0 Date: 07.12.2012

This document describes the following products:

- **C-887.11**
  6-D Hexapod Controller, Control of 2 Additional Servo-Motor Axes Included, TCP/IP and RS-232 Interface, 19'' Chassis
- **C-887.21**
  6-D Hexapod Controller, TCP/IP and RS-232 Interface, Bench-Top

PIEZO NANO POSITIONING | WWW.PI.WS

Physik Instrumente (PI) GmbH & Co. KG · Auf der Römerstr. 1 76228 Karlsruhe, Germany
Telephon +49 721 4846-0 · Telefax +49 721 4846-1019 · E-Mail info@pi.ws

Original instructions
First printing: 07.12.2012
Document number: MS204E, KSch, Version 1.1.0

Subject to change without notice. This manual is superseded by any new release. The latest release is available for download on our website (http://www.pi.ws).

Contents

1 About this Document

In this Chapter

1.1 Goal and Target Audience of this Manual

This manual contains information on the intended use of the C-887.

It assumes that the reader has a fundamental understanding of basic servo systems as well as motion control concepts and applicable safety procedures.

For updated releases of this user manual, or if you have any questions, contact our customer service department (p. 303).

1.2 Symbols and Typographic Conventions

The following symbols and typographic conventions are used in this user manual:

CAUTION

⚠ **Dangerous situation**

If not avoided, the dangerous situation will result in minor injury.

➢ Actions to take to avoid the situation.

NOTICE

Dangerous situation

If not avoided, the dangerous situation will result in damage to the equipment.

➢ Actions to take to avoid the situation.

INFORMATION

Information for easier handling, tricks, tips, etc.

Symbol	Meaning
1. 2.	Action consisting of several steps whose sequential order must be observed
➢	Action consisting of one or several steps whose sequential order is irrelevant
▪	List item
p. 5	Cross-reference to page 5
RS-232	Labeling of an operating element on the product (example: socket of the RS-232 interface)
Start > Settings	Menu path in the PC software (example: to open the menu, the ***Start*** and ***Settings*** buttons must be clicked in succession)
`SV0?`	Command line or a command from PI's General Command Set (GCS) (example: command to get the servo mode).
Device S/N	Parameter name (example: parameter where the serial number is stored)
5	Value that must be entered or selected via the PC software

1.3 Definition

PC software Software that is installed on the PC.

Firmware Software that is installed on the controller.

Volatile memory RAM module in which the controller parameters are saved when the controller is switched on (working memory).

Non-volatile memory CF card from which e.g. settings for communication are loaded to the volatile memory when the controller is started.

Axis Also referred to as the "logical axis". Logical axes reproduce the translations and rotations of the moving platform of the Hexapod and the motions of the optionally usable stages in the firmware of the C-887. Each direction of motion corresponds to a logical axis. Further information on translations and rotations can be found in the manual of the Hexapod.

All motion commands of the C-887 refer to logical axes.

Hexapod strut Several Hexapod struts must be moved in order to move a logical axis of the moving platform.

The C-887 calculates the target positions for the individual struts from the target positions given for the translation and rotation axes. The velocities and accelerations of the struts are calculated in such a way that all struts start and stop at the same time.

Motion profile Also referred to as a "trajectory". Comprises the target position, velocity and acceleration of the axis calculated for each point in time of the motion. The calculated values are called "commanded values". The source of the motion profile can be selected for the Hexapod (p. 32).

XYZ coordinate system The position and orientation of the Cartesian XYZ coordinate system in which the translations of the moving platform of the Hexapod take place cannot be changed, which is why the system is referred to as spatially fixed. The axes X, Y and Z are referred to as translation axes.

The intersection of the axes of the spatially fixed Cartesian XYZ coordinate system (0,0,0) is referred to as the origin.

The Z axis is always perpendicular to the base plate of the Hexapod.

Pivot point The intersection of the rotational axes U, V and W of the moving platform of the Hexapod.
When the default settings for the pivot point coordinates are used, the pivot point is located at the origin of the XYZ coordinate system after a reference move. See the dimensional drawing in the manual of the Hexapod that belongs to the Hexapod system.
The pivot point is displaced along with the platform during translations. Rotations do not change the position of the pivot point. The pivot point coordinates remain unchanged in both cases.
The pivot point coordinates can be changed with the command `SPI` (p. 227).

Working space The entirety of all combinations of translations and rotations that the Hexapod can approach from the current position is referred to as the working space.

The working space can be limited by the following external factors:

- Installation space
- Dimensions and position of the load

Digital Motion Controller Labeling on the C-887 front panel. The C-887 is referred to as "Hexapod controller" in this manual.

1.4 Other Applicable Documents

The devices and software tools which are mentioned in this documentation are described in their own manuals.

For the latest versions of the user manuals contact our customer service department (p. 303).

Description	Document
Short version of the manual for the C-887	MS204Equ User Manual Short Version
C-887 GCS LabVIEW	MS209E Software Manual
PI GCS 2 DLL for the C-887	MS210E Software Manual
PIMikroMove	SM148E Software Manual
GCS Data	SM146E Software Manual
Simulation of Controller and Hexapod	C887T0001 Technical Note
Updating PI Software	A000T0032 Technical Note
PI Update Finder	A000T0028 Technical Note
LabVIEW with Linux	A000T0021 Technical Note

Depending on the Hexapod system ordered, one of the following user manuals for the Hexapod microrobot is included in the scope of delivery.

Model family	Document
H-206 Hexapod for 6D Alignment and Micromanipulation	MS203E
H-811 Miniature Hexapod Microrobot	MS199E
H-820 Hexapod Microrobot	MS207E
H-824 Compact Hexapod Microrobot	MS200E
H-840 Hexapod Microrobot	MS201E
H-850 Hexapod Microrobot	MS202E

2 Safety

In this Chapter

2.1 Intended Use

The C-887 is a laboratory device as defined by DIN EN 61010-1. It is intended to be used in interior spaces and in an environment which is free of dirt, oil and lubricants.

In accordance with its design, the C-887 is intended for closed-loop driving of a Hexapod Microrobot from PI equipped with DC servo motors.

The C-887 must not be used for purposes other than those named in this user manual.

The C-887 may only be used in compliance with the technical specifications and instructions in this user manual. The user is responsible for process validation.

2.2 General Safety Instructions

The C-887 is built according to state-of-the-art technology and recognized safety standards. Improper use can result in personal injury and/or damage to the C-887.

- ➢ Only use the C-887 for its intended purpose, and only use it if it is in a good working order.
- ➢ Read the user manual.
- ➢ Immediately eliminate any faults and malfunctions that are likely to affect safety.

The operator is responsible for the correct installation and operation of the C-887.

2.2.1 Organizational Measures

User Manual

- Always keep this user manual next to the C-887.
  If the user manual is lost or damaged, contact our customer service department (p. 303).
- Add all information given by the manufacturer to the user manual, for example supplements or Technical Notes.
- Only use the device on the basis of the complete user manual. Missing information due to an incomplete user manual can lead to minor injury as well as property damage.
- Only install and operate the C-887 after having read and understood this user manual.

Personnel Qualification

The C-887 may only be started up, operated, maintained and cleaned by authorized and qualified staff.

2.2.2 Safety Measures during Installation

- Install the C-887 near the power source so that the power plug can be quickly and easily disconnected from the mains.
- Use the supplied power cord to connect the C-887 to the power source.
- If the supplied power cord must be replaced, use a power cord with sufficient ratings.

High temperatures can overheat the C-887.

- Set up the C-887 with a distance of at least 10 cm to the front and rear panels and at least 5 cm to the sides. If this is not possible, make sure that the area is cooled sufficiently.
- Ensure sufficient ventilation at the place of installation.
- Keep the ambient temperature to a non-critical level (<40° C).
- Make sure that the ventilation holes of the case are always free.
- Clean the air filter in the front panel of the case if necessary (p. 281).

Impermissible mechanical load and collisions between the Hexapod, the load to be moved and the environment can damage the Hexapod.

- Only hold the Hexapod by the base plate.
- Before installing the load, determine the limit value for the load of the Hexapod with a simulation program (p. 52).
- Before installing the load, determine the working space of the Hexapod with a simulation program (p. 52).
- Make sure that the installed load observes the limit value determined with the simulation program.
- Avoid high forces and torques on the moving platform during installation of the Hexapod and the load.
- Ensure an uninterruptible power supply in order to prevent an unintentional deactivation of the Hexapod system and resulting unintentional position changes of the Hexapod.
- Make sure that no collisions between the Hexapod, the load to be moved and the environment are possible in the working space of the Hexapod.

The C-887 and the optionally available add-on cards contain electrostatic-sensitive devices (ESD) and can be damaged if handled improperly.

- Avoid touching assemblies, pins and PCB traces.
- Before you touch the C-887 or the add-on cards, discharge yourself of any electric charges:
  - Wear an antistatic wrist strap
    or
  - Before touching an electronic component, briefly touch a conducting, grounded object.
- Only handle and store the C-887 and the add-on cards in environments that dissipate existing static charges to earth in a controlled way and prevent electrostatic charges (ESD protected workstation or electrostatically protected area, in short EPA).

The P-611.3SF NanoCube® stage of the F-206.NCU option is driven by piezo actuators. Temperature changes and compressive stresses can induce charges in piezo actuators. After being disconnected from the electronics, piezo actuators can also stay charged for several hours. Touching or short-circuiting the contacts in the connector of the P-611.3SF NanoCube® stage can lead to minor injuries. In addition, the piezo actuators can be destroyed by an abrupt contraction.

- Do **not** pull out the connector from the electronics during operation.
- Do **not** touch the contacts in the connector.
- Secure the connector of the stage with screws against being pulled out of the controller.

2.2.3 Measures during Start-Up and Operation

There is a risk of minor injuries caused by crushing which can occur between the moving parts of the Hexapod and a stationary part or obstacle.

- Keep your fingers away from areas where they can get caught by moving parts.

When the communication between the C-887 and the PC is established via TCP/IP, the PC software offers all controllers present in the same network for selection. After a C-887 has been selected for the connection, all commands are sent to this controller. If the wrong controller is selected, the operating and maintenance staff of the connected Hexapod are at risk of slight injury from crushing due to unexpectedly supplied motion commands.

- If several C-887 are displayed in the PC software, make sure that you select the right C-887.

If a protective earth conductor is not or not properly connected, dangerous touch voltages can occur and there is a risk of electric shock. In the case of a malfunction or failure of the system, touching the P-611.3SF NanoCube® stage of the F-206.NCU option can result in slight injuries.

- Connect the P-611.3SF NanoCube® stage to a protective earth conductor before start-up. Follow the instructions in the Technical Note F206T0021 (included in the scope of delivery of the F-206.NCU option).
- Do **not** remove the protective earth conductor during operation.

- If the protective earth conductor has to be temporarily removed (e.g. for modifications), reconnect the P-611.3SF NanoCube® stage to the protective earth conductor before starting it up again.

The configuration data used by the Hexapod controller (e.g. geometrical data and servo-control parameters) must be adapted to the Hexapod. If incorrect configuration data is used, the Hexapod can be damaged by uncontrolled motions or collisions. The configuration data is adapted before delivery.

- Check whether the Hexapod controller matches the Hexapod.
  A label on the rear panel of the controller indicates for which Hexapod the controller is intended.
- Only operate the Hexapod with a Hexapod controller whose configuration data is adapted to the Hexapod.

Damage can occur to the Hexapod if the transport lock of the Hexapod has been not removed and a motion is commanded.

- Remove the transport lock before you start up the Hexapod system.

Collisions can damage the Hexapod, the load to be moved and the environment.

General measures for avoiding collisions:

- Make sure that no collisions between the Hexapod, the load to be moved and the environment are possible in the working space of the Hexapod.
- Do not place any objects in areas where they can get caught by moving parts.
- If the Hexapod controller malfunctions, stop the motion immediately.
- Please note that the Hexapod moves unpredictably during a reference move and **no** collision check or prevention occurs.

Depending on the source of the motion profile, the platform of the Hexapod can move along an undefined path under certain conditions. As a result, collisions are possible between the Hexapod, the load to be moved and the environment.

When the motion profile is determined by the trajectory generator of the C-887 (default):

- Avoid sending new target positions while the Hexapod (axes X, Y, Z, U, V, W) is still moving.

- If new target positions have to be sent while the Hexapod is still moving (axes X, Y, Z, U, V, W): Only use motion commands to set target positions that maximally deviate from the current position by the value of the ***Path Control Step Size*** parameter (ID 0x19001504).

When the motion profile is determined by consecutive `MOV` commands:

- Only set target positions whose distance from each other is maximally as large as the value of the ***Path Control Step Size*** parameter (ID 0x19001504) with consecutive `MOV` commands.

When an E-760 controller card is installed in the C-887.11, unintentional motions can be caused by a lack of signal transmission. As a result, collisions are possible between the Hexapod, the load to be moved and the environment.

- Before switching on or rebooting the C-887.11, make sure that the P-611.3SF NanoCube® stage is connected to the E-760 controller card via the Sub-D 25 (f) socket (p. 62).

When scanning procedures are carried out, the platform of the Hexapod moves on an undefined path and can tilt if values for paths or angles are too large. As a result, collisions between the Hexapod, the load to be moved and the environment are possible, and the scanning procedure can end with an unsatisfactory result. Measures for avoiding tipping:

- Select suitable values for paths and angles. For the Hexapod models H-810, H-811 and H-206, 0.2 mm or 0.2 degrees should not be exceeded; for other Hexapod models, the ideal values must be determined experimentally.
- Suitably align the moving platform before the scanning procedure.
- Use suitable holders for the inputs and/or outputs of the optical element to be aligned on the moving platform so that the motion during the scanning procedure only takes place over short distances or angles.

During a scanning procedure started with the FSA command, the scanning area can increase up to double the original area. As a result, collisions are possible between the Hexapod, the load to be moved and the environment.

- Make sure that the platform can safely move outside of the originally specified scanning area as well.

When the actual load of the moving platform of the Hexapod exceeds the maximum holding force based on the self-locking of the actuators, switching off the servo mode for the axes of the moving platform of the Hexapod can cause unintentional position changes of the Hexapod. As a result, collisions are possible between the Hexapod, the load to be moved and the environment.

- ➢ Make sure that the actual load of the moving platform of the Hexapod does not exceed the maximum holding force based on the self-locking of the actuators before you switch off the servo mode, reboot or switch off the C-887.
- ➢ Ensure an uninterruptible power supply in order to prevent an unintentional deactivation of the Hexapod system and resulting unintentional position changes of the Hexapod.

Unsuitable parameter settings can lead to improper operation or damage to the connected mechanical system.

- ➢ Only change parameters after careful consideration.

2.2.4 Safety Measures during Maintenance

The C-887 contains electrostatic-sensitive devices (ESD) that can be damaged in the case of short circuits or flashovers.

- ➢ Before cleaning the case and the air filter, disconnect the C-887 from the power source by pulling the power plug.

There is a risk of minor injuries caused by crushing which can occur between the moving parts of the Hexapod and a stationary part or obstacle.

- ➢ Keep your fingers away from areas where they can get caught by moving parts.

Collisions can damage the Hexapod, the load to be moved and the environment.

During a strut test, the Hexapod moves unpredictably. **No** collision check or prevention takes place, even when a configuration for collision prevention has been stored on the C-887 with the PIVeriMove Software for the Collision Check. Soft limits that have been set for the moving platform of the Hexapod with the NLM (p. 212) and PLM (p. 214) commands are ignored during a strut test. As a result, collisions are possible between the Hexapod, the load to be moved and the environment.

- ➢ Make sure that no collisions are possible between the Hexapod, the load to be moved and the environment during a strut test of the Hexapod.
- ➢ Do not place any objects in areas where they can get caught by moving parts during a strut test.
- ➢ Observe the Hexapod during a strut test in order to be able to intervene quickly in the case of malfunctions.

During a strut test, the Hexapod strut can reach a limit switch. As a result, the servo mode is automatically switched off for the axes of the moving platform of the Hexapod. Switching off the servo mode can cause unintentional position changes of the Hexapod. As a result, collisions are possible between the Hexapod, the load to be moved and the environment.

- ➢ Make sure that the actual load of the moving platform of the Hexapod does not exceed the maximum holding force based on the self-locking of the actuators before you start a strut test.

3 Product Description

In this Chapter

3.1 Features and Applications

General

The C-887 Hexapod controller is used to control a Hexapod in six degrees of freedom with a very high positioning accuracy.
The controller forms a Hexapod system together with a cable set and a Hexapod. A Hexapod system offers translations in the direction of the X, Y and Z axes as well as rotations around each of these three axes.

The C-887 drives the DC motors of the six Hexapod struts in closed-loop operation. The Hexapod struts carry the moving platform and bring it into the desired position.

Positioning commands use Cartesian coordinates. The C-887 converts these into the respective positions and velocities of the Hexapod struts before the moving platform moves to the desired position.

The product features of the C-887 include:

- TCP/IP and RS-232 interface for communication
- Four USB connections, e.g. for the C-887.MC operating unit (available as an accessory), a keyboard or a mouse.
- Communication is also possible via a connected keyboard and connected VGA monitor without connecting the C-887 to a PC
- Parameters for configuring the properties of the C-887
- Non-volatile macro memory
- Data Recorder

- Comprehensive software package available, including the ***Hexapod Simulation Software*** program for determining the working space and the maximum loads of the Hexapod.
- Real-time operating system
- Command set available in GCS syntax version 1.0 for downwards compatibility and in the current GCS syntax version 2.0 (default)
- Available in two versions:
  - in a 19" case including the control of two additional axes and extendable by add-on cards (C-887.11) (p. 25)

  or

  - as a compact bench-top (C-887.21)

Simple system connection

Parameters can be set and checked with PC software. The start-up and system configuration takes place via the PIMikroMove program included in the scope of delivery. Connection to the customer's software is possible via LabView driver and shared library The programming of the system is identical for all PI controllers. The combined control of different controllers is therefore easily possible.

Data Recorder

The current position and the target position for the Hexapod axes, among others, can be recorded over time with the data recorder. The recorded values are saved in data recorder tables and can be conveniently read out e.g. with PIMikroMove.

Furthermore, the velocities and the accelerations commanded by C-887 for the individual struts of the Hexapod , and if present, for the axes A and B, for example, can be calculated from the recorded data with PIMikroMove.

Operating system

The C-887 uses a real-time operating system.

Further software

The ***Hexapod Simulation Software*** program is included on the supplied C-887.CD. This program can be used to calculate e.g. the load of the individual struts for the Hexapod depending on the mounted load and the Hexapod position. This prevents the Hexapod from being overloaded.

The controller itself and the connected Hexapod can be simulated on a PC with Windows operating system using various other programs from the supplied C-887.CD.

The PIVeriMove Software for the Collision Check can be ordered as an optional accessory. With PIVeriMove the environment and the load on the moving platform of the Hexapod can be defined and simulated. This configuration can be stored on the controller. Thus the controller can prevent collisions independently of the PC by calculating all distances between the defined components and comparing them with a given safety distance online after it has received move commands.

3.2 Model Overview

Hexapod and Hexapod controller are only available together as a system.

Possible system components

Standard versions of the C-887 Hexapod controller:

Model	Name
C-887.11	6-D Hexapod Controller, Control of 2 Additional Servo-Motor Axes Included, TCP/IP and RS-232 Interface, 19" Chassis
C-887.21	6-D Hexapod Controller, TCP/IP and RS-232 Interface, Bench-Top

Standard versions of the Hexapod:

Model family	Name
H-206	Hexapod for 6D alignment and micromanipulation
H-811	Miniature Hexapod microrobot, direct drive, vacuum-compatible version available
H-820	Hexapod microrobot, basic model
H-824	Compact Hexapod microrobot, versions with gearhead and direct drive each also available with vacuum compatibility
H-840	Hexapod microrobot, versions with gearhead and direct drive available
H-850	Hexapod microrobot with excellent repeatability, gearhead; versions available for higher loads or higher velocities, each also available with vacuum compatibility

For detailed lists of the available models, see the manual of the respective Hexapod model family.

Standard cable sets:

Model	Name
C-887.A03	Hexapod cable set 3 m, consisting of: ▪ Data transmission cable, MDR68 to MDR68, 1:1 (K040B0034) ▪ Power supply cable, M12m 180° to M12f 90° (K060B0111)
C-887.V02	Hexapod cable set 2 m on the vacuum side, vacuum feedthrough, 3 m on the air side, consisting of: ▪ Data transmission cable on the vacuum side, MDR68m to HD Sub-D 78m, 2 m (M824B0010) ▪ Power supply cable on the vacuum side, LEMO 2-pin to LEMO 2-pin, 2 m (K060B0132) ▪ Vacuum feedthrough for data transmission, HD Sub-D 78m/f (4668) ▪ Vacuum feedthrough for power source, LEMO 2-pin to M12m (C887B0002) ▪ Data transmission cable on the air side, HD Sub-D 78f to MDR68m, 3 m (K040B0092) ▪ Power supply cable on the air side, M12m to M12f, 3 m (K060B0112)

Available Hexapod systems

A Hexapod system consists of a Hexapod controller, a Hexapod and a cable set. The order number of the system reflects which system components are combined in the system.

For detailed lists of the available Hexapod systems, see the manual of the respective Hexapod model family. The following table is for general orientation:

System order number	Description
H-xxx.x11	Hexapod system with C-887.11 controller and C-887.A03 cable set; Hexapod not vacuum-compatible
H-xxx.x12	Hexapod system with C-887.21 controller and C-887.A03 cable set; Hexapod not vacuum-compatible
H-xxx.xV1	Hexapod system with C-887.11 controller and C-887.V02 cable set; Hexapod vacuum-compatible
H-xxx.xV2	Hexapod system with C-887.21 controller and C-887.V02 cable set; Hexapod vacuum-compatible

3.3 Product View

3.3.1 Front Panel

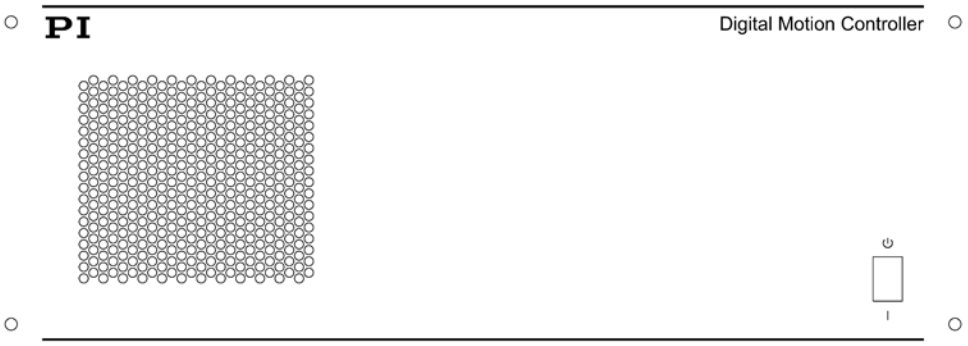


Figure 1: C-887.11 front panel

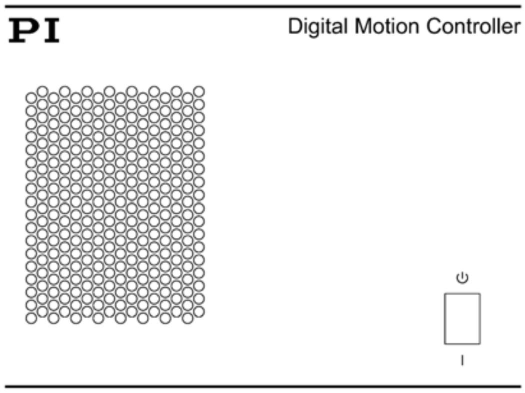


Figure 2: C-887.21 front panel

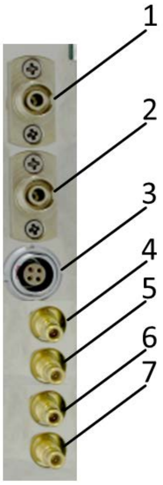
Labeling	Function
	Standby switch: ▪ ⏻ position: C-887 is switched off, various internal components remain in standby mode ▪ I position: C-887 is switched on

3.3.2 Rear Panel

All C-887 models			
	Labeling	**Type, pin assignment**	**Function**
	2x 24 VDC Power Out	A-coded, 4-pin M12 socket (f) (p. 316)	Two connections: ▪ Power source connection for Hexapod ▪ Only when a cable set with line driver boxes is used: power source connection for line driver box The connections can be assigned as desired.
	Universal serial bus	USB type A	Four USB interfaces for connecting peripheral devices, e.g. for keyboard, mouse or C-887.MC control unit
	VGA	Mini-HD Sub-D 15 (f)	VGA connection for monitor
	LAN	RJ45 socket	Network connection via TCP/IP (p. 36)
	RS-232	Sub-D 9 (m) (p. 318)	Serial connection to PC (p. 36)
	Hexapod	MDR68 (f) (p. 316)	Connection for data transmission between Hexapod and controller

Only C-887.11:			
	Labeling	**Type, pin assignment**	**Function**
	A, B	Sub-D 15(f) (p. 319)	Two connections for stages. Only for DC motors! ▪ Output of PWM signals for the stage ▪ Input of the signals of the incremental position sensor ▪ Input of the reference switch signal

Only C-887.11 and only when the E-760.3S0 controller card (F-206.NCU option) is installed:			
	Labeling	**Type, pin assignment**	**Function**
	Nanocube	Sub-D 25 (f) (p. 320)	Connection for P-611.3SF NanoCube® stage: ▪ Output of the operating voltage for the 3 piezo actuators, -20 V to +120 V ▪ Input of the signals of the 3 strain gauge sensors
		RJ45 socket (p. 321)	Connection for an analog input line
		FC with protective cap	Reserved (photodiode not populated)

Only C-887.11 and only when at least one photometer card (F-206.VVU and F-206.iiU options) is installed. Connections per photometer card:				
	No.	**Labeling, when pre-installed**	**Type**	**Function**
	1	**FC1**	FC	Connection for optical input signal 1: for processing with the C-887 the signal must be fed through analog output 1 to analog input 1 (p. 63).
	2	**FC2**	FC	Connection for optical input signal 2: for processing with the C-887 the signal must be fed through analog output 2 to analog input 2 (p. 63).
	3	-	LEMO 4 (f)	Reserved
	4	**IN**	SMB socket	Analog input 1, 0 to 10 V (p. 308), can be used as: ▪ Input for converted signal of optical input 1 or ▪ Input for the analog signal of an external sensor
	5	**IN**	SMB socket	Analog input 2, 0 to 10 V (p. 308), can be used as: ▪ Input for converted signal of optical input 2 or ▪ Input for the analog signal of an external sensor
	6	**OUT**	SMB socket	Analog input 1: ▪ Output of the converted signal of optical input 1; 0 to 10 V (p. 308)
	7	**OUT**	SMB socket	Analog input 2: ▪ Output of the converted signal of optical input 2; 0 to 10 V (p. 308)

3.4 Scope of Delivery

The following table contains the scope of delivery of the Hexapod controller. The scope of delivery of the Hexapod and the cable set of the Hexapod system is listed in the user manual of the Hexapod.

Order Number	Items
C-887	Hexapod controller according to your order
3763	Power cord
C-815.563	Cross-over network cable
C-815.553	Straight-through network cable
C-815.34	Null-modem cable for connection to the PC via RS-232
36711	US keyboard with USB connection
3683	Torx screwdriver for opening the case
	Only C-887.21: 2 mounting brackets for fastening the case on a surface
Documentation, consisting of:	
MS204Equ	Short version of the user manual for the C-887
C-887.CD	CD with PC software and documentation
A000T0028	Technical Note on Using the PI Update Finder
A000T0032	Technical Note on Updating the PI Software using the PI Update Finder
Packaging material	

3.5 Accessories

3.5.1 Overview

Order Number	Description
C-887.A20	Hexapod cable set 20 m, consisting of: ▪ Line driver box for data transmission cable, controller-side (C030B0011) ▪ Line driver box for data transmission cable, Hexapod-side (C030B0012) ▪ 2 data transmission cables, MDR68 to MDR68, 1:1, 3 m (K040B0034) ▪ Data transmission cables, MDR68 to MDR68, 1:1, 3 m (K040B0034) ▪ Power supply cable for line driver box, with M12 coupling/M12 connector, 17 m (K060B0126) ▪ Power supply cable for Hexapod, with M12 coupling/M12 connector, 20 m (K060B0127)
C-887.MC	▪ Hexapod Control Unit, USB Connector, 3 m Cable ▪ Technical Note C887T0003
C-887.VM1	▪ PIVeriMove Software for the Collision Check ▪ Technical Note C887T0002
F-311.LV	PIMotion&Vision LabVIEW Driver Set

Only for C-887.11:

Order Number	Description
M-0xx, **M-1xx,** **M-2xx,** **M-4xx,** **M-5xx**	Stages from the PI product area of micropositioning technology that are equipped with a DC motor and PWM amplifier. Information on suitable models on request.

Order Number	Description
F-206.iiU	▪ Photometer Card, IR Range, 2 Channels ▪ 2 SMB/SMB cables for connecting an analog output with the corresponding analog input ▪ Technical Note F206T0020 for the photometer card
F-206.VVU	▪ Photometer Card, Visible Range, 2 Channels ▪ 2 SMB/SMB cables for connecting an analog output with the corresponding analog input ▪ Technical Note F206T0020 for the photometer card
F-206.NCU	▪ Rapid 3-Axis Piezo Nanopositioning System for Use in Combination with Hexapod Systems. Consists of P-611.3SF NanoCube® XYZ Nanopositioning System, 100 x 100 x 100 µm, Strain Gauge Sensors with Integrated Fiber Adapter Interface and E-760.3S0 NanoCube® Piezo Controller Board, ISA Bus ▪ Mounting bracket for P-611.3SF NanoCube® (F20601127) ▪ Extension cable 1.5 m (P-611K006) ▪ Technical Note F206T0021 on handling the piezo nanopositioning system

➢ To order, contact our customer service department (p. 303).

3.5.2 C-887.VM1 PIVeriMove as an Option for Collision Checking

The PIVeriMove Software for the Collision Check, optionally available under the order number C-887.VM1 (p. 25), offers the following functions:

- Simulation of the arrangement of the Hexapod and its environment on a PC
- Transfer of the configuration created on the PC to the C-887.

With the created configuration, it is checked whether collisions occur between the following groups for any desired target positions:

- Environment incl. base plate of the Hexapod
- Hexapod struts
- Moving platform of the Hexapod incl. load

INFORMATION

The C-887 carries out collision checks on the basis of a configuration created with the PIVeriMove Software for the Collision Check only when the absolute position of the mechanical system is known (Hexapod: after a successful reference move).

The PIVeriMove Software for the Collision Check is installed on a PC and activated there via a license key. For further information on installation and use, see Technical Note C887T0002 (included in the scope of delivery of the PIVeriMove Software for the Collision Check).

3.5.3 F-206.VVU and F-206.iiU Photometer Cards

The C-887.11 can be operated with up to two optionally available 2-channel photometer cards, in order to measure the intensity of optical signals (p. 25).

The photometer cards convert the optical signals fed in via FC connections into analog signals for processing by the C-887.

INFORMATION

The analog inputs are accessible in the firmware of the C-887 as input signal channels with a particular identifier, see "Commandable Items" (p. 28).

For further information, see "Installing Add-on Cards" (p. 45), "Connecting an Optical Signal Source to the C-887.11" (p. 63), "Analog Input Signals" (p. 122) and "Specifications of the F-206.iiU and F-206.VVU Options" (p. 308).

3.5.4 F-206.NCU Piezo Nanopositioning System

The C-887.11 can be optionally operated with the F-206.NCU piezo nanopositioning system. This piezo nanopositioning system consists of the P-611.3SF NanoCube® stage and the E-760.3S0 controller card. When the E-760.3S0 controller card is installed in the C-887, the logical axes K, L and M reproduce the motion of the P-611.3SF NanoCube® in the firmware of the C-887, and an analog input channel is available.

INFORMATION

The analog input is accessible in the firmware of the C-887 as input signal channel with a particular identifier, see "Commandable Items" (p. 28).

Further information can be found in "Installing an Add-on Card in the C-887" (p. 45), "Connecting the P-611.3SF NanoCube® to the C-887.11" (p. 62), "Analog Input Signals" (p. 122) and "Specifications of the F-206.NCU Option" (p. 311).

3.6 Functional Principles

3.6.1 Commandable Items

The following table contains the items that can be accessed with commands of the GCS (p. 127).

Item	Number	Identifier	Description
Logical axis	6	X, Y, Z, U, V, W	The logical axes X to W reproduce the translations and rotations of the moving platform of the Hexapod in the firmware of the C-887. Translation axes: X, Y and Z Rotation axes: U, V and W
Logical axis	2	A, B	Only C-887.11: The logical axes A and B reproduce the motion of additional stages with DC motors in the firmware of the C-887.
Logical axis	3	K, L, M	Only C-887.11 and only when the E-760 controller card (F-206.NCU option) is installed: The logical axes K, L and M reproduce the motion of the P-611.3SF NanoCube® in the firmware of the C-887.

Item	Num-ber	Identi-fier	Description
Hexapod strut	6	1 to 6	The Hexapod struts are not accessible for motion commands but can be selected as data sources for recording with the data recorder for diagnostic purposes, see also `DRC` (p. 155) and `HDR?` (p. 190).
Input signal channels (optional)	Maximum of 6	1 to 6	Only C-887.11 and only when at least one photometer card (F--206.iiU and F-206.VVU options) and/or the E-760.3S0 controller card (F-206.NCU option) is installed. Channels of the analog inputs; the identifier is assigned to the input signal channels in the following order: ■ 1 and 2: first photometer card ■ 3 and 4: second photometer card ■ E-760.3S0 controller card: – 2 when no photometer card is installed* – 4 when only one photometer card is installed* – 6 when two photometer cards are installed* *The identifiers 1, 3 and 5 each are assigned to the FC which is not populated. The `TAC?` command (p. 237) gets the number of installed analog inputs.
Data recorder table	16	1 to 16	The data recorder tables contain the recorded data. The C-887 has 16 data recorder tables (query with `TNR?` (p. 240)) with max. 10,240 data points per table. The number of points per data recorder table can be set with the ***Data Recorder Points Per Table*** parameter (ID 0x16000201).
Overall system	1	1	C-887 as an overall system.

3.6.2 Important Components of the Firmware

The firmware of the C-887 provides the following functional units:

Firmware component	**Description**
ASCII commands	The communication with the C-887 can be carried out with the commands of the PI General Command Set (GCS; syntax version 2.0 or syntax version 1.0). GCS is independent of the hardware (controller, connected Hexapod). Examples of the use of GCS: ▪ Starting motions of the Hexapod ▪ Getting system and motion values The available commands are listed in "Overview of Commands for GCS 2.0" (p. 134).
Configuration files, stage databases, parameters	The settings that are used to adapt the C-887 e.g. to the properties of the connected mechanical system or the communication interfaces used are determined by: ▪ Configuration files and stage databases: only accessible for the customer service department (p. 303) ▪ Interface parameters: Adaptation in the non-volatile memory possible, further information in "Establishing Communication via the TCP/IP Interface" (p. 79) and "Establishing Communication via the RS-232 Interface" (p. 88) ▪ Volatilely configurable parameters: further information in "Adjusting Settings" (p. 269)
Data Recorder	The C-887 contains a real-time data recorder. This can record different input and output signals (e. g. current position, commanded position) from different data sources (e. g. logical axes, input signal channels) (p. 106).
Macros	The C-887 can save macros. Command sequences can be defined and permanently stored via the macro function. A start-up macro can be defined that is executed each time that the C-887 is switched on or rebooted. This simplifies operation without a connected PC. For more information, see "Controller Macros" (p. 110).
User interface	The user interface of the C-887 can be used to get and set the system configuration as well as for test purposes (p. 74).

The firmware can be updated with a tool (p. 284).

3.6.3 Triggering Motions

Motions are triggered by move commands. Possibilities for sending motion commands:

- Direct entry of commands or usage of control elements in the user interface of the C-887 (p. 74)
- Direct entry of commands or usage of control elements in the PC software (connected via TCP/IP or RS-232 interface)
- Control unit C-887.MC for the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W)
- Controller macros (p. 110) with motion commands : Any commands can be sent via the communication interfaces of the C-887 when a macro is running on the controller. The macro content and move commands received via the communication interfaces may overwrite each other.

Commands	Description
MOV,	Motion to absolute target position
MVR	Motion to relative target position Is sent by the C-887.MC control unit. In this case, the step size of the motion equates the setting made with the SST command.
IMP	Starts an impulse and triggers data recording
STE	Starts a step and triggers data recording
FRF	Starts a reference move Can also be sent by the C-887.MC control unit.
AAP, FIO, FLM, FLS, FSA, FSC, FSM	Starting of scanning procedures

INFORMATION

For the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W) and axes A and B, motions can only be commanded after a successful reference move (p. 174) (also referred to as "initialization").

3.6.4 Motions of the Hexapod

Rotations

Rotations take place around the pivot point. The coordinates of the pivot point are determined with the SPI command (p. 227). A given rotation in space is calculated from the individual rotations in the order U -> V- > W. This takes place regardless of whether the values for U, V and W were explicitly given with the current command or result from the previous commands.
Further information can be found in the manual of the Hexapod.

Motion Profile

Depending on the parameter setting, the motion profile (p. 3) for the axes of the Hexapod (X, Y, Z, U, V, W) is specified by one of the two following sources:

- Trajectory generator of the C-887
- Consecutive MOV commands

The motion profile is configured with the following parameters:

Parameter	Description and possible values
Maximum System Velocity (Phys. Unit/s) 0x19001500	Maximum system velocity This parameter is write-protected and is adapted to the Hexapod of the Hexapod system before delivery. Upper limit of the velocity for the moving platform of the Hexapod that can be set with VLS (p. 245).
Minimum System Velocity (Phys. Unit/s) 0x19001501	Minimum system velocity This parameter is write-protected and is adapted to the Hexapod of the Hexapod system before delivery. Depends on the smallest step size of the Hexapod. Lower limit of the velocity for the moving platform of the Hexapod that can be set with VLS.
Maximum System Acceleration (Phys.Unit/s2) 0x19001502	Maximum system acceleration This parameter is write-protected and is adapted to the Hexapod of the Hexapod system before delivery.

Parameter	Description and possible values
Path Control Step Size (mm) 0x19001504	Step size for the trajectory calculation of the platform motion This parameter is write-protected and is adapted to the Hexapod of the Hexapod system before delivery. This parameter is only evaluated when parameter 0x19001900 has the value 0. If parameter 0x19001900 has the value 1: The distance of the target positions set with consecutive `MOV` commands to each other may only be maximally as large as the value of parameter 0x19001504, in order to avoid tilting the Hexapod.
Trajectory Source 0x19001900	Source of the motion profile 0 = Motion profile is determined by the trajectory generator (default). During a motion, any given new motion command sets the target position to a new value and the moving platform immediately approaches the new target position on an undefined path. 1 = Motion profile is determined by consecutive `MOV` commands. Other motion commands (e.g. `MVR`, `IMP` and `STE`, commands for starting scanning procedures) are not permitted.
Trajectory Execution 0x19001901	Execution of the motion profile Determines how the motion profile defined by consecutive `MOV` commands is executed: 0 = Motion profile is immediately executed (default) 1 = Motion profile is stored in a buffer before execution This parameter is only evaluated when parameter 0x19001900 has the value 1.
Maximum Number of Trajectory Points 0x19001902	Maximum number of motion-profile points Indicates the maximum size of the buffer. This parameter is write-protected and only evaluated when the parameters 0x19001900 and 0x19001901 both have the value 1.
Threshold for Trajectory Execution 0x19001903	Threshold value for executing the motion profile Determines how many motion-profile points have to be stored in the buffer (by consecutive `MOV` commands) until the execution of the motion profile begins. This parameter is only evaluated when the parameters 0x19001900 and 0x19001901 both have the value 1.

Parameter	Description and possible values
Current Number Threshold for Trajectory Execution 0x19001904	Shows the current number of motion-profile points in the buffer. The parameter value is always 0 when the motion profile is determined by the trajectory generator or the motion profile defined by consecutive MOV commands is immediately executed. This parameter is write-protected.

The motion profile is configured with the following commands:

Command	Syntax	Function
SCT	SCT "T" <CycleTime>	Determines the cycle time for executing a motion profile. The setting is only in effect when the ***Trajectory Source*** parameter (ID 0x19001900) has the value 1. The cycle time is used to calculate the velocity during the motion so that the specified points of the motion profile are each reached precisely at the end of the time interval (where possible with regard to the velocity and acceleration limits).
VLS	VLS <SystemVelocity>	Sets the velocity for the moving platform of the Hexapod. Limits the velocities of the individual struts when the ***Trajectory Source*** parameter (ID 0x19001900) has the value 1.

The #11 command (p. 142) gets the free memory space of the buffer, the contents of which determine the motion profile of the Hexapod, when the parameters 0x19001900 and 0x19001901 both have the value 1.

INFORMATION

When the ***Trajectory Execution*** parameter (ID 0x19001900) has the value 1: For PC's with a Windows operating system, it is not ensured that consecutive `MOV` commands are sent in the required time unit, due to the lack of real-time capability of the operating system.

- ➢ If the motion profile is already determined, store it in a buffer before execution, in order to ensure that the cycle time is observed. For this purpose, use `SPA` to set the ***Trajectory Execution*** parameter (ID 0x19001901) to the value 1.
- ➢ For a meaningful use of the buffer, increase the value of the ***Threshold for Trajectory Execution*** parameter (0x19001903) with `SPA` (default = 1).

Checking the motion profile created by the trajectory generator

When the motion profile for the Hexapod is created by the trajectory generator (parameter 0x19001900 has the value 0), a check is carried out before the start of each motion whether the moving platform can actually reach the nodes of the calculated trajectory and the commanded target position. If a node or the target position cannot be reached, the motion is not executed. The following is checked:

- Are the nodes and the target position outside of the travel range limits that can be queried with `TMN?` (p. 239) and `TMX?` (p. 239)?
- Have the soft limits set with `NLM` (p. 212) and `PLM` (p. 214) been activated with `SSL` (p. 229), and if yes, are the nodes and the target position outside of these soft limits?
- Are the individual struts able to move the platform to the required nodes and the specified target position?
- When a configuration for collision avoidance has been stored on the C-887 with the optionally available PIVeriMove Software for the Collision Check: Do collisions occur between the following groups?
  - Environment incl. base plate of the Hexapod
  - Hexapod struts
  - Moving platform of the Hexapod incl. load

The `VMO?` command (p. 246) gets whether a specified target position can be reached.

3.7 Communication Interfaces

Available communication interfaces

The C-887 can be controlled with ASCII commands via the following interfaces:

- From the PC:
  - TCP/IP
  - Serial RS-232 connection
- User interface, accessible via keyboard, mouse and monitor connection directly on the C-887
- C-887.MC Hexapod Control Unit (available as accessory), connected directly to the C-887

3.8 Overview of PC Software

The following table shows the PC software that is included in the C-887.CD. The given operating systems stand for the following versions:

- Windows: XP, Vista and 7
- Linux: Kernel 2.6, GTK 2.0, glibc 2.4

PC software	Supporting operating system	Short description	Recommended use
Dynamic program library for GCS	Windows, Linux	Allows software programming for the C-887 with programming languages such as e. g. C++. The functions in the dynamic program library are based on the PI General Command Set (GCS).	For users who would like to use a dynamic program library for their application.

PC software	Supporting operating system	Short description	Recommended use
LabVIEW driver	Windows, Linux	LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments). The C-887 LabVIEW software is a collection of virtual instrument drivers (VI drivers) for the C-887 controller. These drivers support GCS.	For users who want to use LabVIEW to program their application.
PIMikroMove	Windows	Graphic user interface for Windows with which the C-887 and other controllers from PI can be used. ▪ The system can be started without programming effort ▪ Graphic representation of the motions ▪ Macro functionality for storing command sequences on the PC (host macros) ▪ Complete environment for command entry, for testing different commands No command knowledge is necessary to operate the PIMikroMove. PIMikroMove uses the dynamic program library to supply commands to the controller.	For users who want to perform simple automation tasks or test their equipment before or instead of programming an application. A log window showing the commands sent makes it possible to learn how to use the commands.
PITerminal	Windows	Simple graphic user interface for Windows that can be used for nearly all PI controllers (see PIMikroMove user manual).	For users who want to send GCS commands directly to the controller.
PI Update Finder	Windows	Checks the PI software installed on the PC. If more current versions of the PC software are available on the PI server, downloading is offered.	For users who want to update the PC software.

PC software	Supporting operating system	Short description	Recommended use
Hexapod Simulation Software	Windows	Simulation program with which the working space and the load of the Hexapod must be calculated before the Hexapod is installed.	For all users.
Simulation of the controller and Hexapod	Windows	Program with which the C-887 and the connected Hexapod as well as the axes A, B, K, L and M can be simulated. The simulated Hexapod system can be operated with PIMikroMove, for example.	For users who would like to test the behavior of the Hexapod systems when the Hexapod controller and/or the Hexapod are not available.
PI Hexapod Controller Update Wizard	Windows	Program for updating the firmware of the C-887.	For users who want to update the firmware.

INFORMATION

The separately available manuals for PC software are listed in "Other Applicable Documents" (p. 4) and are found on the product CD.

4 Unpacking

1. Unpack the C-887 with care.
2. Compare the contents against the items covered by the contract and against the packing list.
3. Inspect the contents for signs of damage. If parts are missing or you notice signs of damage, contact PI immediately.
4. Keep all packaging materials in case the product needs to be returned.

5 Installation

In this Chapter

5.1 General Notes on Installation

The Hexapod can be mounted in any orientation.

NOTICE

Impermissible mechanical load and collisions!

Impermissible mechanical load and collisions between the Hexapod, the load to be moved and the environment can damage the Hexapod.

- ➢ Only hold the Hexapod by the base plate.
- ➢ Before installing the load, determine the limit value for the load of the Hexapod with a simulation program (p. 52).
  The limit values determined with the simulation program are only valid when the Hexapod controller has the servo mode switched on for the axes of the moving platform of the connected Hexapod.
- ➢ Before installing the load, determine the working space of the Hexapod with a simulation program (p. 52).
  The limits of the working space vary depending on the current position of the Hexapod (translation and rotation coordinates) and the current coordinates of the pivot point.
- ➢ Avoid high forces and torques on the moving platform during installation.
- ➢ Ensure an uninterruptible power supply in order to prevent an unintentional deactivation of the Hexapod system and resulting unintentional position changes of the Hexapod.
- ➢ Make sure that no collisions between the Hexapod, the load to be moved and the environment are possible in the working space of the Hexapod.

5.2 Ensure Ventilation

High temperatures can overheat the C-887.

- ➢ Set up the C-887 with a distance of at least 10 cm to the front and rear panels and at least 5 cm to the sides. If this is not possible, make sure that the area is cooled sufficiently.
- ➢ Ensure sufficient ventilation at the place of installation.
- ➢ Keep the ambient temperature to a non-critical level (<40° C).
- ➢ Make sure that the ventilation holes of the case are always free.
- ➢ Clean the air filter in the front panel of the case if necessary (p. 281).

5.3 Installing PC Software

5.3.1 Performing the Initial Installation

Accessories

- PC with a Windows operating system (XP, Vista, 7) or Linux operating system
- For installation from CD: product CD (included in the scope of delivery)

Installing PC software on Windows

1. Start the installation wizard by double-clicking the icon or the ***setup.exe*** file in the installation directory (root directory of the CD).
2. Follow the instructions on the screen.
   You can choose between the default installation (typical) or the user-defined installation (customized). With the default installation (recommended), the following components are installed among others:
   - LabVIEW drivers
   - Dynamic program library for GCS
   - PIMikroMove
   - ***Hexapod Simulation Software*** simulation program
   - Program for updating the firmware of the C-887

Installing PC software on Linux

1. Unpack the tar archive from the /linux directory of the product CD to a directory on your PC.
2. Open a terminal and go to the directory to which you have unpacked the tar archive.
3. Log on as a superuser (root rights).
4. Enter ./INSTALL to start the installation.
   Pay attention to lower and upper case when entering commands.
5. Follow the instructions on the screen.

You can select individual components for installation.

5.3.2 Installing Updates

Since the PC software is continually improved by PI, your product CD may no longer be up-to-date upon delivery.

➢ Always install the latest version of the PC software, provided that this makes sense for your application.

Prerequisite

✓ Active connection of the PC to the Internet.

– If your PC does **not** have an Internet connection:
You have the Technical Note "Updating PI Software" (A000T0032) at hand. You can find the document either on the product CD or at http://www.update.pi-portal.ws in the zip file for the PI Update Finder.

✓ If your PC uses a Windows operating system:

– If the PI Update Finder program is not on your product CD:
You have downloaded the PI Update Finder from http://www.update.pi-portal.ws and unpacked it from the zip file into a directory on your PC.

– You have the Technical Note "PI Update Finder" (A000T0028) at hand. You can find the document either on the product CD or on in the zip file that you have downloaded for the PI Update Finder.

✓ If your PC uses a Linux operating system:

– You have the user name and password for the C-887 at hand. Both of these can be found in the file "C-887 Releasenews_V_x_x_x.pdf" (x_x_x: version number of the CD) in the \Manuals folder on the product CD.

Updating PC software on Windows

➢ Follow the instructions in the Technical Note for the PI Update Finder (A000T0028).

➢ If your PC does **not** have an Internet connection: Follow the instructions in the Technical Note "Updating PI Software" (A000T0032) as well.

Updating PC software on Linux

1. Open the PI website (http://www.pi.ws).
2. In the ***Resources*** area on the start page, click ***Manuals, Software, ISO Statements*** on the left margin.

   A selection list appears.
3. In the selection list, click ***Software & Manuals on PI Support Server***.

4. In ***User login*** area on the left margin, enter your user name and password from the "xxx Releasenews.pdf" file from the product CD.
5. Click ***Login***.
6. Click the category ***C Motion Controllers / Hexapod Controllers***.
7. Click ***C-887 > Software*** (if you click ***Documents*** the latest versions of the corresponding manuals are displayed).
8. Underneath the latest CD mirror, click the ***Download*** button (also contains the manuals).
9. Save the downloaded archive file on the PC.
10. Unpack the file to a separate installation directory.
11. In the directory with the unpacked files, go to the linux subdirectory.
12. Unpack the archive file in the linux directory by entering the command tar -xypf <Name of the archive file> on the console.
13. Read the accompanying information (readme file) on the software updating.
14. Log onto the PC as a superuser (root rights).
15. Only install the update if it is a good idea for your application.

5.4 Installing an Add-on Card in the C-887

NOTICE

Electrostatic hazard!

The C-887 and the optionally available add-on cards contain electrostatic-sensitive devices (ESD) and can be damaged if handled improperly.

- ➢ Avoid touching assemblies, pins and PCB traces.
- ➢ Before you touch the C-887 or the add-on cards, discharge yourself of any electric charges:
  - – Wear an antistatic wrist strap
    or
  - – Before touching an electronic component, briefly touch a conducting, grounded object.
- ➢ Only handle and store the C-887 and the add-on cards in environments that dissipate existing static charges to earth in a controlled way and prevent electrostatic charges (ESD protected workstation or electrostatically protected area, in short EPA).

NOTICE

Impairment from soiling!

Soiling, e.g. from dust, makes an FC connection for optical fibers inoperative.

➢ Make sure that unused FC connections have a protective cap.

➢ Keep the C-887 free from dust and dirt.

INFORMATION

For the C-887.11 Hexapod controller, optional accessories are available (p. 25) that require the installation of add-on cards in the controller case. Possible add-on cards:

- 2-channel photometer card, IR-range (F-206.iiU option)
- 2-channel photometer card, visible range (F-206.VVU option)
- E-760.3S0 controller card (F-206.NCU option)

INFORMATION

If you want to install add-on cards that you received before May 2012:

➢ Contact our customer service department before installation (p. 303).

INFORMATION

The number of photometer cards that can be installed in the same C-887.11 is limited to two.

INFORMATION

When the optional accessories are ordered together with the Hexapod system, the add-on cards are installed in the controller upon delivery.

If add-on cards are to be installed at a later time:

- The E-760 controller card is completely configured upon delivery and can be installed without further preparations.
- For photometer cards, the address must be set with jumpers before installation (p. 47).

5.4.1 Setting the Address of the Photometer Card

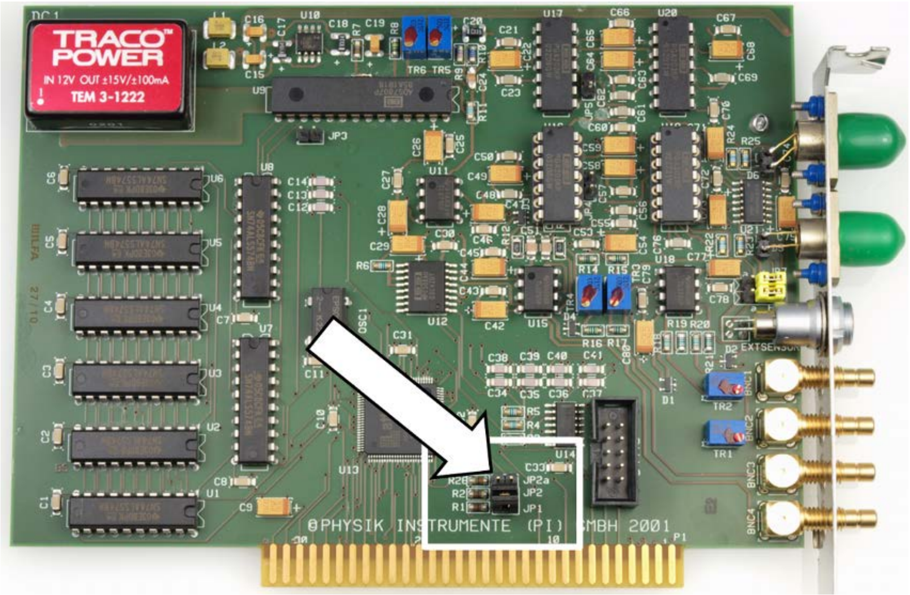


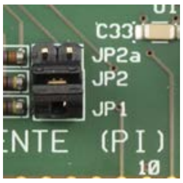
Figure 3: Position of the jumpers for setting the address of the photometer card, example F-206.VVU

Tools and accessories

Photometer card(s) ordered as an optional accessory (p. 25).

Setting the Address of the Photometer Card

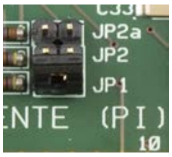
1. Set the address of the first photometer card:



   – Jumper **JP1** is bridged.

   – Jumper **JP2** is bridged.

2. If you want to install a second photometer card in the C-887.11, set the address of the second card:



– Jumper **JP1** is bridged.

– Jumper **JP2** is open.

5.4.2 Installing an Add-on Card

NOTICE

Impermissible mechanical load!

An impermissible mechanical load during installation can damage the add-on card.

➢ Avoid high forces on the add-on card during installation.

➢ Do not use force.

Prerequisite

✓ The C-887 is **not** connected to the power socket via the power cord.

✓ When you install photometer cards: You have correctly set the addresses of the photometer cards (p. 47) .

Tools and accessories

- Supplied Torx screwdriver
- Phillips-head screwdriver
- Open-end wrench for M4 nut, width across flats 7
- Additional card that has been ordered as an optional accessory (p. 25).

Installing an Add-on Card

1. Release the two Torx screws with which the top of the case is fastened to the rear panel of the controller.
2. Carefully remove the top of the case.
3. Use the open-end wrench to loosen the two M4 nuts on the rail that is used to clamp the installed cards.
4. Remove the rail.

5. Loosen the cross-head screw with which the slot bracket of a free ISA slot is fastened to the case of the controller.
6. Remove the slot bracket.
7. Carefully insert the add-on card into the ISA slot whose slot bracket you have removed.
8. Fasten the add-on card to the case of the controller with the cross-head screw.
9. Make sure that all other cards are still firmly in their slots.
10. Repeat steps 5 to 9 for all further add-on cards that you want to install.
11. Insert the rail with which the installed cards are clamped.
12. Place the top of the case so that the locking lugs of the top and the case engage on the front panel.
13. Fasten the top of the case to the rear panel of the controller with the two Torx screws.

5.5 Fastening the Mounting Brackets on the C-887.21

INFORMATION

Mounting brackets for fastening the case to a surface are included in the scope of delivery of the C-887.21 model. The arrangement and the dimensions of the mounting holes in the mounting brackets can be found in the dimensional drawing (p. 314).

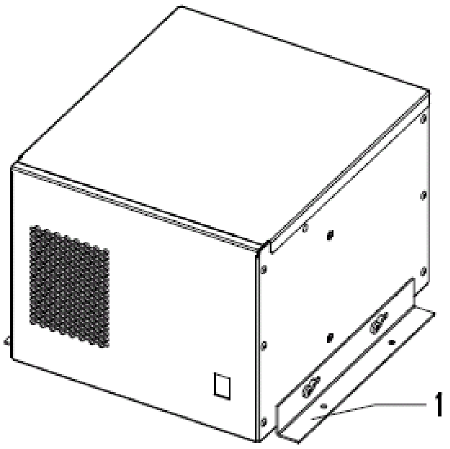


Figure 4: C-887.21 with mounting brackets

1 Mounting bracket

INFORMATION

The C-887.21 can also be mounted on a wall with a horizontal orientation of the mounting brackets, see the following figure.

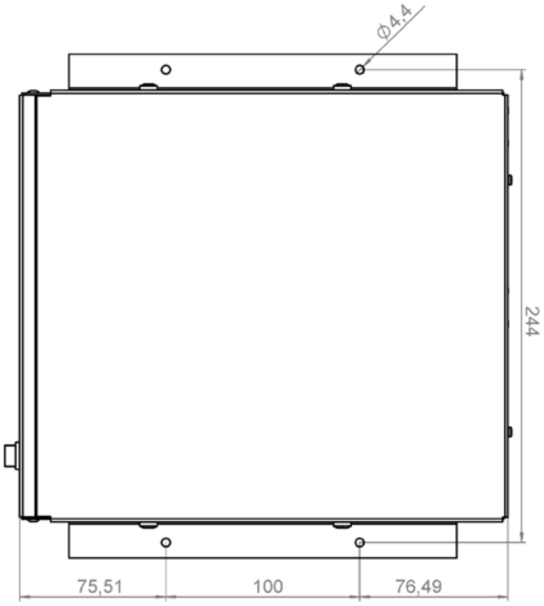


Figure 5: Vertical orientation of the C-887.21 for wall mounting

Prerequisite

✓ The C-887.21 is **not** connected to the power socket via the power cord.

Tools and accessories

- Two mounting brackets (included in the scope of delivery)
- Suitable Phillips-head screwdriver

Fastening the mounting brackets on the C-887.21

1. Loosen the two cross-head screws in the lower area of the side panel of the case until the screw heads protrude by approx. 2 mm.
2. Place the mounting bracket on the side panel of the case so that the screw heads protrude through the two longitudinal holes of the mounting bracket.
3. Push the mounting bracket onto the screws in the direction of the front panel or the rear panel of the case until the correct mounting position has been achieved.
4. Tighten the two screws.
5. Perform steps 1 to 4 for the other side panel of the case.

5.6 Connecting the C-887 to the Power Source

Prerequisites

✓ The C-887 is switched off, i.e. the standby switch is in the ⏻ position.

✓ The C-887 is installed near the power source so that the power plug can be quickly and easily disconnected from the mains.

Tools and accessories

- Supplied power cord
- Alternatively: power cord with sufficient ratings

Connecting the C-887 to the power source

1. Connect the power cord to the panel plug in the rear panel of the C-887.
2. Connect the C-887 to the power socket with the power cord.

5.7 Installing the Hexapod

5.7.1 Determining the Working Space and Permissible Load of the Hexapod

INFORMATION

The limit value for the load of the Hexapod determined with the simulation program is only valid when the servo mode is switched on for the axes of the moving platform. The maximum holding force when the servo mode is switched off is based on the self-locking of the actuators in the Hexapod struts and is lower than the limit value when the servo mode is switched on (see manual of the Hexapod).

INFORMATION

The limit value for the load of the Hexapod varies depending on the following factors:

- Activation state of the servo mode
- Installation position of the Hexapod
- Load to be moved: mass and position of the center of mass
- Position (translation and rotation coordinates) of the moving platform of the Hexapod
- Forces and moments that are applied to the moving platform of the Hexapod.

INFORMATION

The limits of the working space vary depending on the current position of the Hexapod (translation and rotation coordinates) and the current coordinates of the pivot point.

INFORMATION

In the simulation program, the direction of the applied forces and moments corresponds to the axes of an absolute reference coordinate system (also referred to as the "world coordinate system"). The XYZ coordinate system of the Hexapod can deviate from the absolute reference coordinate system depending on the mounting position of the base plate.

Prerequisite

- ✓ The following data for the configuration of your Hexapod system is known:
  - Planned installation position of the Hexapod
  - Load to be moved: mass and position of the center of mass on the moving platform
  - Forces and moments applied to the moving platform
  - Positions and orientations to be approached from the moving platform of the Hexapod during operation (translation and rotation coordinates)
- ✓ The PC is switched on.

Tools and accessories

- PC with Windows operating system on which the ***Hexapod Simulation Software*** simulation program is installed. For further information, see "Installing PC Software" (p. 43).

Determining the Working Space and Permissible Load of the Hexapod

1. Start the simulation program on the PC with the Start menu item ***Programs > PI > C-887 > Hexapod Simulation Software.***
2. Load the geometrical data of the Hexapod for which you want to determine the permissible load:

   a) In the main window, select the menu item ***Main > Load Geometry***.

   b) In the file selection window, go to the directory \PI\C-887\HexapodSimulation. The path in which the \PI directory is located was determined during the installation of the PC software, normally C:\Programs.

   c) In the file selection window, select the file with the extension .dat that corresponds to your Hexapod.

   d) In the file selection window, click ***Open***.

   The Hexapod is displayed in the main window.
3. If you do **not** use the default settings for the pivot point coordinates (R = S = T = 0): enter values for the pivot point coordinates R, S and T in the ***Position*** area.
4. In the ***Position*** area, enter the target positions for the axes X to W that correspond to a position of the moving platform to be approached during operation.

When the ***LIMIT*** display has a red background:
The moving platform **cannot** approach the position resulting from the given target position values.

- Change the target positions for the axes X to W until the ***LIMIT*** display has a green background.

5. Select the ***Forces*** tab in the main window.
6. On the ***Forces*** tab, enter the values for the load and the position of the center of mass as well as the applied forces and moments:
   - ***M***: mass of the load on the moving platform, in kg
   - ***rx M, ry M, rz M***: Position of the center of mass of the load, in relation to the origin of the XYZ coordinate system of the Hexapod (0,0,0), in mm
   - ***Fx, Fy, Fz***: applied force in the respective axis relative to the absolute reference coordinate system, in N.
   - ***rx F, ry, F, rz F***: Position of the application point of the force in relation to the origin of the XYZ coordinate system of the Hexapod (0,0,0), in mm
   - ***Mx, My, Mz***: applied torque in the respective axis of the absolute reference coordinate system, in Nm.
7. On the ***Forces*** tab, select the mounting position of the Hexapod in the ***Mounting*** area. The values for X, Y und Z refer to the XYZ coordinate system of the Hexapod.
8. Read the result of the load calculation:

   The simulation calculates the following values from your entries and displays them on the ***Forces*** tab underneath the ***Mounting*** area:

   - ***F1*** to ***F6***: forces that are applied to the individual struts of the Hexapod, in N
   - ***Fres***: resulting force in the axes X, Y and Z of the absolute reference coordinate system, in N.
   - ***Mres***: resulting moment in the axes X, Y and Z of the absolute reference coordinate system, in Nm.
9. Evaluate the result of the load calculation (see also the two following figures):

   Is at least one of the values for ***F1*** to ***F6*** shown in red?

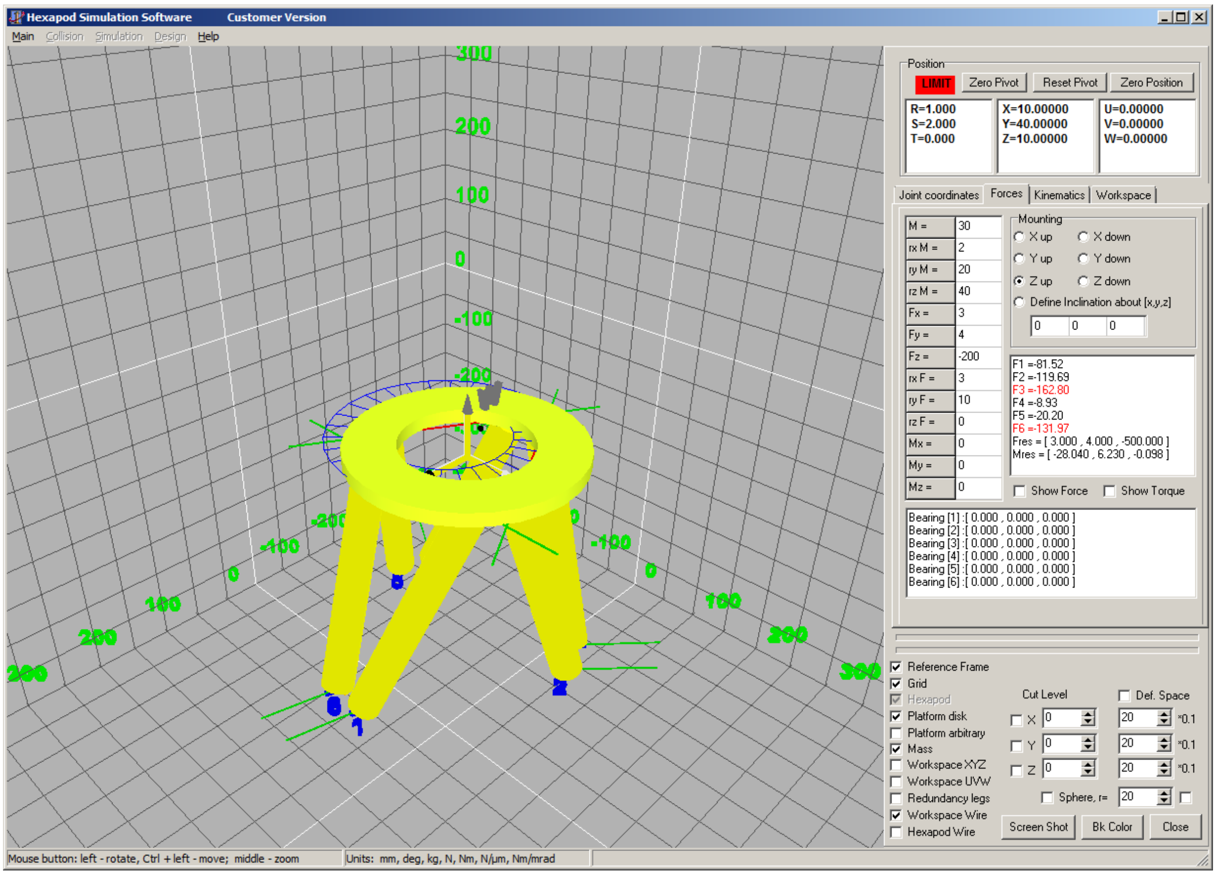
   If so:

- The permissible load of the Hexapod has been exceeded by the values that you have entered. The configuration for the installation of the Hexapod is **not** permissible for the current position of the moving platform. Change the values for the load and/or the position of the center of mass and/or the applied forces and moments and evaluate the result of the load calculation again.

If not:

- The load and the position of the center of mass as well as the applied forces and moments are permissible for the current position of the moving platform.

10. In the main window of the simulation program, select the ***Workspace*** tab.
11. Check the available working space from the current position:
    - On the ***Workspace*** tab, click on the ***Workspace Reserve*** button to find out the maximum distances that the axes can still move from the current position. The values are shown in the field to the right of the list of extreme values.
    - On the ***Workspace*** tab, click the ***Workspace XYZ*** button to graphically display the available working space for the axis combination XYZ from the current position in the main window.
    - On the ***Workspace*** tab, click the ***Workspace UVW*** button to graphically display the available working space for the axis combination UVW from the current position in the main window.
12. Repeat steps 4 to 11 for other positions of the moving platform to be approached during operation.


Hexapod Simulation Software   Customer Version
Main Collision Simulation Design Help
300
200
100
0
-100
-200
-200
-100
0
100
200
300
-100
0
100
200
300
1
2
6
Position
LIMIT
Zero Pivot
Reset Pivot
Zero Position
Joint coordinates
Forces
Kinematics
Workspace
Screen Shot
Bk Color
Close
Mouse button: left - rotate, Ctrl + left - move; middle - zoom
Units: mm, deg, kg, N, Nm, N/µm, Nm/mrad

R=1.000
S=2.000
T=0.000

X=10.00000
Y=40.00000
Z=10.00000

U=0.00000
V=0.00000
W=0.00000

M =	30
rx M =	2
ry M =	20
rz M =	40
Fx =	3
Fy =	4
Fz =	-200
rx F =	3
ry F =	10
rz F =	0
Mx =	0
My =	0
Mz =	0

Mounting
○ X up ○ X down
○ Y up ○ Y down
● Z up ○ Z down
○ Define Inclination about [x,y,z]
0 0 0

```
F1 =-81.52
F2 =-119.69
F3 =-162.80
F4 =-8.93
F5 =-20.20
F6 =-131.97
Fres = [ 3.000 , 4.000 , -500.000 ]
Mres = [ -28.040 , 6.230 , -0.098 ]
```

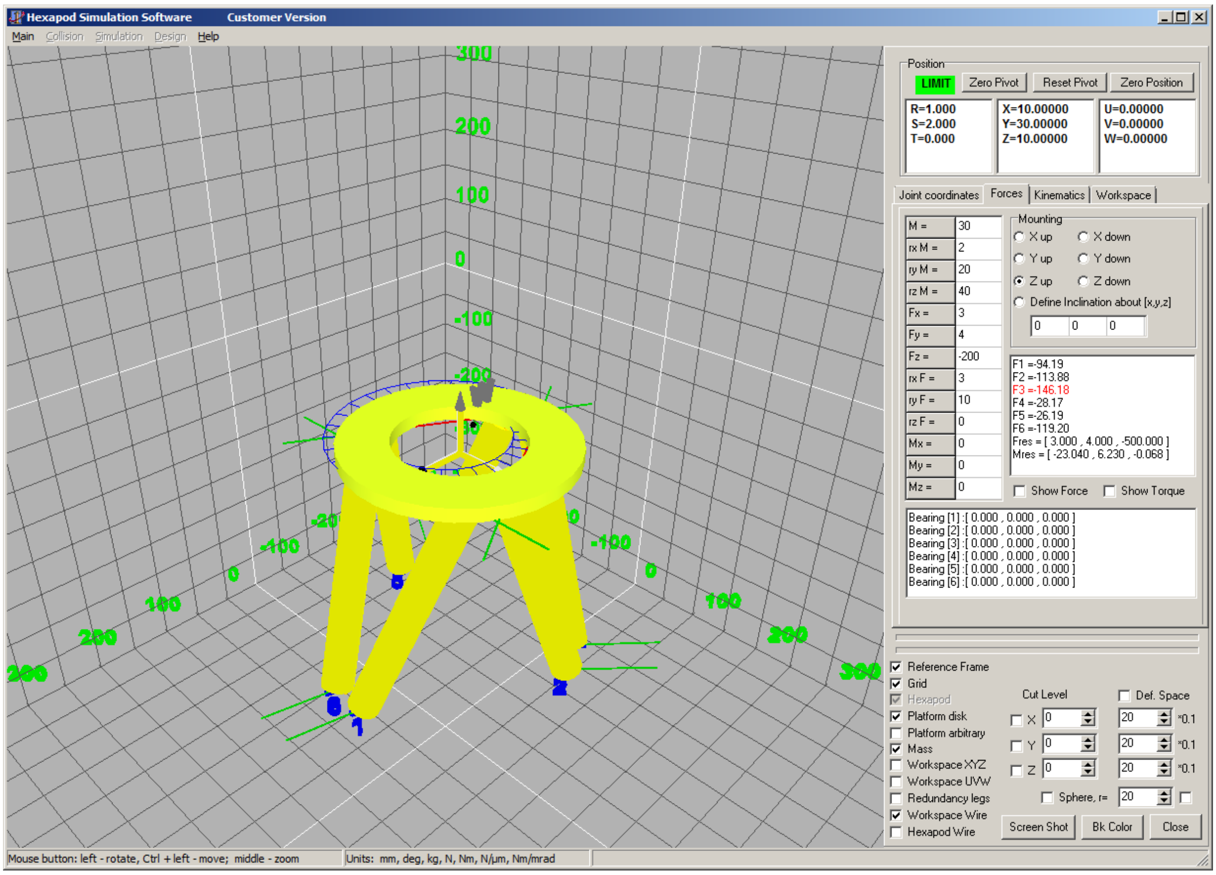
□ Show Force □ Show Torque

```
Bearing [1] :[ 0.000 , 0.000 , 0.000 ]
Bearing [2] :[ 0.000 , 0.000 , 0.000 ]
Bearing [3] :[ 0.000 , 0.000 , 0.000 ]
Bearing [4] :[ 0.000 , 0.000 , 0.000 ]
Bearing [5] :[ 0.000 , 0.000 , 0.000 ]
Bearing [6] :[ 0.000 , 0.000 , 0.000 ]
```

☑ Reference Frame
☑ Grid
☑ Hexapod
☑ Platform disk
□ Platform arbitrary
☑ Mass
□ Workspace XYZ
□ Workspace UVW
□ Redundancy legs
☑ Workspace Wire
□ Hexapod Wire

Cut Level □ Def. Space
□ X 0 20 *0.1
□ Y 0 20 *0.1
□ Z 0 20 *0.1
□ Sphere, r= 20 □

Figure 6: Position cannot be approached



R=1.000
S=2.000
T=0.000

X=10.00000
Y=30.00000
Z=10.00000

U=0.00000
V=0.00000
W=0.00000

M =	30
rx M =	2
ry M =	20
rz M =	40
Fx =	3
Fy =	4
Fz =	-200
rx F =	3
ry F =	10
rz F =	0
Mx =	0
My =	0
Mz =	0

Mounting
○ X up ○ X down
○ Y up ○ Y down
● Z up ○ Z down
○ Define Inclination about [x,y,z]
0 0 0

```
F1 =-94.19
F2 =-113.88
F3 =-146.18
F4 =-28.17
F5 =-26.19
F6 =-119.20
Fres = [ 3.000 , 4.000 , -500.000 ]
Mres = [ -23.040 , 6.230 , -0.068 ]
```

□ Show Force □ Show Torque

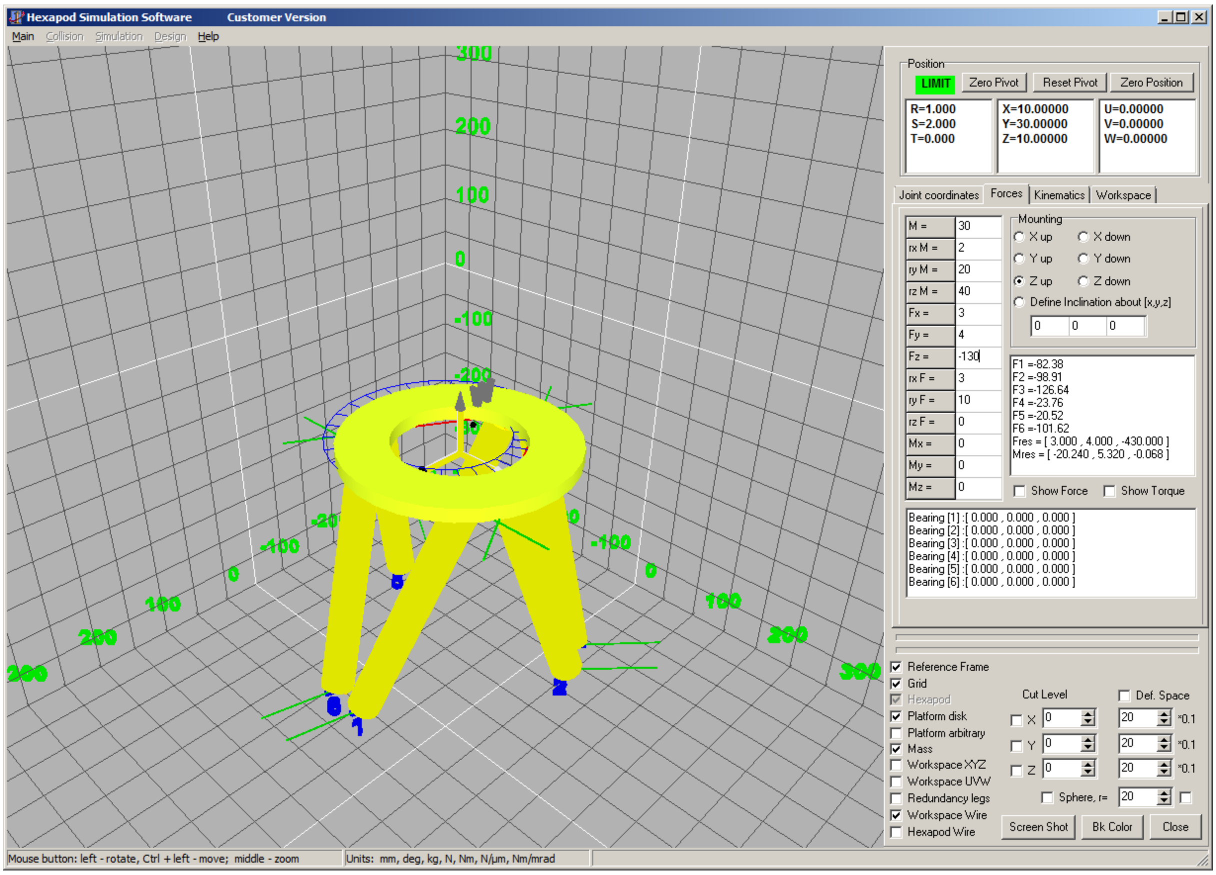
```
Bearing [1] :[ 0.000 , 0.000 , 0.000 ]
Bearing [2] :[ 0.000 , 0.000 , 0.000 ]
Bearing [3] :[ 0.000 , 0.000 , 0.000 ]
Bearing [4] :[ 0.000 , 0.000 , 0.000 ]
Bearing [5] :[ 0.000 , 0.000 , 0.000 ]
Bearing [6] :[ 0.000 , 0.000 , 0.000 ]
```

☑ Reference Frame
☑ Grid
☑ Hexapod
☑ Platform disk
□ Platform arbitrary
☑ Mass
□ Workspace XYZ
□ Workspace UVW
□ Redundancy legs
☑ Workspace Wire
□ Hexapod Wire

Cut Level □ Def. Space
□ X 0 20 *0.1
□ Y 0 20 *0.1
□ Z 0 20 *0.1
□ Sphere, r= 20 □

Mouse button: left - rotate, Ctrl + left - move; middle - zoom
Units: mm, deg, kg, N, Nm, N/µm, Nm/mrad

Figure 7: Permissible load for strut 3 of the Hexapod exceeded; configuration not permissible


Hexapod Simulation Software Customer Version
Main Collision Simulation Design Help
Position
LIMIT
Zero Pivot
Reset Pivot
Zero Position
Joint coordinates
Forces
Kinematics
Workspace
Mounting
300
200
100
0
-100
-200
-100
0
100
200
300
Screen Shot
Bk Color
Close
Mouse button: left - rotate, Ctrl + left - move; middle - zoom
Units: mm, deg, kg, N, Nm, N/µm, Nm/mrad

R=1.000
S=2.000
T=0.000

X=10.00000
Y=30.00000
Z=10.00000

U=0.00000
V=0.00000
W=0.00000

M = 30
rx M = 2
ry M = 20
rz M = 40
Fx = 3
Fy = 4
Fz = -130
rx F = 3
ry F = 10
rz F = 0
Mx = 0
My = 0
Mz = 0

○ X up ○ X down
○ Y up ○ Y down
● Z up ○ Z down
○ Define Inclination about [x,y,z]
0 0 0

```
F1 =-82.38
F2 =-98.91
F3 =-126.64
F4 =-23.76
F5 =-20.52
F6 =-101.62
Fres = [ 3.000 , 4.000 , -430.000 ]
Mres = [ -20.240 , 5.320 , -0.068 ]
```

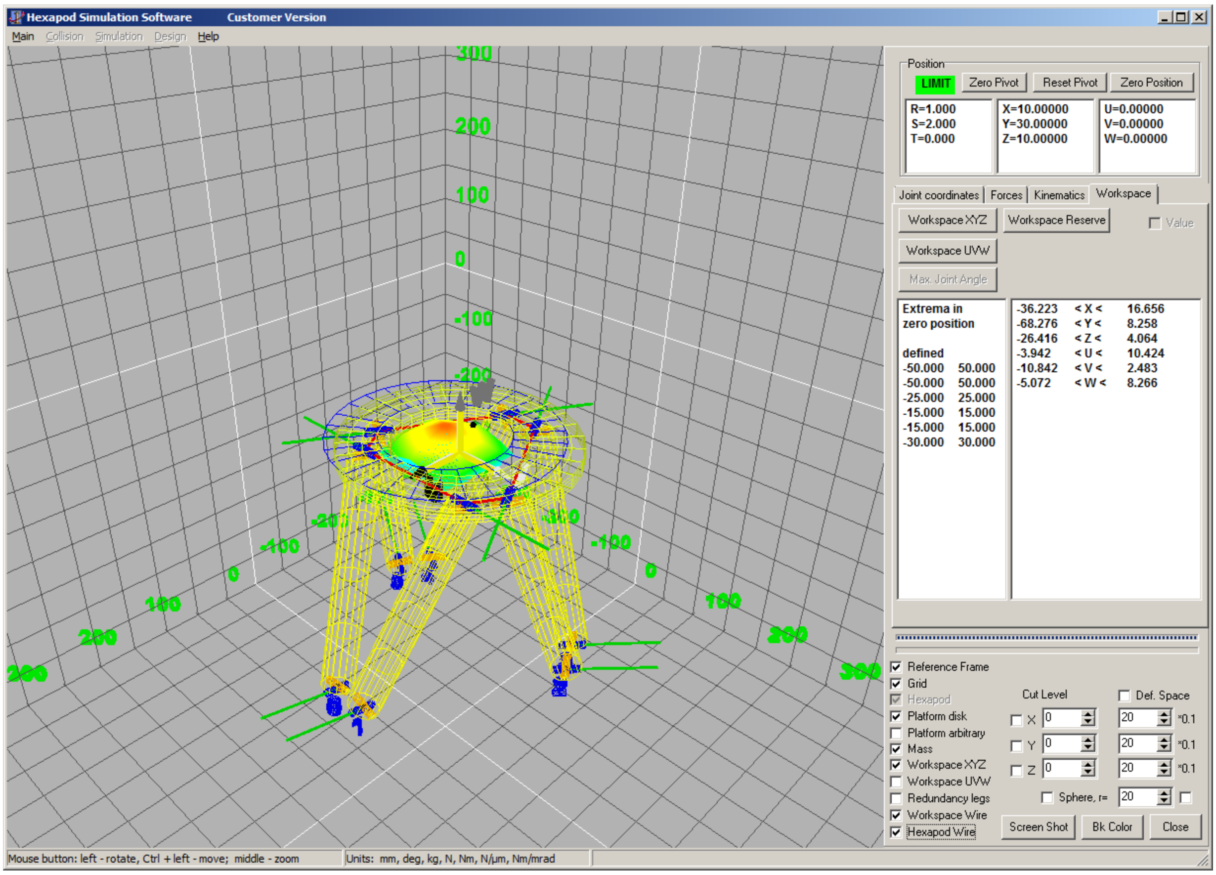
□ Show Force □ Show Torque

```
Bearing [1] :[ 0.000 , 0.000 , 0.000 ]
Bearing [2] :[ 0.000 , 0.000 , 0.000 ]
Bearing [3] :[ 0.000 , 0.000 , 0.000 ]
Bearing [4] :[ 0.000 , 0.000 , 0.000 ]
Bearing [5] :[ 0.000 , 0.000 , 0.000 ]
Bearing [6] :[ 0.000 , 0.000 , 0.000 ]
```

☑ Reference Frame
☑ Grid
☑ Hexapod
☑ Platform disk
□ Platform arbitrary
☑ Mass
□ Workspace XYZ
□ Workspace UVW
□ Redundancy legs
☑ Workspace Wire
□ Hexapod Wire

Cut Level □ Def. Space
□ X 0 20 *0.1
□ Y 0 20 *0.1
□ Z 0 20 *0.1
□ Sphere, r= 20 □

Figure 8: Configuration permissible, limit values for the load not exceeded



R=1.000
S=2.000
T=0.000

X=10.00000
Y=30.00000
Z=10.00000

U=0.00000
V=0.00000
W=0.00000

Extrema in zero position

defined

-50.000	50.000
-50.000	50.000
-25.000	25.000
-15.000	15.000
-15.000	15.000
-30.000	30.000

-36.223	< X <	16.656
-68.276	< Y <	8.258
-26.416	< Z <	4.064
-3.942	< U <	10.424
-10.842	< V <	2.483
-5.072	< W <	8.266

☑ Reference Frame
☑ Grid
☑ Hexapod
☑ Platform disk
□ Platform arbitrary
☑ Mass
☑ Workspace XYZ
□ Workspace UVW
□ Redundancy legs
☑ Workspace Wire
☑ Hexapod Wire

Cut Level □ Def. Space

□ X 0 20 *0.1
□ Y 0 20 *0.1
□ Z 0 20 *0.1
□ Sphere, r= 20 □

Mouse button: left - rotate, Ctrl + left - move; middle - zoom
Units: mm, deg, kg, N, Nm, N/µm, Nm/mrad

Figure 9: Available working space for the axis combination XYZ graphically displayed

INFORMATION

The optionally available PIVeriMove Software for the Collision Check makes it possible to mathematically check possible collisions between the Hexapod, load and environment. The use of the software is recommended when the Hexapod is located in a limited installation space and/or operated with a spatially limiting load. For details on the release and configuration of the PIVeriMove Software for the Collision Check see Technical Note C887T0002 (included in the scope of delivery of the software).

5.7.2 Grounding the Hexapod

➢ Follow the instructions in the manual of the Hexapod.

5.7.3 Mounting the Hexapod on a Surface

- ➢ Follow the instructions in the manual of the Hexapod.

5.7.4 Affixing the Load on the Hexapod

- ➢ Follow the instructions in the manual of the Hexapod.

5.8 Connecting the Hexapod to the C-887 with the Cable Set

Prerequisite

- ✓ The C-887 is switched off, i. e. the standby switch is in the ⏻ position.
- ✓ The cable set is connected to the Hexapod; see the Hexapod manual.

Tools and accessories

- Cable set that belongs to the Hexapod system (p. 17)
- Hexapod that belongs to the Hexapod system (p. 17)

Connecting the Hexapod to the C-887 with the cable set

1. Connect the data transmission cable to the **Hexapod** MDR68 socket in the rear panel of the C-887:

   a) Press the latches together on both sides of the connector.
   b) Insert the connector into the socket on the C-887.
   c) Check that the connector correctly fits.
   d) Release the latches.

2. Connect the M12 connector of the power supply cable for the Hexapod to one of the two 4-pin M12 sockets **2x 24 VDC Power Out** in the rear panel of the C-887.

   – Observe the mechanical coding of the connector and socket.

   – Do not use force.

3. If you are using a cable set with line driver boxes: Connect the M12 connector of the power supply cable for the Hexapod-side line driver box to one of the two 4-pin M12 sockets **2x 24 VDC Power Out** in the rear panel of the C-887.

   – Observe the mechanical coding of the connector and socket.

   – Do not use force

5.9 Connecting Stages for Axes A and B to the C-887.11

NOTICE

Damage if a wrong motor is connected!

Connecting a stage with a stepper motor can cause irreparable damage to the C-887.11.

- ➢ Only connect stages with DC motor and PWM amplifier to sockets **A** and **B** of the C-887.11.

INFORMATION

The C-887.11 has the sockets **A** and **B** (Sub-D 15 (f)) on the rear panel. A stage that is equipped with DC motor and PWM amplifier can be connected to each of these sockets. For details on suitable stages, see "Optional Accessories" (p. 25).

- ➢ After starting up the C-887, send the `VST?` command (p. 248) to obtain a list of suitable stages.

Prerequisite

- ✓ The C-887 is switched off, i. e. the standby switch is in the ⏻ position.
- ✓ You have read and understood the user manual of the stage.
- ✓ You have mounted the stage according to the description in the corresponding user manual.

Tools and accessories

- Stage from PI that is equipped with a DC motor and PWM amplifier, available as an optional accessory (p. 25)

Connecting stages for axes A and B to the C-887.11

1. Connect the stage that you want to command as axis A to socket **A** of the C-887.11.
2. Connect the stage that you want to command as axis B to socket **B** of the C-887.11.

5.10 Connecting the P-611.3SF NanoCube® Stage to the C-887.11

CAUTION

Dangerous voltage and residual charge on piezo actuators!

The P-611.3SF NanoCube® stage, which is part of the F-206.NCU option, is driven by piezo actuators. Temperature changes and compressive stresses can induce charges in piezo actuators. After being disconnected from the electronics, piezo actuators can also stay charged for several hours. Touching or short-circuiting the contacts in the connector of the P-611.3SF NanoCube® stage can lead to minor injuries. In addition, the piezo actuators can be destroyed by an abrupt contraction.

- ➢ Do **not** pull out the connector from the electronics during operation.
- ➢ Do **not** touch the contacts in the connector.
- ➢ Secure the connector of the stage with screws against being pulled out of the controller.

INFORMATION

When the E-760 controller card of the optional accessory F-206.NCU is installed in the C-887.11 (p. 45), the C-887.11 has a Sub-D 25 (f) socket on the rear panel. The settings of the E-760 controller card were calibrated with the P-611.3SF NanoCube® stage included in the scope of delivery of the optional accessory F -206.NCU before delivery.

- ➢ Only connect the P-611.3SF NanoCube® that you have received together with the installed E-760 controller card to the Sub-D 25 (f) socket.

Prerequisite

- ✓ The C-887 is switched off, i. e. the standby switch is in the ⏻ position.
- ✓ An E-760 controller card is installed in the C-887.11 (p. 45).
- ✓ You have read and understood the handling instructions for the P-611.3SF NanoCube® stage, see Technical Note F206T0021 (included in the scope of delivery of the F-206.NCU option).
- ✓ You have mounted the P-611.3SF NanoCube® according to the handling instructions.

Tools and accessories

- P-611.3SF NanoCube® stage that was delivered together with the installed E-760 controller card as part of the F-206 option
- If the cable of the P-611.3SF NanoCube® stage is too short: 1.5 m extension cable that was supplied together with the stage as part of the F-206.NCU option

Connecting the P-611.3SF NanoCube® stage to the C-887.11

➢ Connect the P-611.3SF NanoCube® stage to the Sub-D 25 (f) socket in the rear panel of the controller. If necessary, use the extension cable.

5.11 Connecting an Optical Signal Source to the C-887.11

NOTICE

Impairment from soiling!

Soiling, e.g. from dust, makes an FC connection for optical fibers inoperative.

➢ Make sure that unused FC connections have a protective cap.

➢ Keep the C-887 free from dust and dirt.

INFORMATION

If photometer cards (F-206.VVU and F-206.iiU options) are installed in the C-887.11 (p. 45), two FC connections for optical fibers are present for each photometer card on the rear panel. The photometer cards convert the optical signals fed in via FC connections into analog signals for processing by the C-887.

- The converters of the optical signals do not have any connection to the ISA bus via which the photometer card is connected to the C-887.
- The analog signals resulting from the conversion are fed into the C-887 by each being conveyed through an analog output of the photometer card to an analog input of the photometer card and transmitted from there via the ISA bus.

INFORMATION

The analog inputs are accessible in the firmware of the C-887 as input signal channels with a particular identifier, see "Commandable Items" (p. 28).

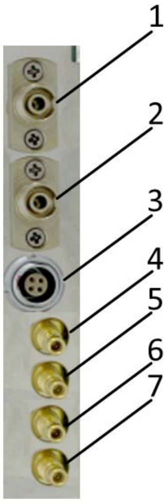
Prerequisite

✓ At least one of the following add-on cards is installed in the C-887.11 (p. 45):

- 2-channel photometer card, IR-range (F-206.iiU option)
- 2-channel photometer card, visible range (F-206.VVU option)

Tools and accessories

- One optical fiber per FC connection that is to be used; observe the specifications of the photometer card (p. 308).
- One SMB/SMB cable per FC connection that is to be used (included in the scope of delivery of the photometer card)



1 *FC connection for optical input signal 1*
2 *FC connection for optical input signal 2*
3 *Reserved*
4 *SMB socket of analog input 1*
5 *SMB socket of analog input 2*
6 *SMB socket of analog output 1*
7 *SMB socket of analog output 2*

Figure 10: Connections of a photometer card

Connecting an optical signal source

1. Remove the protective cap from an FC connection of a photometer card on the rear panel of the C-887.11.
2. Connect the optical fiber to the FC connection via its FC connector.
3. Use a SMB/SMB cable to connect the analog output of the FC connection with the corresponding analog input of the photometer card.

The assignment of the FC connections to the SMB sockets can be found in the figure above.

4. Repeat steps 1 to 3 for all further optical fibers to be connected.

5.12 Connecting an Analog Signal Source to the C-887.11

INFORMATION

The analog inputs are accessible in the firmware of the C-887 as input signal channels with a particular identifier, see "Commandable Items" (p. 28).

5.12.1 Connecting via a Photometer Card

INFORMATION

Each photometer card that is installed in the C-887.11 (F-206.VVU and F-206.iiU options) provides two SMB sockets for analog input signals on the rear panel. Each SMB socket for analog input signals can be used for one of the following purposes:

- Input for the signal converted from optical to analog, which is fed into the photometer card via an FC connection; for details see "Connecting an Optical Signal Source to the C-887.11" (p. 63).

Or

- Input for the analog signal of an external sensor

Prerequisite

✓ At least one of the following add-on cards is installed in the C-887.11 (p. 45):

- 2-channel photometer card, IR-range (F-206.iiU option)
- 2-channel photometer card, visible range (F-206.VVU option)

Tools and accessories

- Suitable signal source with corresponding connector: 0 to 10 V, maximum 10 kHz, SMB connector

Connecting an analog signal source a photometer card in the C-887.11

- If you want to feed in the signal of an external sensor: Connect the signal source to one of the two SMB sockets marked with **IN** on the photometer card (positions 4 and 5 in the figure in "Connecting an Optical Signal Source to the C-887.11" (p. 63)).
- If you want to feed the signal of an FC connection that has been converted from optical to analog into the SMB socket of a photometer card, follow the instructions in "Connecting an Optical Signal Source to the C-887.11" (p. 63).

5.12.2 Connecting via an E-760 Controller Card

INFORMATION

An E-760 controller card that is installed in the C-887.11 (F-206.NCU option) provides an unlabelled RJ45 socket on the rear panel. An analog signal can be fed into the C-887.11 via pin 7 of this socket; for the resolution of the A/D converter and pin assignment, see p. 321.

Prerequisite

✓ An E-760 controller card (F-206.NCU option) is installed in the C-887.11 (p. 45).

Tools and accessories

- Suitable signal source with corresponding connector: -5 V to +5 V on pin 7 of an RJ45 connector

Connecting an analog signal source to the E-760 controller card in the C-887.11

- Connect the signal source via the unlabelled RJ45 socket of the E-760 controller card. The RJ45 socket is located on the E-760 controller card underneath the Sub-D 25 (f) socket.

5.13 Connecting the C-887 to a PC

The communication between the C-887 and a PC can be used to configure the C-887 and send motion commands with the commands of the GCS. The C-887 has the following interfaces for this purpose:

- TCP/IP
- RS-232 interface

In this section, you learn how to establish the corresponding cable connections between the C-887 and a PC. All further steps that are necessary for establishing communication between the C-887 and a PC can be found in "Establishing Communication via a TCP/IP Interface" (p. 79) and in "Establishing Communication via an RS-232 Interface" (p. 88).

5.13.1 Connecting the C-887 via the TCP/IP Interface

Prerequisites

✓ If the C-887 is to be directly connected to the PC:
The PC has a free Ethernet connection socket.

✓ If the C-887 and a PC are to be operated together in a network:
A free access point to the network is available for the C-887; a suitable hub or switch is connected to the network for this purpose if necessary.

Tools and accessories

- If the C-887 is to be directly connected to the PC:
  cross-over network cable (C-815.563 included in the scope of delivery)
- If the C-887 is to be connected to a network access point: straight-through network cable (C-815.553 included in the scope of delivery)

Connecting the C-887 directly to the PC

➢ Connect the **LAN** RJ45 socket on the rear panel of the C-887 to the RJ45 Ethernet connection socket of the PC via the cross-over network cable.

Connecting the C-887 to the network in which the PC is also located

➢ Connect the **LAN** RJ45 socket on the rear panel of the C-887 with the network access point via the straight-through network cable.

5.13.2 Connecting the C-887 via the RS-232 Interface

Prerequisite

✓ The PC has a free RS-232 interface (also called a "serial interface" or "COM port", e. g. COM1 or COM2).

Tools and accessories

- RS-232 null-modem cable (C-815.34 included in the scope of delivery)

Connecting the C-887 to the PC

➢ Connect the **RS-232** panel plug on the rear panel of the C-887 and the RS-232 interface of the PC (one Sub-D 9(m) panel plug) with the null-modem cable.

5.14 Connecting the Keyboard, Mouse and Monitor to the C-887

A keyboard, a mouse and a monitor can be used to work with the user interface of the C-887; for details see "Using the User Interface of the C-887" (p. 74). The C-887 has the following peripheral interfaces for that purpose:

- VGA connection
- USB connections

Tools and accessories

- Keyboard with USB connection, included in the scope of delivery (p. 24)
- Mouse with USB connection, not included in the scope of delivery
- Monitor with VGA connection, not included in the scope of delivery

Connecting the keyboard, mouse and monitor to the C-887

➢ Connect the keyboard to one of the **Universal Serial Bus** USB connections on the rear panel of the C-887.

➢ Connect the mouse to one of the **Universal Serial Bus** USB connections on the rear panel of the C-887.

➢ Connect the monitor to the **VGA** socket on the rear panel of the C-887.

6 Start-Up

In this Chapter

6.1 General Notes on Start-Up

CAUTION

⚠ **Risk of crushing by moving parts!**

There is a risk of minor injuries caused by crushing which can occur between the moving parts of the Hexapod and a stationary part or obstacle.

➢ Keep your fingers away from areas where they can get caught by moving parts.

NOTICE

! **Incorrect configuration of the Hexapod controller!**

The configuration data used by the Hexapod controller (e.g. geometrical data and servo-control parameters) must be adapted to the Hexapod. If incorrect configuration data is used, the Hexapod can be damaged by uncontrolled motions or collisions. The configuration data is adapted before delivery.

➢ Check whether the Hexapod controller matches the Hexapod. A label on the rear panel of the controller indicates for which Hexapod the controller is intended.

➢ When you have established the communication via TCP/IP (p. 79) or RS-232 (p. 88) or use the user interface of the C-887 (p. 74), send the `CST?` command (p. 152). The response shows the Hexapod to which the controller is adapted.

➢ Only operate the Hexapod with a Hexapod controller whose configuration data is adapted to the Hexapod.

NOTICE

Damage from collisions!

Collisions can damage the Hexapod, the load to be moved and the environment.

- ➢ Make sure that no collisions between the Hexapod, the load to be moved and the environment are possible in the working space of the Hexapod.
- ➢ Do not place any objects in areas where they can get caught by moving parts.
- ➢ If the Hexapod controller malfunctions, stop the motion immediately.

NOTICE

Damage from transport lock that has not been removed!

Damage can occur to the Hexapod if the transport lock (p. 39) of the Hexapod has not been removed and a motion is commanded.

- ➢ Remove the transport lock before you start up the Hexapod system.

NOTICE

Damage from unintentional position changes!

The limit value for the load of the Hexapod determined with the simulation program is only valid when the servo mode is switched on (p. 52) for the axes of the moving platform. The maximum holding force when the servo mode is switched off is based on the self-locking of the actuators in the Hexapod struts and is lower than the limit value when the servo mode is switched on (see manual of the Hexapod).

When the actual load of the Hexapod exceeds the maximum holding force based on the self-locking of the actuators, unintentional position changes of the Hexapod can occur in the following cases:

- Switching off the C-887
- Rebooting the C-887
- Switching off the servo mode for the axes of the moving platform of the Hexapod.

As a result, collisions are possible between the Hexapod, the load to be moved and the environment. Collisions can damage the Hexapod, the load to be moved or the environment.

- ➢ Make sure that the actual load of the moving platform of the Hexapod does not exceed the maximum holding force based on the self-locking of the actuators before you switch off the servo mode, reboot or switch off the C-887.

NOTICE

❗

Damage from collisions during the reference move!

During a reference move, the Hexapod moves unpredictably. **No** collision check or prevention takes place, even when a configuration for collision prevention has been stored on the C-887 with the PIVeriMove Software for the Collision Check.

As a result, collisions are possible between the Hexapod, the load to be moved and the environment. Collisions can damage the Hexapod, the load to be moved and the environment.

➢ Make sure that no collisions between the Hexapod, the load to be moved and the environment are possible during the reference move of the Hexapod.

➢ Do not place any objects in areas where they can get caught by moving parts during the reference move.

➢ After a successful reference move, supply a command for the corresponding target position in order to return to the reference position (default: zero position) from any given position. Do **not** start a new reference move.

NOTICE

❗

Damage from uncontrolled motion of the Hexapod!

In the following cases, the velocity and acceleration of the moving platform of the Hexapod are not specified by the trajectory generator of the C-887:

- The ***Trajectory Source*** parameter (ID 0x19001900) has the value 1.
- The ***Trajectory Source*** parameter has the value 0, and the Hexapod (axes X, Y, Z, U, V, W) is still moving while a new motion command is sent. The previous target position is thereby overwritten without the velocity and acceleration of the moving platform of the Hexapod being recalculated.

The platform of the Hexapod then moves on an undefined path. On this undefined path, collisions with the environment of the Hexapod are possible. Collisions can damage the Hexapod, the load to be moved and the environment.

When the ***Trajectory Source*** parameter has the value 1:

➢ Only set target positions whose distance from each other is maximally as large as the value of the ***Path Control Step Size (mm)*** parameter (ID 0x19001504) with consecutive `MOV` commands.

When the ***Trajectory Source*** parameter has the value 0:

➢ Avoid sending new target positions while the Hexapod (axes X, Y, Z, U, V, W) is still moving.

➢ If new target positions have to be sent while the Hexapod is still moving (axes X, Y, Z, U, V, W): Only use motion commands to set target positions that maximally deviate from the current position by the value of the ***Path Control Step Size (mm)*** parameter (ID 0x19001504).

NOTICE

Damage from uncontrolled motion of the Hexapod!

If an E-760 controller card is installed in the C-887.11 (F-206.NCU option), the P-611.3SF NanoCube® stage must be correctly connected in order to start the operating system and the firmware. If the P-611.3SF NanoCube® stage is not connected when the C-887.11 is switched on or rebooted, the Hexapod system can demonstrate the following behavior:

- The operating system and the firmware of the C-887.11 do not start.
- The operating system and the firmware of the C-887.11 start, and the Hexapod carries out uncontrolled motions. As a result, collisions are possible between the Hexapod, the load to be moved and the environment. Collisions can damage the Hexapod, the load to be moved and the environment.

➢ Before switching on or rebooting the C-887.11, make sure that the P-611.3SF NanoCube® stage is connected to the E-760 controller card via the Sub-D 25 (f) socket (p. 62).

INFORMATION

The optionally available PIVeriMove Software for the Collision Check makes it possible to mathematically check possible collisions between the Hexapod, load and environment. The use of the software is recommended when the Hexapod is located in a limited installation space and/or operated with a spatially limiting load. For details on the release and configuration of the PIVeriMove Software for the Collision Check see Technical Note C887T0002 (included in the scope of delivery of the software).

INFORMATION

The communication between the C-887 and a PC can be used to configure the C-887 and send motion commands with the commands of the GCS.

- Communication is possible via the RS-232 interface without further settings.
- For communication via TCP/IP, it may be necessary to adjust the interface parameters accordingly once (p. 81).

INFORMATION

The communication interfaces of the C-887 (e.g. user interface, TCP/IP and RS-232 interfaces) are active at the same time. Commands are executed in the order that the complete command lines arrive. The simultaneous use of several communication interfaces can cause problems with the PC software, however.

- Always only use one interface of the C-887.

INFORMATION

When switched on or rebooted, the C-887 automatically switches on the servo mode for all axes. When the servo mode is switched off for the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W), it is automatically switched on when the reference move is started.

INFORMATION

For the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W), motions can be commanded only after a successful reference move (also referred to as "initialization").

A common reference move is always performed for the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W).

6.2 Switching on the C-887

Prerequisites

- ✓ You have read and understood the General Notes on Start-Up (p. 69).
- ✓ The C-887 has been installed properly (p. 41).

Switching on the C-887

- Move the standby switch on the front panel of the C-887 into the | position.

The standby switch lights up green.

The C-887 starts the operating system and the firmware. The starting procedure lasts approx. 36 seconds; the beginning and end are each indicated by a signal tone.

During the starting procedure, the C-887 performs the following actions, among others:

- Switching on the servo mode for all axes
- Activation of the factory default settings for the volatilely configurable parameters
- Activation of the settings stored in the non-volatile memory for the interface parameters
- Starting the start-up macro, if present

6.3 Using the User Interface of the C-887

The user interface of the C-887 can be used for the following purposes:

- Sending commands in the ***Terminal Panel*** (p. 76)
- Starting motions of the Hexapod platform in the main window of the user interface (p. 77)
- Assigning a stage type for axes A and B — only C-887.11 (p. 78)
- Getting voltages at the analog inputs — only C-887.11 (p. 79)
- Performing tests for Hexapod struts (p. 79)

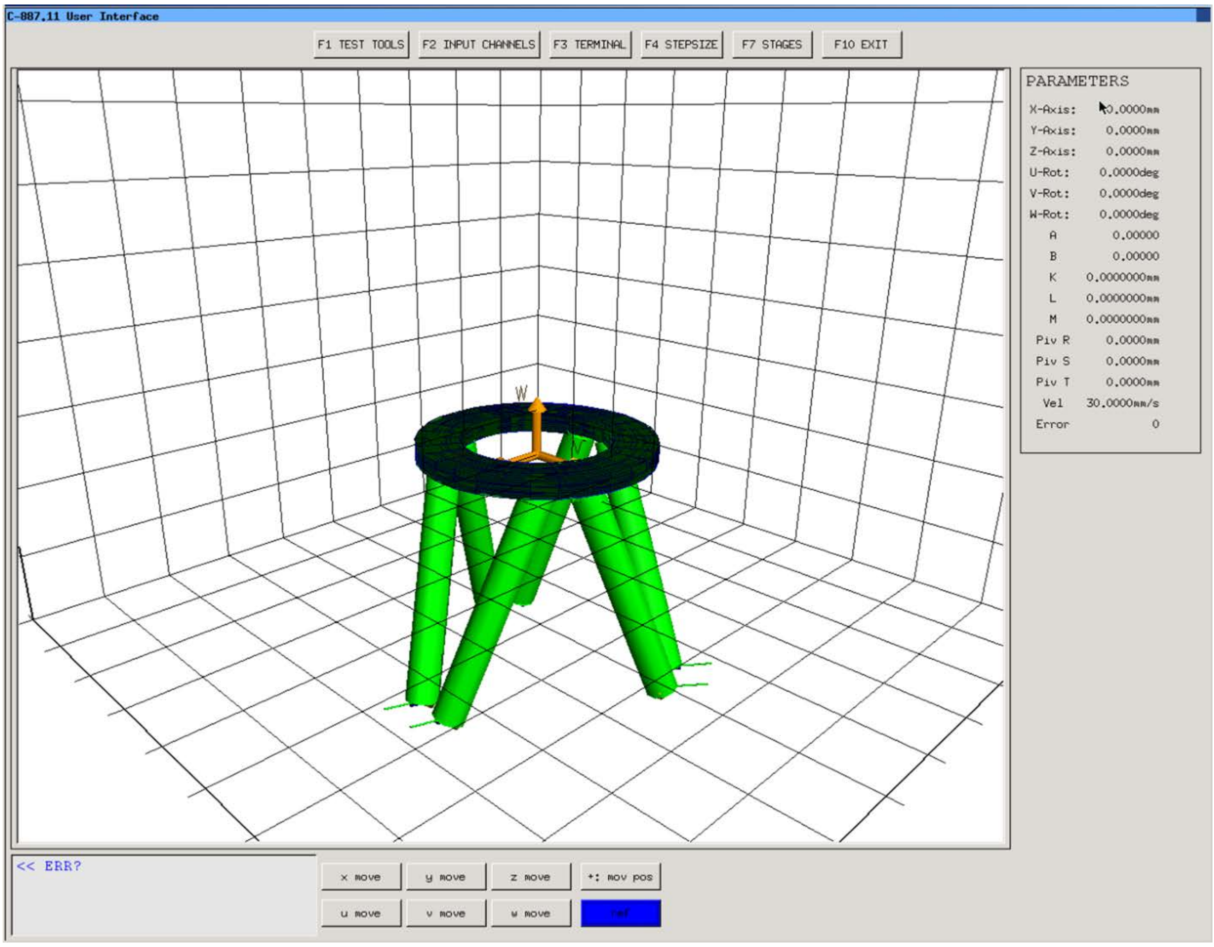
INFORMATION

The operating system of the C-887 may not recognize the layout of the connected keyboard until the C-887 has been rebooted or after the keyboard has been removed and reconnected while the C-887 is switched on.

➢ If possible, only use the US keyboard included in the scope of delivery.

INFORMATION

When the starting procedure of the C-887 has finished (p. 73), the user interface of the C-887 automatically opens.



```
PARAMETERS
X-Axis:      0.0000mm
Y-Axis:      0.0000mm
Z-Axis:      0.0000mm
U-Rot:       0.0000deg
V-Rot:       0.0000deg
W-Rot:       0.0000deg
  A          0.00000
  B          0.00000
  K      0.0000000mm
  L      0.0000000mm
  M      0.0000000mm
 Piv R       0.0000mm
 Piv S       0.0000mm
 Piv T       0.0000mm
  Vel     30.0000mm/s
 Error             0
```

```
<< ERR?
```

Figure 11: Main window of the user interface of the C-887

Prerequisites

- ✓ You have read and understood the General Notes on Start-Up (p. 69).
- ✓ A keyboard and a monitor are connected to the C-887 (p. 68).
- ✓ Optional: A mouse is connected to the C-887 (p. 68).
- ✓ The C-887 is switched on and the starting procedure of the C-887 has finished (p. 73).
- ✓ The monitor is switched on and shows the main window of the user interface (see above figure).

Sending commands in the Terminal Panel

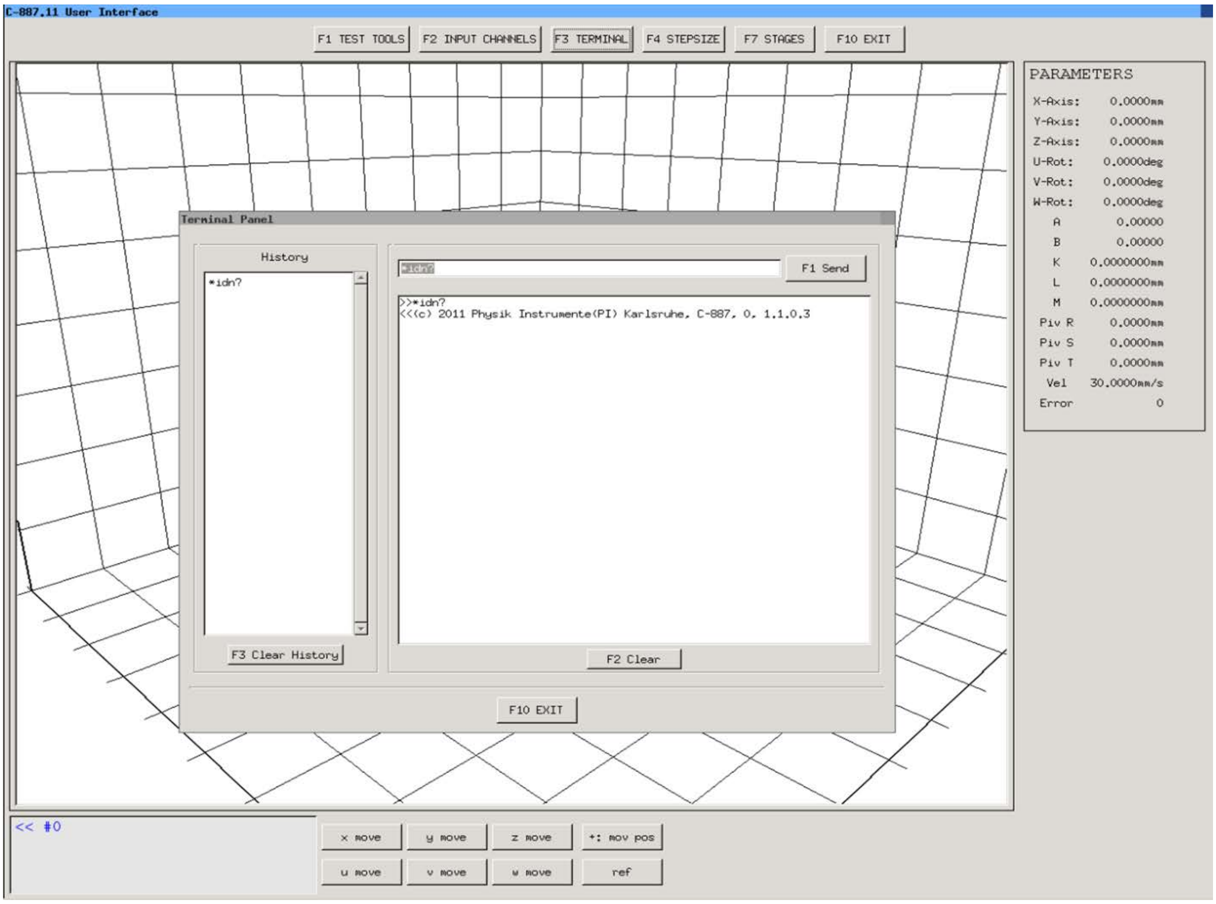
1. Press the F3 key on the keyboard or click the ***F3 TERMINAL*** button in the main window of the user interface to open the ***Terminal Panel*** window.

   The ***Terminal Panel*** window contains the following elements:

   – Input line for commands (on the left of the ***F1 Send*** button)

   – Display field for sent commands and the responses of the C-887 (underneath the input line)

   – ***History*** field with a list of sent commands (to the left of the input line and display field)



C-887.11 User Interface

Terminal Panel

History

*idn?

```
>>*idn?
<<(c) 2011 Physik Instrumente(PI) Karlsruhe, C-887, 0, 1.1.0.3
```

PARAMETERS

X-Axis:	0.0000mm
Y-Axis:	0.0000mm
Z-Axis:	0.0000mm
U-Rot:	0.0000deg
V-Rot:	0.0000deg
W-Rot:	0.0000deg
A	0.00000
B	0.00000
K	0.0000000mm
L	0.0000000mm
M	0.0000000mm
Piv R	0.0000mm
Piv S	0.0000mm
Piv T	0.0000mm
Vel	30.0000mm/s
Error	0

<< #0

2. Send a command:

   a) Type the command in the input line in the ***Terminal Panel*** window.
   b) Press the F1 key or ENTER on the keyboard or click the ***F1 Send*** button.

3. When the sent command requires a response from the C-887: Read the response of the C-887 in the display field of the ***Terminal Panel*** window.

INFORMATION

Sending the following query commands in the ***Terminal Panel*** window gives you an overview of the system configuration:

- `*IDN?` (p. 144) gets the device identification string of the C-887.
- `VER?` (p. 244) gets the versions of the firmware components.
- `IFS?` (p. 195) gets the values of the interface parameters in the non-volatile memory.
- `MAC DEF?` (p. 201) gets the start-up macro.
- `SAI?` (p. 221) gets the axis identifiers.
- `CST?` (p. 152) gets the names of the connected stage types.
- `SVO?` (p. 236) gets the servo mode.
- `FRF?` (p. 175) gets whether the axes are referenced.

Starting motions of the Hexapod platform in the main window of the user interface

1. Make sure that the main window of the user interface is displayed.
2. Press the I or R button on the keyboard or click the ***ref*** button in the main window of the user interface to start a reference move of the Hexapod (the corresponding command is `FRF` (p. 174)).
3. If you want to change the step size for motions of the platform of the Hexapod (default value is 0.01 mm or degrees; the corresponding command is `SST` (p. 231)):
   a) Press the F4 key on the keyboard or click the ***F4 STEPSIZE*** button in the main window of the user interface to open the ***step size configuration*** window.
   b) Type in values for the step sizes of the individual axes. The permissible range of values for the step size is 0.0001 to 0.5 mm or degrees.
   c) Return to the main window of the user interface with the F1 key (save values) or F10 key (do not save values).

4. If you want to reverse the direction of motion of the axes: Press the + or - key on the keyboard or click the ***+: mov pos*** or ***-:mov pos*** button in the main window of the user interface (labeling of the button depends on the currently set orientation).

5. Press the X, Y, Z, U, V, W keys on the keyboard or click the ***x move***, ***y move***, ***z move***, ***u move***, ***v move***, ***w move*** buttons in the main window of the user interface to start motions of the corresponding axis of the platform of the Hexapod (the corresponding command is `MOV` (p. 206)). Each press of a key or click increases the target position by the set step size.

Assigning a stage type for axes A and B — only C-887.11

If stages for the axes A and B are subsequently added to the Hexapod system, the appropriate stage types must be assigned to the axes A and B each time that the C-887.11 is switched on and rebooted; for further information see "Starting Motions of the Axes A, B, K, L and M - Only C-887.11" (p. 94).

1. Press the F7 key on the keyboard or click the ***F7 STAGES*** button in the main window of the user interface to open the ***Select additional stages*** window.

2. Select the suitable stage type from the ***Stage database entries*** list with the arrow keys on the keyboard or by clicking.

3. Allocate the selected stage type to an axis (the corresponding command is `CST` (p. 152)):

   – Press the A key or click ***'A' assign to channel A -->*** for allocation to axis A.

   – Press the B key or click ***'B' assign to channel B -->*** for allocation to axis B.

4. If you want to allocate the stage type NOSTAGE to an axis and thereby deactivate the axis:

   – Press the F1 key or click ***<-- F1 disconnect channel A*** to deactivate axis A.

   – Press the F2 key or click ***<-- F2 disconnect channel B*** to deactivate axis B.

5. Return to the main window of the user interface with the F10 key.

Getting voltages at the analog inputs — only C-887.11

1. Press the F2 key on the keyboard or click the ***F2 INPUT CHANNELS*** button in the main window of the user interface to open the ***InputChannels*** window.
2. Read the voltage values for the analog inputs (the corresponding command is `TAV?` (p. 238)). The ***InputChannels*** window only displays the actually present analog input channels (p. 28).
3. Return to the main window of the user interface with the Esc key.

Performing tests for Hexapod struts

The ***hexapod strut tests*** window allows the testing of Hexapod struts 1 to 6 for error diagnosis (accessible from the main window of the user interface with the F1 key or the ***F1 TEST TOOLS*** button).

➢ For tests of the Hexapod struts with the user interface, follow the instructions in "Carrying out a Strut Test" (p. 291).

INFORMATION

When the E-760 controller card of the optional accessory F-206.NCU is installed in the C-887.11 (p. 45), measurements of the impulse response and the step response for the axes K, L, and M of the P-611.3SF NanoCube® stage can also be carried out in the ***hexapod strut tests*** window (p. 100).

6.4 Establishing Communication via the TCP/IP Interface

Adaptation of the interface parameters

Before communication is established, it can be necessary to adapt the factory settings of the interface parameters once. The following interface parameters for the TCP/IP communication can be adapted in the non-volatile memory of the C-887 with the `IFS` command (p. 195):

Interface parameter	Factory setting	Note
Default IP address (IPADR)	192.168.1.28:50000	Allows the definition of a static address. This static address is **not** used when the C-887 is configured to obtain the IP address via a DHCP server or AutoIP (factory setting of the start-up behavior for configuration of the IP address). If the static address is to be used: ▪ The start-up behavior must be changed so that the C-887 uses the IP address defined with “IPADR". ▪ The IP addresses and subnet masks of the C-887 and PC as well as all other network participants must be coordinated with each other. For details, see "Preparing the PC and C-887 for Using Static IP Addresses" (p. 81).
Start-up behavior for configuring the IP address for TCP/IP communication (IPSTART)	DHCP or AutoIP is used to obtain the IP address	The IP address of the C-887 is obtained via DHCP or automatically configured using AutoIP with the factory setting of the start-up behavior. The factory setting of the start-up behavior only has to be changed if the network participants are to use static addresses instead.
Subnet mask (IPMASK)	255.255.255.0	The factory setting of the subnet mask may have to be changed if the network participants are to use static addresses. For details, see "Preparing the PC and C-887 for Using Static IP Addresses" (p. 81).

After switching on or rebooting the C-887

The starting procedure of the C-887 must be finished before the communication between the C-887 and PC can be established. The starting procedure takes around 36 seconds (second signal tone indicates the end).

When the IP addresses of the network participants are configured with AutoIP, it will take up to 2 minutes after the end of the starting procedure of the C-887 (p. 73) until communication is possible via TCP/IP.

Connection of the network cable when the controller is switched on

The establishment of communication via TCP/IP can fail if the network cable was connected to the **LAN** RJ45 socket on the rear panel of the C-887 while the C-887 was switched on.

➢ If the establishment of communication fails, switch the C-887 off and back on again while the network cable is plugged in.

Port setting

For communication via TCP/IP, the C-887 only has a single unchangeable port (50000) available that cannot be used by more than one connection at a time.

6.4.1 Preparing the PC and C-887 for Using Static IP Addresses

If a network does not have a DHCP server or the C-887 is directly connected to the Ethernet connection socket of the PC **and** static IP addresses are to be used, the following adaptations of the interface parameters are necessary:

- Set the start-up behavior for configuring the IP address of the C-887 so that a static address is used
- Coordinate IP addresses and subnet masks of the C-887 and PC as well as all other network participants with each other.

To adapt IP addresses and subnet masks, you can choose one of the two following options:

- Adapt the PC settings and if necessary the settings of other network participants. The settings of the C-887 remain unchanged.
- Adapt the settings of the C-887. The PC settings and if necessary the settings of other network participants remain unchanged.

Prerequisite

✓ You have established communication between the C-887 and the PC via RS-232 in order to determine the settings of the C-887 and to be able to change them if necessary (p. 88).

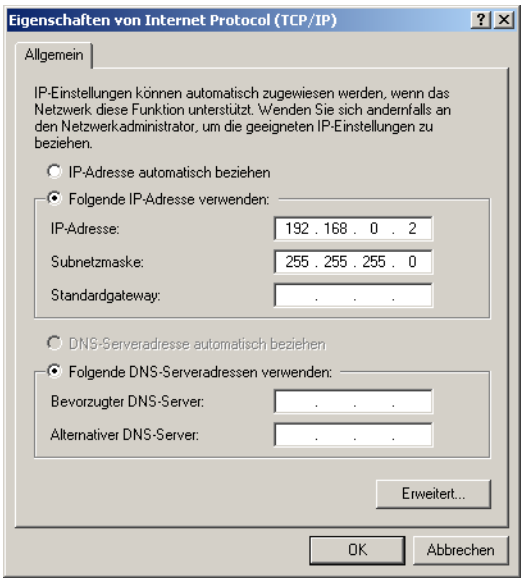
or:

✓ You have made preparations for using the user interface of the C-887 (p. 74), in order to determine the settings of the C-887 and be able to change them if necessary.

Determining the IP address and subnet mask of the PC

1. Open the window in which the properties of the Internet protocol TCP/IP are displayed and set, in a suitable way on your PC. The necessary steps depend on the operating system used.

   If your operating system distinguishes between Internet protocol version 4 (TCP/IPv4) and version 6 (TCP/IPv6) (e.g. Windows 7), open the window for version 4.



Eigenschaften von Internet Protocol (TCP/IP)

Allgemein

IP-Einstellungen können automatisch zugewiesen werden, wenn das Netzwerk diese Funktion unterstützt. Wenden Sie sich andernfalls an den Netzwerkadministrator, um die geeigneten IP-Einstellungen zu beziehen.

○ IP-Adresse automatisch beziehen

● Folgende IP-Adresse verwenden:

IP-Adresse: 192 . 168 . 0 . 2

Subnetzmaske: 255 . 255 . 255 . 0

Standardgateway: . . .

○ DNS-Serveradresse automatisch beziehen

● Folgende DNS-Serveradressen verwenden:

Bevorzugter DNS-Server: . . .

Alternativer DNS-Server: . . .

Figure 12: "Internet Protocol (TCP/IP) Properties" window with example settings (not necessarily suitable for your system)

   The figure shows example settings that may not apply to your system.

2. Write down the settings.

Determining the IP address and subnet mask of the C-887, adapting the start-up behavior of the C-887

1. Prepare for entering commands:

   – If you have established communication between the C-887 and PC via the RS-232 interface, open the window for entering commands in the used program:

   or

   – If you use the user interface of the C-887: Press the F3 key on the keyboard or click the ***F3 TERMINAL*** button in the main window of the user interface to open the ***Terminal Panel*** window.

2. Enter the command `IFS?`.

   This command gets the values of the interface parameters in the non-volatile memory.

3. Write down the settings ***IPMASK*** and ***IPADR***.

4. Make sure that the ***IPSTART*** parameter is correctly set:

   – If ***IPSTART*** = 0 (the static IP address defined with ***IPADR*** is used), the setting is correct.

   – If ***IPSTART*** ≠ 0: Send the command `IFS 100 IPSTART 0`.

   –

INFORMATION

In PIMikroMove® you can determine the IP address and subnet mask of the C-887 and adapt the start-up behaviour of the C-887 in the ***Configure Interface*** window without sending commands.

➢ Open the ***Configure Interface*** window using the ***Hexapod > Configure interface...*** menu sequence in the main window.

–

Adapting IP settings of the PC

➢ If you want to leave the PC settings unchanged, continue with the section "Adapting C-887 settings" (p. 84).

1. Activate ***Use following IP address*** in the window in which the properties of the Internet protocol TCP/IP (TCP/IPv4) are displayed and set.

2. Adapt the IP address and the subnet mask to the settings of the C-887:

   a) Copy the first three sections of the IP address of the C-887 for the IP address on the PC.
   b) Make sure that the last section of the IP address on the PC differs from the last section of the IP address of the C-887 and is not "255" or "0".
   c) Copy the subnet mask of the C-887 for the subnet mask on the PC.

   Example:
   New IP address of the PC: 192.168.1.29 (if the C-887 has the IP address 192.168.1.28)
   New subnet mask of the PC: 255.255.255.0 (if the C-887 has the subnet mask 255.255.255.0)

3. Confirm the settings with the ***OK*** button.

4. If further network participants have to be adapted:
   Adapt the IP addresses and subnet masks as in the previous steps.
   Assign a separate, clear IP address to each network participant.
   IP addresses must not occur twice in the same network.
5. If you have established communication between the C-887 and the PC via RS-232: Close the connection via the RS-232 interface, e. g. in PIMikroMove in the main window via the ***Connections > Close > Hexapod*** menu item.
6. Switch off the C-887.
7. Continue with the section "Establishing Communication via TCP/IP in the PC Software" (p. 85).

Adapting C-887 settings

1. Adapt the settings of the C-887 to those of the PC with the `IFS` command:
   a) Change the subnet mask with the `IFS 100 IPMASK xxx.xxx.xxx.xxx` command, whereby *xxx.xxx.xxx.xxx* is the subnet mask of the PC.
   b) Change the IP address with the `IFS 100 IPADR xxx.xxx.xxx.yyy:50000` command, whereby the following applies:

   – *xxx.xxx.xxx.* matches the first three sections of the IP address of the PC

   – *yyy* differs from the last section of the IP address of the PC and every other device in the same network

   – *yyy* is not "255" and not "0" and is in the address range which is given by the last section of the subnet mask

   – The port address "50000" must not be changed

   Example:
   If the IP address of the PC is 192.168.0.1 and no other device has the IP address 192.168.0.2, send the `IFS 100 IPADR 192.168.0.2:50000` command.
2. If you have established communication between the C-887 and the PC via RS-232: Close the connection via the RS-232 interface, e. g. in PIMikroMove in the main window via the ***Connections > Close > Hexapod*** menu item.
3. Switch off the C-887.
4. Continue with the section "Establishing Communication via TCP/IP in the PC Software" (p. 85).

6.4.2 Establishing Communication via TCP/IP in the PC Software

CAUTION

Risk of crushing from unexpected motion

When the communication between the C-887 and the PC is established via TCP/IP, the PC software offers all controllers present in the same network for selection. After a C-887 has been selected for the connection, all commands are sent to this controller. If the wrong controller is selected, the operating and maintenance staff of the connected Hexapod are at risk of slight injury from crushing due to unexpectedly supplied motion commands.

➢ If several C-887 are displayed in the PC software, make sure that you select the right C-887.

INFORMATION

In the list of controllers found in the same network, the C-887 is **not** shown as ***C-887*** but as ***HEXAPOD*** or ***F-HEX***, followed by the 9-digit serial number of the C-887.

Example of the display of a C-887 to which no connection has been established via TCP/IP yet:

HEXAPOD SN 111160398 -- listening on port 50000 -- 172.17.72.55 50000

111160398 is the serial number of the C-887 in this example.

➢ If several C-887 are connected with the same network via TCP/IP, identify the C-887 to be connected in the list of found controllers on the basis of its serial number. The serial number of the controller can be found on the type plate on the rear panel of the C-887.

➢ If you establish (p. 88) communication in advance via RS-232 or use (p. 74) the user interface of the C-887, you can also find the serial number of the C-887 in the response to the `*IDN?` (p. 144) command or get using the `SSN?` command (p. 230).

Prerequisites

✓ You have read and understood the General Notes on Start-Up (p. 69).

✓ The C-887 is connected to the network via the **LAN** RJ45 socket on its rear panel or directly to the PC.

- ✓ If the C-887 is connected to a network:
  The PC to be used for communication with the C-887 is suitably connected to the same network as the C-887.
- ✓ If the used network does not have a DHCP server or if the C-887 is directly connected to the Ethernet connection socket of the PC **and** static IP addresses are to be used:
  You have set the correct start-up behavior for configuring the IP address of the C-887 by adapting the interface parameters and coordinated the IP addresses and subnet masks of the C-887 and PC as well as all other network participants with each other (p. 81).
- ✓ If several C-887's are connected to the same network via their TCP/IP interfaces: You have the serial number of the C-887 at hand with which the communication is to be established. The serial number can be found on the type plate on the rear panel of the C-887.
- ✓ The PC is switched on.
- ✓ The required software is installed on the PC (p. 43).
- ✓ You have read and understood the manual of the used PC software. The software manuals are found on the product CD.
- ✓ The C-887 is switched off.

Establishing communication via TCP/IP

The procedure for PIMikroMove is described in the following. The procedure for other PC software programs (PITerminal, LabVIEW driver) is similar.

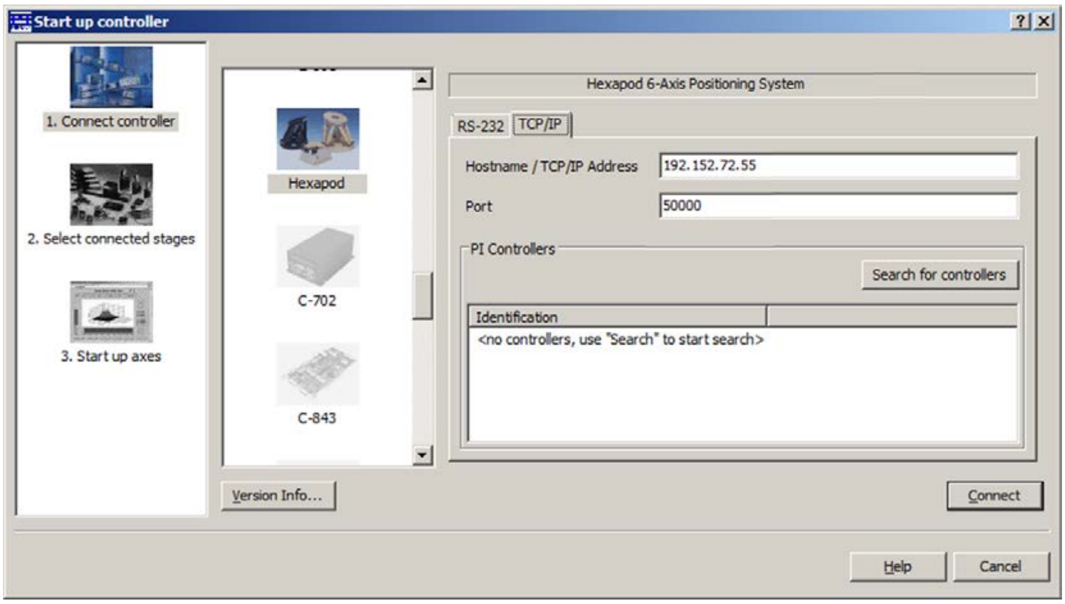
1. Switch on the C-887.

   When the IP addresses of the network participants are configured with AutoIP, it will take up to 2 minutes after the end of the starting procedure of the C-887 (p. 73) until communication is possible via TCP/IP.

2. Start PIMikroMove.

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

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



Hostname / TCP/IP Address: 192.152.72.55

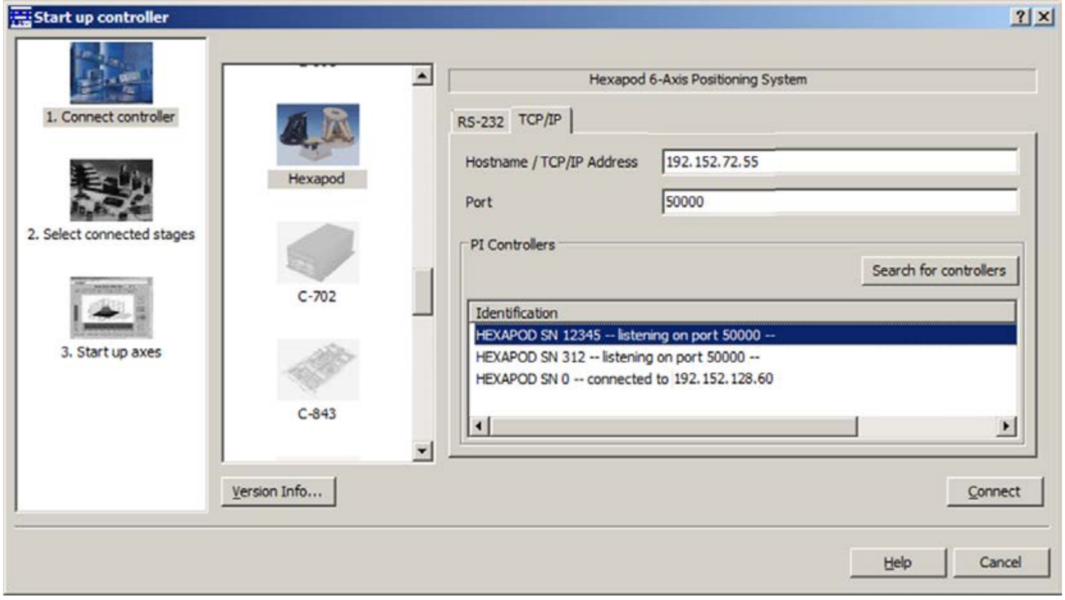
Port: 50000

PI Controllers

Identification	
<no controllers, use "Search" to start search>	

3. Select ***Hexapod*** in the field for controller selection.
4. Select the ***TCP/IP*** tab on the right side of the window.
5. Click the ***Search for controllers*** button.

   All controllers in the same network are shown in the field underneath the button.
6. Click the ***HEXAPOD SN ...*** or ***F-HEX SN ...*** entry in the list of controllers found (*SN* stands for *serial number*).
   - If several ***HEXAPOD SN ...*** or ***F-HEX SN ...*** entries are shown, identify your C-887 on the basis of its nine-digit serial number.
   - If the C-887 is not displayed in the list of the controllers found, check the network settings (p. 299). Consult your network administrator if necessary.



Start up controller

Hexapod 6-Axis Positioning System

Hostname / TCP/IP Address: 192.152.72.55

Port: 50000

PI Controllers

Identification
HEXAPOD SN 12345 -- listening on port 50000 --
HEXAPOD SN 312 -- listening on port 50000 --
HEXAPOD SN 0 -- connected to 192.152.128.60

Do **not** select a controller with which a connection via TCP/IP already exists. Otherwise, an error message will be displayed as soon as you want to establish communication with this controller.

7. Check the IP address in the ***Hostname / TCP/IP Address*** field and the port number in the ***Port*** field.
8. Click the ***Connect*** button to establish communication.

When communication has been successfully established, the ***Start up controller*** window goes to the ***Start up axes*** step.

6.5 Establishing Communication via the RS-232 Interface

6.5.1 Changing the Baud Rate

The interface parameters for RS-232 communication are set at the factory as in the table below. The values in the non-volatile memory can be queried with the `IFS?` (p. 197) command.

Interface parameter	Factory setting	Note
Port for RS-232 communication (RSPORT)	1	Write-protected Indicates the port of the C-887 used for RS-232 communication.

Handshake for RS-232 communication (RSHSHK)	RTS/CTS	Write-protected Indicates the handshake setting of the C-887 for RS-232 communication.
Baud rate (RSBAUD)	115200	Indicates the baud rate of the C-887 for RS-232 communication. Further possible values are 9600, 19200, 38400, 57600. To successfully establish communication, the baud rates of the C-887 and PC must match.

Before communication is established, it can be necessary to change the factory baud rate setting of the C-887. The procedure for the user interface of the C-887 is described in the following. Alternatively, the PC software can also be used for entering commands after the communication has been established via TCP/IP or RS-232. The baud rate is changed in the non-volatile memory and does not go into effect until the C-887 is rebooted.

Prerequisite

✓ You have made the preparations for using the user interface of the C-887 (p. 74).

Changing the baud rate for the RS-232 connection

1. Press the F3 key on the keyboard or click the ***F3 TERMINAL*** button in the main window of the user interface to open the ***Terminal Panel*** window.
2. Send the `IFS 100 RSBAUD xxxxx` command, whereby *xxxxx* is the new baud rate.

6.5.2 Establishing Communication via RS-232 in the PC Software

Prerequisites

✓ You have read and understood the General Notes on Start-Up (p. 69).

✓ The C-887 is connected to the RS-232 interface of the PC (p. 68)

✓ The C-887 is switched on and the starting procedure of the C-887 has finished (p. 73).

✓ The PC is switched on.

- ✓ The required software is installed on the PC.
- ✓ You have read and understood the manual of the used PC software. The software manuals are found on the product CD.

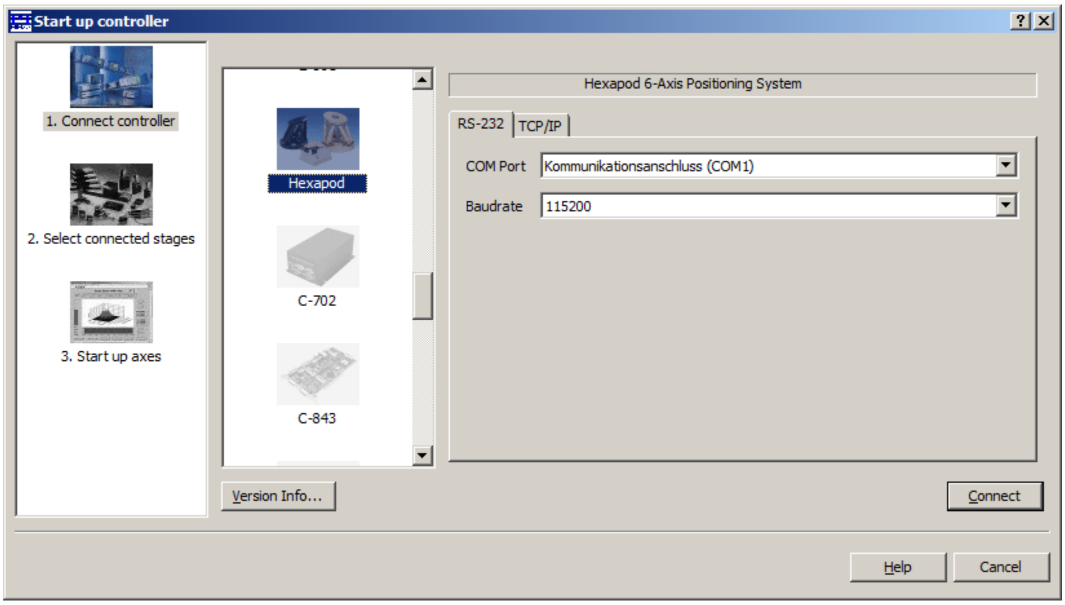
Establishing communication via RS-232

The procedure for PIMikroMove is described in the following. The procedure for other PC software programs (PITerminal, LabVIEW driver) is similar.

1. Start PIMikroMove.

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

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



COM Port Kommunikationsanschluss (COM1)

Baudrate 115200

2. Select ***Hexapod*** in the field for controller selection.
3. Select the ***RS-232*** tab on the right side of the window.
4. In the ***COM Port*** field, select the COM port of the PC to which you have connected the C-887.
5. In the ***Baud rate*** field, set the value 115200 (factory setting upon delivery of the C-887).

   This adapts the baud rate of the PC to the baud rate of the C-887.

   If you have changed the baud rate of the C-887, you have to enter the new value in the ***Baud rate*** field instead (p. 88).
6. Click ***Connect*** to establish communication.

When communication has been successfully established, the ***Start up controller*** window goes to the ***Start up axes*** step.

6.6 Starting Motions of the Platform of the Hexapod

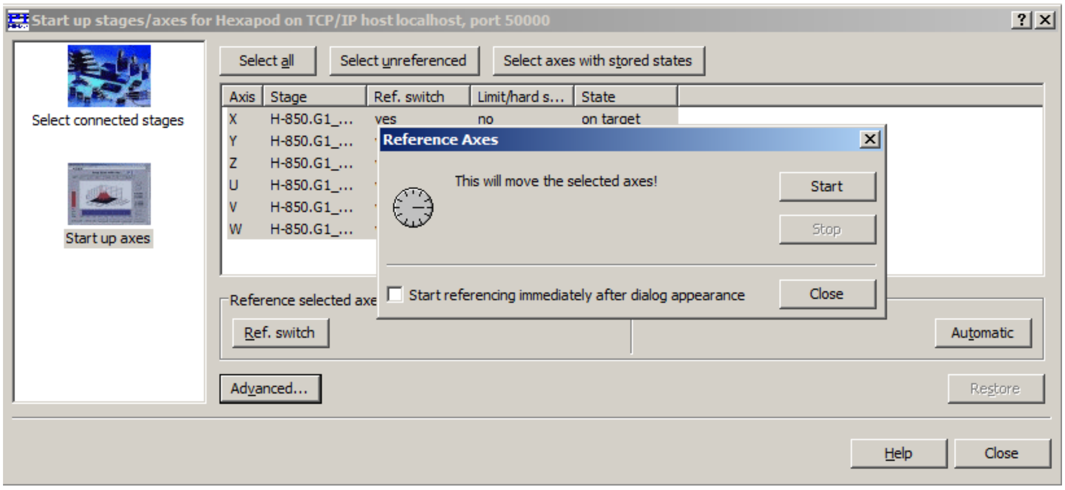
In the following, PIMikroMove is used to move the platform of the Hexapod. For starting motions with the user interface, see "Using the User Interface of the C-887" (p. 74).

Prerequisites

- ✓ You have read and understood the General Notes on Start-Up (p. 69).
- ✓ PIMikroMove is installed on the PC (p. 43).
- ✓ You have read and understood the PIMikroMove manual. The manual is found on the product CD.
- ✓ You have correctly installed the Hexapod system (C-887, Hexapod and cable set) (p. 41).
- ✓ You have read and understood the user manual of the Hexapod.
- ✓ You have established communication between the C-887 and the PC with PIMikroMove via the TCP/IP interface (p. 85) or the RS-232 interface (p. 89).

Starting motions of the platform of the Hexapod with PIMikroMove

1. Carry out the reference move for the moving platform of the Hexapod in the ***Start up axes*** step. To do this, click ***Ref.switch*** or ***Automatic***.



Start up stages/axes for Hexapod on TCP/IP host localhost, port 50000

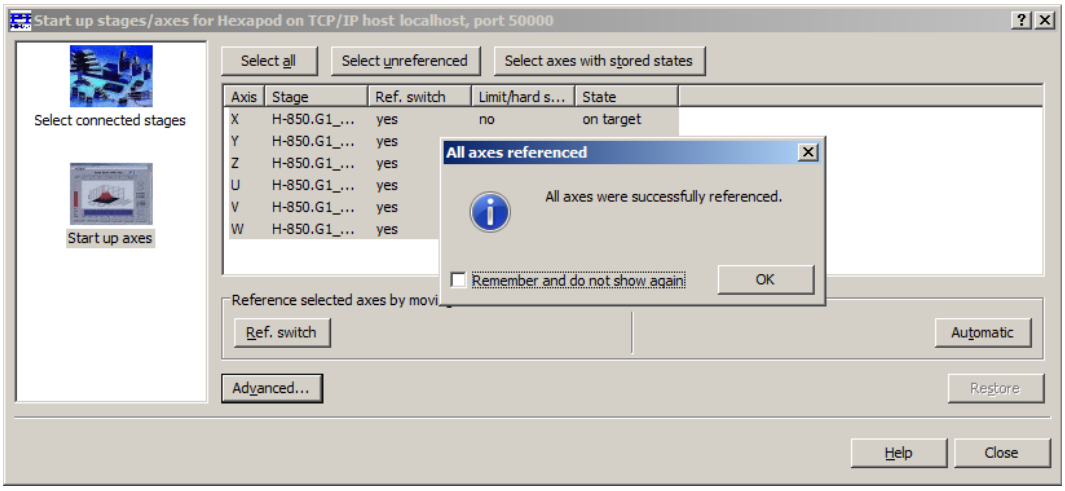
Axis	Stage	Ref. switch	Limit/hard s...	State
X	H-850.G1_...	yes	no	on target
Y	H-850.G1_...			
Z	H-850.G1_...			
U	H-850.G1_...			
V	H-850.G1_...			
W	H-850.G1_...			

Reference Axes

This will move the selected axes!

□ Start referencing immediately after dialog appearance

2. After a successful reference move, click ***OK > Close***.


Start up stages/axes for Hexapod on TCP/IP host localhost, port 50000
Select connected stages
Start up axes
Select all
Select unreferenced
Select axes with stored states
Reference selected axes by movi
Ref. switch
Automatic
Advanced...
Restore
Help
Close

Axis	Stage	Ref. switch	Limit/hard s...	State
X	H-850.G1_...	yes	no	on target
Y	H-850.G1_...	yes		
Z	H-850.G1_...	yes		
U	H-850.G1_...	yes		
V	H-850.G1_...	yes		
W	H-850.G1_...	yes		

All axes referenced

All axes were successfully referenced.

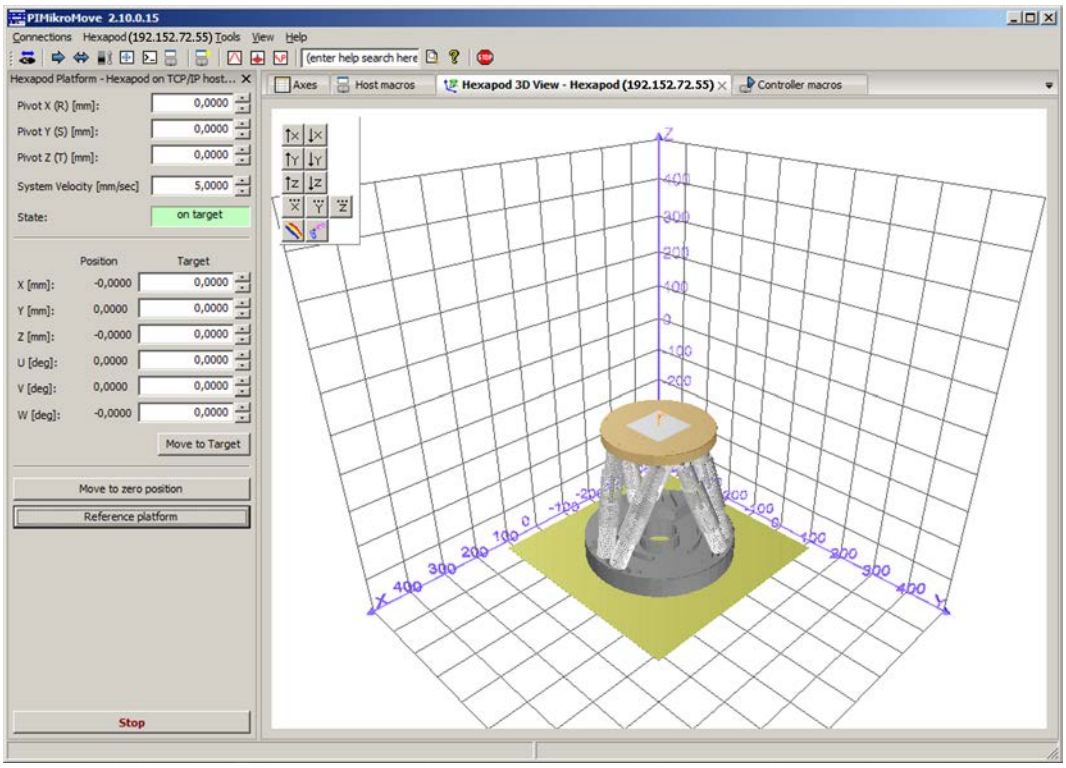
☐ Remember and do not show again OK

3. If the ***Hexapod Platform*** window (left, docked) and the ***Hexapod 3D View*** card are **not** displayed in the main window of PIMikroMove:

   – Display the ***Hexapod Platform*** window with the ***Hexapod > Show Hexapod platform settings*** menu item.

   – Display the ***Hexapod 3D View*** card with the ***Hexapod > Hexapod 3D View > Show*** menu item.

   – If the ***Hexapod 3D View*** card is not displayed in the foreground, click on the corresponding tab.



Pivot X (R) [mm]: 0,0000

Pivot Y (S) [mm]: 0,0000

Pivot Z (T) [mm]: 0,0000

System Velocity [mm/sec] 5,0000

State: on target

	Position	Target
X [mm]:	-0,0000	0,0000
Y [mm]:	0,0000	0,0000
Z [mm]:	-0,0000	0,0000
U [deg]:	0,0000	0,0000
V [deg]:	0,0000	0,0000
W [deg]:	-0,0000	0,0000

4. Start a few test motions of the axes of the moving platform of the Hexapod:

   a) In the ***Hexapod Platform*** window, enter a target position for at least one axis of the moving platform of the Hexapod in the corresponding ***Target*** field.

   b) Click ***Move to Target*** to start the motion towards the given target position.

   c) When the motion is finished, repeat steps a and b for a new target position.

If a node of the calculated trajectory or the target position cannot be reached, the motion is not executed and a window opens with an error message.

A new target position cannot be entered during a motion.

The motion is graphically displayed on the ***Hexapod 3D View*** card.

6.7 Starting Motions of the Axes A, B, K, L and M - Only C-887.11

CAUTION

Dangerous voltage and residual charge on piezo actuators!

The P-611.3SF NanoCube® stage of the F-206.NCU option is driven by piezo actuators. Temperature changes and compressive stresses can induce charges in piezo actuators. After being disconnected from the electronics, piezo actuators can also stay charged for several hours. Touching or short-circuiting the contacts in the connector of the P-611.3SF NanoCube® stage can lead to minor injuries. In addition, the piezo actuators can be destroyed by an abrupt contraction.

- Do **not** pull out the connector from the electronics during operation.

CAUTION

Risk of electric shock if the protective earth conductor is not connected!

If a protective earth conductor is not or not properly connected, dangerous touch voltages can occur and there is a risk of electric shock. In the case of a malfunction or failure of the system, touching the P-611.3SF NanoCube® stage of the F-206.NCU option can result in slight injuries.

- Connect the P-611.3SF NanoCube® stage to a protective earth conductor before start-up. Follow the instructions in the Technical Note F206T0021 (included in the scope of delivery of the F-206.NCU option).
- Do **not** remove the protective earth conductor during operation.
- If the protective earth conductor has to be temporarily removed (e.g. for modifications), reconnect the P-611.3SF NanoCube® stage to the protective earth conductor before starting it up again.

NOTICE

Damage to the P-611.3SF NanoCube® stage from oscillations!

Oscillations can be detected by a humming of the stage and indicate an incorrect setting of the notch filter and/or the servo-control parameters.

- Perform measurements of the impulse and step response for the axes K, L and M before the first start-up of the P-611.3SF NanoCube® stage and optimize the notch filter frequency (parameter 0x08000100) and the P-Term of the controller (parameter 0x07000300) (p. 100) if necessary.
- If oscillations occur during operation, immediately switch the servo mode off for the axes K, L and M.

NOTICE

Damage from assigning an incorrect stage type!

If stages for axes A and B are subsequently added to the Hexapod system, the appropriate stage types must be assigned to axes A and B each time that the C-887.11 is switched on and rebooted. The assignment of an incorrect stage type can cause damage to the stage.

- Make sure that the assigned stage type matches the connected stage.

NOTICE

Damage from unsuitable settings of travel range limits and soft limits!

When the reference point for axis A and/or B is defined by setting the ***Sensor Reference Mode*** parameter (ID 0x02000A00) since no reference switch is present, the value of the current position and the settings of the travel range limits and soft limits of the axis (TMN?, TMX?; NLM?, PLM?) are not changed. This may result in target positions which are allowed by the C-887 but cannot be approached by the hardware. Target positions are also possible which can be approached by the hardware but are denied by the C-887. When motions are commanded, due to unsuitable limit settings the axis can hit the hard stop or move in areas which are to be protected by the soft limits. As a result, the stage, the load to be moved or the environment can be damaged.

- Check the current position with POS? after you have set the ***Sensor Reference Mode*** parameter for reference point definition and command only target positions which can be approached by the hardware.
- Set the soft limits to suitable values with NLM and PLM.

INFORMATION

A stage with DC motor and PWM amplifier can be connected to each of the sockets **A** and **B** on the C-887.11 (p. 61).

Assignment of the stage type for axes A and B

In the firmware of the C-887.11 these stages can be accessed as axes A and B. The C-887.11 is adapted to the properties of the used stage types by various settings (p. 269).

If the stage types used for axes A and B are known upon delivery of the C-887.11, you will receive the C-887.11 suitably preconfigured.

If stages for axes A and B are subsequently added to the Hexapod system, the appropriate stage types must be assigned to axes A and B each time that the C-887.11 is switched on and rebooted. The assignment is done with the command CST (p. 152). CST loads the operating parameters of the assigned stage type from a stage database on the controller into the working memory. The permissible stage types can be listed with the VST? command (p. 248). The currently assigned stage types can be queried with the command CST? (p. 152).

- ➢ If you want to avoid entering commands, use the corresponding functions of the PC software or the user interface of the C-887, e.g. the ***Select connected stages*** (p. 99) step in PIMikroMove to assign the stage type.
- ➢ If you want to automatically assign the correct stage type to axes A and B when the C-887.11 is switched on or rebooted, use CST in the start-up macro (p. 118).

If the stage type "NOSTAGE" has been assigned to axes A and B (in PIMikroMove: ***No Stage ->*** button in the ***Select connected stages*** step), they are considered "deactivated" and are not displayed. Exception: the responses to the CST? (p. 152) and SAI? ALL commands include the deactivated axes.

Reference point definition for axes A und B

Axes A and B are intended for the use of incremental sensors. For this reason, motions cannot be commanded for axes A and B until after a successful reference point definition.

With the factory settings of the C-887.11, the reference point definition **must** take place with a reference move to the reference point switch of the stage. If a stage without a reference point switch is connected, a reference move **cannot** be started for the axis. Determine the reference value for an axis to which a stage without a reference point switch is connected as follows:

1. Open the window for entering commands in the PC software or on the user interface of the C-887.
2. Send the command `CCL 1 advanced` to go to command level 1.
3. Change the value of the ***Sensor Reference Mode*** parameter (ID 0x02000A00) to the value 1 for the axis with the SPA command (p. 224) so that the reference

point definition is considered successfully completed.

Example: To change the parameter value for axis A, send the `SPA A 0x02000A00 1` command

INFORMATION

If the E-760 controller card of the optional accessory F-206.NCU is installed in the C-887.11 (p. 45), a P-611.3SF NanoCube® stage can be connected (p. 62). In the firmware of the C-887.11, this stage is accessible through the axes K, L and M. The settings of the E-760 controller card were calibrated with the supplied stage before delivery. Adaptations may be necessary to the notch filter frequency and the P-Term of the controller, depending on the load used (p. 100).

The P-611.3SF NanoCube® stage is equipped with absolute-measuring sensors. A reference move is therefore not necessary and not intended for the axes of this stage.

INFORMATION

When switched on or rebooted, the C-887 automatically switches on the servo mode for all axes.

When the servo mode is switched off for axes A and B, it is automatically switched on when a stage type is assigned with CST.

PIMikroMove is used in the following.

Prerequisites

- ✓ You have read and understood the General Notes on Start-Up (p. 69).
- ✓ PIMikroMove is installed on the PC (p. 43).
- ✓ You have read and understood the PIMikroMove manual. The manual is found on the product CD.
- ✓ If you want to use axes A and B:
  - – You have correctly installed the corresponding stages (p. 61).
  - – You have read and understood the user manual of the respective stage.
- ✓ If you want to use the axes K, L and M supplied with the optional accessory F-206.NCU:
  - – You have read and understood the handling information for the P-611.3SF NanoCube® stage in the Technical Note F206T0021.

- – You have installed the P-611.3SF NanoCube® stage in the same manner as it will be used in your application (corresponding load and orientation).
- – You have connected the P-611.3SF NanoCube® stage when the C-887.11 was switched off (p. 62).

✓ You have established communication between the C-887 and the PC with PIMikroMove via the TCP/IP interface (p. 85) or the RS-232 interface (p. 89).

✓

Referencing axes A and B with PIMikroMove

➢ If the ***Start up controller*** window is **not** displayed in PIMikroMove with the ***Start up axes*** step, open it with the ***Hexapod > Start up axes...*** menu item.

➢ For each of the two axes A and B that you want to use, check whether the axis is in the axis list in the ***Start up axes*** step.

If the axis is **not** contained in the axis list:

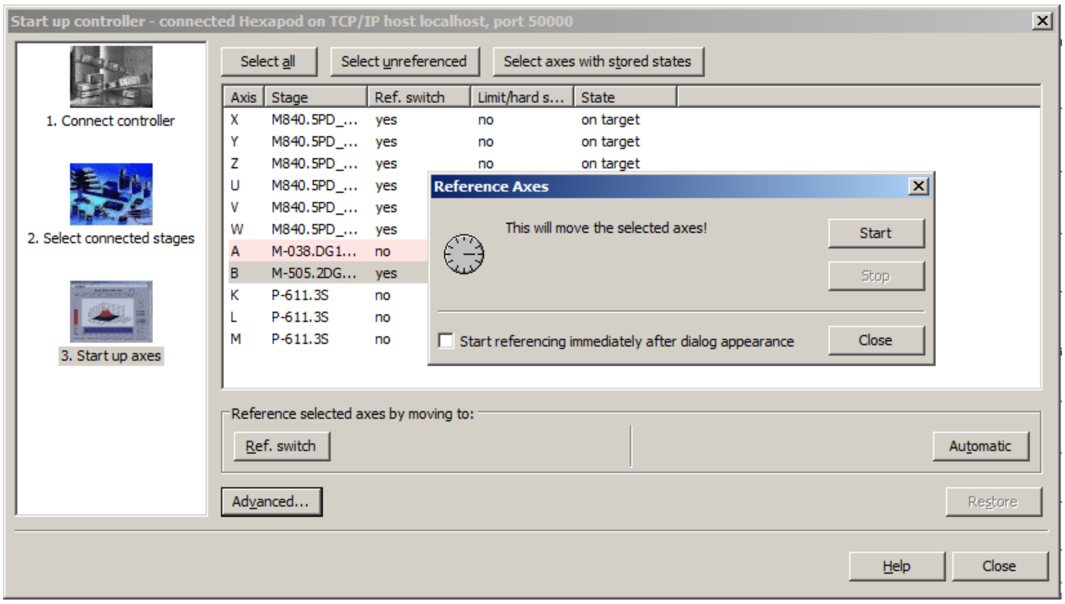
➢ Click ***Select connected stages*** in the left area of the window and follow the instructions in "Assigning a stage type for axes A and B" (p. 99).

If the axis is contained in the axis list:

➢ Check the entry for the axis in the ***Ref.switch*** column of the axis list to find out whether the stage has a reference point switch.

If the entry is ***yes*** :

1. Mark the axis in the list.
2. Click ***Ref.switch*** or ***Automatic*** to open the window for starting the reference move.
3. Click ***Start*** to start the reference move (see following figure).
4. After a successful reference move, click ***OK > Close***.



Start up controller - connected Hexapod on TCP/IP host localhost, port 50000

Axis	Stage	Ref. switch	Limit/hard s...	State
X	M840.5PD_...	yes	no	on target
Y	M840.5PD_...	yes	no	on target
Z	M840.5PD_...	yes	no	on target
U	M840.5PD_...	yes		
V	M840.5PD_...	yes		
W	M840.5PD_...	yes		
A	M-038.DG1...	no		
B	M-505.2DG...	yes		
K	P-611.3S	no		
L	P-611.3S	no		
M	P-611.3S	no		

Reference Axes

This will move the selected axes!

☐ Start referencing immediately after dialog appearance

If the entry is ***no*** (i.e. the stage cannot perform a reference move):

1. Close the ***Start up axes*** step with ***Close***.
2. Open the window for entering commands with the ***Tools > Command entry*** menu item in the main window of PIMikroMove.
3. Send the `CCL 1 advanced` command.
4. Send the `SPA AxisID 0x02000A00 1` command, whereby *AxisID* is the identifier of the affected axis (A or B).
5. Close the ***Command entry*** window.

After a successful reference move or after the parameter value has been changed with the SPA command, the axis is considered to be referenced and motion commands can be given.

➢ Continue with "Starting motions of the axes A, B, K, L and M with PIMikroMove" (p. 103).

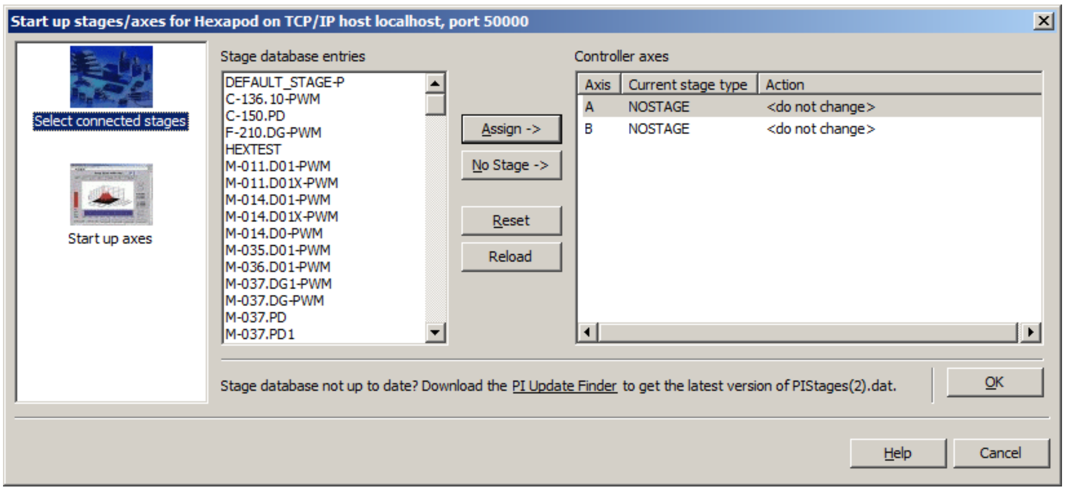
Assigning a stage type for axes A and B with PIMikroMove

➢ In the ***Select connected stages*** step, select the stage type of the connected stage:

   a) Mark the axis to which you want to assign a stage type in the ***Controller axes*** list.
   b) Mark the stage type in the ***Stage database entries*** list.
   c) Click ***Assign***.

d) If necessary, repeat steps a to c for the second axis.

e) Confirm the selection with ***OK*** to load the operating parameters of selected stage type from a stage database on the C-887 into the working memory of the C-887.



Start up stages/axes for Hexapod on TCP/IP host localhost, port 50000

Stage database entries

DEFAULT_STAGE-P
C-136.10-PWM
C-150.PD
F-210.DG-PWM
HEXTEST
M-011.D01-PWM
M-011.D01X-PWM
M-014.D01-PWM
M-014.D01X-PWM
M-014.D0-PWM
M-035.D01-PWM
M-036.D01-PWM
M-037.DG1-PWM
M-037.DG-PWM
M-037.PD
M-037.PD1

Controller axes

Axis	Current stage type	Action
A	NOSTAGE	<do not change>
B	NOSTAGE	<do not change>

Stage database not up to date? Download the PI Update Finder to get the latest version of PIStages(2).dat.

After you have clicked ***OK***, the ***Start up controller*** window goes to the ***Start up axes*** step.

After the correct stage type has been assigned, axes A and B must be referenced before motions can be commanded:

➢ Continue with "Referencing axes A and B with PIMikroMove" (p. 98).

Optimizing the notch filter frequency and P-Term for axes K, L and M

Adaptations may be necessary for the notch filter frequency and the P-Term of the controller of the axes K, L, and M after delivery of the F-206.NCU option, depending on the used load. By measuring the impulse and step response for axes K, L and M, you can determine whether the notch filter frequency and the P-Term of the controller are correctly set for your application.

The procedure for the user interface of the C-887 is described in the following.

INFORMATION

In PIMikroMove, the ***Data Recorder*** window can be used for measurements of the impulse and step response, and parameter changes can be made in the ***Command entry*** window. For details, see the PIMikroMove manual

1. Close the connection via the TCP/IP or RS-232 interface in the main window of PIMikroMove with the ***Connections > Close > Hexapod*** menu item.
2. Make the preparations for using the user interface of the C-887 (p. 74).
3. Press the F1 key on the keyboard or click the ***F1 TEST TOOLS*** button in the main window of the user interface to open the ***hexapod strut tests*** window.
4. Record an impulse response for an axis of the P-611.3SF NanoCube® stage in the ***hexapod strut tests*** window (the corresponding command is `IMP` (p. 198)):
   a) Press the F2 key or click the ***F2 Impulse Response*** button.
   b) Press the F5 key or click the ***F5 START TEST*** button to open the ***Test program*** window.
   c) In the ***Test program*** window, enter the identifier of the axis for which the impulse response is to be recorded (K, L or M), and confirm the entry with the Enter key or by clicking the ***OK*** button. The ***Impulse response*** window opens.
   d) Do not change the value for the impulse height in the ***Impulse response*** window, and press the Enter key or click on the ***OK*** button to go to where you can enter the pulse width of the impulse.
   e) Enter the pulse width of the impulse in milliseconds (recommended value: 100 ms).
   f) Press the Enter key or click the ***OK*** button to start the impulse and the recording of the current position of the axis. The axis moves in accordance with the set pulse width of the impulse.
   g) Evaluate the impulse response on the basis of the curves in the graphics field of the ***hexapod strut tests*** window.
   h) If the impulse response in the ***Frequency Response*** graphics field (FFT display) shows a noticeable peak at a particular frequency, note down this frequency as the new value for the notch filter frequency.
5. Record a step response for an axis of the P-611.3SF NanoCube® stage in the ***hexapod strut tests*** window (the corresponding command is `STE` (p. 234)):
   a) Press the F1 key or click the ***F1 Step Response*** button.

b) Press the F5 key or click the ***F5 START TEST*** button to open the ***Test program*** window.

c) In the ***Test program*** window, enter the identifier of the axis for which the step response is to be recorded (K, L or M), and confirm the entry with the Enter key or by clicking the ***OK*** button. The ***Step response*** window opens.

d) In the ***Step response*** window, enter the step size of the step in millimeters (recommended value: 0.01 mm).

e) Press the Enter key or click the ***OK*** button to start the step and the recording of the current position of the axis. The axis moves in accordance with the set step size.

f) Evaluate the step response on the basis of the curves in the graphics field of the ***hexapod strut tests*** window.

g) If the step response in the ***Step/Impulse Response*** graphic field shows strong overshooting, oscillations or too slow of a step-and-settle of the current position, the P-Term of the controller must be optimized.

6. If necessary, adjust the parameter values for the notch filter frequency and the P-Term of the controller according to the measuring results:

a) Press the F10 key to exit the ***hexapod strut tests*** window.

b) Press the F3 key on the keyboard or click the ***F3 TERMINAL*** button in the main window of the user interface to open the ***Terminal Panel*** window.

c) To change the notch filter frequency, give the `SPA AxisID 0x08000100 notch filter frequency` command, whereby *AxisID* indicates the identifier of the axis and *notch filter frequency* is the value in Hz determined in step 2.h (value range: 100 to 333 Hz).

d) To change the P-Term, enter the `SPA AxisID 0x08000100 P-Term` command, whereby *AxisID* indicates the identifier of the axis and *P-Term* is the new value for the P-Term (value range: 0.000045 to 0.046).

For further information on changing parameters, see "Adapting Settings" (p. 269).

7. Repeat steps 1 to 4 for all axes of the P-611.3SF NanoCube® stage until you obtain satisfactory measuring results.

8. If you want to move axes K, L and M with PIMikroMove:

a) Establish communication between the C-887 and the PC with PIMikroMove via the TCP/IP interface (p. 85) or the RS-232 interface (p. 89).

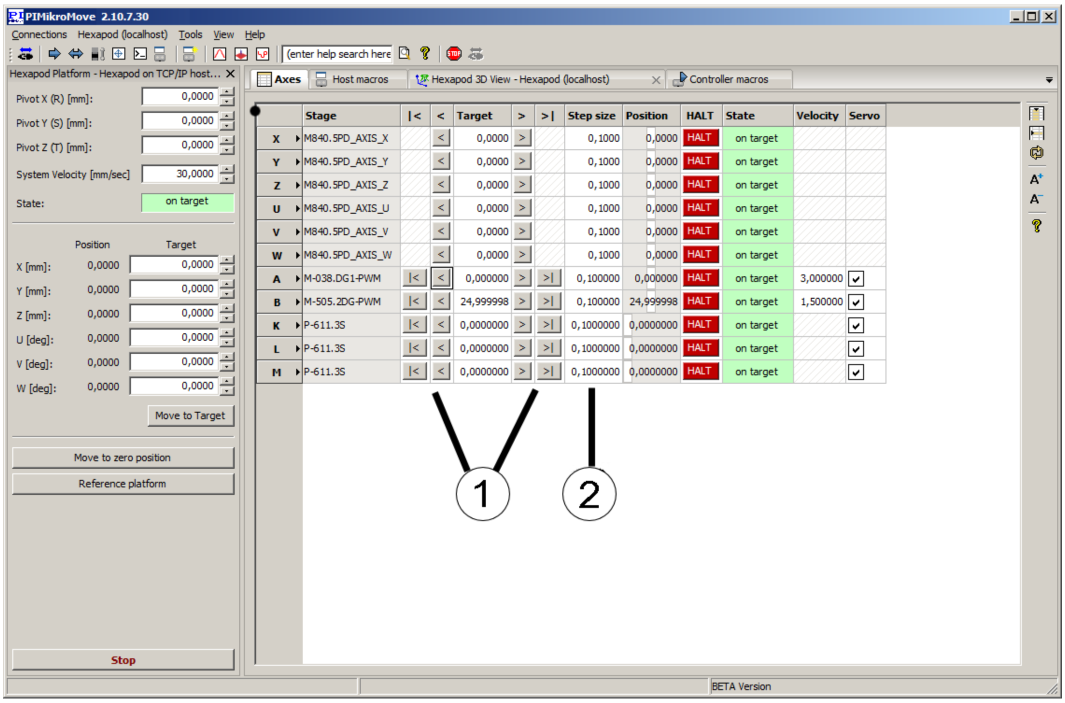
b) Continue with "Starting motions of the axes A, B, K, L and M with PIMikroMove" (p. 103).

Starting motions of the axes A, B, K, L and M with PIMikroMove

1. If the ***Axes*** card is **not** displayed in the foreground of the main window of PIMikroMove, click on the corresponding tab.

2. In the main window of PIMikroMove, start a few test motions of the axes A, B, K, L and M on the ***Axes*** card.

   You can, for example, execute steps with a particular step size (2) by clicking the corresponding arrow keys (1) for an axis.



Pivot X (R) [mm]: 0,0000
Pivot Y (S) [mm]: 0,0000
Pivot Z (T) [mm]: 0,0000
System Velocity [mm/sec] 30,0000
State: on target

	Position	Target
X [mm]:	0,0000	0,0000
Y [mm]:	0,0000	0,0000
Z [mm]:	0,0000	0,0000
U [deg]:	0,0000	0,0000
V [deg]:	0,0000	0,0000
W [deg]:	0,0000	0,0000

	Stage	|<	<	Target	>	>|	Step size	Position	HALT	State	Velocity	Servo
X	M840.5PD_AXIS_X		<	0,0000	>		0,1000	0,0000	HALT	on target		
Y	M840.5PD_AXIS_Y		<	0,0000	>		0,1000	0,0000	HALT	on target		
Z	M840.5PD_AXIS_Z		<	0,0000	>		0,1000	0,0000	HALT	on target		
U	M840.5PD_AXIS_U		<	0,0000	>		0,1000	0,0000	HALT	on target		
V	M840.5PD_AXIS_V		<	0,0000	>		0,1000	0,0000	HALT	on target		
W	M840.5PD_AXIS_W		<	0,0000	>		0,1000	0,0000	HALT	on target		
A	M-038.DG1-PWM	|<	<	0,000000	>	>|	0,100000	0,000000	HALT	on target	3,000000	☑
B	M-505.2DG-PWM	|<	<	24,999998	>	>|	0,100000	24,999998	HALT	on target	1,500000	☑
K	P-611.3S	|<	<	0,0000000	>	>|	0,1000000	0,0000000	HALT	on target		☑
L	P-611.3S	|<	<	0,0000000	>	>|	0,1000000	0,0000000	HALT	on target		☑
M	P-611.3S	|<	<	0,0000000	>	>|	0,1000000	0,0000000	HALT	on target		☑

7 Operation

In this Chapter

7.1 General Notes on Operation

CAUTION

⚠

Risk of crushing by moving parts!

There is a risk of minor injuries caused by crushing which can occur between the moving parts of the Hexapod and a stationary part or obstacle.

➢ Keep your fingers away from areas where they can get caught by moving parts.

NOTICE

❗

Damage from collisions!

Collisions can damage the Hexapod, the load to be moved and the environment.

➢ Make sure that no collisions between the Hexapod, the load to be moved and the environment are possible in the working space of the Hexapod.

➢ Do not place any objects in areas where they can get caught by moving parts.

➢ If the Hexapod controller malfunctions, stop the motion immediately.

NOTICE

Damage from unintentional position changes!

The limit value for the load of the Hexapod determined with the simulation program is only valid when the servo mode is switched on (p. 52) for the axes of the moving platform. The maximum holding force when the servo mode is switched off is based on the self-locking of the actuators in the Hexapod struts and is lower than the limit value when the servo mode is switched on (see manual of the Hexapod).

When the actual load of the Hexapod exceeds the maximum holding force based on the self-locking of the actuators, unintentional position changes of the Hexapod can occur in the following cases:

- Switching off the C-887
- Rebooting the C-887
- Switching off the servo mode for the axes of the moving platform of the Hexapod.

As a result, collisions are possible between the Hexapod, the load to be moved and the environment. Collisions can damage the Hexapod, the load to be moved or the environment.

➢ Make sure that the actual load of the moving platform of the Hexapod does not exceed the maximum holding force based on the self-locking of the actuators before you switch off the servo mode, reboot or switch off the C-887.

7.2 Data Recorder

7.2.1 Properties

The C-887 contains a real-time data recorder. The data recorder can record different values for axes (e.g. current position) as well as the signals of the analog inputs.

The recorded data is temporarily stored in 16 data recorder tables with up to 10,240 points each. Each data recorder table contains the data of one data source.

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

7.2.2 Setting up the Data Recorder

INFORMATION

The settings for setting up the data recorder can only be changed in the volatile memory of the C-887. After the C-887 has been switched on or rebooted, factory settings will be active unless a configuration takes place with a start-up macro.

Reading out general information about the data recorder

➢ Send the `HDR?` command (p. 190).

The available record options and trigger options as well as information about additional parameters and commands for data recording are displayed.

Configuring the data recorder

You can assign the data sources and record options to the data recorder tables.

➢ Send the `DRC?` command (p. 158) to read out the current configuration. Data recorder tables with the record option 0 are deactivated, i.e. nothing is recorded. In the default setting, the data recorder tables of the C-887 record the following:

- Data recorder table 1, 3, 5, 7, 9, 11: commanded position of the axes X, Y, Z, U, V and W
- Data recorder table 2, 4, 6, 8, 10, 12: current position of the axes X, Y, Z, U, V and W
- Data recorder table 13: the time

➢ Configure the data recorder with the `DRC` command (p. 155).

INFORMATION

Record option 80 (***Status register of axis***) can be used to record bits 0 to 15 of the signal status register for struts 1 to 6 of the Hexapod and for axes A and B. Only the following bits are relevant for the evaluation:

- Bit 0: Encoder A
- Bit 1: Encoder B
- Bit 3: Reference switch

➢ To record the signal status register, use the ***Data Recorder*** window in PIMikroMove, which allows the targeted selection of individual bits for the graphic display of the recorded data. For details, see the PIMikroMove manual.

You can specify how the recording is to be triggered.

➢ Get the current trigger option with `DRT?` (p. 162).

➢ Change the trigger option with the `DRT` command (p. 161). The trigger option applies to all data recorder tables whose record option is not set to 0.

You can specify the maximum number of points that are to be recorded for each data recorder table.

➢ Use the `SPA` command (p. 224) to change the ***Data Recorder Points Per Table*** parameter, ID 0x16000201. Default value: 1024 points. Maximum value: 10240 points.

Setting the record table rate

➢ Send the `RTR?` command (p. 220) to read out the record table rate.
The record table rate indicates via a factor the frequency with which data points are recorded. The possible range of values of the factors for RTR is 1 to 1,000,000. The factor 1 corresponds to the frequency 50 kHz. The default value is 250 and corresponds to 200 Hz.

➢ Change the record table rate with the `RTR` command (p. 219).

The greater the record table rate set with `RTR`, the greater the maximum duration of the data recording.

INFORMATION

The data recorder records with a maximum frequency of 50 kHz (can be set with `RTR`). The individual hardware components of the C-887 provide the values to be recorded with different clock rates, however.

If a recording contains data that could not be recorded with the frequency of the data recorder, the data points can no longer be allocated to the respective recording time.

The "Sample time" information will then be lacking in the recording. The time itself should therefore always be recorded as well (record option 8 "Measurement time" preset for data recorder table 13).

7.2.3 Starting the Recording

- ➢ Start the recording with the trigger option set with `DRT`.

Regardless of the trigger option set, the data recording is always triggered when a step response measurement is started with `STE` (p. 234) or an impulse response measurement with `IMP` (p. 198).

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.

7.2.4 Reading out Recorded Data

INFORMATION

Reading out the recorded data can take some time, depending on the number of data points and the record options set.

The data can also be read out while data is being recorded.

- ➢ Read out the last recorded data with the `DRR?` command.

The data is output in the GCS array format (see the SM146E user manual on the product CD).

7.3 Controller Macros

7.3.1 Overview: Macro Functionalities and Example Macros

The C-887 can save and process command sequences as macros.

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

- Several macros can be saved at the same time.
- Any macro can be defined as the start-up macro. The start-up macro is executed each time that the C-887 is switched on or rebooted.
- Processing a macro and stopping macro execution can be linked to conditions. In this way, loops can be realized as well.
- Macros can call up themselves or other macros in 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.

In this manual, you will find example macros for the following tasks:

- Moving an axis back and forth (p. 114)
- Moving an axis with a variable travel back and forth (p. 117)
- Triggering a reference move for the Hexapod with a start-up macro (p. 119)

7.3.2 Commands and Parameters

Commands

The following commands are specially available for handling macros or for use in macros:

Command	Syntax	Function
ADD (p. 148)	ADD <Variable> <FLOAT1> <FLOAT2>	Can only be used in macros. Adds two values and saves the result to a variable.
CPY (p. 151)	CPY <Variable> <CMD?>	Can only be used in macros. Copies a command response to a variable.
DEL (p. 154)	DEL <uint>	Can only be used in macros. Causes a delay of <uint> milliseconds.

Command	Syntax	Function
JRC (p. 199)	JRC <Jump> <CMD?> <OP> <Value>	Can only be used in macros. Triggers a relative jump of the macro execution pointer depending on a condition.
MAC (p. 201)	MAC BEG <macroname>	Starts the recording of a macro with the name *macroname* on the controller. *macroname* can consist of up to 8 characters.
	MAC DEF <macroname>	Defines the given macro as the start-up macro.
	MAC DEF?	Gets the start-up macro.
	MAC DEL <macroname>	Deletes the given macro.
	MAC END	Ends the macro recording.
	MAC ERR?	Reports the last error which occurred during macro execution.
	MAC FREE?	Gets the free memory space for the macro recording
	MAC NSTART <macroname> <uint> [{<String>}]	Starts the given macro n times in succession (n = number of executions). The values of local variables can be set for the macro with <String>.
	MAC START <macroname> [{<String>}]	Starts one execution of specified macro. The values of local variables can be set for the macro with <String>.
MAC? (p. 205)	MAC? [<macroname>]	Lists all macros or the content of a given macro.
MEX (p. 206)	MEX <CMD?> <OP> <value>	Can only be used in macros. Stops the macro execution depending on a condition.
RMC? (p. 217)	RMC?	Lists macros which are currently running.
VAR (p. 241)	VAR <Variable> <String>	Sets a variable to a certain value or deletes it.
VAR? (p. 243)	VAR? [{<Variable>}]	Gets variable values.
WAC (p. 248)	WAC <CMD?> <OP> <value>	Can only be used in macros. Waits until a condition is met.
#8 (p. 142)	-	Tests if a macro is running on the controller.

INFORMATION

A maximum of 256 characters are permitted per command line.

Parameter

The following parameters are available for working with macros:

Parameter	Description and possible values
Ignore Macro Error? 0x72	Determines whether the controller macro is stopped when an error occurs while it is being executed. ▪ 0 = Stop macro when error occurs (default) ▪ 1 = Ignore error

7.3.3 Working with Macros

Work with macros comprises the following:

- Recording of macros (p. 112)
- Starting macro execution (p. 115)
- Stopping macro execution (p. 118)
- Setting up a start-up macro (p. 118)
- Deleting of macros (p. 120)

INFORMATION

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. Details are found in the PIMikroMove manual.

Recording a macro

INFORMATION

The C-887 can save an unlimited number of macros at the same time. A maximum of 10 nesting levels are possible in macros.

INFORMATION

Basically all GCS commands (p. 127) can be included in a macro. Exceptions:

- `RBT` for rebooting the C-887
- `MAC BEG` and `MAC END` for macro recording
- `MAC DEL` for deleting a macro
- `#27` for stopping the C-887

Query commands can be used in macros in combination with the `CPY`, `JRC`, `MEX` and `WAC` commands. Otherwise they have no effect, since macros do not transmit any responses to interfaces.

INFORMATION

To make the use of macros more flexible, you can use local and global variables in macros. See “Variables” for more information.

INFORMATION

The number of write cycles in the non-volatile memory is restricted by the limited lifetime of the memory chip.

- ➢ Only record macros when it is necessary.
- ➢ Use variables in macros to make macros more flexible, and give the corresponding variable values when starting macro execution.
- ➢ Contact our customer service department (p. 303) if the C-887 shows unpredictable behavior.

1. Start the macro recording.
   - – If you are working with the user interface, PITerminal or in the ***Command entry*** window of PIMikroMove: Send the `MAC BEG macroname` command, whereby *macroname* refers to the name of the macro.
   - – If you are working on the ***Controller macros*** tab of PIMikroMove: Click the ***Create new empty macro*** icon to create a tab for entering a new macro. Do **not** enter the `MAC BEG macroname` command.
2. Enter the commands to be included in the *macroname* macro line by line, using the normal command syntax.

Macros can call up themselves or other macros in several nesting levels.

3. End the macro recording.

   – If you are working with the user interface, PITerminal or in the ***Command entry*** window of PIMikroMove: Send the `MAC END` command.

   – If you are working in the ***Controller macros*** tab of PIMikroMove: Do **not** enter the `MAC END` command. Click the ***Send macro to controller*** icon and enter the name of the macro in a separate dialog window.

   The macro has been stored in the non-volatile memory of the C-887.

4. If you want to check whether the macro has been correctly recorded:

   If you are working with the user interface, PITerminal or in the ***Command entry*** window of PIMikroMove:

   – Get which macros are saved in the C-887 by sending the `MAC?` command.

   – Get the contents of the *macroname* macro by sending the `MAC? macroname` command.

   If you are working in the ***Controller macros*** tab of PIMikroMove:

   – Click the ***Read list of macros from controller*** icon.

   – Mark the macro to be checked in the list on the left side and click the ***Load selected macro from controller*** icon.

Example: Moving an axis back and forth

INFORMATION

When macros are recorded on the ***Controller macros*** tab in PIMikroMove, the `MAC BEG` and `MAC END` commands must be omitted.

The axis X is to move back and forth. For this purpose, 3 macros are recorded. 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 X 12.5

WAC ONT? X = 1

MAC END

MAC BEG macro2

MVR X -12.5

WAC ONT? X = 1

MAC END

MAC BEG macro3

MAC START macro1

MAC START macro2

MAC END
```

Starting a macro execution

INFORMATION

When a macro is running on the C-887, commands can be sent to the C-887 via all communication interfaces. Commands are processed in the order that they arrive. The processing time for the individual commands depends on the command. The execution of a time-consuming command which was sent via a communication interface can therefore pause the macro execution. Thus macros are not suitable for time-critical running of defined motion profiles.

- Do **not** use a macro for running a time-critical motion profile but consecutive MOV commands which are stored in a buffer, see „Motions of the Hexapod“ (p. 32).
- Avoid using the C-887.MC control unit when a macro is running.
- If possible, do **not** start or use PC software like PIMikroMove or LabVIEW driver when a macro is running.
- If possible, avoid sending commands or using operating elements in the user interface of the C-887 when a macro is running.

INFORMATION

Simultaneous execution of multiple macros is not possible. Only one macro can be executed at a time.

INFORMATION

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

In the following, the user interface or the PITerminal or the ***Command entry*** window of PIMikroMove is used to enter commands. Details on working with the ***Controller macros*** tab in PIMikroMove are found in the PIMikroMove manual.

1. If the macro execution is to be continued despite the occurrence of an error:

   – Set the ***Ignore Macro Error?*** parameter (ID 0x72) correspondingly: Send the `SPA 1 0x72 Status` command, whereby *Status* can have the value 0 or 1 (0 = Stop macro when error occurs (default); 1 = Ignore macro error).

   Further information on changing parameters can be found in "Adapting Settings" (p. 269).

2. Start the macro execution:

   – If the macro is to be executed once, send the `MAC START macroname string` command, whereby *macroname* refers to the name of the macro.

   – If the macro is to be executed n times, send the `MAC NSTART macroname n string` command, whereby *macroname* refers to the name of the macro and *n* indicates the number of executions.

   string stands for the values of local variables. The values only have to be given 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 with spaces.

3. If you want to check the macro execution:

   – Get whether a macro is being executed on the controller by sending the `#8` command.

   – Get the name of the macro that is currently being executed on the controller by sending the `RMC?` command.

Example: Moving an axis with a variable travel back and forth

INFORMATION

When macros are recorded on the ***Controller macros*** tab in PIMikroMove, the `MAC BEG` and `MAC END` commands must be omitted.

The axis X 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 used macros. Local and global variables are therefore used.

1. Create the global variables LEFT and RIGHT by sending:

   ```
   VAR LEFT 5
   ```

   ```
   VAR RIGHT 15
   ```

   LEFT thus 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-887 is switched on or rebooted, since they are only written to the volatile memory of the C-887.

2. 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*).

3. Record the MOVWAI macro by sending:

   ```
   MAC BEG movwai
   ```

   ```
   MOV X $1
   ```

   ```
   WAC ONT? X = 1
   ```

```
MAC END
```

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

4. 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 X alternately moves to the positions 5 and 15 five times. You can also select any other value for the number of executions.

Stopping a macro execution

INFORMATION

You can link the stopping of the macro execution to a condition with the `MEX` command. The command must be included in the macro.

In the following, the user interface or the PITerminal or the ***Command entry*** window of PIMikroMove is used to enter commands. Details on working with the ***Controller macros*** tab in PIMikroMove are found in the PIMikroMove manual.

- Stop the macro execution with the `#24` or `STP` commands.
- If you want to check whether an error has occurred during the macro execution, send the `MAC ERR?` command. The response shows the last error that has occurred.

INFORMATION

The macro execution is also stopped by the `CSV 1` command.

Setting up a start-up macro

Any macro can be defined as the start-up macro. The start-up macro is executed each time that the C-887 is switched on or rebooted.

INFORMATION

Deleting a macro does not delete its selection as a start-up macro.

In the following, the user interface or the PITerminal or the ***Command entry*** window of PIMikroMove is used to enter commands. Details on working with the ***Controller macros*** tab in PIMikroMove are found in the PIMikroMove manual.

- Define a macro as the start-up macro with the `MAC DEF macroname` command, whereby *macroname* refers to the name of the macro.
- If you want to cancel the selection of the start-up macro and do not want to define another macro as the start-up macro, only send `MAC DEF`.
- Get the name of the currently defined start-up macro by sending the `MAC DEF?` command.

Example: triggering a reference move for the Hexapod with a start-up macro

INFORMATION

When macros are recorded on the ***Controller macros*** tab in PIMikroMove, the `MAC BEG` and `MAC END` commands must be omitted.

The STARTCL macro starts a reference move. STARTCL is defined as a start-up macro so that the Hexapod immediately executes the reference move after being switched on.

- Send:

```
MAC BEG startcl

DEL 1000

FRF X

MAC END

MAC DEF startcl
```

Deleting a macro

INFORMATION

A running macro may not be deleted.

In the following, the user interface or the PITerminal or the ***Command entry*** window of PIMikroMove is used to enter commands. Details on working with the ***Controller macros*** tab in PIMikroMove are found in the PIMikroMove manual.

- ➢ Delete a macro with the `MAC DEL macroname` command, whereby *macroname* refers to the name of the macro.

7.3.4 Variables

For more flexibility in programming, the C-887 supports the use of variable in macros. While global variables can be used for all macros, local variables are only valid for a particular macro. The number of global or local variables is not restricted.

Variables are present in volatile memory (RAM) only. The variable values are of the STRING data type.

Working with variables comprises the following:

- Generating variables:
  - Local variables: see "Specifics of Local Variables"
  - Global variables: within a macro with the `ADD` (p. 148), `CPY` (p. 151) and `VAR` (p. 241) commands, outside a macro with `VAR`
- Changing variable values: within a macro with the `ADD`, `CPY` and `VAR` commands
- Getting variable values: with `VAR?` within or outside a macro
- Deleting variables: with `VAR` within a macro; global variables even outside a macro

The following conventions apply to variable names:

- Variable names must not contain special characters (especially no “$”).
- The maximum number of characters is 8.
- Names of global variables can consist of characters A to Z and 0 to 9. They must start with an alphabetic character.

- Names of local variables must not contain alphabetic characters. Possible characters are 0 to 9.
- The variable name can also be given 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 curly brackets.

If the variable name consists of a single character, the curly brackets can be omitted.

Note that if the curly brackets are omitted with variable names consisting of multiple characters, the first character after the “$” is interpreted as the variable name.

Specifics of Local Variables

- The names of local variables used in a macro must form a continuous series. Example of permissible designation: 1, 2, 3, 4. Designation with, e.g., 1, 2, 5, 6 is not allowed.
- The values of local variables are given as arguments <String> of the `MAC START` or `MAC NSTART` command when starting the macro.

  The command formats are:

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

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

  <String> stands for the value of a local variable contained in the macro. 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 with spaces. A maximum of 256 characters are permitted per command line. <String> can be given directly or via the value of another variable.

  <uint> gives the number of times the macro is to be run. See the `MAC` command description (p. 201) for more information.

- The local variable 0 is read-only. Its value gives the number of arguments (i. e. values of local variables) set when starting the macro.
- As long as the macro is running, the values of local variables can be queried with `VAR?`.

7.4 Analog Input Signals

The C-887.11 has analog inputs when photometer cards (F-206.VVU and F-206.iiU options) and/or the E-760 controller card (F-206.NCU option) are installed. The corresponding connections are located on the rear panel of the C-887.11 (p. 21).

INFORMATION

The analog inputs are accessible in the firmware of the C-887 as input signal channels with a particular identifier, see "Commandable Items" (p. 28).

➢ Get the number of the analog inputs available on the C-887.11 with the `TAC?` (p. 237) command.

Application possibilities of the analog input signals:

- Use in macros:
  - Conditional execution of the macro: see WAC (p. 248)
  - Conditional stopping of the macro execution: see MEX (p. 206)
  - Conditional jump of the macro execution pointer: see JRC (p. 199)
  - Copying the input state to a variable: see CPY (p. 151)

  Details and examples of macros are found in "Controller Macros" (p. 110).
- Use in scanning procedures:
  - Scanning procedures carried out by the C-887: see the AAP (p. 144), FIO (p. 164), FLM (p. 168), FLS (p. 171), FSA (p. 176), FSC (p. 180), FSM (p. 185) commands
  - Scanning procedures carried out by PIMikroMove (available in the ***Tools*** menu of PIMikroMove): see the PIMikroMove manual

The following commands are available for the use of analog inputs:

Command	Syntax	Function
AAP	AAP <AxisID> <Distance> <AxisID> <Distance> ["SA" <StepSize>] ["N" <NumberOfRepetitions>] ["A" <AnalogInputID>]	Starts a scanning procedure for fine determination of the maximum intensity of an analog input signal.
CPY	CPY <Variable> <CMD?>	In combination with the query command TAV?, copies the voltage value of an analog input line to a variable. Use in macros to set local variables.
DRC	DRC {<RecTableID> <Source> <RecOption>}	Determines the data source to be used and the kind of data to be recorded (record option) for the given data recorder table. The values of the analog input lines can be recorded with the record options 17 and 18.
FIO	FIO <AxisID> <Distance> <AxisID> <Distance> ["S" <LinearSpiralStepSize>] ["AR" <AngularScanSize>] ["L" <Threshold>] ["A" <AnalogInputID>]	Starts a scanning procedure for the alignment of optical elements (e.g. optical fibers), the input and output of which are on the same side. Within the element, light is transmitted from the input to the output. In order for a sensor that is connected with an analog input of the controller to receive the signal at the output of the optical element with maximum intensity, the input and output of the optical element must be aligned at the same time.
FLM	FLM <AxisID> <Distance> ["L" <Threshold>] ["A" <AnalogInputID>] ["D" <ScanDirection>]	Starts a scanning procedure to determine the global maximum intensity of an analog input signal along a distance.

Command	Syntax	Function
FLS	FLS <AxisID> <Distance> ["L" <Threshold>] ["A" <AnalogInputID>] ["D" <ScanDirection>]	Starts a scanning procedure which scans a specified distance along an axis until the analog input signal reaches a specified intensity threshold.
FSA	FSA <Axis1ID> <Distance1> <Axis2ID> <Distance2> ["L" <Threshold>] ["S" <ScanLineDistance>] ["SA" <StepSize>] ["A" <AnalogInputID>]	Starts a scanning procedure to determine the maximum intensity of an analog input signal in a plane.
FSC	FSC <Axis1ID> <Distance1> <Axis2ID> <Distance2> ["L" <Threshold>] ["S" <ScanLineDistance>] ["A" <AnalogInputID>]	Starts a scanning procedure which scans a specified area until the analog input signal reaches a specified intensity threshold.
FSM	FSM <Axis1ID> <Distance1> <Axis2ID> <Distance2> ["L" <Threshold>] ["S" <ScanLineDistance>] ["A" <AnalogInputID>]	Starts a scanning procedure to determine the global maximum intensity of an analog input signal in a plane.
JRC	JRC <Jump> <CMD?> <OP> <Value>	Can only be used in macros. Triggers a relative jump of the macro execution pointer depending on the voltage at an analog input line when used in combination with the TAV? query command.
MEX	MEX <CMD?> <OP> <value>	Can only be used in macros. Stops the macro execution depending on the voltage at an analog input line when used in combination with the TAV? query command.
NAV	NAV {<AnalogInputID> <NumberOfReadings>}	Determines the number of readout values of the analog input that are averaged.

Command	Syntax	Function
`NAV?`	NAV? [{<AnalogInputID>}]	Gets the number of readout values of the analog input that are averaged.
`SGA`	SGA {<AnalogInputID> <Gain>}	Determines the gain value for the optical input signal of the given analog input.
`SGA?`	SGA? [{<AnalogInputID>}]	Gets the gain value for the optical input signal of the given analog input.
`TAC?`	TAC?	Gets the number of installed analog inputs.
`TAD?`	TAD? [{<InputSignalID>}]	Gets the current value of the A/D converter of the specified analog input. Using this command it is possible to check for sensor overflow.
`TAV?`	TAV? [{<AnalogInputID>}]	Gets the voltage at the analog input.
`WAC`	WAC <CMD?> <OP> <value>	Can only be used in macros. Waits until an analog input line reaches a particular voltage, when used in combination with the TAV? query command.

8 GCS Commands

In this Chapter

8.1 Selection of the GCS Syntax Version

The C-887 uses the PI General Command Set (GCS). This command set is available in two syntax versions for the C-887: GCS syntax version 1.0 (GCS 1.0) and GCS syntax version 2.0 (GCS 2.0).

When the default settings of the C-887 are used, GCS 2.0 is active. GCS 2.0 has been expanded and revised compared to GCS 1.0. GCS 2.0 is also standard for other PI controllers.

INFORMATION

The `CSV` (p. 153) command can be used to switch to GCS 1.0, in order to remain compatible with older Hexapod controllers. These older Hexapod controllers were sold together in a Hexapod system with one of the following Hexapod models: F-206, M-810, M-824, M-840 or M-850.

Since important syntax differences must be observed, however, it is recommended to only use GCS 1.0 when compatibility with existing Hexapod systems is necessary.

For the important differences between the two syntax versions, see "Differences between GCS 2.0 and GCS 1.0" (p. 131).

The present manual only describes the syntax and commands of GCS syntax version 2.0 (p. 138).

The syntax and commands of GCS syntax version 1.0 are described in the manuals MS54E (for M-824, M-840 and M-850 Hexapod systems), MP36E (for F-206 Hexapod systems) and MS186E (for M-810 Hexapod systems). These manuals are available for downloading at www.pi.ws or can be requested from the customer service department.

INFORMATION

The communication interfaces (TCP/IP, RS-232) work independently of the selected GCS syntax version.

To select the GCS syntax version, proceed as follows:

- To activate GCS 1.0, send:
  `CSV 1`

- To activate GCS 2.0, send:
  `CSV 2`

The activated GCS syntax version can be determined with `CSV?` (p. 154). The only possible response is „2“ (for GCS 2.0). In GCS syntax version 1.0, the `CSV?` command is not present and will cause an error.

INFORMATION

A start-up macro can be used to activate GCS 1.0 during switching-on or rebooting (p. 118).

8.2 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 item identifier or a command-specific parameter
[…]	Square brackets indicate an optional entry
{…}	Curly brackets indicate a repetition of entries, i.e. that it is possible to access more than one item (e.g. several axes) in one command line.
LF	LineFeed (ASCII char #10), is the default termination character (character at the end of a command line)
SP	Space (ASCII char #32), indicates a space character
"..."	Quotation marks indicate that the characters enclosed are returned or to be entered.

8.3 GCS Syntax for Syntax Version 2.0

A GCS command consists of 3 characters, e.g. CMD. The corresponding query command has a question mark appended, e.g. CMD?. Command mnemonic:

CMD ::= character1 character2 character3 [?]

Exceptions:

- Single-character commands, e.g. fast polling commands, consist only of one ASCII character. The ASCII character is written as combination of # and the character code in dezimal format, e.g. `#24`. Single-character commands are not followed by a termination character. The reply to a single-character commands is followed by a termination character, however.
- `*IDN?` (for GPIB compatibility).

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

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

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

The <AxisID> argument is used for the logical axes of the controller. Depending on the controller, an axis can be identified with up to 16 characters. All alphanumeric characters and the underscore are allowed. See "Commandable Items" for the identifiers supported by the C-887.

Example 1:

Axis 1 is to be moved to position 10.0. The unit depends on the controller (e.g. µm or mm).

Send: `MOV`SP`1`SP`10.0`LF

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

Example 2:

Two axes which are connected to the same controller are to be moved.

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

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

When all arguments are optional and are omitted, the command is executed for all possible argument values.

Example 3:

All parameters in volatile memory are to be reset.

Send: `RPA`LF

Example 4:

The position of all axes is to be queried.

Send: `POS?`LF

The reply syntax is as follows:

[<argument>[{SP<argument>}]"="]<value>LF

With multi-line replies, the space preceding the termination character is omitted in the last line:

{[<argument>[{SP<argument>}]"="]<value>SP LF}

[<argument>[{SP<argument>}]"="]<value>LF   for the last line!

In the reply the arguments are listed in the same order as in the query command.

Query command:

CMD?SP<arg3>SP <arg1>SP <arg2>LF

Reply to this command:

<arg3>"=">value3>SP LF

<arg1>"="<value1>SP LF

<arg2>"="<value2>LF

Example 5:

Send: `TSP?`SP`2`SP`1`LF

Receive: `2=-1158.4405`SP LF

`1=+0000.0000`LF

8.4 Differences between GCS 2.0 and GCS 1.0

The differences between GCS syntax version 2.0 and 1.0 concern the syntax of the commands and the use of commands and parameters.

The most important difference in the syntax of the commands is the use of spaces:

- GCS 2.0: The command mnemonic and all arguments (e.g. axis identifiers, channel identifiers, parameters, etc.) must be separated from each other by a space.
  For further information, seeGCS Syntax for Syntax Version 2.0" (p. 129)
- GCS 1.0: The space between the axis identifier and the corresponding arguments is optional.
  For further information, see the manuals MS54E (for M-824, M-840 and M-850 Hexapod systems), MP36E (for F-206 Hexapod systems) and MS186E (for M-810 Hexapod systems)

The most important differences in the use of commands and parameters:

- Starting the reference move: in GCS 2.0 with `FRF`, in GCS 1.0 with `INI`

- Query whether a target position can be approached from the current position: in GCS 2.0 with `VMO?`, in GCS 1.0 with `VMO`. The meaning of the response is inverted between GCS 2.0 and GCS 1.0.
- Data recorder: not available in GCS 1.0
- Scanning commands: in GCS 1.0 only available for the Hexapod models H-206, H-810 and H-811 as well as the corresponding predecessor models F-206, M-810 and M-811; in GCS 2.0 available for all Hexapod models and improved compared to GCS 1.0
- Parameters: can be configured in GCS 2.0 with SPA in the volatile memory, not available in GCS 1.0
- Velocity of the moving platform: specified in GCS 2.0 with `VLS`, in GCS 1.0 with `VEL`

Further details can be found in the following comparison of commands in GCS syntax version 1.0 and 2.0.

Commands that have been replaced or omitted

GCS 2.0	GCS 1.0	Note
-	DRV	Omitted without replacement in GCS 2.0
FLM	FAA	FLM also supports U, V and W
FSM	FAM	FSM also supports UW, VW and UV
FSC	FAS	FSC also supports UW, VW and UV
FLM, FLS	FSN	
HLP?	HELP	
FRF <AxisID>	INI	The reference move can be started for a specific axis with FRF in GCS 2.0. A common reference move is always performed for the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W). It is therefore always sufficient to enter a single axis, e.g. It is therefore sufficient to enter a single axis, e. g. FRF X, to start the reference move of the moving platform.
MOV	MOV!	In GCS 2.0 the behavior of MOV corresponds to the behavior of the command MOV!, which is only available in GCS 1.0 when the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1 (p. 32).
-	MWG	Omitted without replacement in GCS 2.0: In GCS 2.0, the graphic display is automatically updated on a connected monitor.
STP	STOP	

GCS 2.0	GCS 1.0	Note
-	TAV	Omitted without replacement in GCS 2.0
VLS	VEL	In GCS 2.0, VLS sets the velocity of the moving platform; VEL sets the velocity for axes A and B. In GCS 1.0, VEL sets the velocity for the moving platform and / or axes A and B.
VLS?	VEL?	In GCS 2.0, VLS? gets the velocity of the moving platform; VEL? gets the velocity of axes A and B. In GCS 1.0, VEL? gets the velocity of the moving platform and / or axes A and B.
VMO?	VMO	GCS 2.0: The query command is VMO? (with question mark) The response to VMO? in GCS 2.0 is inverted to the response to VMO in GCS 1.0 In GCS 2.0: 0 = specified position cannot be approached 1 = specified position can be approached
-	MAR!	Omitted without replacement in GCS 2.0
-	READ	Omitted without replacement in GCS 2.0

Commands with different behavior

Command	GCS 2.0	GCS 1.0
DRR?	Gives the recorded values of the data recorder as the response	Gives the result of the last DRV test command as the response
MOV	In the case of consecutive MOV commands, the target position of a new MOV overwrites the target position of the previous MOV.	If MOV is sent during a motion, the target position of the sent MOV is temporarily stored. When the motion has ended, the new target position is approached.
MOV?	Gives the last set valid target position as the response	Gives the motion status of the moving platform as the response

Command	GCS 2.0	GCS 1.0
POS?, #3	Gives the current position as the response. The current position of the moving platform is calculated from the measured positions of the individual struts.	If the Hexapod is not moving: Gives the last valid target position as the response. If the Hexapod is moving: Gives an interpolated position as the response.
VEL	Sets the velocity for axes A and B	Sets the velocity for the moving platform and / or axes A and B
VEL?	Gets the velocity of axes A and B	Gets the velocity of the moving platform and / or axes A and B
#5	The response contains the motion status of each individual axis of the moving platform.	Gives the motion status of the moving platform as a whole and / or of axes A and B as the response.
#6	The response indicates the axes for which the position has changed since the last POS? or #3 sent.	Gives whether the moving platform has changed its position since the last sent POS? as a response.

8.5 Command Overview for GCS 2.0

In this user manual, the commands of the GCS syntax version 2.0 are described. This syntax version is activated by default when the controller is switched on or rebooted.

GCS syntax version 1.0 can be selected with the CSV (p. 153) command. You can find the descriptions of the command for the GCS syntax version 1.0 in the manuals MS54E (for M-824, M-840 and M-850 Hexapod systems), MP36E (for F-206 Hexapod systems) and MS186E (for M-810 Hexapod systems).

#3 (p. 138)	Get Real Position
#4 (p. 139)	Request Status Register
#5 (p. 140)	Request Motion Status
#6 (p. 141)	Query for Position Change
#7 (p. 141)	Request Controller Ready Status
#8 (p. 142)	Query if Macro is Running
#11 (p. 142)	Get Memory Space for Trajectory Points

#24 (p. 143)	Stop All Axes
#27 (p. 143)	System Abort
*IDN? (p. 144)	Get Device Identification
AAP (p. 144)	Automated Alignment Part
ADD (p. 148)	Add and Save to Variable
CCL (p. 149)	Set Command Level
CCL? (p. 150)	Get Command Level
CPY (p. 151)	Copy into Variable
CST (p. 152)	Set Assignment of Stages to Axes
CST? (p. 152)	Get Assignment of Stages to Axes
CSV (p. 153)	Set Syntax Version
CSV? (p. 154)	Get Current Syntax Version
DEL (p. 154)	Delay the Command Interpreter
DRC (p. 155)	Set Data Recorder Configuration
DRC? (p. 158)	Get Data Recorder Configuration
DRL? (p. 158)	Get Number of Recorded Points
DRR? (p. 159)	Get Recorded Data Values
DRT (p. 161)	Set Data Recorder Trigger Source
DRT? (p. 162)	Get Data Recorder Trigger Source
ECO? (p. 163)	Echo a String
ERR? (p. 163)	Get Error Number
FIO (p. 164)	Fast Input-Output Alignment Procedure
FLM (p. 168)	Fast Line Scan to Maximum
FLS (p. 171)	Fast Line Scan
FRF (p. 174)	Fast Reference Move To Reference Switch
FRF? (p. 175)	Get Referencing Result
FSA (p. 176)	Fast Scan with Automated Alignment
FSC (p. 180)	Fast Scan with Abort
FSM (p. 185)	Fast Scan to Maximum
FSS? (p. 189)	Get Status of Fast Scan Routines

HDR? (p. 190)	Get All Data Recorder Options
HLP? (p. 192)	Get List of Available Commands
HLT (p. 193)	Halt Motion Smoothly
HPA? (p. 194)	Get List of Available Parameters
IFS (p. 195)	Set Interface Parameters as Default Values
IFS? (p. 197)	Get Interface Parameters as Default Values
IMP (p. 198)	Start Impulse and Response Measurement
JRC (p. 199)	Jump Relatively Depending on Condition
LIM? (p. 200)	Indicate Limit Switches
MAC (p. 201)	Call Macro Function
MAC? (p. 205)	List Macros
MEX (p. 206)	Stop Macro Execution due to Condition
MOV (p. 206)	Set Target Position
MOV? (p. 208)	Get Target Position
MVR (p. 209)	Set Target Relative to Current Position
NAV (p. 210)	Set Number of Readout Values to be Averaged?
NAV? (p. 211)	Get Number of Readout Value Values to be Averaged?
NLM (p. 212)	Set Low Position Soft Limit
NLM? (p. 213)	Get Low Position Soft Limit
ONT? (p. 213)	Get On Target State
PLM (p. 214)	Set High Position Soft Limit
PLM? (p. 215)	Get High Position Soft Limit
POS? (p. 216)	Get Real Position
PUN? (p. 217)	Get Position Unit
RBT (p. 217)	Reboot System
RMC? (p. 217)	List Running Macros
RON? (p. 218)	Get Reference Mode
RTR (p. 219)	Set Record Table Rate
RTR? (p. 220)	Get Record Table Rate
SAI? (p. 221)	Get List Of Current Axis Identifiers

SCT (p. 221)	Set Cycle Time
SCT? (p. 222)	Get Cycle Time
SGA (p. 223)	Set Gain
SGA? (p. 224)	Get Gain
SPA (p. 224)	Set Volatile Memory Parameters
SPA? (p. 226)	Get Volatile Memory Parameters
SPI (p. 227)	Set Pivot Point
SPI? (p. 229)	Get Pivot Point
SSL (p. 229)	Set Soft Limit
SSL? (p. 230)	Get Soft Limit Status
SSN? (p. 230)	Get Device Serial Number
SST (p. 231)	Set Step Size
SST? (p. 232)	Get Step Size
STA? (p. 232)	Query Status Register Value
STE (p. 234)	Start Step And Response Measurement
STP (p. 235)	Stop All Axes
SVO (p. 235)	Set Servo Mode
SVO? (p. 236)	Get Servo Mode
TAC? (p. 237)	Tell Analog Channels
TAD? (p. 237)	Get ADC Value of Input Signal
TAV? (p. 238)	Get Analog Input Voltage
TMN? (p. 239)	Get Minimum Commandable Position
TMX? (p. 239)	Get Maximum Commandable Position
TNR? (p. 240)	Get Number of Record Tables
TRS? (p. 241)	Indicate Reference Switch
VAR (p. 241)	Set Variable Value
VAR? (p. 243)	Get Variable Value
VEL (p. 243)	Set Closed-Loop Velocity
VEL? (p. 244)	Get Closed-Loop Velocity
VER? (p. 244)	Get Version

VLS (p. 245)	Set System Velocity
VLS? (p. 246)	Get System Velocity
VMO? (p. 246)	Virtual Move
VST? (p. 248)	Get Connectable Stages
WAC (p. 248)	Wait for Condition

8.6 Command Descriptions for GCS 2.0

#3 (Get Real Position)

Description: Gets the current axis position.

Format: #3 (single ASCII character number 3)

Arguments: None

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

where

<float> is the current axis position in physical units.

Notes: This command is identical in function to POS? (p. 216), but only one character has to be sent via the interface. Therefore #3 can also be used while the controller is performing time-consuming tasks.

The current position of axes X, Y, Z, U, V and W is calculated from the measured positions of the individual struts.

Between the switching-on of the controller and the reference point definition of the Hexapod with FRF (p. 174), the current position of the Hexapod and axes A and B is unknown. Nevertheless, the response to #3 gives the position value 0 for all axes.

The physical unit in which the axis position is given can be queried with PUN? (p. 217).

#4 (Request Status Register)

Description: Requests system status information.

Format: #4 (single ASCII character number 4)

Arguments: None

Response: The answer is bit-mapped. See below for the individual codes.

Notes: This command is identical in function to STA? (p. 232), but only one character has to be sent via the interface.

For axes K, L and M, there is no status query.

The response is the sum of the following codes in hexadecimal format. When analyzing the response, the following must be taken into account:

- Bits 14 and 15 for the motion status of axes A and B are only set when the motion has been triggered by a command.
- Unassigned bits have the value 0.

Bit:	23	22	21	20	19	18	17	16
	-	-	-	-	Refe rence move is being carried out	Reference-move axis B successful	Reference-move axis A successful	Reference-move Hexapod successful
Bit:	15	14	13	12	11	10	9	8
	Axis B in motion	Axis A in motion	Strut 6 in motion	Strut 5 in motion	Strut 4 in motion	Strut 3 in motion	Strut 2 in motion	Strut 1in motion
Bit:	7	6	5	4	3	2	1	0
	Motion error axis B	Motion error axis A	Motion error strut 6	Motion error strut 5	Motion error strut 4	Motion error strut 3	Motion error strut 2	Motion error strut 1

Example: Send: #4

Receive: 0x71804

Note: The response is given in hexadecimal format. This means: a motion error was reported for strut 3; strut 4 and 5 are in motion. The reference move of the Hexapod and axes A and B has been successfully completed.

#5 (Request Motion Status)

Description: Requests motion status of the axes.

Format: #5 (single ASCII character number 5)

Arguments: none

Response: The answer <uint> is bit-mapped and returned as the hexadecimal sum of the following codes:

1=first axis is moving
2=second axis is moving
4=third axis is moving
...

Examples: 0 indicates motion of all axes complete
3 indicates that the first and the second axis are moving

49 indicates that axes X, U and A are moving.

Note: Axes 1 to 8 correspond to the X, Y, Z, U, V, W, A and B axes in this order.

Exception: When the "NOSTAGE" stage type is assigned to an axis (possible for axes A and B; query with the CST? (p. 152) command), this axis is not included in the bit-mapped answer. In this case, it is skipped when counting the axes.

The motion status of axes K, L and M is **not** given.

#6 (Query for Position Change)

Description: Queries whether the axis positions have changed since the last position query was sent.

Format: #6 (single ASCII character number 6)

Arguments: None

Response: The response <uint> is bit-mapped and returned as the hexadecimal sum of the following codes:

1 = Position of the first axis has changed
2 = Position of the second axis has changed
4 = Position of the third axis has changed
...

Examples: 0 indicates that no axis position has changed.

3 indicates that the positions of the first and second axis have changed.

49 indicates that the positions of axes X, U and A have changed.

401 indicates that the positions of axes X and M have changed.

Notes: When the "NOSTAGE" stage type is assigned to an axis (possible for axes A and B; query with the CST? (p. 152) command), this axis is not included in the bit-mapped answer. In this case, it is skipped when counting the axes.

It is considered a position change when a new valid target position is specified with a command - even within a macro - or via a connected control unit (p. 25) (C-887.MC) since the last POS? (p. 216) or #3 (p. 138) was sent.

#7 (Request Controller Ready Status)

Description: Asks controller for ready status (tests if controller is ready to perform a new command).

Note: Use #5 (p. 140) instead of #7 to verify if motion has finished.

Format: #7 (single ASCII character number 7)

Arguments: None

Response: B1h (ASCII character 177 = "±" in Windows) if controller is ready

B0h (ASCII character 176 = "°" in Windows) if controller is not ready
(e.g., performing a reference move)

Troubleshooting: The response characters may appear differently in non-Western character sets or other operating systems.

#8 (Query If Macro Is Running)

Description: Tests if a macro is running on the controller.

Format: #8 (single ASCII character number 8)

Arguments: none

Response: <uint>=0 no macro is running
<uint>=1 a macro is currently running

#11 (Get Memory Space for Trajectory Points)

Description: Gets the free memory space for the points of the motion profile.

Format: #11 (single ASCII character number 11)

Arguments: None

Response: <uint> is the free memory space, given as the number of the motion profile points.

Notes: #11 gets the free memory space of a buffer that contains the motion profile points for the Hexapod. One motion profile point corresponds to a set of target positions for the axes of the Hexapod (X, Y, Z, U, V, W). The content of the buffer is only used to define the motion profile when the 0x19001900 and 0x19001901 parameters both have the value 1.

For more information, see "Motions of the Hexapod" (p. 32).

#24 (Stop All Axes)

Description:	Stops all axes abruptly. For details see the notes below. Sets error code to 10. This command is identical in function to STP (p. 235), but only one character must be send via the interface.
Format:	#24 (ASCII character 24)
Arguments:	none
Response:	none
Notes:	#24 stops all axis motions that are caused by motion commands (MOV (p. 206), MVR (p. 209), FRF (p. 174), AAP (p. 144), FIO (p. 164), FLM (p. 168), FLS (p. 171), FSA (p. 176), FSC (p. 180), FSM (p. 185), IMP (p. 198), STE (p. 234)). #24 stops macros. After the axes are stopped, their target positions are set to their current positions.

#27 (System Abort)

Description:	Stops the controller.
Format:	#27 (single ASCII character number 27)
Arguments:	None
Response:	None

Notes: #27 triggers the following:

- Motors of the struts are switched off
- Servo mode is switched off
- Commands are no longer processed
- Parameters in the volatile memory are set to default values
- Power supply of the controller remains switched on

For a reboot, the controller must be switched off and back on again with the standby switch.

*IDN? (Get Device Identification)

Description: Reports the device identity number.

Format: *IDN?

Arguments: None

Response: Single-line text terminated with a termination character (line feed) with controller name, serial number, and firmware version

Note: For C-887, *IDN? responds something like:

```
©2012 Physik Instrumente(PI) Karlsruhe, C-
887,111160398,1.2.0.0
```

AAP (Automated Alignment Part)

Description: Starts a scanning procedure for a detailed determination of the maximum intensity of an analog input signal.

The scanning procedure started with AAP corresponds to the "fine portion" of the scanning procedure that was started with the FSA (p. 176) command.

Scans a specified area ("scanning area") using a gradient formation in the intensity until a local maximum of the analog input signal is found.

The plane in which the scanning area is located can be defined by one of the following axis pairs:

X Y

Y Z

X Z

U V

U W

V W

The scanning procedure starts at the current position at the time the command is received ("initial position"). The scanning area is centered around the initial position.

The step size of the motion is automatically adjusted in order to detect the local maximum as precisely as possible.

AAP (and thus the motion of the platform) has been successfully completed when the number of checks specified with <NumberOfRepetitions> has confirmed that the local maximum is present at the current position. The influence of a superimposed noise signal can be reduced by multiple checks.

When <NumberOfRepetitions> is set to zero, a gradient of the intensity is continually sought. With this setting, a moving local maximum can be continuously tracked. The scanning procedure does not end until it is aborted with #24 (p. 143), STP (p. 235) or HLT (p. 193) and is then considered to be unsuccessfully completed.

AAP has been unsuccessfully completed in the following cases:

- No gradient of the intensity was found: the moving platform returns to the initial position.
- The scanning area would be exceeded: the moving platform returns to the initial position.
- #24 (p. 143), STP (p. 235) or HLT (p. 193) has been sent: the moving platform remains in the current position.

FSS? (p. 189) can be used to check whether a scanning procedure has been successfully completed.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).

Format: AAP <AxisID> <Distance> <AxisID> <Distance> ["SA" <StepSize>] ["N" <NumberOfRepetitions>] ["A" <AnalogInputID>]

Arguments: <AxisID>: Is one axis of the controller, format: string

Axes X, Y, Z, U, V, W are permissible.

<Distance>: distance to be scanned along the axis, format: Double

"SA": Required keyword for entering <StepSize>

<StepSize>: Starting value for the step size, format: double

"N": Required keyword for entering <NumberOfRepetitions>

<NumberOfRepetitions>: Number of successful checks of the local maximum at the current position that is required for successfully completing AAP. Format: integer

"A": Required keyword for entering <AnalogInputID>

<AnalogInputID>: is the identifier of the analog input signal whose maximum intensity is sought, format: Integer

Response: None

Notes:

The physical unit in which <Distance> and <Stepsize> are to be given can be queried with PUN?

The values for <Distance> must be identical for both axes.

The following default values are used when the corresponding arguments are omitted:

<StepSize>: 0.001

<NumberOfRepetitions>: 3

<AnalogInputID>: 1

These default values have only been checked for the Hexapods H-810, H-811 and H-206. For other Hexapods, the ideal parameters must be determined experimentally.

If there are several local maxima in the scanning area, FSM (p. 185) can be used to find the global maximum intensity before AAP is called. It is recommended to first perform an FSM scanning procedure with a large step size over the entire scanning area, and then to scan a smaller area with FSM using a smaller step size, and finally to optimize the alignment of the moving platform with the found global maximum intensity with AAP.

The greater the noise of the signal to be scanned, the more frequently it should be checked whether a local maximum has been reached, i.e. a large value should be selected for <NumberOfRepetitions> if possible.

The greater the noise of the signal to be scanned, the higher the number of readout values of the analog signal should be set with NAV (p. 210) for averaging.
At the same time, it should be noted that this setting increases the time required for the scanning procedure.

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands, and AAP is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example:

Send: AAP Y 0.1 Z 0.1 SA 0.001 N 3 A 2

Starts a scanning procedure in the YZ plane:

- Side length of the square area: 0.1 mm
- Start value of the step size: 1 µm
- Number of successful checks of the local maximum after which AAP has been successfully completed: 3.
- Identifier of the analog input channel whose maximum intensity is sought: 2

ADD (Add and Save to Variable)

Description: Adds two values and saves the result to a variable.

The variable is present in volatile memory (RAM) only.

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

Arguments: <Variable> is the name of the variable to which the result is to be saved.

<FLOAT1> is the first summand.

<FLOAT2> is the second summand.

For the summands, floating point numbers are expected. They can be given directly or via the value of a variable.

Response: None

Note: ADD can only be used in macros.

Example 1: Value $B is added to value $A, and the result is saved to variable C:

Send: `ADD C $A $B`

Example 2: The name of the variable to which the result is to be copied is given via the value of another variable:

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

CCL (Set Command Level)

Description: Changes the active "command level" and thus determines the availability of commands and of write access to system parameters.

Format: CCL <Level> [<PSWD>]

Arguments:	<Level> is a command level of the controller <PSWD> is the password required for changing to the appropriate command level The following command levels and passwords are valid: Level = 0 is the default setting, all commands provided for "normal" users are available, read access to all parameters, no password required. Level = 1 adds additional commands and write access to level-1 parameters (commands and parameters from level 0 are included). The required password is "advanced". Level > 1 is provided for PI service personnel only. Users cannot change to a level > 1. Contact the customer service department if there seem to be problems with level 2 or higher parameters.
Response:	None
Troubleshooting:	Invalid password
Notes:	HLP? (p. 192) lists all commands available in the current command level. HPA? (p. 194) lists the parameters including the information about which command level allows write access to them. For more information about using parameters, see "Adjusting Settings" (p. 269). After controller power-on or reboot, the active command level is always level 0.

CCL? (Get Command Level)

Description:	Get the active "command level".
Format:	CCL?

Arguments: none

Response: <Level> is the currently active command level; uint.

Notes: <Level> should be 0 or 1.

<Level> = 0 is the default setting, all commands provided for "normal" users are available, as is read access to all parameters

<Level> = 1 adds additional commands and write access to level 1 parameters (commands and parameters from level 0 are included).

CPY (Copy into Variable)

Description: Copies a command response to a variable.

The variable is present in volatile memory (RAM) only.

Format: CPY <Variable> <CMD?>

Arguments: <Variable> is the name of the variable to which the command response is to be copied.

<CMD?> is one query command in its usual notation. The response has to be a single value and not more.

Response: None

Note: CPY can only be used in macros.

Example: It is possible to copy the value of one variable (e.g. SOURCE) to another variable (e.g. TARGET):

Send: `CPY TARGET VAR? SOURCE`

CST (Set Assignment of Stages to Axes)

Description: Assigns a stage type to an axis.

Format: CST {<AxisID> <StageName>}

Arguments: <AxisID> is one axis of the controller

<StageName> is the name of the stage type.

Notes: The corresponding stage type is automatically assigned to all axes when the controller is switched on or rebooted.

The assignment of a stage type with CST is only permissible for axes A and B. In order to change the standard assignment for A and B in the volatile memory, CST can be used e.g. in a start-up macro.

CST loads the operating parameters of the assigned stage type from a stage database on the controller into the working memory. The permissible stage types can be listed with the VST? (p. 248) command.

CST also switches on servo mode for axes A and B.

CST? (Get Assignment of Stages to Axes)

Description: Gets the name of the stage type that is connected to the given axis.

Format: CST? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<string> LF}

where

<string> is the name of the stage type that is assigned to the axis.

Notes: The corresponding stage type is automatically assigned to all axes when the controller is switched on or rebooted. For axes A and B, the assignment can be changed with CST (p. 152) in the volatile memory.

CSV (Set Syntax Version)

Description: Set current GCS syntax version used in the firmware.

Format: CSV <SyntaxVersion>

Arguments: <SyntaxVersion> is the GCS syntax version to be used, can be 1.0 (for GCS 1.0) or 2.0 (for GCS 2.0)

Response: None

Notes: When the defaults of the C-887 are used, GCS syntax version 2.0 is active after a switch-on or reboot.

When CSV is used to switch to GCS syntax version 1.0, only the commands of this syntax version are available.

A description of GCS syntax version 1.0 and the corresponding commands can be found in the manuals MS54E (for M-824, M-840 and M-850 Hexapod systems), MP36E (for F-206 Hexapod systems) and MS186E (for M-810 Hexapod systems).

In the case of questions, contact our customer service department.

In order to activate GCS syntax version 1.0 directly after the controller is switched on or rebooted, a start-up macro can be used that contains the corresponding CSV command. The macro execution is stopped when the CSV command is processed. Thus all commands following the CSV command in the macro are ineffective. Further information can be found in "Controller Macros" (p. 110).

CSV? (p. 154) can be used to query which GCS syntax version is active:

When GCS syntax version 2.0 is active, the response is "2".

When GCS syntax version 1.0 is active, an error will occur since the CSV? command is not present in GCS syntax version 1.0.

The HLP? (p. 192) command can be used to list the commands available for the current GCS syntax version.

In order to switch back to GCS syntax version 2.0, send:

```
CSV 2
```

Further information can be found "Selection of the GCS Syntax Version" (p. 127).

CSV? (Get Current Syntax Version)

Description: Gets current GCS syntax version used in the firmware.

Format: CSV?

Arguments: None

Response: The current GCS syntax version

Note: The only possible response is 2.0 (for GCS 2.0). In GCS syntax version 1.0, the CSV? command is not present and will cause an error.

DEL (Delay The Command Interpreter)

Description: Delays <uint> milliseconds.

Format: DEL <uint>

Arguments: <uint> is the delay value in milliseconds.

Response: none

Note: DEL should only be used in macros.

DRC (Set Data Recorder Configuration)

Description:	Determines the data source to be used and the kind of data to be recorded (record option) for the data recorder table given.
Format:	DRC {<RecTableID> <Source> <RecOption>}
Arguments:	<RecTableID>: is one data recorder table of the controller, see below. <Source>: is the data source, for example, an axis or a channel of the controller. The required source depends on the selected record option. <RecOption>: is the kind of data to be recorded (record option). See below for a list of the available record options and the corresponding data sources.
Response:	None
Notes:	The number of available data recorder tables can be read with TNR? (p. 240). The C-887 has 16 data recorder tables. The maximum number of points per data recorder table available for data recording is 10240. With HDR? (p. 190) you will obtain a list of available record options and information about additional parameters and commands concerned with data recording. For more information see "Data Recording" (p. 106).

Record options for the corresponding data sources:	<Source>	<RecOption>
	Axis	0 = Nothing is recorded
		1 = Commanded position of axis
		2 = Real position of axis
		3 = Position error of axis
		8 = Measurement time
		70 = Commanded velocity of axis
		71 = Commanded acceleration of axis
		72 = Real velocity of axis
		73 = Motor output of axis
		76 = Current integrated position error of axis
		80 = Status register of axis Further information can be found in the section "Signal Status Register".
	Input signal channel	17 = Input value of channel, calculated, in volt.
		18 = Input value of channel, directly from channel, without dimension.

HDR? indicates for which axes or channels the individual record option can be used. For details on existing axes and channels, see"Commandable Items" (p. 28).

The data recorder records with a maximum frequency of 50 kHz (can be set with RTR (p. 219)). The individual hardware components of the C-887 provide the values to be recorded with different clock rates, however.

If a recording contains data that could not be recorded with the frequency of the data recorder, the data points can no longer be allocated to the respective recording time.

The "Sample time" information will then be lacking in the recording. The time itself should therefore always be recorded as well (record option 8 "Measurement time" preset for data recorder table 13).

The defaults of the data recorder are as following:

```
drc?
1=X 1
2=X 2
3=Y 1
4=Y 2
5=Z 1
6=Z 2
7=U 1
8=U 2
9=V 1
10=V 2
11=W 1
12=W 2
13=1 8
14=0 0
15=0 0
16=0 0
```

Example: Send: DRC 4 X 2

The current position of axis X is to be recorded in data recorder table 4.

DRC? (Get Data Recorder Configuration)

Description: Returns settings made with DRC (p. 155).

Format: DRC? [{<RecTableID>}]

Arguments: <RecTableID>: is one data recorder table of the controller; if omitted settings for all tables are given.

Response: The current DRC settings:

{<RecTableID>"="<Source> <RecOption> LF}

where

<Source>: is the data source, for example an axis or an output signal channel of the controller. The source type depends on the record option.

<RecOption>: is the kind of data to be recorded (record option).

See DRC for a list of the available record options and the corresponding data sources.

DRL? (Get Number of Recorded Points)

Description: Reads the number of points comprised by the last recording.

Format: DRL? [{<RecTableID>}]

Arguments: <RecTableID> is one data recorder table of the controller

Response: {<RecTableID>"="<uint> LF}

where

<uint> gives the number of points recorded with the last recording

Notes: The number of points is reset to zero for a data recorder table when changing its configuration with DRC (p. 155).

DRR? (Get Recorded Data Values)

Description: Reading of the last recorded data set.

Reading can take some time depending on the number of points to be read!

It is possible to read the data while recording is still in progress.

Format: DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]

Arguments: <StartPoint>: is the first point to be read from the data recorder table, starts with index 1

<NumberOfPoints>: is the number of points to be read per table

<RecTableID>: is one data recorder table of the controller

Response: The recorded data in GCS array format, see the separate manual for GCS array, SM146E, and the example below

Notes: If the value -1 is set for <NumberOfPoints>, all valid points of the selected tables will be read out

If <RecTableID> is omitted, all tables whose record option set with DRC (p. 155) is not equal to zero are read out.

With HDR? (p. 190) you will obtain a list of available record options and trigger options and information about additional parameters and commands concerned with data recording.

For more information see "Data Recording" (p. 106).

Example:

```
drc 1 X 1 2 X 2
drt 1 1 1
mov x 2
drr? 1 5 1 2 13
# REM data recorded with C-887 controller
#
# TYPE = 0
# SEPARATOR = 9
# DIM = 3
# NDATA = 5
# SAMPLE_TIME = 0.005000
#
# NAME0 = TARGET POSITION X
# NAME1 = REAL POSITION X
# NAME2 = MEASUREMENT TIME Strut 1
# DISPUNIT0 = mm
# DISPUNIT1 = mm
# DISPUNIT2 = sec
# END_HEADER
0.004220971838 -0.000364157866 0
0.064530499279 0.036015000194 0.005040000193
0.145174950361 0.140960231423 0.010040000081
0.232607111335 0.216114446521 0.015200000256
0.309781700373 0.30354321003 0.020020000637
```

DRC (p. 155) is used to determine that the current position of axis X is to be recorded in data recorder table 1 and that the commanded position of axis X is to be recorded in data recorder table 2.

DRT (p. 161) is used to determine that a recording is to be triggered by the next motion command, e.g. by MOV (p. 206).

MOV triggers the motion of axis X to position 2.

DRR? is used to get the first five points of data recorder tables 1, 2 and 13.

The default setting of the data recorder is that the time is recorded in data recorder table 13.

DRT (Set Data Recorder Trigger Source)

Description: Defines a trigger source for the given data recorder table.

Format: DRT <RecTableID> <TriggerSource> <Value>

Arguments: <RecTableID> is one data recorder table of the controller. See below for details.

<TriggerSource> ID of the trigger source, see below for a list of available options

<Value> depends on the trigger source, can be a dummy, see below

Response: none

Notes: Regardless of the data recorder table selected with <RecTableID>, the trigger option selected with <Triggersource> is always set for all data recorder tables.

With HDR? (p. 190) you will obtain a list of available record and trigger options and additional information about data recording.

IMP (p. 198) and STE (p. 234) each trigger a recording by the data recorder regardless of the DRT settings.

The data recorder configuration, i.e. the assignment of data sources and record options to the recorder tables, is set with DRC (p. 155). A recording is actually only started for data recorder tables whose record option set with DRC is not equal to zero.

The recording ends when the maximum number of points is reached (specified by the ***Data Recorder Points Per Table*** parameter, ID 0x16000201).

For more information see "Data Recording" (p. 106).

Available trigger options:

0 = No trigger. Exception: STE and IMP always trigger data recording;
Data recording is not triggered, and data recording in progress is continued until the maximum number of points is reached. Exception: STE and IMP always trigger a recording. <Value> must be a dummy.

1 = Trigger with next command that changes the position; e.g. MOV (p. 206), MVR (p. 209); <Value> must be a dummy. Default setting

2 = Trigger with next command;
sets trigger to the trigger option 0 after execution; <Value> must be a dummy.

4 = Trigger immediately;
sets trigger to the trigger option 0 after execution; <Value> must be a dummy.

6 = Trigger with next command that changes the position, e.g. MOV, MVR; sets trigger to the trigger option 0 after execution; <Value> must be a dummy.

If trigger option 2, 4 or 6 is set, then trigger option 0 is automatically activated after the recording is triggered. As a result, the recording continues until the maximum number of points of the data recorder table(s) is reached. In this way, several motions can be recorded in succession. Trigger options 2, 4 or 6 are a good idea e.g. when a motion profile is to be defined by consecutive MOV commands (see MOV).

DRT? (Get Data Recorder Trigger Source)

Description: Gets the trigger source for the data recorder tables.

Format: DRT? [{<RecTableID>}]

Arguments: <RecTableID> is a data recorder table of the controller.

Response: {<RecTableID>"="<TriggerSource> <Value> LF}

where

<TriggerSource> is the identifier of the trigger source.

<Value> depends on the trigger source; 0 is a dummy.

Further information is found in the description of the DRT (p. 161) command.

ECO? (Echo a String)

Description: Returns a string.

ECO? can be used to test the communication.

Format: ECO? <String>

Arguments: <String> is any given combination of characters consisting of letters and numbers

Response: <String> LF

Note: <String> can consist of up to 100 characters.

ERR? (Get Error Number)

Description: Get error code <int> of the last occurred error and reset the error to 0.

Only the last error is buffered. Therefore you should call ERR? after each command.

The error codes and their descriptions are fully listed in "Error Codes" (p. 250).

Format: ERR?

Arguments: none

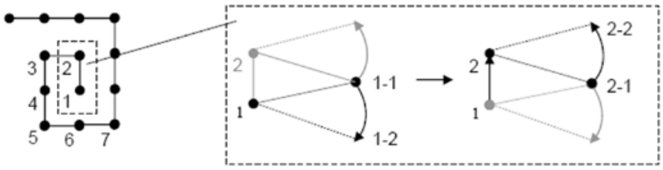
Response: The error code of the last occurred error (int).

Troubleshooting: Communication breakdown

FIO (Fast Input-Output Alignment Procedure)

Description: Starts a scanning procedure for the alignment of optical elements (e.g. optical fibers), the input and output of which are on the same side. Within the element, light is transmitted from the input to the output. In order for a sensor that is connected with the analog input channel of the controller to receive the signal at the output of the optical element with maximum intensity, the input and output of the optical element must be aligned at the same time.

Moves along a linear spiral in a specified area with a specified step length, see figure



The level in which the specified area is located can be defined by one of the following axis pairs:

X Z

Y Z

X Y

The scanning procedure starts at the current position at the time the command is received ("initial position").

At the start point and after each step, scanning is done at a specified angle around the pivot point until an analog input signal reaches a specified intensity threshold. This corresponds to an angular scan around the axis that is perpendicular to the scanned area.

The pivot point must be at the position that corresponds to the center of the input or the center of the output of the optical element. The coordinates of the pivot point can be set with SPI (p. 227).

FIO (and thus the motion of the platform) has been successfully completed when the analog input signal reaches the specified intensity threshold.

FIO has been unsuccessfully completed in the following cases:

- The intensity threshold has not been reached in the specified area: the moving platform returns to the initial position.
- #24 (p. 143), STP (p. 235) or HLT (p. 193) has been sent: the moving platform remains in the current position.

FSS? (p. 189) can be used to check whether a scanning procedure has been successfully completed.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).

Format:

FIO <AxisID> <Distance> <AxisID> <Distance>
["S" <LinearSpiralStepSize>] ["AR" <AngularScanSize>]
["L" <Threshold>] ["A" <AnalogInputID>]

Arguments:

<AxisID>: Is one axis of the controller, format: string

Axes X, Y, and Z are permissible.

<Distance>: Distance to be moved along the axis, format: double

"S": Required keyword for entering <LinearSpiralStepSize>

<LinearSpiralStepSize>: Step size in which the platform moves along the spiral path, format: double

"AR": Required keyword for entering <AngularScanSize>

<AngularScanSize>: Angle around the pivot point at which scanning is done, in degrees, format: double

"L": Required keyword for entering <Threshold>

<Threshold>: Intensity threshold of the analog input signal, in V, format: Double

"A": Required keyword for entering <AnalogInputID>

<AnalogInputID>: is the identifier of the analog input signal whose maximum intensity is sought, format: Integer

Response: None

Notes: The physical unit in which <Distance> and <LinearSpiralStepSize> are to be given can be queried with PUN? (p. 217).

The area specified by the values for <Distance> must be square.

The following default values are used when the corresponding arguments are omitted:

<LinearSpiralStepSize>: 0.01 mm

<AngularScanSize: 0.2 degrees

<Threshold>: 1.0 V

<AnalogInputID>: 1

These default values have only been checked for the H-810, H-811, and H-206 Hexapod models. For other Hexapod models, the ideal parameters must be determined experimentally.

The lower the system velocity is set with VLS (p. 245) during a scanning procedure, the greater the accuracy with which the threshold of the intensity is found.

Velocities in the range of under 1 mm/s are recommended.

Note:

If the values for distances or angles are too large, the platform moves on an undefined path during a scanning procedure and it can tilt. Thus collisions and an unsatisfying result of the scanning procedure are possible. Measures for avoiding tipping:

- ➢ Select values of up to 0.2 degrees for <AngularScanSize>
- ➢ Suitably align the moving platform before starting the scanning procedure.
- ➢ Use suitable holders for the inputs and/or outputs of the optical element to be aligned on the moving platform so that the motion during the scanning procedure only takes place over short distances or angles.

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands, and FIO is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example: FIO Y 0.1 Z 0.1 S 0.01 AR 0.1 L 1 A 2

Starts a scanning procedure along a linear spiral in the YZ level:

- Side length of the square area: 0.1 mm
- Step size: 10 µm
- Angle around the pivot point: 0.1 degrees (for axis U)
- Intensity threshold: 1 V
- Identifier of the analog input channel whose maximum intensity is sought: 2

FLM (Fast Line Scan to Maximum)

Description:

Starts a scanning procedure to determine the global maximum intensity of an analog input signal.

Completely scans a specified distance along an axis for the intensity of the analog input signal. In the case of several maximum intensities, this prevents that only a local maximum is found instead of the global maximum.

The direction of the scanning procedure as well as the starting and end position of the distance can be specified with the argument <ScanDirection>.

FLM (and thus the motion of the platform) has been successfully completed when the two following conditions have been met:

- The analog input signal has reached the specified intensity threshold on the specified distance at least once.
- The moving platform has returned from the end position of the distance to the position with the maximum intensity.

FLM has been unsuccessfully completed in the following cases:

- The intensity threshold has not been reached on the specified distance: the moving platform returns from the end position of the distance to the initial position (current position at the time the command is received).
- #24 (p. 143), STP (p. 235) or HLT (p. 193) has been sent: the moving platform remains in the current position.

FSS? (p. 189) can be used to check whether a scanning procedure has been successfully completed.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).
A similar scanning procedure can be started with FLS (p. 171). In contrast to FLM, FLS has already been

successfully completed when the intensity threshold has been reached for the first time.

Format: FLM <AxisID> <Distance> ["L" <Threshold>] ["A" <AnalogInputID>] ["D" <ScanDirection>]

Arguments: <AxisID>: Is one axis of the controller, format: string

Axes X, Y, Z, U, V, W are permissible.

<Distance>: distance to be scanned along the axis, format: Double

"L": Required keyword for entering <Threshold>

<Threshold>: Intensity threshold of the analog input signal, in V, format: Double

"A": Required keyword for entering <AnalogInputID>

<AnalogInputID>: is the identifier of the analog input signal whose maximum intensity is sought, format: Integer

"D": Required keyword for entering <ScanDirection>

<ScanDirection>: Specifies the direction of the scanning procedure as well as the starting and ending position of the distance:

0: Scanning procedure is done centered around the current position, in positive direction.

Starting position = current position - <Distance>/2

Ending position = current position + <Distance>/2

1: Scanning procedure in positive direction:

Starting position = current position

Ending position = current position + <Distance>

-1: Scanning procedure in negative direction:

Starting position = current position

Ending position = current position - <Distance>

Response: None

Notes:

The physical unit in which <Distance> is to be given can be queried with PUN? (p. 217).

The following default values are used when the corresponding arguments are omitted:

<Threshold>: 1.0 V

<AnalogInputID>: 1

<ScanDirection>: 0 (scanning procedure centered around the current position, in positive direction)

The lower the system velocity is set with VLS (p. 245) during a scanning procedure, the greater the accuracy with which the maximum intensity is found.

Velocities in the range of under 1 mm/s are recommended.

If the values for distances or angles are too large, the platform moves on an undefined path during a scanning procedure and it can tilt. Thus collisions and an unsatisfying result of the scanning procedure are possible. Measures for avoiding tipping:

- Select suitable values for <Distance>. For the Hexapods H-810, H-811 and H-206, 0.2 mm or 0.2 degrees should not be exceeded; for other Hexapod models, the ideal values must be determined experimentally.
- Suitably align the moving platform before starting the scanning procedure.
- Use suitable holders for the inputs and/or outputs of the optical element to be aligned on the moving platform so that the motion during the scanning procedure only takes place over short distances or angles.

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands, and FLM is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example:

FLM Z 0.2 L 1.2 A 2 D 0

Starts a scanning procedure along the Z axis:

- Distance: 0.2 mm
- Intensity threshold: 1.2 V
- Identifier of the analog input channel whose maximum intensity is sought: 2
- Scanning procedure takes place centered around the current position, in positive direction

FLS (Fast Line Scan)

Description:

Starts a scanning procedure which scans a specified distance along an axis until the analog input signal reaches a specified intensity threshold.

The direction of the scanning procedure as well as the start position of the distance can be specified with the argument <ScanDirection>.

When the intensity threshold has already been reached at the initial position (current position at the time the command is received), a different start position will not be approached.

FLS (and thus the motion of the platform) has been successfully completed when the analog input signal reaches the specified intensity threshold. If necessary due to the braking distance, the platform will return to the position at which the intensity threshold was reached after stopping.

FLS has been unsuccessfully completed in the following cases:

- The intensity threshold has not been reached on the specified distance: the moving platform returns from the end position of the distance to the initial position.
- #24 (p. 143), STP (p. 235) or HLT (p. 193) has been sent: the moving platform remains in the current position.

FSS? (p. 189) can be used to check whether a scanning procedure has been successfully completed.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).

A similar scanning procedure can be started with FLM (p. 168). In contrast to FLS, FLM scans along the entire specified distance, so that the global maximum is found if several local maxima are present.

Format:

FLS <AxisID> <Distance> ["L" <Threshold>]
["A" <AnalogInputID>] ["D" <ScanDirection>]

Arguments:

<AxisID>: Is one axis of the controller, format: string

Axes X, Y, Z, U, V, and W are permissible.

<Distance>: distance to be scanned along the axis, format: Double

"L": Required keyword for entering <Threshold>

<Threshold>: Intensity threshold of the analog input signal, in V, format: Double

"A": Required keyword for entering <AnalogInputID>

<AnalogInputID>: is the identifier of the analog input signal whose maximum intensity is sought, format: Integer

"D": Required keyword for entering <ScanDirection>

<ScanDirection>: Specifies the direction of the scanning procedure as well as the start and end position of the distance:

0: Scanning procedure is done centered around the current position, in positive direction.

Starting position = current position - <Distance>/2

Ending position = current position + <Distance>/2

1: Scanning procedure in positive direction:
Starting position = current postion
Ending position = current position + <Distance>

-1: Scanning procedure in negative direction:
Starting position = current postion
Ending position = current position - <Distance>

Response: None

Notes: The physical unit in which the <Distance> is to be given can be queried with PUN?

The following default values are used when the corresponding arguments are omitted:
<Threshold>: 1.0 V
<AnalogInputID>: 1.
<ScanDirection>: 0 (scanning procedure centered around the current position, in positive direction)

The lower the system velocity is set with VLS (p. 245) during a scanning procedure, the greater the accuracy with which the maximum intensity is found.
Velocities in the range of under 1 mm/s are recommended.

If the values for distances or angles are too large, the platform moves on an undefined path during a scanning procedure and it can tilt. Thus collisions and an unsatisfying result of the scanning procedure are possible. Measures for avoiding tipping:

- Select suitable values for <Distance>. For the Hexapods H-810, H-811 and H-206, 0.2 mm or 0.2 degrees should not be exceeded; for other Hexapod models, the ideal values must be determined experimentally.
- Suitably align the moving platform before starting the scanning procedure.
- Use suitable holders for the inputs and/or outputs of the

optical element to be aligned on the moving platform so that the motion during the scanning procedure only takes place over short distances or angles.

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands, and FLS is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example: FLS Z 0.2 L 1 A 2 D 0

Starts a scanning procedure along the Z axis:

- Distance: 0.2 mm
- Intensity threshold: 1 V
- Identifier of the analog input channel whose maximum intensity is sought: 2
- Scanning procedure takes place centered around the current position, in positive direction

FRF (Fast Reference Move To Reference Switch)

Description: Performs a reference move.

Moves the given axis to the reference point switch and sets the current position to a defined value. See below for details.

If multiple axes are given in the command, they are moved synchronously.

Format: FRF [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller, if omitted, all axes are affected.

Response: none

Troubleshooting: Illegal axis identifier

Notes: For the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W) and axes A and B, motions can only be commanded after a successful reference move (also referred to as "initialization").

During a reference move, the Hexapod moves unpredictably. A collision check or prevention is **not** carried out, even if a configuration for preventing collisions was stored on the C-887 with the PIVeriMove Software for the Collision Check.

It is therefore always sufficient to enter a single axis, e.g. It is therefore always sufficient to enter a single axis, e.g. FRF X, to start the reference move of the moving platform.

FRF also switches the servo mode on for the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W).

The P-611.3SF NanoCube® stage is equipped with absolute-measuring sensors. A reference move is therefore not intended for the axes of this stage (K, L, M).

FRF can be aborted by #24 (p. 143), STP (p. 235) and HLT (p. 193).

Use FRF? (p. 175) to check whether the reference move was successful.

FRF? (Get Referencing Result)

Description: Indicates whether the given axis is referenced or not.

Format: FRF? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<uint> LF}

where

<uint> indicates whether the axis has been successfully referenced (=1) or not (=0).

Troubleshooting: Illegal axis identifier

Note: Axes X, Y, Z, U, V, W, A and B are considered to be "referenced" when a reference move has been successfully completed with FRF (p. 174).

Axes A and B are considered to be "referenced" even if the parameter ***Sensor Reference Mode*** (ID 0x02000A00) was changed to value 1 (p. 98) with SPA (p. 224). It is necessary to change the parameter value when the connected stage is not equipped with a reference point switch and can therefore not carry out a reference move.

Axes K, L and M are equipped with absolute-measuring sensors and do not need a reference move. For this reason, FRF? always responds with 1 for these axes.

FSA (Fast Scan with Automated Alignment)

Description: Starts a scanning procedure to determine the maximum intensity of an analog input signal in a plane. The search consists of two subprocedures:

"Coarse portion"; corresponds to the procedure that is started with the FSC (p. 180) command

"Fine portion"; corresponds to the procedure that is started with the AAP (p. 144) command

The fine portion is only executed when the coarse portion has previously been successfully completed. For more detailed descriptions of the two subprocedures, see AAP and FSC.

Note:

For the fine portion, the scanning area is reset using the values for <Distance1> and <Distance2> specified with FSA, so that the start position of the fine portion is in the center of the specified area. As a result, the scanning area can be enlarged to up to double the original area.

- ➢ Make sure that the moving platform can safely move outside of the originally specified scanning area as well.

The plane in which the scanning area is located can be defined by one of the following axis pairs:

X Z

Y Z

X Y

FSA (and thus the motion of the platform) has been successfully completed when the fine portion has been successfully completed, i.e. when three consecutive checks have confirmed that the local maximum is present at the current position. The purpose of the checks is to reduce the influence of any superimposed noise signal.

FSA has been unsuccessfully completed in the following cases:

- During the coarse portion, the intensity threshold has not been reached in the scanning area: the moving platform returns to the initial position.
- No gradient of the intensity was found during the fine portion: the moving platform returns to the start position of the fine portion.
- The specified area would be exceeded during the fine portion: the moving platform returns to the start position of the fine portion.
- #24 (p. 143), STP (p. 235) or HLT (p. 193) has been sent: the moving platform remains in the current position.

FSS? (p. 189) can be used to check whether a scanning procedure has been successfully completed.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).

Format: FSA <Axis1ID> <Distance1> <Axis2ID> <Distance2>
["L" <Threshold>] ["S" <ScanLineDistance>]
["SA" <StepSize>] ["A" <AnalogInputID>]

Arguments:

<Axis1ID>: Is one axis of the controller, format: string

Axes X, Y, and Z are permissible.

During the coarse portion, the platform moves in this axis from scanning line to scanning line by the distance given by <ScanlineDistance>.

<Distance1>: Side length of the scanning area along the axis <Axis1ID>, format: double

<Axis2ID>: Is one axis of the controller, format: string

Axes X, Y, and Z are permissible.

During the coarse portion, the scanning lines are located on this axis.

<Distance2>: Side length of the scanning area along the axis, format: double

"L": Required keyword for entering <Threshold>

<Threshold>: Intensity threshold of the analog input signal, in V, format: Double

"S": Required keyword for entering <ScanLineDistance>

<ScanLineDistance>: Distance between the scanning lines, is only used during the coarse portion. Format: double

"SA": Required keyword for entering <StepSize>

<StepSize>: Starting value for the step size, is only used during the fine portion, format: double

"A": Required keyword for entering <AnalogInputID>

<AnalogInputID>: is the identifier of the analog input signal whose maximum intensity is sought, format: Integer

Response: None

Notes:

The physical unit in which <Distance1>, <Distance2>, <ScanLineDistance> and <StepSize> are to be given can be queried with PUN? (p. 217).

The values for <Distance1> and <Distance2> must be identical.

The following default values are used when the corresponding arguments are omitted:
<Threshold>: 1.0 V
<ScanLineDistance>: 0.01 mm
<StepSize>: 0.001 mm
<AnalogInputID>: 1
These default values have only been checked for the H-810, H-811, and H-206 Hexapod models. For other Hexapod models, the ideal parameters must be determined experimentally.

The lower the system velocity is set with VLS (p. 245) during a scanning procedure, the greater the accuracy with which the maximum intensity is found.
Velocities in the range of under 1 mm/s are recommended.

If the values for distances are too large, the platform moves on an undefined path during a scanning procedure and it can tilt. Thus collisions and an unsatisfying result of the scanning procedure are possible. Measures for avoiding tipping:

- Select values of up to 0.2 mm for <Distance1> and <Distance2>.
- Suitably align the moving platform before starting the scanning procedure.
- Use suitable holders for the inputs and/or outputs of the optical element to be aligned on the moving platform so that the motion during the scanning procedure only takes place over short distances.

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands, and FSA is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example: FSA Y 0.2 Z 0.2 L 1 S 0.01 SA 0.001 A 2

Starts a scanning procedure in the YZ plane:

- Side length of the square area: 0.2 mm
- Intensity threshold for the coarse portion: 1 V
  When 1 V is reached, the moving platform stops, the scanning area is then reset, and the fine portion is started.
- Distance between the scanning lines (in the Z axis) for the coarse portion: 10 µm
- Start value of the step size for the fine portion: 1 µm
- Identifier of the analog input channel whose maximum intensity is sought: 2

FSC (Fast Scan with Abort)

Description: Starts a scanning procedure which scans a specified area ("scanning area") until the analog input signal reaches a specified intensity threshold.

The scanning procedure started with FSC corresponds to the "coarse portion" of the scanning procedure that is started with the FSA (p. 176) command.

The plane in which the scanning area is located can be defined by one of the following axis pairs:

X Z
Y Z
X Y
U W
V W
U V

The scanning area is centered around the current position at the time the command was received ("initial position"). The size of the scanning area is determined by the values for <Distance1> and <Distance2>.

The scanning procedure starts in the corner of the scanning area where the following holds true:

- For <Axis1ID>:
  start position1 = initial position1 - <Distance1>/2
- For <Axis2ID>:
  start position2 = initial position2 - <Distance2>/2

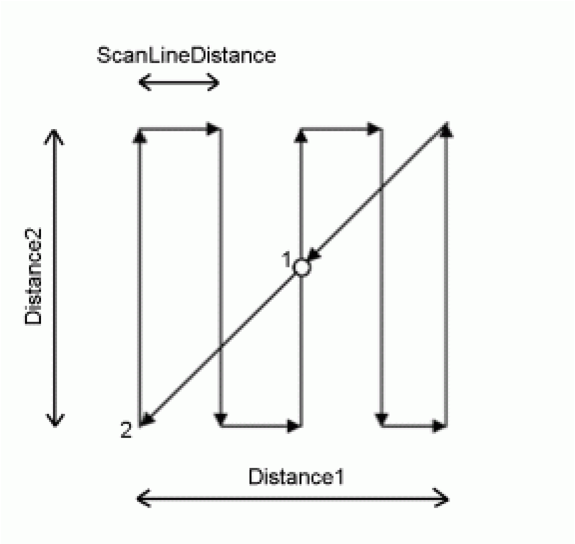
If the intensity threshold has already been reached at the initial position, the start position will not be approached.

<Axis1ID> specifies the axis in which the platform moves from one scanning line to the next, <Axis2ID> specifies the axis in which the scanning lines are located. The distance between the scanning lines can be specified.

FSC (and thus the motion of the platform) has been successfully completed when the analog input signal reaches the specified intensity threshold. If necessary due to the braking distance, the platform will return to the position at which the intensity threshold was reached after stopping.

FSC has been unsuccessfully completed in the following cases:

- The intensity threshold has not been reached in the scanning area: the moving platform returns to the initial position.
- #24 (p. 143), STP (p. 235) or HLT (p. 193) has been sent: the moving platform remains in the current position.



Tab. 1: Motion sequence when the intensity threshold has not been reached in the scanning area

1 Initial position (= end position)

2 Start position

FSS? (p. 189) can be used to check whether a scanning procedure has been successfully completed.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).

A similar scanning procedure can be started with FSM (p. 185). In contrast to FSC, FSM scans the entire scanning area so that the global maximum is found if several local maxima are present.

Format: FSC <Axis1ID> <Distance1> <Axis2ID> <Distance2> ["L" <Threshold>] ["S" <ScanLineDistance>] ["A" <AnalogInputID>]

Arguments: <Axis1ID>: Is the axis in which the platform moves from scanning line to scanning line by the distance given by <ScanlineDistance>. Format: String

Axes X, Y, Z, U, V, and W are permissible.

<Distance1>: Side length of the scanning area along the axis, format: double

<Axis2ID>: Is the axis in which the scanning lines are located, format: string

Axes X, Y, Z, U, V, and W are permissible.

<Distance2>: Side length of the scanning area along the axis, format: double

"L": Required keyword for entering <Threshold>

<Threshold>: Intensity threshold of the analog input signal, in V, format: Double

"S": Required keyword for entering <ScanLineDistance>

<ScanLineDistance>: Distance between two scanning lines. Format: double

"A": Required keyword for entering <AnalogInputID>

<AnalogInputID>: is the identifier of the analog input signal whose maximum intensity is sought, format: Integer

Response: None

Notes: The physical unit in which <Distance1>, <Distance2>, and <ScanLineDistance> are to be given can be queried with PUN? (p. 217).

The values for <Distance1> and <Distance2> must be identical.

The following default values are used when the corresponding arguments are omitted:

<Threshold>: 1.0 V

<ScanLineDistance>: 0.01

<AnalogInputID>: 1

These default values have only been checked for the H-810, H-811, and H-206 Hexapod models. For other Hexapod models, the ideal parameters must be determined experimentally.

The lower the system velocity is set with VLS (p. 245) during a scanning procedure, the greater the accuracy with which the threshold of the intensity is found.

Velocities in the range of under 1 mm/s are recommended.

If the values for distances or angles are too large, the platform moves on an undefined path during a scanning procedure and it can tip over. Thus collisions and an unsatisfying result of the scanning procedure are possible. Measures for avoiding tipping:

- Select suitable values for <Distance1> and <Distance2>. For the Hexapod models H-810, H-811 and H-206, 0.2 mm or 0.2 degrees should not be exceeded; for other Hexapod models, the ideal values must be determined experimentally.
- Suitably align the moving platform before starting the scanning procedure.
- Use suitable holders for the inputs and/or outputs of the optical element to be aligned on the moving platform so that the motion during the scanning procedure only takes place over short distances or angles.

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands, and FSC is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example: FSC Y 0.2 Z 0.2 L 1 S 0.05 A 2

Starts a scanning procedure in the YZ plane:

- Side length of the square area: 0.2 mm
- Intensity threshold: 1 V
- Distance between the scanning lines (in the Z axis): 50 µm
- Identifier of the analog input channel whose maximum intensity is sought: 2

FSM (Fast Scan to Maximum)

Description:

Starts a scanning procedure to determine the global maximum intensity of an analog input signal in a plane.

Completely scans a specified area ("scanning area") for the intensity of the analog input signal. In the case of several maximum intensities in the scanning area, this prevents only a local maximum from being found instead of the global maximum.

The plane in which the scanning area is located can be defined by one of the following axis pairs:

X Z
Y Z
X Y
U W
V W
U V

The scanning area is centered around the current position at the time the command was received ("initial position"). The size of the scanning area is determined by the values for <Distance1> and <Distance2>.

The scanning procedure starts in the corner of the scanning area where the following holds true:

- For <Axis1ID>:
  start position1 = initial position1 - <Distance1>/2
- For <Axis2ID>:
  start position2 = initial position2 - <Distance2>/2

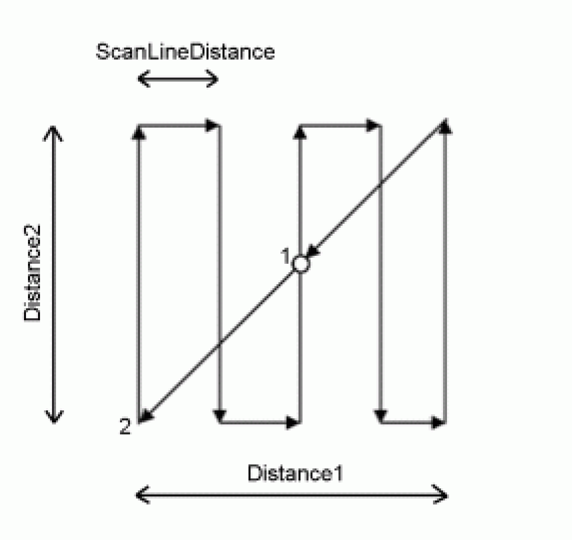
<Axis1ID> specifies the axis in which the platform moves from one scanning line to the next, <Axis2ID> specifies the axis in which the scanning lines are located. The distance between the scanning lines can be specified.

FSM (and thus the motion of the platform) has been successfully completed when the two following conditions have been met:

- The analog input signal has reached the specified intensity threshold in the scanning area at least once.
- The moving platform has returned from the end position of the scanning area to the position with maximum intensity.

FSM has been unsuccessfully completed in the following cases:

- The intensity threshold has not been reached in the scanning area: the moving platform returns to the initial position.
- #24 (p. 143), STP (p. 235) or HLT (p. 193) has been sent: the moving platform remains in the current position.



Tab. 2: Motion sequence when the intensity threshold has not been reached in the scanning area

1 Initial position (= end position)

2 Start position

FSS? (p. 189) can be used to check whether a scanning procedure has been successfully completed.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).

A similar scanning procedure can be started with FSC (p. 180). In contrast to FSM, FSC is already successfully completed when the intensity threshold is reached for the first time.

Format: FSM <Axis1ID> <Distance1> <Axis2ID> <Distance2> ["L" <Threshold>] ["S" <ScanLineDistance>] ["A" <AnalogInputID>]

Arguments: <Axis1ID>: Is the axis in which the platform moves from scanning line to scanning line by the distance given by <ScanlineDistance>. Format: String

Axes X, Y, Z, U, V, and W are permissible.

<Distance1>: Side length of the scanning area along the axis, format: double

<Axis2ID>: Is the axis in which the scanning lines are located, format: string

Axes X, Y, Z, U, V, and W are permissible.

<Distance2>: Side length of the scanning area along the axis, format: double

"L": Required keyword for entering <Threshold>

<Threshold>: Intensity threshold of the analog input signal, in V, format: Double

"S": Required keyword for entering <ScanLineDistance>

<ScanLineDistance>: Distance between two scanning lines, format: double

"A": Required keyword for entering <AnalogInputID>

<AnalogInputID>: is the identifier of the analog input signal whose maximum intensity is sought, format: Integer

Response: None

Notes:

The physical unit in which <Distance1>, <Distance2>, and <ScanLineDistance> are to be given can be queried with PUN? (p. 217).

The values for <Distance1> and <Distance2> must be identical.

The following default values are used when the corresponding arguments are omitted:
<Threshold>: 1.0 V
<ScanLineDistance>: 0.01
<AnalogInputID>: 1

These default values have only been checked for the H-810, H-811, and H-206 Hexapod models. For other Hexapod models, the ideal parameters must be determined experimentally.

In the case of intensity distributions with secondary maxima, FSM with a subsequent AAP (p. 144) is to be preferred over FSC (p. 180) or FSA (p. 176).

The lower the system velocity is set with VLS (p. 245) during a scanning procedure, the greater the accuracy with which the maximum intensity is found.
Velocities in the range of under 1 mm/s are recommended.

If the values for distances or angles are too large, the platform moves on an undefined path during a scanning procedure and it can tip over. Thus collisions and an unsatisfying result of the scanning procedure are possible. Measures for avoiding tipping:

- Select suitable values for <Distance1> and <Distance2>. For the Hexapod models H-810, H-811 and H-206, 0.2 mm or 0.2 degrees should not be exceeded; for other Hexapod models, the ideal values must be determined experimentally.
- Suitably align the moving platform before starting the scanning procedure.
- Use suitable holders for the inputs and/or outputs of the optical element to be aligned on the moving platform so that the motion during the scanning procedure only takes place over short distances or angles.

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands, and FSM is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example: FSM Y 0.2 Z 0.2 L 1 S 0.05 A 2

Starts a scanning procedure in the YZ plane:

- Side length of the square area: 0.2 mm
- Intensity threshold: 1 V
- Distance between the scanning lines (in the Z axis): 50 µm
- Identifier of the analog input channel whose maximum intensity is sought: 2

FSS? (Get Status of Fast Scan Routines)

Description: Gets the status of the last scanning procedure that was started.

In order to check whether a scanning procedure is still going on, the motion status of the axes can be queried with #5 (p. 140).

Format: FSS?

Arguments: None

Response: <uint> indicates the status of the last scanning procedure that was started.

1: Scanning procedure has been successfully completed

0: Scanning procedure is still going on or has been unsuccessfully completed.

For details on successful and unsuccessful completion, see the descriptions of the scanning procedures.

Note: FSS? gets the status of scanning procedures that are started with the following commands:
AAP (p. 144), FIO (p. 164), FLM (p. 168), FLS (p. 171), FSA (p. 176), FSC (p. 180), FSM (p. 185)

Example: Send: AAP Y 0.1 Z 0.1 SA 0.001 N 3 A 2

Send: FSS?

Receive: 0

Note: The scanning procedure is still running or has not been successfully completed.

Send: FSS?

Receive: 1

Note: The scanning procedure started with AAP has been successfully completed, i.e., a determined number of checks (number ≠ zero) has confirmed that the local maximum intensity is present at the current position.

HDR? (Get All Data Recorder Options)

Description: List a help string which contains all information available about data recording (record options and trigger options, information about additional parameters and commands concerned with data recording).

Format: HDR?

Arguments: None

Response #RecordOptions

{<RecOption>"="<DescriptionString>[of <Channel>]}

#TriggerOptions

[{<TriggerOption>"="<DescriptionString>}]

#Parameters to be set with SPA

[{<ParameterID>"="<DescriptionString>}]

#Additional information

[{<Command description>"("<Command>")"}]

#Sources for Record Options

[{<RecOption>"="<Source>}]

end of help

Note: The HDR response lists all the possible record options for the C-887, regardless of whether the corresponding accessories are installed in the controller.

Example:

```
hdr?
#RecordOptions
0=Nothing is recorded
1=Commanded position of axis
2=Real position of axis
3=Position error of axis
8=Measurement time
17=Input value of channel, calculated, in
volt
18=Input value of channel, directly from
channel, without dimension
70=Commanded velocity of axis
71=Commanded acceleration of axis
72=Real velocity of axis
73=Motor output of axis
76=Current integrated position error of axis
80=Status register of axis
#TriggerOptions
0=No trigger. Exception: STE and IMP always
trigger data recording
1=Trigger with next command that changes the
position, default setting
2=Trigger with next command, resets trigger
settings to 0
4=Trigger immediately, resets trigger
settings to 0
6=Trigger with next command that changes the
position, resets trigger settings to 0
#Parameters to be set with SPA
0x16000000=Data Recorder Table Rate
0x16000201=Data Recorder Points Per Table
#Additional information
```

```
Set data recorder configuration with DRC
Get data recorder configuration with DRC?
Get number of recorded points with DRL?
Get recorded data values with DRR?
Set data recorder trigger source with DRT
Get data recorder trigger source with DRT?
Set data recorder table rate (in cycles)
with RTR
Get data recorder table rate with RTR?
Tell number of data recorder tables with
TNR?
#Sources for Record Options
0 = X Y Z U V W A B K L M 1 2 3 4 5 6
1 = X Y Z U V W A B K L M 1 2 3 4 5 6
2 = X Y Z U V W A B K L M 1 2 3 4 5 6
3 = A B 1 2 3 4 5 6
8 = 1 2 3 4 5 6
17 = 1 2 3 4 5 6
18 = 1 2 3 4 5 6
70 = A B 1 2 3 4 5 6
71 = A B 1 2 3 4 5 6
72 = A B 1 2 3 4 5 6
73 = A B 1 2 3 4 5 6
76 = A B 1 2 3 4 5 6
80 = A B 1 2 3 4 5 6
end of help
```

HLP? (Get List Of Available Commands)

Description: List a help string which contains all commands available.

Format: HLP?

Arguments: none

Response: List of commands available

Troubleshooting: Communication breakdown

Note: The HLP? response contains the commands provided by the current command level. See CCL (p. 149) for more information.

HLT (Halt Motion Smoothly)

Description: Halts the motion of given axes smoothly. For details see the notes below.

Error code 10 is set.

#24 (p. 143) and STP (p. 235) in contrast abort current motion as fast as possible for the controller without taking care of maximum velocity and acceleration.

Format: HLT [{<AxisID>}]

Arguments: <AxisID>: is one axis of the controller, if omitted all axes are halted

Response: none

Troubleshooting: Illegal axis identifier

Notes: HLT stops all axis motions that are caused by motion commands (MOV (p. 206), MVR (p. 209), FRF (p. 174), AAP (p. 144), FIO (p. 164), FLM (p. 168), FLS (p. 171), FSA (p. 176), FSC (p. 180), FSM (p. 185), IMP (p. 198), STE (p. 234)).

After the axes are stopped, their target positions are set to their current positions.

HLT does not stop macros.

HPA? (Get List Of Available Parameters)

Description: Responds with a help string which contains all available parameters with short descriptions. For more information, see "Controller Parameters" (p. 273).

Format: HPA?

Arguments: None

Response {<PamID>"="<string> LF}

where

<PamID> is the ID of one parameter, hexadecimal format

<string> is a string which describes the corresponding parameter.

Notes: The string has following format:

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

where

<CmdLevel> is the command level which allows write access to the parameter value

<MaxItem> is the maximum number of items of the same type that the parameter affects. (The meaning of "item" depends on the parameter; this can refer to an axis, a sensor channel or the entire system).

<DataType> is the data type of the parameter value, can be INT, FLOAT or CHAR

<FunctionGroupDescription> is the name of the function group to which the parameter belongs. (Parameters are grouped according to their purpose in order to clarify their interrelation.)

<ParameterDescription> is the parameter name

<PossibleValue> is one value from the allowed data range

<ValueDescription> is the meaning of the corresponding value

The parameter listing varies depending on the optional accessories that are installed.

The listed parameters can be changed with SPA (p. 224). SPA affects the parameter settings in volatile memory (RAM).

IFS (Set Interface Parameters As Default Values)

Description: Stores interface parameters.

The default parameters for the interface are changed in the non-volatile memory, but the currently active parameters are not. Settings made with IFS become active with the next switching-on or reboot.

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

Arguments: <Pswd> is the password for writing to the non-volatile memory, default is "100"

<InterfacePam> is the interface parameter to be changed, see below

<PamValue> gives the value of the interface parameter, see below

The following interface parameters can be set:

RSBAUD
<PamValue> indicates the baud rate to be used for RS-232 communication. Possible values are:
9600, 19200, 38400, 57600 and 115200. Default is 115200

IPADR
The first four parts of <PamValue> specify the default IP address for TCP/IP communication, the last part specifies the default port to be used, default is 192.168.1.28:50000;
Note: While the IP address can be changed, the port must always be 50000!

IPSTART
<PamValue> defines the startup behavior for configuration of the IP address for TCP/IP communication,
0 = IP address is defined with IPADR
1 = DHCP or Auto-IP is used to obtain IP address. (default);

IPMASK
<PamValue> indicates the subnet mask to be used for TCP/IP communication, in the form uint.uint.uint.uint, default is 255.255.255.0;

Response: None

Notes: **Note that the number of write cycles of non-volatile memory is limited. Write default settings only when necessary.**

When IPSTART = 1, the C-887 uses the Auto-IP functionality for TCP/IP communication in a network where no DHCP server is present or directly with the PC. Using Auto-IP the network participants automatically configure their interfaces so that no manual setting of the IP addresses is necessary.

Further interface parameters of the C-887 are write-protected: The presettings of these parameters can be queried with IFS? (p. 197).

For more information, see "Establishing Communication via TCP/IP Interfaces" (p. 79)

IFS? (Get Interface Parameters As Default Values)

Description: Get the interface configuration parameter values stored in non-volatile memory (i.e. the current power-on default)

Format: IFS? [{<InterfacePam>}]

Arguments: <InterfacePam> is the interface parameter to be queried. See below for possible values.

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

where

<PamValue> is the value of the interface parameter in non-volatile memory.

<InterfacePam> can be RSPORT, RSBAUD, RSHSHK, IPADR, IPSTART, IPMASK, MACADR und TERMSTR

The following interface parameters are write-protected:

For <InterfacePam> = RSPORT, <PamValue> returns the port used for RS-232 communication:
1

For <InterfacePam> = RSHSHK, <PamValue> returns the handshake setting for RS-232 communication:
1 = RTS/CTS

For <InterfacePam> = MACADR, <PamValue> returns the unique address of the network hardware in the C-887.

For <InterfacePam> = TERMSTR, <PamValue> returns the termination character for the commands of the GCS:
0 = LineFeed (ASCII character 10)

For all other forms of <InterfacePam>, see IFS (p. 195).

IMP (Start Impulse And Response Measurement)

Description: Starts performing an impulse and recording the impulse response for the given axis.

The data recorder configuration, i.e., the assignment of data sources and record options to the recorder tables, can be set with DRC (p. 155).

The recorded data can be read with the DRR? command (p. 159).

Format: IMP <AxisID> <Amplitude>

Arguments: <AxisID> is one axis of the controller

<Amplitude> is the height of the impulse. See below for details.

Response: None

Troubleshooting: The target position resulting from the specified impulse height is out of limits.

Notes: An "impulse" consists of a relative motion with the given amplitude, followed by an equally large motion in the opposite direction. The impulse is executed in relation to the current position.

The pulse width of the impulse results from the value of the ***Pulse Length Factor*** parameter (ID 0x0E000900) multiplied by the axis-dependent cycle time (p. 306).

The physical unit in which <Amplitude> is to be given can be queried with PUN? (p. 217).

For the axes of the Hexapod (X, Y, Z, U, V, W), the following is valid:

- Before the start of every motion, it is checked whether the moving platform can actually reach the commanded target position. With VMO? (p. 246) you can query whether the target position can be reached.
- Rotations take place around the pivot point, whose coordinates can be set with SPI (p. 227).
- When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands. IMP is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

JRC (Jump Relatively Depending on Condition

Description: Jumps relatively depending on a given condition of the following type: one given value is compared with a queried value according to a given rule.

Can only be used in macros.

Format: JRC <Jump> <CMD?> <OP> <Value>

Arguments:	<Jump> is the size of the relative jump. -1 means the macro execution pointer jumps back to the previous line, 0 means the command is executed again, which is the same behavior as with WAC (p. 248). 1 jumps to the next line, making the command unnecessary, and 2 jumps over the next command. Only jumps within the current macro are allowed. <CMD?> is one query command in its usual notation. The response has to be a single value and not more. For an example see below. <OP> is the operator to be used. The following operators are possible: = <= < > >= != Important: There must be a blank space before and after the operator! <Value> is the value to be compared with the response of <CMD?>
Response:	None
Troubleshooting:	Check proper jump target

LIM? (Indicate Limit Switches)

Description:	Indicates whether axes have limit switches.
Format:	LIM? [{<AxisID>}]
Arguments:	<AxisID>: is one axis of the controller
Response:	{<AxisID>"="<uint> LF} where <uint> indicates whether the axis has limit switches (=1) or not (=0).

Troubleshooting: Illegal axis identifier

Note: The struts of the Hexapod and axes A and B have limit switches. These are not connected to the controller, however. For this reason, LIM? always responds with 0.

MAC (Call Macro Function)

Description: Calls a macro function. Permits recording, deleting and running macros on the controller.

Format: MAC <keyword> {<parameter>}

in particular:

MAC BEG <macroname>

MAC DEF <macroname>

MAC DEF?

MAC DEL <macroname>

MAC END

MAC ERR?

MAC FREE?

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

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

Arguments

<keyword> determines which macro function is called. The following keywords and parameters are used:

MAC BEG <macroname>

Starts recording a macro to be named *macroname* on the controller; may not be used in a macro; the commands that follow become the macro. End the recording with MAC END. Note that erroneous macro content cannot be detected by sending the ERR? command.

MAC DEF <macroname>

Sets specified macro as start-up macro. This macro will be automatically executed with the next power-on or reboot of the controller. If <macroname> is omitted, the current start-up macro selection is canceled.

MAC DEF?

Asks for the start-up macro
Response: <macroname>
If no start-up macro is defined, the response is an empty string with the terminating character.

MAC DEL <macroname>

Deletes specified macro.

MAC END

Stops macro recording (cannot become part of a macro)

MAC ERR?

Reports the last error which occurred during macro execution.

Response: <macroname> <uint1>”=”<uint2> <”<”CMD”>”>

where <macroname> is the name of the macro, <uint1> is the line in the macro, <uint2> is the error code and <”<”CMD”> is the erroneous command which was sent to the parser.

MAC FREE?

Gets the free memory space for macro recording

Response: <uint> is the number of characters in bytes for which free memory is still available

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

Repeats the specified macro <uint> times. Another execution is started when the last one is finished.

<String> stands for the value of a local variable contained in the macro. 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 with spaces. A maximum of 256 characters are permitted per command line. <String> can be given directly or via the value of another variable. See “Variables” for further details.

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

Starts one execution of specified macro. <String> has the same function as with MAC NSTART.

Response: None

Troubleshooting: Macro recording is active (keywords BEG, DEL) or inactive (END)

Macro contains a disallowed MAC command

Notes: During macro recording no macro execution is allowed.

When macros are recorded on the ***Controller macros*** tab in PIMikroMove, the `MAC BEG` and `MAC END` commands must be omitted.

Macros can contain local and global variables. The names of local variables used in a macro must form a continuous series. Example of permissible designation: 1, 2, 3, 4. Designation with e.g. 1, 2, 5, 6 is not allowed. Further information can be found in the section "Variables".

A running macro sends no responses to any interface.

Depending on the value of parameter 0x72 (Ignore Macro Error), there are the following possibilities when an error is caused by the running macro:

0 = macro execution is aborted
1 = the error is ignored and macro execution will be continued

Irrespective of the parameter setting, MAC ERR? always reports the last error that occurred during a macro execution.

The following commands provided by the C-887 can only be used in macros:
ADD (p. 148), CPY (p. 151), DEL (p. 154), JRC (p. 199), MEX (p. 206) and WAC (p. 248).

A macro can start another macro. The maximum number of nesting levels is 10. A macro can call itself to form an infinite loop.

Any commands can be sent from the command line when a macro is running. The macro content and move commands received from the command line may overwrite each other, and only the last move command will be executed, irrespective of its source.

Macro execution can be stopped with #24 (p. 143) and STP (p. 235).

If the GCS syntax version is changed to 1.0 with CSV (p. 153) during a macro execution, the macro execution will be aborted.

Simultaneous execution of multiple macros is not possible. Only one macro can be executed at a time.

A running macro may not be deleted.

You can query with #8 (p. 142) if a macro is currently running on the controller.

Warning: The number of write cycles of non-volatile memory is limited.

MAC? (List Macros)

Description: Lists macros or content of a given macro.

Format: MAC? [<macroname>]

Arguments <macroname>: name of the macro whose content shall be listed; if omitted, the names of all stored macros are listed.

Response: <string>

if <macroname> was given, <string> is the content of this macro;

if <macroname> was omitted, <string> is a list with the names of all stored macros

Troubleshooting: Macro <macroname> not found

MEX (Stop Macro Execution Due To Condition)

Description: Stops macro execution due to a given condition of the following type: a specified value is compared with a queried value according to a specified rule.

Can only be used in macros.

When the macro interpreter accesses this command the condition is checked. If it is true the current macro is stopped, otherwise macro execution is continued with the next line. Should the condition be fulfilled later, the interpreter will ignore it.

See also the WAC (p. 248) command.

Format: MEX <CMD?> <OP> <value>

Arguments <CMD?> is one query command in its usual notation. The response has to be a single value and not more. For an example see below.

<OP> is the operator to be used. The following operators are possible:
= <= < > >= !=
Important: There must be a blank space before and after the operator!

<value> is the value to be compared with the response to <CMD?>.

Response: None

MOV (Set Target Position)

Description: Set new absolute target position for given axis.

Servo mode must be switched on for the commanded axis prior to using this command (closed-loop operation).

Format: MOV {<AxisID> <Position>}

Arguments <AxisID> is one axis of the controller

<Position> is the new absolute target position in physical units.

Response: none

Troubleshooting:

Target position outside of the current working space.

Illegal axis identifier

See section "Commandable Items" (p. 28) for more information.

Servo mode is OFF for one of the axes specified.

The reference move has not been successfully completed for at least one axis.

Notes: The physical unit in which <Position> is to be given can be queried with PUN? (p. 217).

In order to determine whether a motion has been completed, it is recommended to send #5 (p. 140).

The motion can be aborted by #24 (p. 143), STP (p. 235) and HLT (p. 193).

For the axes of the Hexapod (X, Y, Z, U, V, W), the following is valid:

Before the start of every motion, it is checked whether the moving platform can actually reach the commanded target position. With VMO? (p. 246) you can query whether the target position can be reached.

Rotations take place around the pivot point, whose coordinates can be set with SPI (p. 227).

Depending on the setting of the ***Trajectory Source*** parameter (ID 0x19001900), the motion profile for the axes of the Hexapod (X, Y, Z, U, V, W) is defined by one of the following two sources:

Trajectory generator of the C-887

Consecutive MOV commands

For more information, see "Motions of the Hexapod" (p. 32).

Example 1: Send: MOV X 10 U 5

Note: Axis X moves to 10 (target position in mm), axis U moves to 5 (target position in °)

Example 2: Send: MOV X 4 Y 2.3 Z -3 U -5.3 V 3 W 1

Note: Target positions can be given for all six axes of the Hexapod with a single motion command.

Example 3: Send: MOV Z 100

Send: ERR?

Receive: 7

Note: The axis does not move. The error code "7" in the response to the ERR? (p. 163) command indicates that the target position given in the motion command is out of limits.

MOV? (Get Target Position)

Description: Returns last valid commanded target position.

Format: MOV? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

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

where

<float> is the last commanded target position in physical units

Troubleshooting: Illegal axis identifier

Notes: The target position can be changed with various sources, e.g. with commands that cause motion (MOV (p. 206), MVR (p. 209), IMP (p. 198), STE (p. 234)), or with a control unit (C-887.MC).

MOV? gets the commanded positions. Use POS? (p. 216) to get the current positions.

MVR (Set Target Relative To Current Position)

Description: Moves given axes relative to the last commanded target position.

Servo mode must be switched on for the commanded axis prior to using this command (closed-loop operation).

Format: MVR {<AxisID> <Distance>}

Arguments: <AxisID> is one axis of the controller.

<Distance> gives the distance to move; the sum of the distance and the last commanded target position is set as new target position (in physical units).

Response: none

Troubleshooting:

- Target position outside of the current working space.
- Illegal axis identifier
  Further information can be found in the section "Commandable Items" (p. 28).
- Servo mode is OFF for one of the axes specified.
- The reference move has not been successfully completed for at least one axis.

Notes: The physical unit in which <Distance> is to be given can be queried with PUN? (p. 217).

In order to determine whether a motion has been completed, it is recommended to send #5 (p. 140).

The motion can be aborted by #24 (p. 143), STP (p. 235) and HLT (p. 193).

For the axes of the Hexapod (X, Y, Z, U, V, W), the following is valid:

- Before the start of every motion, it is checked whether the moving platform can actually reach the commanded target position. With VMO? (p. 246) you can query whether the target position can be reached.
- Rotations take place around the pivot point, whose coordinates can be set with SPI (p. 227).

- When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands. MVR is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

Example:

Send: MOV X 0.5

Note: This is an absolute move.

Send: POS? X

Receive: X=0.500000

Send: MOV? X

Receive: X=0.500000

Send: MVR X 2

Note: This is a relative move.

Send: POS? X

Receive: X=2.500000

Send: MVR X 2000

Note: New target position of axis X would exceed the motion range. Command is ignored, i.e. the target position remains unchanged, and the axis does not move.

Send: MOV? X

Receive: X=2.500000

Send: POS? X

Receive: X=2.500000

NAV (Set Number of Readout Values to be Averaged)

Description: Determines the number of readout values of the analog input that are averaged.

Format: NAV {<AnalogInputID> <NumberOfReadings>}

Arguments: <AnalogInputID> is the identifier of the analog input channel.

<NumberOfReadings> is the number of readout values of the analog signal

Troubleshooting: Number of readout values too high
Illegal identifier of the analog input channel.

Notes: The default value for <NumberOfReadings> is 1; the permissible range of values is 1 to 10000.

NAV can reduce the influence of noise at the analog input on queries (TAV? (p. 238), TAD? (p. 237)) or on scanning procedures (e.g. AAP (p. 144)). The greater the noise of the analog input signal, the higher the number of readout values of the analog signal should be set with NAV for averaging.

This command requires at least one analog input channel in the controller, see "Optional Accessories" (p. 25) and "Commandable Items" (p. 28).

Example: Send: NAV 2 200
Send: TAV? 2
Note: The response to the TAV? 2 command is the average of 200 readout values of the analog input channel 2.

NAV? (Get Number of Readings to be Averaged)

Description: Gets the number of readout values of the analog input used for averaging.

Format: NAV? [{<AnalogInputID>}]

Arguments: <AnalogInputID> is the identifier of the analog input channel

Response: {<AnalogInputID>"="<int> LF}

where

<int> is the number of readout values.

The response consists of a line feed when the controller does not contain an analog input channel.

NLM (Set Low Position Soft Limit)

Description:	Limits the low end of the axis travel range in closed-loop operation ("soft limit").
Format:	NLM {<AxisID> <LowLimit>}
Arguments:	<AxisID> is one axis of the controller <LowLimit> is the limit position for the low end of the travel range, in physical units
Response:	None
Notes:	When the defaults are used, the soft limits correspond to the values that can be queried with TMN? (p. 239) and TMX? (p. 239). The value set with NLM must be smaller than the current position value. Only negative values are allowed for axes X, Y, Z, U, V and W. The soft limits are activated and deactivated with SSL (p. 229). Soft limits can only be set when the axis is not moving (query with #5 (p. 140)). The physical unit in which <LowLimit> is to be given can be queried with PUN? (p. 217). When the pivot point is changed with SPI (p. 229), the soft limits for the rotational axes U, V and W are not adjusted.

Example:

```
Send: NLM? X
Receive: X = -22.5
Send: POS? X
Receive: X = -10
Send: NLM X -5
Send: ERR?
Receive: 27  -  (error 27 - "Soft limit out of range")
Send: NLM? X
Receive: X = -22.5
```

NLM? (Get Low Position Soft Limit)

Description: Get the position "soft limit" which determines the low end of the axis travel range in closed-loop operation.

Format: NLM? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<LowLimit> LF}

where

<LowLimit> is the limit position for the low end of the travel range, in physical units.

ONT? (Get On Target State)

Description: Gets on-target state of given axis.

If all arguments are omitted, gets state of all axes.

Format: ONT? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<uint> LF}

where

<uint> = "1" when the specified axis is on target, "0" otherwise.

Troubleshooting: Illegal axis identifier

PLM (Set High Position Soft Limit)

Description: Limits the high end of the axis travel range in closed-loop operation ("soft limit").

Format: PLM {<AxisID> <HighLimit>}

Arguments: <AxisID> is one axis of the controller

<HighLimit> is the limit position for the high end of the travel range, in physical units.

Response: None

Notes: When the defaults are used, the soft limits correspond to the values that can be queried with TMN? (p. 239) and TMX? (p. 239).

The value set with PLM must be greater than the current position value. Only positive values are allowed for axes X, Y, Z, U, V and W.

The soft limits are activated and deactivated with SSL (p. 229).

Soft limits can only be set when the axis is not moving (query with #5 (p. 140)).

The physical unit in which <HighLimit> is to be given can be queried with PUN? (p. 217).

When the pivot point is changed with SPI (p. 229), the soft limits for the rotational axes U, V and W are not adjusted.

Example:

Send: PLM? X
Receive: X = 22.5
Send: POS? X
Receive: X = 10
Send: PLM X 5
Send: ERR?
Receive: 27 - (error 27 - "Soft limit out of range")
Send: PLM? X
Receive: X = 22.5

PLM? (Get High Position Soft Limit)

Description: Gets the position "soft limit" which determines the high end of the axis travel range in closed-loop operation.

Format: PLM? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<HighLimit> LF}

where

<HighLimit> is the limit position for the high end of the travel range, in physical units.

POS? (Get Real Position)

Description: Returns the current axis position.

If all arguments are omitted, gets current position of all axes.

Format: POS? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

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

where

<float> is the current axis position in physical units

Troubleshooting: Illegal axis identifier

Notes: This command is identical in function to #3 (p. 138) which should be preferred when the controller is performing time-consuming tasks.

The current position of axes X, Y, Z, U, V and W is calculated from the measured positions of the individual struts.

Between the switching-on of the controller and the reference point definition of the Hexapod with FRF (p. 174), the current position of the Hexapod and axes A and B is unknown. Nevertheless, the response to POS? gives the position value 0 for all axes.

The physical unit in which the axis position is given can be queried with PUN? (p. 217).

PUN? (Get Position Unit)

Description: Gets the current unit of the position.

If all arguments are omitted, gets current unit of the position for all axes.

Format: PUN? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<string> LF}

where

<string> is the current unit of the position

Troubleshooting: Illegal axis identifier

Note: The following units are used for the positions of the axes:

X, Y, Z, A, B, K, L, M: millimeters

U, V, W: degrees

The units cannot be changed.

RBT (Reboot System)

Description: Reboots system. Controller behaves just like after power-on.

Format: RBT

Arguments: none

Response: none

RMC? (List Running Macros)

Description: Lists macros which are currently running.

Format: RMC?

Arguments: none

Response: {<macroname> LF}

where

<macroname> is the name of one macro which is saved on the controller and currently running. The response is an empty line when no macro is running.

RON? (Get Reference Mode)

Description: Gets mode of reference point definition of given axes.

Format: RON? [{ <AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<ReferenceOn> LF}

where

<ReferenceOn> is the currently set mode of reference point definition for the axis

Troubleshooting: Illegal axis identifier

Notes: RON? always returns 1. For axes whose position is measured by incremental sensors (i.e. for X, Y, Z, U, V, W, A und B), this means:

- Motion commands are only executed after a successful reference point definition.
- The reference point definition must take place with a reference move.

With the C-887, the reference move takes place to the reference point switch (start with FRF (p. 174)).

Only for axes A and B:

When a stage is connected without a reference point switch, no reference move can be started for the axis. In order for the reference point definition to be considered successfully completed, the value of the ***Sensor Reference Mode*** parameter (ID 0x02000A00) must be set to 1 with SPA (p. 224) for the axis, see „Starting Motions of the Axes A, B, K, L and M – Only C-887.11“ (p. 103).

RTR (Set Record Table Rate)

Description: Sets the record table rate, i.e., the number of cycles to be used in data recording operations. Settings larger than 1 make it possible to cover longer time periods.

Format: RTR <RecordTableRate>

Arguments: <RecordTableRate> is the record table rate to be used for recording operations (unit: number of cycles), must be an integer value larger than zero.

Response: None

Notes: RTR sets the value of the ***Data Recorder Table Rate*** parameter (ID 0x16000000).

Values of between 1 and 1000000 are possible. The value of the parameter is set to 250 by default.

The duration of the recording can be calculated as follows:

Rec. duration = cycle time * RTR value * number of points

where

the cycle time of the data recorder is 20 µs

the number of points for the C-887 is max. 10240 (default: 1024 points)

The record table rate set with RTR is saved in volatile memory (RAM) only.

For more information see "Data Recording" (p. 106).

RTR? (Get Record Table Rate)

Description: Gets the current record table rate, i.e., the number of cycles used in data recording operations.

Format: RTR?

Arguments: None

Response: <RecordTableRate> is the table rate used for recording operations (unit: number of cycles).

Note: Gets the ***Data Recorder Table Rate*** parameter value in the volatile memory (ID 0x16000000).

For more information see "Data Recording" (p. 106).

SAI? (Get List Of Current Axis Identifiers)

Description: Gets the axis identifiers.

See also "Commandable Items" (p. 28).

Format: SAI? [ALL]

Arguments: [ALL] is optional. For controllers which allow for axis deactivation, [ALL] ensures that the answer also includes the axes which are "deactivated".

Response: {<AxisID> LF}

<AxisID> is one axis of the controller.

Note: Axes A and B can be "deactivated" by assigning them the stage type "NOSTAGE" with the CST (p. 152) command.

Deactivated axes are not shown.

Exception: the responses to the CST? (p. 152) and SAI? ALL commands include the deactivated axes.

SCT (Set Cycle Time)

Description: Determines the cycle time for performing a motion profile.

Format: SCT "T" <CycleTime>

Arguments: "T" is the required keyword for the argument <CycleTime>

<CycleTime> is the cycle time in ms, format: float.

Troubleshooting: Permissible value range exceeded

Notes: The permissible range of values for <CycleTime> is 1 to 10000 ms; the standard value is 50 ms.

The cycle time set with SCT is only valid when the ***Trajectory Source*** parameter (ID 0x19001900) has the value 1 (= the motion profile is defined by consecutive MOV commands).

The cycle time is used to calculate the velocity during the motion so that the given points of the motion profile are reached precisely at the end of the time interval.

When the ***Trajectory Execution*** parameter (ID 0x19001901) has the value 1, the intermediate storage of the motion profile points guarantees that the motion profile is performed at the specified time intervals.

When the ***Trajectory Execution*** parameter (ID 0x19001901) has the value 0, the MOV commands are executed immediately after being sent.

➢ Make sure that the MOV commands are sent in the time intervals specified in the cycle time so that the motion profile can be maintained.

Example: Send: SCT T 30

Sets the cycle time for performing a motion profile that is defined by consecutive MOV commands to 30 ms.

SCT? (Get Cycle Time)

Description: Gets the current cycle time for running a defined motion profile.

Format: SCT? [<T>]

Arguments: "T" serves as a keyword and can be omitted for the query.

Response: T=<float> LF

<float> is the cycle time in ms.

SGA (Set Gain)

Description: Determines the gain value for the given analog input channel.

Format: SGA {<AnalogInputID> <Gain>}

Arguments: <AnalogInputID> is the identifier of the analog input channel.

<Gain> is the gain factor

Troubleshooting: Illegal value

Note: SGA sets the gain value for optical input signals. After being transformed, these signals are fed to the C-887 as analog signals (p. 27).

The standard value for <Gain> is 100; permissible values are 1, 10, 100 and 1000.

This command requires at least one optical analog input channel in the controller, see "Optional Accessories" (p. 25) and "Commandable Items" (p. 28).

Example: Send: SGA 1 10

Sets the gain value of the analog input channel 1 to 10.

SGA? (Get Gain)

Description: Gets the gain value for the given analog input channel.

Format: SGA? [{<AnalogInputID>}]

Arguments: <AnalogInputID> is the identifier of the analog input channel, see SGA (p. 224)

Response: {<AnalogInputID>"="<int> LF}

where

<int>is the gain value of the analog input channel.

The response consists of a line feed when the controller does not contain an optical analog input channel.

Example: Send: SGA? 2

Receive: 2 = 10

The gain value of the analog input channel 2 is 10.

SPA (Set Volatile Memory Parameters)

Description: Sets a parameter of the given item in the volatile memory (RAM) to a specific value. Parameter changes are lost when the controller is switched off or rebooted.

Format: SPA {<ItemID> <PamID> <PamValue>}

Arguments: <ItemID> is the item for which a parameter is changed in the volatile memory. See below for details.

<PamID> is the parameter identifier, can be written in hexadecimal or decimal format. See below for details.

<PamValue> is the value to which the parameter of the given item is set.

Response: None

Note that this command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware!

With HPA? (p. 194) you can obtain a list of the available parameters.

Troubleshooting: Illegal item identifier, wrong parameter ID,value out of range, command level too low for write access

Parameter values can only be set when the axis is not moving (query with #5 (p. 140)).

Available item IDs and parameter IDs: The item can be an axis, a sensor channel or the whole system; the item type depends on the parameter.
See "Parameter Overview" (p. 273) for the item type concerned; for item identifiers, see "Commandable Items" (p. 28).

Valid parameter IDs are given in "Parameter Overview" (p. 273).

Example: Send: SPA 1 0x16000000 8

Note: Sets the record table rate for the controller to 8, parameter ID written in hexadecimal format

Send: SPA 1 369098752 2

Note: Sets the record table rate for the controller to 2, parameter ID written in decimal format

SPA? (Get Volatile Memory Parameters)

Description:	Gets the value of a parameter of a given item from volatile memory (RAM). With HPA? (p. 194) you can obtain a list of the available parameters.
Format:	SPA? [{<ItemID> <PamID>}]
Arguments:	<ItemID> is the item for which a parameter is to be queried in volatile memory. See below for details. <PamID> is the parameter identifier, can be written in hexadecimal or decimal format. See below for details.
Response:	{<ItemID> <PamID>"="<PamValue> LF} where <PamValue> is the value of the given parameter for the given item
Troubleshooting:	Illegal item identifier, wrong parameter ID
Available item IDs and parameter IDs:	The item can be an axis, a sensor channel or the whole system; the item type depends on the parameter. See "Parameter Overview" (p. 273) for the item type concerned; for item identifiers, see "Commandable Items" (p. 28). Valid parameter IDs are given in "Parameter Overview" (p. 273).

SPI (Set Pivot Point)

Description: Sets the pivot point coordinates in the volatile memory.

Can only be set when the following holds true for the rotation coordinates of the moving platform:
U = V = W = 0

Format: SPI {<PPCoordinate> <Position>}

Arguments: <PPCoordinate> is a pivot point coordinate, see below.

<Position> is the value of the pivot point coordinate, see below.

Response: None

Troubleshooting: At least one of the rotation coordinates U, V and W is not equal to 0

Notes: <PPCoordinate> can be R, S and T. X, Y and Z can also be used as alias identifiers for R, S and T.

<Position> is given in mm.

The pivot point moves along with the moving platform during translation motions; the pivot point coordinates do not change, though. The reason for this is that the physical position of the pivot point consists of the sum of the values of the pivot point coordinates set with SPI and the X, Y, Z coordinates of the moving platform.
This also holds true when SPI was set with the alias identifiers X, Y and Z instead of R, S and T.

This behavior is advantageous, e.g., for the scanning procedure initiated with FIO (p. 164): The pivot point must only once be set to the position that corresponds to the center of the input or the output of the optical element. Subsequently, it moves along during the linear alignment of the moving platform of the Hexapod.

When the controller is switched on or rebooted, the pivot point coordinates are set to the values (0,0,0). This applies to all standard Hexapods.

When the defaults are used for the pivot point coordinates, the pivot point lies as shown in the dimensional drawing of the connected Hexapod after a reference move; see corresponding user manual of the Hexapod.

If the pivot point is moved with SPI, the possible travel ranges for the rotational axes U, V and W change and thus the working space as well. The following values are **not** adapted to the changed travel ranges, however:

- Responses to TMN? (p. 239) and TMX? (p. 239)
- Soft limits that are set with NLM (p. 212) and PLM (p. 214)

Use VMO? (p. 246) to query whether a target position can be reached.

A startup macro can be used for changed values of the pivot point coordinates directly after the controller is switched on.
Further information on the use of macros can be found in "Controller Macros" (p. 110).

Example:

```
Send: SPI?
Receive: R=0
S=0
T=0
Send: SPI S 2
Send: SPI?

Receive: R=0
S=2
T=0
Send: SPI Z 2
Send: SPI?
Receive: R=0
S=2
T=2
```

SPI? (Get Pivot Point)

Description: Gets the pivot point coordinates.

Format: SPI? [{<PPCoordinate>}]

Arguments: <PPCoordinate> is a pivot point coordinate, see below

Response: {<PPCoordinate>"="<Position> LF}

where

<Position> is the value of the pivot point coordinate in physical units.

Note: <PPCoordinate> can be R, S and T. X, Y and Z can also be used as alias identifiers for R, S and T.

SSL (Set Soft Limit)

Description: Activates or deactivates the soft limits that are set with NLM (p. 212) and PLM (p. 214).

Format: SSL {<AxisID> <SoftLimitsOn>}

Arguments: <AxisID> is one axis of the controller

<SoftLimitsOn> is the status of the soft limits:

0 = soft limits deactivated

1 = soft limits activated

Response: None

Note: Soft limits can only be activated/deactivated when the axis is not moving (query with #5 (p. 140)).

Example: Send: SSL X 1

The soft limits for axis X are activated.

Send: SSL Y 0 Z 1 W 1

The soft limits are deactivated for the axis Y and activated for axes Z and W.

SSL? (Get Soft Limit Status)

Description: Gets the status of the soft limits that are set with NLM (p. 212) and PLM (p. 214).

If all arguments are omitted, the status is queried for all axes.

Format: SSL? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<SoftLimitsOn> LF}

where

<SoftLimitsOn> is the status of the soft limits:

0 = soft limits deactivated

1 = soft limits activated

Troubleshooting: Illegal axis identifier

SSN? (Get Device Serial Number)

Description: Gets the serial number of the C-887.

Format: SSN?

Arguments: None

Response: <SerialNumber> is the serial number of the device.

SST (Set Step Size)

Description:	Sets the step size for motions of the given axis that are triggered by a manual control unit.
Format:	SST {<AxisID> <StepSize>}
Arguments:	<AxisID> is one axis of the controller <StepSize> is the step size, format: float
Response:	None
Troubleshooting:	Illegal value Illegal axis identifier
Note:	The physical unit in which <StepSize> is to be given can be queried with PUN? (p. 217). The permissible range of values for <StepSize> is 0.0001 to 0.5; the default value is 0.01 The step size set with SST is used for motions of the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W) which are triggered by the optionally available C-887.MC control unit (p. 25) or in the main window of the user interface of the C-887 (p. 74).

Example:	Send: SST Y 0.002 U 0.05
	Send: SST?
	Receive: X=0.01
	Y=0.002
	Z=0.01
	U=0.05
	V=0.01
	W=0.01

```
Send: SST X 0.09 W 0.09
Send: SST?
Receive: X=0.09
Y=0.002
Z=0.01
U=0.05
V=0.01
W=0.09
```

SST? (Get Step Size)

Description: Gets the step size for motions of the given axis that are triggered by a manual control unit.

Format: SST? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<StepSize> LF}

where

<StepSize> is the step size in physical units, see SST (p. 231).

STA? (Query Status Register Value)

Description: Requests system status information.

Format: STA?

Arguments: None

Response: The answer is bit-mapped. See below for the individual codes.

Notes: This command is identical in function to #4 (p. 139).

For axes K, L and M, there is no status query.

The response is the sum of the following codes in hexadecimal format. When analyzing the response, the following must be taken into account:

- Bits 14 and 15 for the motion status of axes A and B are only set when the motion has been triggered by a command.
- Unassigned bits have the value 0.

Bit:	23	22	21	20	19	18	17	16
	-	-	-	-	Refe rence move is being carried out	Refe-rence-move axis B success-ful	Refe-rence-move axis A success-ful	Refe-rence-move Hexapod success-ful
Bit:	15	14	13	12	11	10	9	8
	Axis B in motion	Axis A in motion	Strut 6 in motion	Strut 5 in motion	Strut 4 in motion	Strut 3 in motion	Strut 2 in motion	Strut 1in motion
Bit:	7	6	5	4	3	2	1	0
	Motion error axis B	Motion error axis A	Motion error strut 6	Motion error strut 5	Motion error strut 4	Motion error strut 3	Motion error strut 2	Motion error strut 1

Example: Send: STA?

Receive: 0x71804

Note: The response is given in hexadecimal format. This means: A motion error was reported for strut 3; strut 4 and 5 are in motion. The reference move of the Hexapod and axes A and B has been successfully completed.

STE (Start Step And Response Measurement)

Description: Starts performing a step and recording the step response for the given axis.

The data recorder configuration, i.e. the assignment of data sources and record options to the recorder tables, can be set with DRC (p. 155).

The recorded data can be read with the DRR? (p. 159) command.

Format: STE <AxisID> <Amplitude>

Arguments <AxisID> is one axis of the controller

<Amplitude> is the height of the step. See below for details.

Response: none

Troubleshooting: The target position resulting from the specified step height is out of limits.

Notes: A "step" consists of a relative move of the specified amplitude. The step is executed in relation to the current position.

The physical unit in which <Amplitude> is to be given can be queried with PUN? (p. 217).

For the axes of the Hexapod (X, Y, Z, U, V, W), the following is valid:

Before the start of every motion, it is checked whether the moving platform can actually reach the commanded target position. With VMO? (p. 246) you can query whether the target position can be reached.

Rotations take place around the pivot point, whose coordinates can be set with SPI (p. 227).

When the ***Trajectory Source*** parameter (ID 0x19001900) is set to 1, the motion profile must be defined by consecutive MOV commands. STE is not allowed.

For more information, see "Motions of the Hexapod" (p. 32).

STP (Stop All Axes)

Description: Stops all axes abruptly. For details see the notes below.

Sets error code to 10.

This command is identical in function to #24 (p. 143).

Format: STP

Arguments: None

Response: None

Troubleshooting: Communication breakdown

Notes: STP stops all axis motions that are caused by motion commands (MOV (p. 206), MVR (p. 209), FRF (p. 174), AAP (p. 144), FIO (p. 164), FLM (p. 168), FLS (p. 171), FSA (p. 176), FSC (p. 180), FSM (p. 185), IMP (p. 198), STE (p. 234)).

STP stops macros.

After the axes are stopped, their target positions are set to their current positions.

SVO (Set Servo Mode)

Description: Sets the servo mode for given axes (open-loop or closed-loop operation).

Format: SVO {<AxisID> <ServoState>}

Arguments: <AxisID> is one axis of the controller

<ServoState> can have the following values:
0 = servo mode off (open-loop operation)
1 = servo mode on (closed-loop operation)

Response: None

Troubleshooting: Illegal axis identifier

Notes: For the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W), the servo mode is always switched on or off together. For this reason, it is sufficient to enter a single axis, e.g. SVO X 1, to set the servo mode for the axes of the moving platform.

Servo mode is automatically switched on in the following cases:

- Switching-on or reboot of the C-887 – servo mode is switched on for all axes.
- For the axes of the moving platform of the Hexapod (X, Y, Z, U, V, W): start of a reference move with FRF (p. 174)
- For axes A and B: assignment of a stage type with CST (p. 152)

Motions of the axes can only be triggered when servo mode is switched on.

Servo mode can only be switched off when the axis is not moving (query with #5 (p. 140)).

SVO? (Get Servo Mode)

Description: Gets the servo mode for the axes specified.

If all arguments are omitted, gets the servo mode of all axes.

Format: SVO? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

Response: {<AxisID>"="<ServoState> LF}

where

<ServoState> is the current servo mode for the axis:
0 = servo mode off (open-loop operation)
1 = servo mode on (closed-loop operation)

Troubleshooting: Illegal axis identifier

TAC? (Tell Analog Channels)

Description: Get the number of installed analog lines.

Format: TAC?

Arguments: None

Response: <uint> indicates the total number of analog lines (inputs and outputs).

Note: With the C-887, all analog lines included in the response to TAC? are analog input channels. These input channels are only present when the controller is equipped with corresponding optional accessories. See "Optional Accessories" (p. 25) for available options.

TAD? (Get ADC Value Of Input Signal)

Description: Get the current value from the specified input signal channel's A/D converter. Using this command it is possible to check for sensor overflow.

Format: TAD? [{<InputSignalID>}]

Arguments: <InputSignalID> is one input signal channel of the controller

Response: {<InputSignalID>"="<uint> LF}

where

<uint> is the current A/D value, dimensionless

Notes: The TAD? response represents the digitized signal value without filtering and linearization.

The greater the noise of the analog input signal, the higher the number of readout values of the analog signal should be set with NAV (p. 210) for averaging.

The response consists of a line feed when the controller does not contain an analog input channel.

See "Optional Accessories" (p. 25) for available options.

TAV? (Get Analog Input Voltage)

Description: Get voltage at analog input.

Format: TAV? [{<AnalogInputID>}]

Arguments: <AnalogInputID> is the identifier of the analog input channel; see below for details.

Response: {<AnalogInputID>"="<float> LF}

where

<float> is the current voltage at the analog input in volts

Note: The greater the noise of the analog input signal, the higher the number of readout values of the analog signal should be set with NAV (p. 210) for averaging.

The response consists of a line feed when the controller does not contain an analog input channel.

See "Optional Accessories" (p. 25) for available options.

TMN? (Get Minimum Commandable Position)

Description: Get the minimum commandable position in physical units.

Format: TMN? [{ <AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response {<AxisID>"="<float> LF}

where

<float> is the minimum commandable position in physical units

Notes: The travel ranges in X, Y, Z, U, V, W are dependent on each other. Depending on the current position of the moving platform of the Hexapod, the actually available travel range for axes X, Y, Z, U, V and W can turn out to be smaller than given in the response to TMN?. The response to TMN? only corresponds to the actually available travel range of an axis when the following two conditions are met:

All other axes are at zero position.

The defaults for the pivot point coordinates apply.

If the pivot point is moved with SPI (p. 227), the available travel ranges for the rotational axes U, V and W change. The values for the minimum commandable positions are **not** adapted to the changed travel ranges, however. Use VMO? (p. 246) to query whether a target position can be reached.

The physical unit in which the minimum commandable position is given can be queried with PUN? (p. 217).

TMX? (Get Maximum Commandable Position)

Description: Get the maximum commandable position in physical units.

Format: TMX? [{ <AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response {<AxisID>"="<float> LF}

where

<float> is the maximum commandable position in physical units

Notes: The travel ranges in X, Y, Z, U, V, W are dependent on each other. Depending on the current position of the moving platform of the Hexapod, the actually available travel range for the axes X, Y, Z, U, V and W can turn out to be smaller than given in the response to TMX?. The response to TMX? only corresponds to the actually available travel range of an axis when the following two conditions are met:

All other axes are at zero position.

The defaults for the pivot point coordinates apply.

If the pivot point is moved with SPI (p. 227), the available travel ranges for the rotational axes U, V and W change. The values for the maximum commandable positions are **not** adapted to the changed travel ranges, however. Use VMO? (p. 246) to query whether a target position can be reached.

The physical unit in which the maximum commandable position is given can be queried with PUN? (p. 217).

TNR? (Get Number of Record Tables)

Description: Gets the number of data recorder tables currently available on the controller.

Format: TNR?

Arguments: none

Response <uint> is the number of data recorder tables which are currently available

Notes: The response indicates the value of the ***Data Recorder Channel Number*** parameter (ID 0x16000300).

For more information see "Data Recording" (p. 106).

TRS? (Indicate Reference Switch)

Description: Indicates whether axes have a reference point switch with direction sensing.

Format: TRS? [{<AxisID>}]

Arguments: <AxisID>: is one axis of the controller

Response: {<AxisID>"="<uint> LF}

where

<uint> indicates whether the axis has a direction-sensing reference point switch (=1) or not (=0).

Troubleshooting: Illegal axis identifier

VAR (Set Variable Value)

Description: Sets a variable to a certain value.

Local variables can be set using VAR in macros only. See “Variables” for details regarding local and global variables.

The variable is present in RAM only.

Format: VAR <Variable> <String>

Arguments: <Variable> is the name of the variable whose value is to be set.

<String> is the value to which the variable is to be set. If omitted, the variable is deleted.

The value can be given directly or via the value of a variable.

See “Variables” for conventions regarding variable names and values.

Response: None

Example: It is possible to set the value of one variable (e.g. TARGET) to that of another variable (e.g. SOURCE):

```
VAR TARGET ${SOURCE}
```

Use curly brackets if the name of the variable is longer than one character:

```
VAR A ONE
VAR VARB TWO
VAR $A 1
VAR ${VARB} 2
VAR $VARB 2 // this will result in an unwanted behavior
VAR?
A=ONE
VARB=TWO
ONE=1
TWO=2 // ${VARB}: is replaced by its value “TWO”
ARB=2 // $VARB: $V is replaced by its (empty) value
```

See ADD (p. 148) for another example.

VAR? (Get Variable Values)

Description: Gets variable values.

If VAR? is combined with CPY (p. 151), JRC (p. 199), MEX (p. 206) or WAC (p. 248), the response to VAR? has to be a single value and not more.

More information regarding local and global variables can be found in “Variables”.

Format: VAR? [{<Variable>}]

Arguments: <Variable> is the name of the variable to be queried. More information on name conventions can be found in "Variables".

If <Variable> is omitted, all global variables present in the RAM are listed.

Response: {<Variable>”=”<String>LF}

where

<String> gives the value to which the variable is set.

Note: Within a macro, VAR? can only be meaningfully used in combination with CPY (p. 151), JRC (p. 199), MEX (p. 206) or WAC (p. 248).

VEL (Set Closed-Loop Velocity)

Description: Set velocity of given axes.

The velocity can be changed with VEL while the axis is moving.

Format: VEL {<AxisID> <Velocity>}

Arguments: <AxisID> is one axis of the controller.

<Velocity> is the velocity value in physical units/s.

Response: None

Troubleshooting: Illegal axis identifiers

Notes: The command is only permissible for axes A and B.

<Velocity> must be ≥ 0. The upper limit depends on the type of stage that is assigned to the axes A and B (see CST (p. 152)).

The velocity set with VEL is saved in volatile memory (RAM) only.

VEL? (Get Closed-Loop Velocity)

Description: Gets the velocity value commanded with VEL (p. 243).

If all arguments are omitted, the value of all axes commanded with VEL is queried.

Format: VEL? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller.

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

where

<float> is the velocity value commanded with VEL in physical units/s.

VER? (Get Versions Of Firmware And Drivers)

Description: Gets the versions of the firmware of the C-887 as well as of further components like, for example, drivers and libraries.

Format: VER?

Arguments: None

Response {<string1>":" <string2> [<string3>]LF}

where

<string1> is the name of the component;
<string2> is the version information of the component <string1>;
<string3> is an optional note.

Example: For C-887, VER? responds something like:

```
VER?
PI PI_GCS2_DLL.dll: V2.9.100.49
FW: V 1.1.0.1
GUI: V 1.0.1.646
Macro: V 0.8.0.78
OS: V #5 SMP PREEMPT RT Tue Jul 5 09:02:42
CEST 2011
Hexdata: V 1.0.0.0
PIStages: V 21/6/2011 14:15:26 V200.3
C842.80: V 2.3 2 4 1 536
Collision: V 2.15
```

VLS (Set System Velocity)

Description: Sets the velocity for the moving platform of the Hexapod.

Format: VLS <SystemVelocity>

Arguments: <SystemVelocity> is the velocity value in physical units.

Response: None

Notes: The unit for <SystemVelocity> is mm/s.

The lower limit for <SystemVelocity> is due to the smallest step size of the Hexapod (model-dependent) and is specified by the value of the ***Minimum System Velocity*** parameter (ID 0x19001501). The parameter can be queried with SPA? (p. 226).

The velocity can only be set with VLS when the Hexapod is not moving (axes X, Y, Z, U, V, W; query with #5 (p. 140)).

For axes A and B, the velocity can be set with VEL (p. 243).

VLS? (Get System Velocity)

Description: Gets the velocity of the moving platform of the Hexapod that is set with VLS (p. 245).

Format: VLS?

Arguments: None

Response: <SystemVelocity> is the velocity value in physical units, see VLS.

VMO? (Virtual Move)

Description: Checks whether the moving platform of the Hexapod can approach a specified position from the current position.

VMO? does not trigger any motion.

Format: VMO? {<AxisID> <Position>}

Arguments: <AxisID> is an axis of the controller, see below

<Position> is a target position value to be checked

Response: <uint> indicates whether the moving platform can approach the position resulting from the given target position values:

0 = specified position cannot be approached

1 = specified position can be approached

Notes: Axes X, Y, Z, U, V, W are permissible.

VMO? checks the following:

- Are the nodes and the target position outside of the travel range limits that can be queried with TMN? (p. 239) and TMX? (p. 239)?
- Have the soft limits set with NLM (p. 212) and PLM (p. 214) been activated with SSL (p. 229), and if yes, are the nodes and the target position outside of these soft limits?
- Are the individual struts able to move the platform to the required nodes and the specified target position?
- When a configuration for collision avoidance has been stored on the C-887 with the optionally available PIVeriMove Software for the Collision Check: Do collisions occur between the following groups?
  - Environment incl. base plate of the Hexapod
  - Hexapod struts
  - Moving platform of the Hexapod incl. load

In order to receive a reliable response, send VMO? not until after a successful reference move (start with FRF (p. 174)) and only when the Hexapod is not moving (query with #5 (p. 140)).

If you have changed to GCS syntax version 1.0 with CSV (p. 153), note the following:

- The query command is "VMO" (without a question mark).
- The response to VMO is inverted, i.e. 0 means that a position can be approached.

VST? (Get Connectable Stages)

Description: Lists the stage types that can be connected to the C-887 from stage databases on the C-887.

Format: VST?

Arguments: None

Response: <string> is a list of all stage types that are contained in stage databases on the C-887 and can be used with the C-887.

Notes: The stage database PIStage2.dat is stored on the C-887. If you would like to update this stage database, contact the customer service department (p. 303).

The version of the stage database stored on the C-887 can be found in the response of the VER? (p. 244) command.

The stage types listed with VST? can be assigned to the axes A and B with the CST (p. 152) command.

WAC (Wait For Condition)

Description: Waits until a given condition of the following type occurs: a specified value is compared with a queried value according a specified rule.

Can only be used in macros.

See also the MEX (p. 206) command.

Format: WAC <CMD?> <OP> <value>

Arguments

<CMD?> is one query command in its usual notation. The response has to be a single value and not more. For an example see below.

<OP> is the operator to be used. The following operators are possible:
= <= < > >= !=
Important: There must be a blank space before and after the operator!

<value> is the value to be compared with the response to <CMD?>.

Response: None

Example: Send:

```
MAC BEG LPMOTION
MVR 1 1
WAC ONT? 1 = 1
MVR 1 -1
WAC ONT? 1 = 1
MAC START LPMOTION
MAC END
MAC START LPMOTION
```

Note: Macro LPMOTION is first recorded and then started. WAC ONT? 1 = 1 waits until the response to ONT? 1 is 1=1. To form an infinite loop, the macro calls itself. To form an infinite loop, the macro calls itself.

8.7 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

20	PI_CNTR_MACRO_NOT_FOUND	Macro not found
21	PI_CNTR_AXIS_HAS_NO_BRAKE	Axis has no brake
22	PI_CNTR_DOUBLE_AXIS	Axis identifier specified more than once
23	PI_CNTR_ILLEGAL_AXIS	Illegal axis
24	PI_CNTR_PARAM_NR	Incorrect number of parameters
25	PI_CNTR_INVALID_REAL_NR	Invalid floating point number
26	PI_CNTR_MISSING_PARAM	Parameter missing
27	PI_CNTR_SOFT_LIMIT_OUT_OF_RANGE	Soft limit out of range
28	PI_CNTR_NO_MANUAL_PAD	No manual pad found
29	PI_CNTR_NO_JUMP	No more step-response values
30	PI_CNTR_INVALID_JUMP	No step-response values recorded
31	PI_CNTR_AXIS_HAS_NO_REFERENCE	Axis has no reference sensor
32	PI_CNTR_STAGE_HAS_NO_LIM_SWITCH	Axis has no limit switch
33	PI_CNTR_NO_RELAY_CARD	No relay card installed
34	PI_CNTR_CMD_NOT_ALLOWED_FOR_STAGE	Command not allowed for selected stage(s)
35	PI_CNTR_NO_DIGITAL_INPUT	No digital input installed
36	PI_CNTR_NO_DIGITAL_OUTPUT	No digital output configured
37	PI_CNTR_NO_MCM	No more MCM responses
38	PI_CNTR_INVALID_MCM	No MCM values recorded
39	PI_CNTR_INVALID_CNTR_NUMBER	Controller number invalid
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave
42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter)

		missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error
49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	MOV! motion still in progress
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data Record Table does not exist
58	PI_CNTR_INVALID_RECORDER_SRC_OPT	Source does not exist; number too low or too high
59	PI_CNTR_INVALID_RECORDER_SRC_CHAN	Source Record Table number too low or too high
60	PI_CNTR_PARAM_PROTECTION	Protected Param: current Command Level (CCL) too low
61	PI_CNTR_AUTOZERO_RUNNING	Command execution not possible while Autozero is running
62	PI_CNTR_NO_LINEAR_AXIS	Autozero requires at least one linear axis
63	PI_CNTR_INIT_RUNNING	Initialization still in progress
64	PI_CNTR_READ_ONLY_PARAMETER	Parameter is read-only
65	PI_CNTR_PAM_NOT_FOUND	Parameter not found in non-volatile memory
66	PI_CNTR_VOL_OUT_OF_LIMITS	Voltage out of limits

67	PI_CNTR_WAVE_TOO_LARGE	Not enough memory available for requested wave curve
68	PI_CNTR_NOT_ENOUGH_DDL_MEMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DELAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_ARRAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN_SINGLE_MODE_RESTART	Attempt to restart the generator while it is running in single step mode
72	PI_CNTR_ANALOG_TARGET_ACTIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERATOR_ACTIVE	Motion commands are not allowed when wave generator is active
74	PI_CNTR_AUTOZERO_DISABLED	No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)
75	PI_CNTR_NO_WAVE_SELECTED	Generator started (WGO) without having selected a wave table (WSL).
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly
77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_SERVO_ON	Open-loop commands (SVA, SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM
81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command

82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
90	PI_CNTR_DIO_AND_TRACING_NOT_POSSIBLE	No tracing possible while digital IOs are used on this HW revision. Reconnect to switch operation mode.
91	PI_CNTR_COLLISION	Move not possible, would cause collision
92	PI_CNTR_SLAVE_NOT_FAST_ENOUGH	Stage is not capable of following the master. Check the gear ratio.
93	PI_CNTR_CMD_NOT_ALLOWED_WHILE_AXIS_IN_MOTION	This command is not allowed while the affected axis or its master is in motion.
94	PI_CNTR_OPEN_LOOP_JOYSTICK_ENABLED	Servo cannot be switched on when open-loop joystick control is enabled.
95	PI_CNTR_INVALID_SERVO_STATE_FOR_PARAMETER	This parameter cannot be changed in current servo mode.
96	PI_CNTR_UNKNOWN_STAGE_NAME	Unknown stage name
100	PI_LABVIEW_ERROR	PI LabVIEW driver reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on

206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error
210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller
212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken
216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, some controllers need CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop
218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATIO N	Position was calculated during MOV motion
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_SE RVO_ON	Open-loop motion attempted when servo ON
304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long

305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus
307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro
310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined
402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
500	PI_CNTR_EMERGENCY_STOP_BUTTON_ACTIVATED	The \"red knob\" is still set and disables system
501	PI_CNTR_EMERGENCY_STOP_BUTTON_WAS_ACTIVATED	The \"red knob\" was activated and still disables system - reanimation required
502	PI_CNTR_REDUNDANCY_LIMIT_EXCEEDED	Position consistency check failed
503	PI_CNTR_COLLISION_SWITCH_ACTIVATED	Hardware collision sensor(s) are activated
504	PI_CNTR_FOLLOWING_ERROR	Strut following error occurred, e.g. caused by overload or encoder failure
505	PI_CNTR_SENSOR_SIGNAL_INVALID	One sensor signal is not valid
506	PI_CNTR_SERVO_LOOP_UNSTABLE	Servo loop was unstable due to wrong parameter setting and switched off to avoid damage.

555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
601	PI_CNTR_NOT_ENOUGH_MEMORY	not enough memory
602	PI_CNTR_HW_VOLTAGE_ERROR	hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	hardware temperature out of range
604	PI_CNTR_POSITION_ERROR_TOO_HIGH	Position error of any axis in the system is too high
606	PI_CNTR_INPUT_OUT_OF_RANGE	Maximum value of input signal has been exceeded
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
1006	PI_CNTR_INVALID_IDENTIFIER	Invalid identifier (invalid special characters, ...)
1007	PI_CNTR_UNKNOWN_VARIABLE_OR_ARGUMENT	Variable or argument not defined
1008	PI_CNTR_RUNNING_MACRO	Controller is (already) running a macro
1009	PI_CNTR_MACRO_INVALID_OPERATOR	Invalid or missing operator for condition. Check necessary spaces around operator.
1010	PI_CNTR_MACRO_NO_ANSWER	No answer was received while executing WAC/MEX/JRC/...
1011	PI_CMD_NOT_VALID_IN_MACRO_MODE	Command not valid during macro execution
1024	PI_CNTR_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
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
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
5000	PI_CNTR_INVALID_PCC_SCAN_DATA	PicoCompensation scan data is not valid
5001	PI_CNTR_PCC_SCAN_RUNNING	PicoCompensation is running, some actions can not be executed during scanning/recording
5002	PI_CNTR_INVALID_PCC_AXIS	Given axis can not be defined as PPC axis
5003	PI_CNTR_PCC_SCAN_OUT_OF_RANGE	Defined scan area is larger than the travel range

5004	PI_CNTR_PCC_TYPE_NOT_EXISTING	Given PicoCompensation type is not defined
5005	PI_CNTR_PCC_PAM_ERROR	PicoCompensation parameter error
5006	PI_CNTR_PCC_TABLE_ARRAY_TOO_LARGE	PicoCompensation table is larger than maximum table length
5100	PI_CNTR_NEXLINE_ERROR	Common error in NEXLINE® firmware module
5101	PI_CNTR_CHANNEL_ALREADY_USED	Output channel forNEXLINE® 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

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
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended operation
-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost
-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full

-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits
-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)
-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources
-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device
-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present
-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error

-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port: was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again
-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)

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
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?
-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed

-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup
-1032	PI_USER_CANCELLED	The user cancelled a dialog
-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL
-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter file open. Close it before opening a new file
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid
-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis
-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started

-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC
-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication with OPM (Optical Power Meter), maybe no OPM connected
-1057	PI_WAVE_EDITOR_WRONG_PARAMNUM	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	PI_WAVE_EDITOR_FREQUENCY_OUT_OF_RANGE	WaveEditor: Frequency out of range
-1059	PI_WAVE_EDITOR_WRONG_IP_VALUE	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	PI_WAVE_EDITOR_WRONG_DP_VALUE	WaveEditor: Error during wave creation, incorrect index for floating point parameter

-1061	PI_WAVE_EDITOR_WRONG_ITEM_VALUE	WaveEditor: Error during wave creation, could not calculate value
-1062	PI_WAVE_EDITOR_MISSING_GRAPH_COMPONENT	WaveEditor: Graph display component not installed
-1063	PI_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
-1064	PI_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
-1065	PI_EXT_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
-1066	PI_EXT_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
-1067	PI_PROFILE_GENERATOR_NO_PROFILE	ProfileGenerator: No profile has been created yet
-1068	PI_PROFILE_GENERATOR_OUT_OF_LIMITS	ProfileGenerator: Generated profile exceeds limits of one or both axes
-1069	PI_PROFILE_GENERATOR_UNKNOWN_PARAMETER	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	PI_PROFILE_GENERATOR_PAR_OUT_OF_RANGE	ProfileGenerator: Parameter out of allowed range
-1071	PI_EXT_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
-1072	PI_EXT_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
-1073	PI_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.

-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by another function. Please try again later.
-1077	PI_PARAM_DAT_FILE_INVALID_VERSION	Version of parameter DAT file does not match the required version. Current files are available at www.pi.ws.
-1078	PI_CANNOT_WRITE_TO_PARAM_DAT_FILE	Cannot write to parameter DAT file to store user defined stage type.
-1079	PI_CANNOT_CREATE_PARAM_DAT_FILE	Cannot create parameter DAT file to store user defined stage type.
-1080	PI_PARAM_DAT_FILE_INVALID_REVISION	Parameter DAT file does not have correct revision.
-1081	PI_USERSTAGES_DAT_FILE_INVALID_REVISION	User stages DAT file does not have correct revision.
-1082	PI_SOFTWARE_TIMEOUT	Timeout Error. Some lengthy operation did not finish within expected time.

9 Adapting Settings

In this Chapter

9.1 Changing volatilely configurable parameters in the C-887

INFORMATION

The settings that are used to adapt the C-887 e.g. to the properties of the connected mechanical system or the communication interfaces used are determined by:

- Configuration files and stage databases: only accessible for the customer service department (p. 303)
- Interface parameters: Adaptation in the non-volatile memory possible with the `IFS` command (p. 195), further information in "Establishing Communication via the TCP/IP Interface" (p. 79) and "Establishing Communication via the RS-232 Interface" (p. 88)
- Volatile-configurable parameters: information in this chapter

INFORMATION

After the C-887 has been switched on or rebooted, factory settings for the volatilely configurable parameters will be active unless a configuration takes place with a start-up macro

9.1.1 General Commands

The following general commands are available for changing volatilely configurable parameters:

Command	Function
SPA (p. 224)	Change parameters in the volatile memory.
CCL (p. 149)	Go to a higher command level, e. g. to obtain write permission for particular parameters.
SPA? (p. 226)	Get parameter values from the volatile memory.

9.1.2 Special Commands

The following special commands only change the corresponding parameters in the volatile memory.

Command	Adjustable parameters
RTR (p. 219)	Record table rate (ID 0x16000000), can also be changed with SPA (p. 224).
CST (p. 152)	Only for axes A and B, **cannot** be changed with SPA: ▪ P-Term (ID 0x1) ▪ I-Term (ID 0x2) ▪ D-Term (ID 0x3) ▪ I-Limit (ID 0x4) ▪ Maximum Position Error (ID 0x8) ▪ Numerator and denominator of the factor for impulses per physical length unit (IDs 0xE and 0xF) ▪ Is this a rotary stage? (ID 0x13)

INFORMATION

The command CST does not set parameter values directly but assigns a stage type to axes A and B . This loads the corresponding parameter values from a stage database on the controller to the volatile memory.

9.1.3 Changing Parameter Values

NOTICE

Unsuitable parameter settings can damage the connected mechanical system!

Unsuitable parameter settings can lead to improper operation or damage to the connected mechanical system.

➢ Only change the following parameters after careful consideration.

- For axes A and B:
  - ***Sensor Reference Mode*** (ID 0x02000A00)
- For axes K, L and M:
  - ***Servo-Loop P-Term*** (ID 0x07000300)
  - ***Notch Frequency*** (ID 0x08000100)
- For the Hexapod system:
  - ***Ignore Macro Error?*** (ID 0x72)
  - ***Security Distance*** (ID 0x01900952)

Determining available parameters

The parameters available for adapting the C-887 to your application depend in part on which optional accessories your C-887 has (p. 25).

➢ Send the `HPA?` command (p. 194) to obtain a list of all available parameters with a short description.

Changing parameter values in the volatile memory

➢ Send the `SPA?` command (p. 226) to obtain a list of parameter values in the volatile memory.

➢ Change the parameter values in the volatile memory with the `SPA` command (p. 224).

- and/or -

➢ Change selected parameter values in the volatile memory with special commands (p. 270).

INFORMATION

The following parameters can be changed in the volatile memory with `SPA`:

- For axes A and B:
  - ***Sensor Reference Mode*** (ID 0x02000A00)
    Before changing this parameter, it is necessary to go to command level 1, see `CCL` (p. 149).
- For axes K, L and M:
  - ***Servo-Loop P-Term*** (ID 0x07000300)
  - ***Notch Frequency*** (ID 0x08000100)
- For the Hexapod system:
  - ***Ignore Macro Error?*** (ID 0x72)
  - ***Pulse Length Factor*** (ID 0x0E000900)
  - ***Security Distance*** (ID 0x01900952)
  - ***Data Recorder Table Rate*** (ID 0x16000000)
  - ***Data Recorder Points Per Table*** (ID 0x16000201)
  - ***Trajectory Source*** (ID 0x19001900)
  - ***Trajectory Execution*** (ID 0x19001901)
  - ***Threshold for Trajectory Execution*** (ID 0x19001903)

INFORMATION

In order to change parameter values in the volatile memory directly after the controller is switched on or rebooted, a start-up macro can be used. For more information, see "Controller Macros" (p. 110).

9.1.4 Saving Parameter Values

INFORMATION

If you want to permanently save the changed settings of volatilely configurable parameters:

➢ Create a backup copy of the changed parameter settings of the C-887 as a text file on the PC.

Prerequisites

✓ You have read and understood the General Notes on Start-Up (p. 69).

✓ PIMikroMove or PITerminal is installed on the PC.

✓ You have established communication between the C-887 and the PC via the RS-232 interface (p. 88) or the TCP/IP interface (p. 79).

Saving parameter values in a text file

1. Start PITerminal or PIMikroMove on the PC.
2. If you use PIMikroMove, open the ***Command entry*** window with the ***Tools > Command entry*** menu item or the F4 key.
3. Get the parameter values from the volatile memory with the `SPA?` command.
4. Save the queried parameter values in a text file on the PC:

   a) Click the ***Save...*** button to open a file selection window.
   b) In the file selection window, select the directory to which you want to save the text file.
   c) Enter the filename of the text file in the file selection window.
   d) In the file selection window, click ***Save***.

9.2 Parameter Overview

INFORMATION

The parameters listed in the following, for which write access is granted to the command levels 0 and 1, can be volatilely configured.

Parameters with a command level that is higher than 1 for write access can only be changed by the PI customer service department or loaded with the `CST` command from a stage database on the controller to the volatile memory.

Parameter ID (hexa-decimal)	**Com-mand level for write access**	**Affected item type**	**Data type**	**Parameter name** (Unit)	**Description**
0x1	3	Hexapod struts 1 to 6, axes A and B	INT	P-term	Is changed for axes A and B in the volatile memory when a stage type is assigned with `CST` (p. 152).

Parameter ID (hexa-decimal)	Com-mand level for write access	Affected item type	Data type	Parameter name (Unit)	Description
0x2	3	Hexapod struts 1 to 6, axes A and B	INT	I-term	Is changed for axes A and B in the volatile memory when a stage type is assigned with `CST` (p. 152).
0x3	3	Hexapod struts 1 to 6, axes A and B	INT	D-term	Is changed for axes A and B in the volatile memory when a stage type is assigned with `CST` (p. 152).
0x4	3	Hexapod struts 1 to 6, axes A and B	INT	I-limit	Is changed for axes A and B in the volatile memory when a stage type is assigned with `CST` (p. 152).
0x8	3	Hexapod struts 1 to 6, axes A and B	FLOAT	Maximum Position Error (mm)	Maximum position error Is used for the detection of motion errors. If the position deviation (i.e. the absolute value of the difference between the current position and the commanded position) exceeds the specified maximum value in closed-loop operation, the C-887 sets the error code +1024 ("Motion error"), the servo mode is switched off and the moving platform or the axis is stopped. Is changed for axes A and B in the volatile memory when a stage type is assigned with `CST`.

Parameter ID (hexa-decimal)	Command level for write access	Affected item type	Data type	Parameter name (Unit)	Description
0xE	3	Hexapod struts 1 to 6, axes A and B	INT	Numerator Of The Counts-Per-Physical-Unit Factor	Numerator and denominator of the factor for impulses per physical length unit 1 to 1,000,000 for each parameter. The factor for impulses per physical length unit determines the unit for motion commands. Is changed for axes A and B in the volatile memory when a stage type is assigned with CST (p. 152).
0xF	3	Hexapod struts 1 to 6, axes A and B	INT	Denominator Of The Counts-Per-Physical-Unit Factor	
0x13	3	Axes A and B	INT	Is Rotary Stage?	Is this a rotary stage? 0 = Not a rotary stage 1 = Rotary stage Is changed for axes A and B in the volatile memory when a stage type is assigned with CST (p. 152).
0x72	0	Hexapod system	INT	Ignore Macro Error?	Ignore macro error? Determines whether the controller macro is stopped when an error occurs while it is being executed. 0 = Stop macro when error occurs (default) 1 = Ignore macro error
0x02000101	3	Axes K, L and M	FLOAT	Sensor Board Gain	Sensor gain factor
0x02000102	3	Axes K, L and M	FLOAT	Sensor Offset Factor	Offset factor of the sensor

Parameter ID (hexa-decimal)	Com-mand level for write access	Affected item type	Data type	Parameter name (Unit)	Description
0x02000A00	1	Axes A and B	INT	Sensor Reference Mode	Defines how the reference value for the incremental sensor is determined 1 = Reference move not necessary, see also RON? (p. 218) and "Referencing Axes A and B with PIMikroMove" (p. 98) 4 = Reference move to direction-sensing reference point switch necessary (default)
0x07000300	0	Axes K, L and M	FLOAT	Servo-Loop P-Term	P-Term of the controller Minimum value = 0.000045 Maximum value = 0.046
0x08000100	0	Axes K, L and M	FLOAT	Notch Frequency (Hz)	Notch filter frequency 100 to 333 Hz
0x0E000900	0	Hexapod system	INT	Pulse Length Factor	Multiplied by the cycle time (p. 306) Product is the pulse width of the impulse Default: 5
0x0E000B00	3	Hexapod system	INT	Number Of Input Channels	Number of installed analog input channels Input channels are only present when the controller is equipped with corresponding optional accessories, see "Optional Accessories" (p. 25). For details on the identifiers of the input channels, see "Commandable Items" (p. 28). Corresponds to the response to TAC? (p. 237)

Parameter ID (hexa-decimal)	Com-mand level for write access	Affected item type	Data type	Parameter name (Unit)	Description
0x16000000	0	Hexapod system	INT	Data Recorder Table Rate	Record table rate 1 to 1000000 Default: 250 See also RTR (p. 219).
0x16000100	3	Hexapod system	INT	Max. Number of Data Recorder Channels	Maximum number of data recorder tables
0x16000200	3	Hexapod system	INT	Data Recorder Max. Points	Maximum number of all points of the data recorder tables
0x16000201	0	Hexapod system	INT	Data Recorder Points Per Table	Number of points per data recorder table 1 to 10240 Default: 1024
0x16000300	3	Hexapod system	INT	Channel Number	Number of data recorder tables
0x19000951	3	Hexapod system	INT	Status	License status of the PIVeriMove Software for the Collision Check (C-887.VM1, see "Optional Accessories" (p. 25)) -2 = license invalid -1 = configuration file invalid, e. g. incorrectly formatted 0 = not activated 1 = activated
0x19000952	0	Hexapod system	FLOAT	Security Distance (mm)	Security distance for use in the PIVeriMove Software for the Collision Check Parameter value must be ≥ 0

Parameter ID (hexa-decimal)	**Com-mand level for write access**	**Affected item type**	**Data type**	**Parameter name** (Unit)	**Description**
0x19001500	3	Hexapod system	FLOAT	Maximum System Velocity (mm/s)	Maximum system velocity See specifications in the user manual of the Hexapod of the Hexapod system and "Motions of the Hexapod" (p. 32).
0x19001501	3	Hexapod system	FLOAT	Minimum System Velocity (mm/s)	Minimum system velocity For details see "Motions of the Hexapod" (p. 32).
0x19001502	3	Hexapod system	FLOAT	Maximum System Acceleration (mm/s^2)	Maximum system acceleration For details see "Motions of the Hexapod" (p. 32).
0x19001504	3	Hexapod system	FLOAT	Path Control Step Size (mm)	Step size for the trajectory calculation of the platform motion For details see "Motions of the Hexapod" (p. 32).
0x19001800	0	Hexapod system	INT	Coordination Mode	Coordination mode 0 = multi-axis mode 1 = Hexapod mode (default) The parameter value must not be changed.
0x19001900	0	Hexapod system	INT	Trajectory Source	Source of the motion profile 0 = Motion profile is determined by trajectory generator (default) 1 = Motion profile is determined by consecutive `MOV` commands For details see "Motions of the Hexapod" (p. 32).

Parameter ID (hexa-decimal)	Com-mand level for write access	Affected item type	Data type	Parameter name (Unit)	Description
0x19001901	0	Hexapod system	INT	Trajectory Execution	Execution of the motion profile 0 = Motion profile is immediately executed (default) 1 = Motion profile is stored in a buffer before execution For details see "Motions of the Hexapod" (p. 32).
0x19001902	3	Hexapod system	INT	Maximum Number of Trajectory Points	Maximum number of motion-profile points For details see "Motions of the Hexapod" (p. 32).
0x19001903	0	Hexapod system	INT	Threshold for Trajectory Execution	Threshold value for executing the motion profile For details see "Motions of the Hexapod" (p. 32).
0x19001904	3	Hexapod system	INT	Current Number of Trajectory Points	Shows the current number of motion-profile points in the buffer. For details see "Motions of the Hexapod" (p. 32).

10 Maintenance

In this Chapter

10.1 Cleaning the C-887

NOTICE

Pull the power plug before cleaning!

The C-887 contains electrostatic-sensitive devices (ESD) that can be damaged in the case of short circuits or flashovers.

➢ Before cleaning, disconnect the C-887 from the power source by pulling the power plug.

➢ When necessary, clean the surfaces of the C-887 case with a towel that has been lightly dampened with a mild cleanser or disinfectant.

➢ Do **not** use any organic solvents.

10.2 Cleaning the Air Filter

INFORMATION

The intervals for cleaning the air filter depend on the operating conditions of the C-887.

INFORMATION

The figures in the following section show a C-887.11 controller. In the case of the model C-887.21, the top of the case is fastened and removed in the same way as with the C-887.11, and the air filter is also located on the inside of the front panel of the case.

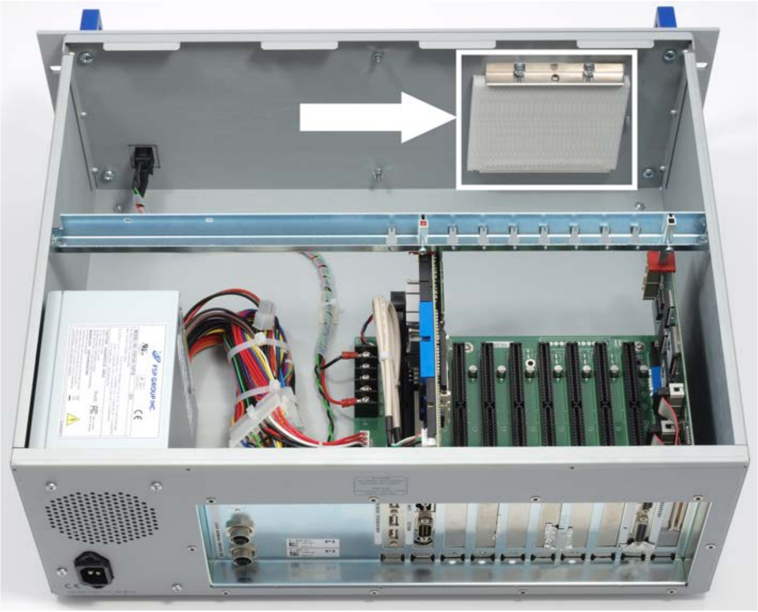


Figure 13: C-887.11 opened for air filter cleaning

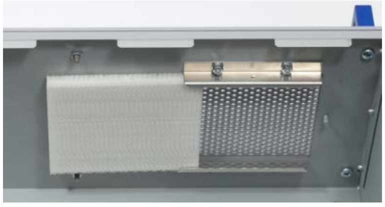


Figure 14: Air filter, already moved, and ventilation holes of the C-887

Prerequisite

- ✓ The C-887 is **not** connected to the power socket via the power cord.

Tools and accessories

- Compressed air, oil-free
- Supplied Torx screwdriver

Cleaning the air filter

1. Release the two Torx screws with which the top of the case is fastened to the rear panel of the controller.
2. Carefully remove the top of the case.
3. Remove the air filter from the rails in front of the ventilation holes in the front panel of the C-887, see figures above:

   a) Loosen the screw which locks the air filter in the upper rail.
   b) Remove the screw.
   c) Carefully push the air filter out of the rails.
4. Clean the air filter with oil-free compressed air.
5. Insert the air filter into the rails so that it completely covers the ventilation holes.
6. Lock the air filter in the upper rail using the screw.
7. Place the top of the case so that the locking lugs of the top and the case engage on the front panel.
8. Fasten the top of the case to the rear panel of the controller with the two Torx screws.

10.3 Updating Firmware

NOTICE

Damage from unintentional position changes!

The limit value for the load of the Hexapod determined with the simulation program is only valid when the servo mode is switched on (p. 52) for the axes of the moving platform. The maximum holding force when the servo mode is switched off is based on the self-locking of the actuators in the Hexapod struts and is lower than the limit value when the servo mode is switched on (see manual of the Hexapod).

When the actual load of the Hexapod exceeds the maximum holding force based on the self-locking of the actuators, unintentional position changes of the Hexapod can occur in the following cases:

- Switching off the C-887
- Rebooting the C-887
- Switching off the servo mode for the axes of the moving platform of the Hexapod.

As a result, collisions are possible between the Hexapod, the load to be moved and the environment. Collisions can damage the Hexapod, the load to be moved or the environment.

➢ Make sure that the actual load of the moving platform of the Hexapod does not exceed the maximum holding force based on the self-locking of the actuators before you switch off the servo mode, reboot or switch off the C-887.

INFORMATION

The firmware of the C-887 consists of several components that can be updated separately.

INFORMATION

To update the firmware, the C-887 and PC must communicate via the TCP/IP interface.

INFORMATION

The following firmware components should be kept up to date:

- FW
- GUI (user interface)
- Macro

Device identification string of the C-887

The `*IDN?` command reads out the device identification string of the C-887, which contains the version of the firmware component FW among other things.

Example: `(c) 2011 Physik Instrumente (PI) Karlsruhe, C-887, 107000001, 1.1.0.67`

- `C-887` Device name
- `107000001` Serial number of the C-887, contains the encrypted date of manufacture (for the serial numbers of the Hexapod system and the Hexapod, see the labeling on the devices)
- `1.1.0.67` Version of the firmware component FW

Firmware components

The versions of the firmware components can be read out with the `VER?` command.

Example:

- `FW: V 1.1.0.67` Version of the firmware component FW
- `GUI: V 1.1.0.26` Version of the user interface
- `Macro: V 0.8.0.106` Version of the macro functionality
- `OS: V #5 SMP PREEMPT RT Tue Jul 5 09:02:42 CEST 2011` Version of the operating system of the C-887
- `Hexdata: V 1.0.0.0` Version of the configuration file with the geometrical data for the Hexapod of the Hexapod system
- `PIStages: V 21/6/2011 14:15:26 V200.3` Version of the stage database (only necessary for axes A and B of the C-887.11)
- `C842.80: V 2.3 2 4 1 536` Version of the motor control for the Hexapod struts and the axes A and B available with the C-887.11
- `Nanocube: V E760 1.0.0.0` Firmware version of the E-760 controller card, is only displayed when the E-760 controller card is installed in the C-887.11 (F-206.NCU option)

- `Collision: V 2.15` Version of the library for the optionally available PIVeriMove Software for the Collision Check; is also displayed when the option is not activated

Ordering current firmware of the C-887

➢ Please contact our customer service department (p. 303) to obtain current versions of the firmware components.

Tools and accessories

- PC with Windows operating system that has been prepared as follows:
  - The PI Hexapod Controller Update Wizard program is installed. For further information, see "Installing PC Software" (p. 43).
  - The current files of the firmware components that you have received from our customer service department (file type "Update Package", file extension "IPK") are located in a directory on the PC.

Prerequisite

✓ You have made all necessary preparations for communication via the TCP/IP interface, see "Establishing Communication via TCP/IP in the PC-Software" (p. 85).

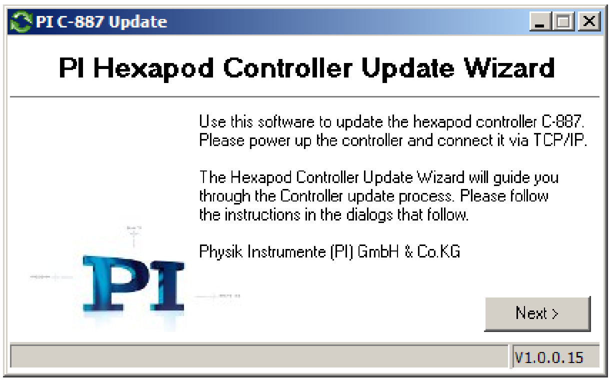
✓ The C-887 is switched on and the starting procedure of the C-887 has finished (p. 73).

✓ The PC is switched on.

Updating firmware components of the C-887

1. Start the PI Hexapod Controller Update Wizard program on the PC with the ***All Programs > PI > C-887 > Firmware Update Tool*** Start menu item.

   The ***PI C-887 Update*** window opens.



PI C-887 Update

PI Hexapod Controller Update Wizard

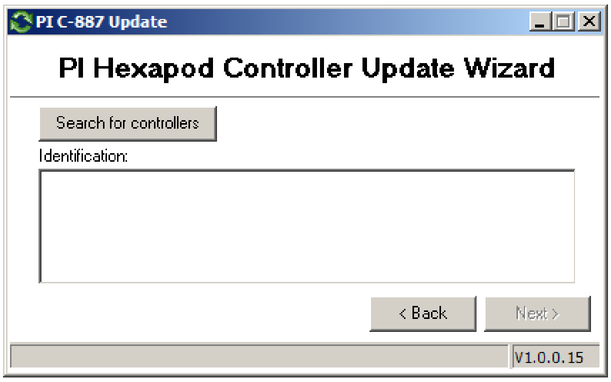
Use this software to update the hexapod controller C-887.
Please power up the controller and connect it via TCP/IP.

The Hexapod Controller Update Wizard will guide you through the Controller update process. Please follow the instructions in the dialogs that follow.

Physik Instrumente (PI) GmbH & Co.KG

2. Click the ***Next >*** button.

   The ***Search for controllers*** button is displayed.



PI C-887 Update

PI Hexapod Controller Update Wizard

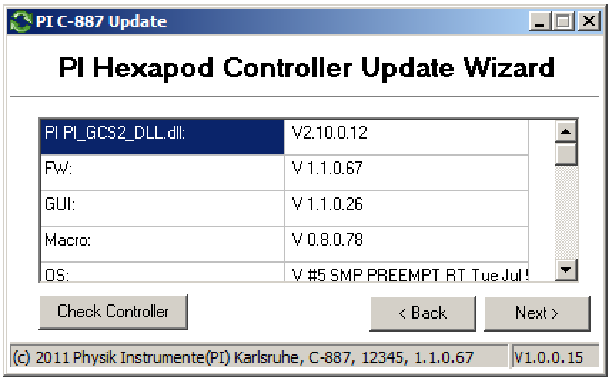
3. Click the ***Search for controllers*** button.

   The found C-887 is displayed in the ***Identification:*** field (serial number, port, IP address).

4. Mark the line of the C-887 in the ***Identification:*** field.

5. Click the ***Next >*** button.

   The currently active firmware components of the C-887 as well as the dynamic program library PI_GCS2_DLL installed on the PC are listed with their version numbers.



PI C-887 Update

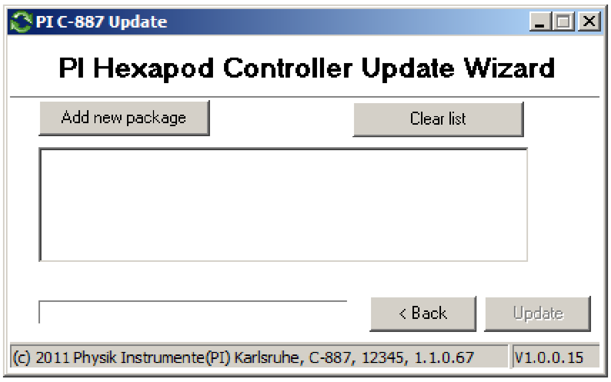
PI Hexapod Controller Update Wizard

PI PI_GCS2_DLL.dll:	V2.10.0.12
FW:	V 1.1.0.67
GUI:	V 1.1.0.26
Macro:	V 0.8.0.78
OS:	V #5 SMP PREEMPT RT Tue Jul !

(c) 2011 Physik Instrumente(PI) Karlsruhe, C-887, 12345, 1.1.0.67 V1.0.0.15

6. Click the ***Next >*** button.

   The ***Add new package*** button for selecting the firmware components to be updated is displayed.



PI C-887 Update

PI Hexapod Controller Update Wizard

(c) 2011 Physik Instrumente(PI) Karlsruhe, C-887, 12345, 1.1.0.67 V1.0.0.15

7. Click the ***Add new package*** button.

   A file selection window opens

8. In the file selection window, go to the directory that has the files that you have received from the customer service department.

9. In the file selection window, select the files for the firmware components to be updated. Filenames:

   – FW component: hexapod-firmware_x.x.x.x.ipk

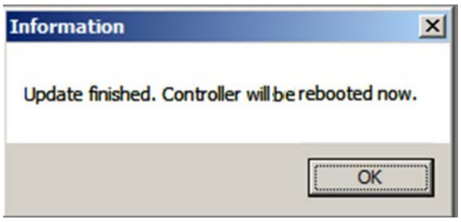
   – GUI component (user interface): hexapod-gui_x.x.x.x.ipk

   – Macro component: hexapod-macro-manager_x.x.x.x.ipk

10. Click the ***Open*** button in the file selection window to confirm the selection.

11. Click the ***Update*** button to start updating the selected firmware components on the C-887.

    The ***Information*** window is displayed.



Update finished. Controller will be rebooted now.

12. Click the ***OK*** button.

    The C-887 reboots.

    The updating of the firmware components is finished.

10.4 Maintaining and Checking the Hexapod

10.4.1 Carrying out a Maintenance Run

CAUTION

Risk of crushing by moving parts!

There is a risk of minor injuries caused by crushing which can occur between the moving parts of the Hexapod and a stationary part or obstacle.

- Keep your fingers away from areas where they can get caught by moving parts.

NOTICE

Damage from collisions!

Collisions can damage the Hexapod, the load to be moved and the environment.

- Make sure that no collisions between the Hexapod, the load to be moved and the environment are possible in the working space of the Hexapod.
- Do not place any objects in areas where they can get caught by moving parts.
- If the Hexapod controller malfunctions, stop the motion immediately.

Frequent motions over a limited travel range can cause the lubricant to be unevenly distributed on the drive screws of the Hexapod struts. This can result in increased wear. A maintenance run across the entire travel range of the Hexapod struts evenly distributes the lubricant again.

INFORMATION

The more often motions are carried out over a limited travel range, the shorter the time between the maintenance runs has to be.

➢ Carry out the maintenance run at appropriate intervals according to the requirements of your application.

Prerequisites

✓ You have read and understood the General Notes on Start-Up (p. 69).

✓ You have correctly installed the Hexapod system (C-887, Hexapod and cable set) (p. 41).

✓ You have read and understood the user manual of the Hexapod.

✓ If you want to perform the maintenance run with PIMikroMove:

- PIMikroMove is installed on the PC (p. 43).
- You have read and understood the PIMikroMove manual. The manual is found on the product CD.
- You have established communication between the C-887 and the PC with PIMikroMove via the TCP/IP interface (p. 85) or the RS-232 interface (p. 89).
- You have read and understood the section "Starting Motions of the Platform of the Hexapod" (p. 91).

✓ If you want to perform the maintenance run with the user interface of the C-887: You have read and understood the section "Using the User Interface of the C-887" and made the preparations for using the user interface of the C-887 mentioned there (p. 74).

Carrying out a Maintenance Run

1. Set the velocity for the moving platform of the Hexapod with the `VLS` command (p. 245) to about 30 % of the default value which is valid after switching on or rebooting the C-887.
2. Command a motion of the Hexapod to the following target position:

   Z = Greatest commandable position (query with the `TMX?` command (p. 239) possible)

   X = Y = U = V = W = 0
3. Command a motion of the Hexapod to the following target position:

Z = Smallest commandable position (query with the `TMN?` command (p. 239) possible)

X = Y = U = V = W = 0

4. Repeat steps 2 and 3 three times each.
5. Command a motion of the Hexapod to the target position X = Y = Z = U = V = W = 0.

10.4.2 Performing a Strut Test

CAUTION

⚠

Risk of crushing by moving parts!

There is a risk of minor injuries caused by crushing which can occur between the moving parts of the Hexapod and a stationary part or obstacle.

➢ Keep your fingers away from areas where they can get caught by moving parts.

NOTICE

❗

Damage from collisions during the strut test!

During a strut test, the Hexapod moves unpredictably. **No** collision check or prevention takes place, even when a configuration for collision prevention has been stored on the C-887 with the PIVeriMove Software for the Collision Check. Soft limits that have been set for the moving platform of the Hexapod with the `NLM` (p. 212) and `PLM` (p. 214) commands are ignored during a strut test.

As a result, collisions are possible between the Hexapod, the load to be moved and the environment. Collisions can damage the Hexapod, the load to be moved and the environment.

➢ Make sure that no collisions are possible between the Hexapod, the load to be moved and the environment during a strut test of the Hexapod.

➢ Do not place any objects in areas where they can get caught by moving parts during a strut test.

➢ Observe the Hexapod during a strut test in order to be able to intervene quickly in the case of malfunctions.

NOTICE

Damage from unintentional position changes during the strut test!

During a strut test, the strut can reach a limit switch. As a result, the servo mode is automatically switched off for the axes of the moving platform of the Hexapod.

When the actual load of the Hexapod exceeds the maximum holding force based on the self-locking of the actuators, switching off the servo mode for the axes of the moving platform of the Hexapod can cause unintentional position changes of the Hexapod.

As a result, collisions are possible between the Hexapod, the load to be moved and the environment. Collisions can damage the Hexapod, the load to be moved and the environment.

➢ Make sure that the actual load of the moving platform of the Hexapod does not exceed the maximum holding force based on the self-locking of the actuators before you start a strut test.

INFORMATION

Recommendations for performing a strut test:

➢ If possible, command the Hexapod to move to the reference position before a strut test, in order for each strut to have the greatest possible travel range in the positive and negative direction of motion available.

➢ Set the appropriate velocity with the `VLS` command (p. 245) before a strut test.

If malfunctions occur in the Hexapod system, the following tests of the Hexapod struts can make error diagnosis easier:

- Measurement of the impulse response of the strut. Signs of malfunctions are:
  - Strong overshooting of the current position
  - Oscillations of the current position
  - Significant deviation between the graphs of the current position and the target position
- Measurement of the motor control during one drive screw revolution of the Hexapod strut, given as a percentage of the maximum value; allows evaluation of the control reserve, among other things. Signs of malfunctions are:
  - Motor control continuously >80 %

- – Unusually high motor control value at a particular point in the travel range

The following tools can be used for the tests:

- ***Hexapod Service Tools*** window in PIMikroMove
- ***hexapod strut tests*** window of the user interface of the C-887

INFORMATION

As an alternative to the impulse response, you can also record and evaluate a step response of the Hexapod strut. After the step has been executed, the Hexapod strut does **not** move back to the initial position, however. If a step is executed several times, the Hexapod strut can reach the limit switch. When the limit switch is reached, the servo mode is switched off for the axes of the moving platform of the Hexapod.

Prerequisites

- ✓ You have read and understood the General Notes on Start-Up (p. 69).
- ✓ You have correctly installed the Hexapod system (C-887, Hexapod and cable set) (p. 41).
- ✓ You have read and understood the user manual of the Hexapod.
- ✓ If you want to perform the strut test in the ***Hexapod Service Tools*** window of PIMikroMove:
  - – PIMikroMove is installed on the PC (p. 43).
  - – You have read and understood the PIMikroMove manual. The manual is found on the product CD.
  - – You have established communication between the C-887 and the PC with PIMikroMove via the TCP/IP interface (p. 85) or the RS-232 interface (p. 89).
- ✓ If you want to perform the strut test with the user interface of the C-887: You have made the preparations for using the user interface of the C-887 (p. 74).

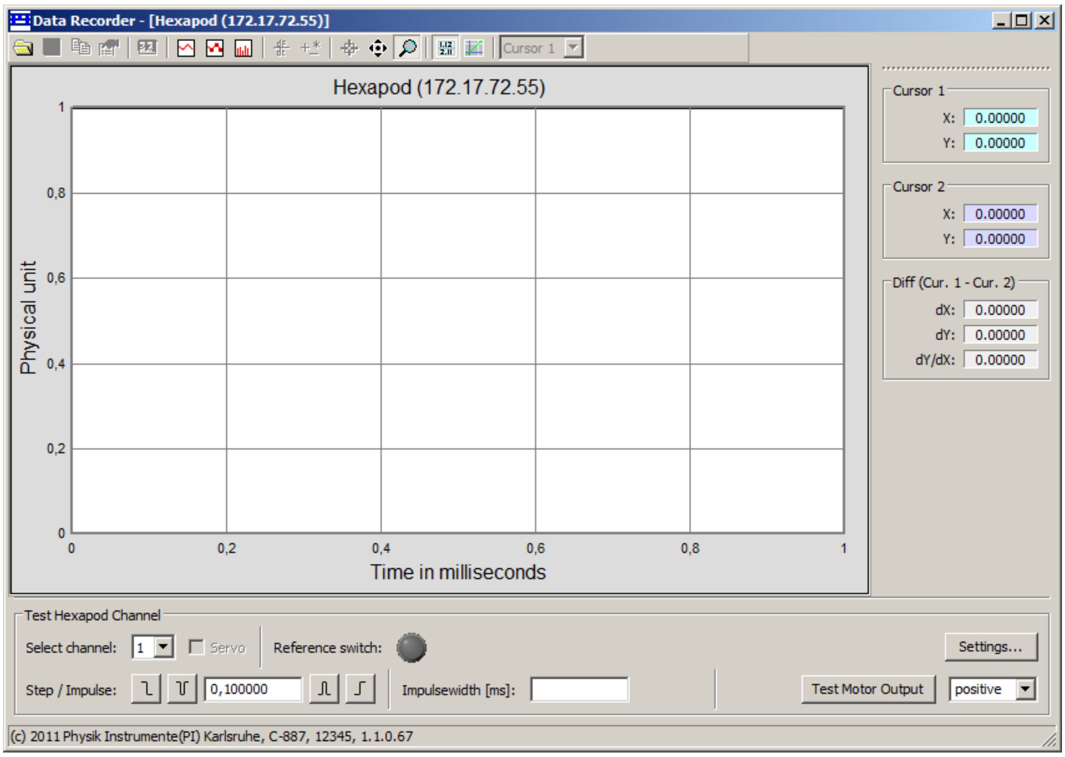
Performing the strut test in the *Hexapod Service Tools* window of PIMikroMove

1. In the main window of PIMikroMove, select the ***Hexapod > Show service tools...*** menu item.

   A window opens with information on possible property damage when the ***Hexapod Service Tools*** window is used.

2. In the information window, click ***Show service tools***.

   The ***Hexapod Service Tools*** window opens.



Cursor 1 X: 0.00000 Y: 0.00000

Cursor 2 X: 0.00000 Y: 0.00000

Diff (Cur. 1 - Cur. 2) dX: 0.00000 dY: 0.00000 dY/dX: 0.00000

Select channel: 1

Step / Impulse: 0,100000

positive

(c) 2011 Physik Instrumente(PI) Karlsruhe, C-887, 12345, 1.1.0.67

3. Record an impulse response for a Hexapod strut in the ***Hexapod Service Tools*** window:

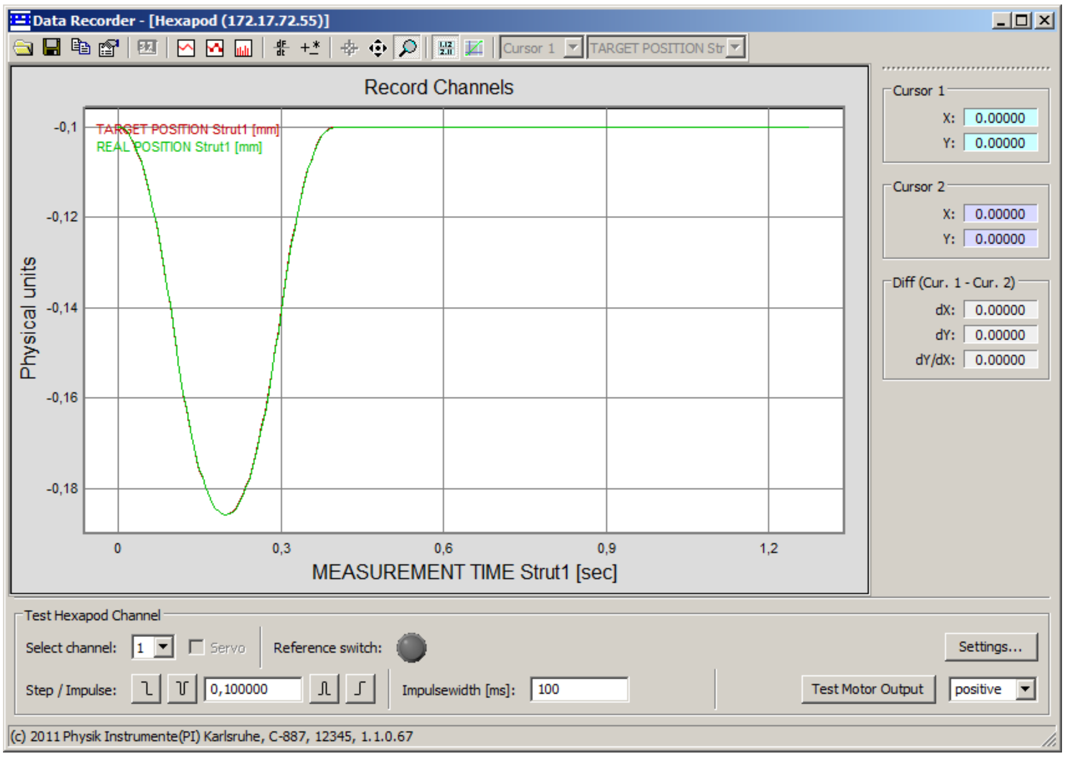
   a) In the ***Select channel*** field, select the strut for which the impulse response is to be recorded.

   b) In the ***Impulsewidth [ms]*** field, enter the pulse width of the impulse in milliseconds.

   c) Click the ⎍ or ⊔ button to start the impulse in a positive or negative direction.
   Starting the impulse also starts the recording of the current position and the target position of the strut. The strut moves according to the set pulse width of the impulse and then returns back to the initial position.

   d) Evaluate the impulse response using the curves in the graphic field of the ***Hexapod Service Tools*** window (see following figure).

A step response with a corresponding recording can be started with the ⎽⎺ or ⎺⎽ button. The step size of the step is determined by the entry in the field between the buttons for starting the impulses.



Data Recorder - [Hexapod (172.17.72.55)]

Cursor 1
X: 0.00000
Y: 0.00000

Cursor 2
X: 0.00000
Y: 0.00000

Diff (Cur. 1 - Cur. 2)
dX: 0.00000
dY: 0.00000
dY/dX: 0.00000

Test Hexapod Channel
Select channel: 1
☐ Servo
Reference switch: ●
Settings...
Step / Impulse: 0,100000
Impulsewidth [ms]: 100
Test Motor Output
positive

(c) 2011 Physik Instrumente(PI) Karlsruhe, C-887, 12345, 1.1.0.67

4. Record the motor control during one drive screw revolution of a Hexapod strut in the ***Hexapod Service Tools*** window:

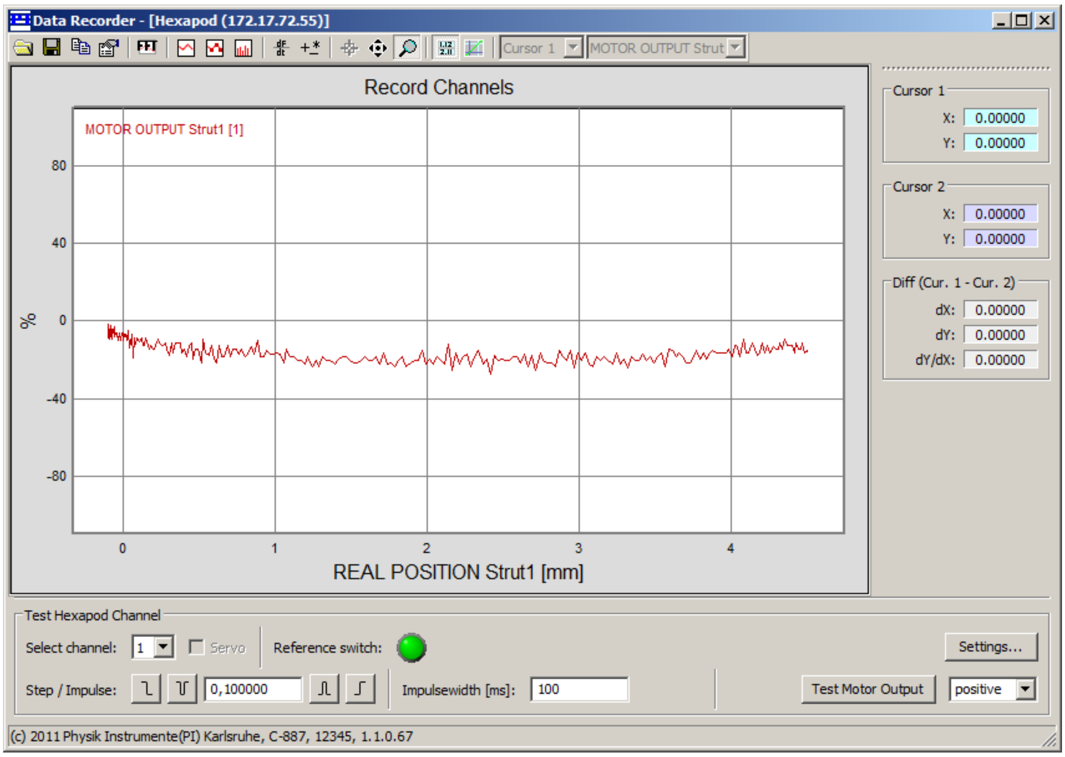
   a) In the ***Select channel*** field, select the strut for which the motor control is to be recorded.

   b) In the field on the right side of the ***Test Motor Output*** button, select the direction (positive or negative) in which the drive screw revolution is to be executed.

   c) Click the ***Test Motor Output*** button to start one drive screw revolution of the Hexapod strut in the selected direction.
   Starting the drive screw revolution also starts the recording of the current motor control and the current position of the Hexapod strut. The Hexapod strut does **not** move back to the initial position after the drive screw revolution.

   d) Evaluate the recorded motor control using the curve in the graphic field of the ***Hexapod Service Tools*** window (see following figure).

   If a drive screw revolution is executed several times in the same direction, the Hexapod strut can reach the limit switch. When the limit switch is reached, the servo mode is switched off for the axes of the moving platform of the Hexapod.



Cursor 1 X: 0.00000 Y: 0.00000

Cursor 2 X: 0.00000 Y: 0.00000

Diff (Cur. 1 - Cur. 2) dX: 0.00000 dY: 0.00000 dY/dX: 0.00000

Select channel: 1 ☐ Servo

Step / Impulse: 0,100000

Impulsewidth [ms]: 100

positive

(c) 2011 Physik Instrumente(PI) Karlsruhe, C-887, 12345, 1.1.0.67

5. If the test results indicate a malfunction:

   a) Do **not** operate the Hexapod system further.

   b) Communicate the test results for the error diagnosis to our customer service department (p. 303).

Performing a strut test with the user interface of C-887

1. Press the F1 key on the keyboard or click the ***F1 TEST TOOLS*** button in the main window of the user interface to open the ***hexapod strut tests*** window.

2. In the ***hexapod strut tests*** window, record an impulse response for a strut:

   a) Press the F2 key or click the ***F2 Impulse Response*** button.

   b) Press the F5 key or click the ***F5 START TEST*** button to open the ***Test program*** window.

   c) In the ***Test program*** window, enter the identifier of the strut for which the impulse response is to be recorded (1 to 6), and confirm the entry with the Enter key or by clicking the ***OK*** button. The ***Impulse response*** window opens.

   d) Do not change the value for the impulse height in the ***Impulse response*** window, and press the Enter key or click on the ***OK*** button to go to where you can enter the pulse width of the impulse.

   e) Enter the pulse width of the impulse in milliseconds.

f) Press the Enter key or click the ***OK*** button to start the impulse and the recording of the current position of the Hexapod strut. The Hexapod strut moves according to the set pulse width of the impulse and then returns back to the initial position.

g) Evaluate the impulse response using the curve in the ***Step/Impulse response*** graphic field of the ***hexapod strut tests*** window.

A recording of the step response can be started in the ***hexapod strut tests*** window with the F1 key or the ***F1 Step Response*** button.

3. Record the motor control during one drive screw revolution of a Hexapod strut in the ***hexapod strut tests*** window :

a) Press the F3 key or click the ***F3 Motor Output*** button.

b) Press the F5 key or click the ***F5 START TEST*** button to open the ***Test program*** window.

c) In the ***Test program*** window, enter the identifier of the strut for which the motor control is to be recorded (1 to 6), and confirm the entry with the Enter key or by clicking the ***OK*** button. The ***Motor Output*** window opens.

d) In the ***Motor Output*** window, enter the direction (1 = positive, -1 = negative) in which the drive screw revolution is to be executed.

e) Press the Enter key or click the ***OK*** button to start the drive screw revolution and the recording of the current motor control of the Hexapod strut. The Hexapod strut does **not** move back to the initial position after the drive screw revolution.

f) Evaluate the recorded motor control using the curve in the graphic field of the ***hexapod strut tests*** window.

If a drive screw revolution is executed several times in the same direction, the Hexapod strut can reach the limit switch. When the limit switch is reached, the servo mode is switched off for the axes of the moving platform of the Hexapod.

4. If the test results indicate a malfunction:

a) Do **not** operate the Hexapod system further.

b) Communicate the test results for the error diagnosis to our customer service department (p. 303).

11 Troubleshooting

Problem	Possible Causes	Solution
Hexapod does not move	The cable is not connected correctly	➢ Check the cable connections.
	Incorrect command or incorrect syntax	➢ Send the `ERR?` command (p. 163) and check the error code this returns.
	Incorrect configuration	➢ Check the parameter settings in the volatile memory with the `SPA?` command (p. 226). The configuration data used by the C-887 must be adapted to the Hexapod. ➢ Check with the `CST?` command (p. 152) whether the axes of the moving platform of the Hexapod have been assigned the correct stage type. ➢ If the stage type does not match the connected Hexapod, contact our customer service department (p. 303).
One Hexapod strut does not move or is difficult to move	Wear of the drive screw	➢ If only small motions are executed over a long period of time: Carry out a maintenance run at regular intervals (p. 289).
	▪ Foreign body has entered the drive screw ▪ Faulty motor ▪ Blocked joint due to wear or foreign body ▪ Wear of the drive screw	➢ Carry out a strut test (p. 291) and communicate the test results to our customer service department (p. 303).
Reduced accuracy of the Hexapod	Warped base plate	➢ Mount the Hexapod on an even surface (see user manual of the Hexapod).

Problem	Possible Causes	Solution
Communication with the C-887 failed	The operating system and firmware of the C-887 do not start because the permissible operating temperature has been exceeded	1. Switch off the C-887. 2. Wait a few minutes until the C-887 has cooled off. 3. Switch on the C-887. ➢ Ensure sufficient ventilation at the place of installation (p. 42) to prevent the C-887 from overheating in the future.
	Communication cable is incorrect or defective	➢ Check the cable. – Use a null-modem cable for the RS-232 connection. – Use the straight-through network cable for TCP/IP connection via a hub or a router (with DHCP server). – Use the cross-over network cable for direct connection with the Ethernet connection socket of the PC. ➢ If necessary, also check whether the cable works on a fault-free system.
	Communication interface is not correctly configured	**When using the RS-232 interface:** ➢ Check the port settings, the baud rate and the handshake setting of the PC. **When using the TCP/IP connection:** ➢ Connect the controller to the network **before** you switch it on. Otherwise, you will have to switch the controller off and then on again. ➢ Check the network settings (p. 79). ➢ Make sure that the network is not blocked for unknown devices. ➢ Make sure that several PC software applications cannot access the C-887 at the same time. ➢ Make sure that you have selected the correct C-887 when establishing communication. ➢ If you cannot solve the problems, consult your network administrator if necessary.
	The starting procedure of the operating system and the firmware of the C-	1. Wait approx. 36 seconds after switching on or rebooting the C-887 (second signal tone has been emitted). 2. Try to establish communication or send commands.

Problem	Possible Causes	Solution
	887 has not finished yet.	When the IP addresses of the network participants are configured with AutoIP, it will take up to 2 minutes after the end of the starting procedure of the C-887 (p. 73) until communication is possible via TCP/IP.
	A different program is accessing the interface	➢ Close the other program.
	Problems with special PC software	➢ Check whether the system works with different PC software, such as e. g. a terminal program or a development environment. You can test the communication by starting a terminal program (such as e. g. PI Terminal) and entering `*IDN?` or `HLP?`. ➢ Make sure that you end the command with an LF (line feed). A command is not executed until the LF has been received.
The customer software does not run with the PI drivers	Incorrect combination of driver routines/ Vis	➢ Check whether the system works with a terminal program. If so: ➢ Read the information in the corresponding software manual and compare the sample code on the C-887.CD with your program code.

If the problem that occurred with your system is not listed in the table above or it cannot be solved as described, contact our customer service department (p. 303).

12 Customer Service

For inquiries and orders, contact your PI sales engineer or send us an e-mail (mailto:info@pi.ws).

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

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

The latest versions of the relevant user manuals for your system are available for download on our website (http://www.pi.ws).

13 Technical Data

In this Chapter

13.1 Specifications

13.1.1 Data Table

	C-887.11	C-887.21
Function	6-D controller for Hexapods, 19", incl. control of two additional single axes, can be enhanced with many options	6-D controller for Hexapods, compact bench-top for a lower system price
Drive type	Servo motors (Hexapod and additional axes) Optional: Piezo drives	Servo motors
Motion and control		
Servo characteristics	32-bit PID filter	
Trajectory profile modes	Trapezoid, linear interpolation	
Processor	CPU: 1.6 GHz, motion control chip with 2.5 kHz servo update rate	
Encoder input	AB (quadrature) differential TTL signal, 5 MHz	
Stall detection	Servo off, triggered by position error	
Reference point switch	TTL signal	
Electrical properties		
Max. output power per channel	10-bit output for PWM drivers, 24 kHz	
Max. output voltage per channel	TTL in PWM operation for SIGN and MAGN	

Interface and operation		
Communication interfaces	TCP/IP, RS-232 VGA (monitor), USB (keyboard, manual control unit)	
Hexapod connection	MDR, 68-pin for data transmission M12 4-pin. for power supply	
Connection for additional single axes	15-pin sub-D	–
I/O ports	Optional: Analog inputs (photometer cards)	–
Command set	PI General Command Set (GCS)	
User software	PIMikroMove	
Software drivers	LabVIEW driver, shared libraries for Windows and Linux	
Manual control	Optional: C-887.MC control unit for Hexapods	
Miscellaneous		
Operating voltage	100 to 240 VAC, 50 / 60 Hz	
Operating temperature range	5 to 40°C	
Mass	11 kg	5 kg
Dimensions	395 × 483 × 185 mm	255 × 226 × 185 mm

13.1.2 Cycle Times

The differences in the cycle times of axes and Hexapod struts result from the different drive concepts:

- Axes X, Y, Z, U, V and W of the moving platform of the Hexapod: motion results from the motions of struts 1 to 6
- Hexapod struts 1 to 6 and axes A and B: DC servo motors
- Axes K, L, M: piezo actuators

Axis	**X, Y, Z, U, V, W**	**A, B, Hexapod struts 1 to 6**	**K, L, M**
Cycle time in ms	1	0.403	0.02

13.1.3 Maximum Ratings

The C-887 is designed for the following maximum ratings:

Input on:	Maximum Operating Voltage	Maximum Operating Frequency (Unloaded)	Maximum Current Consumption
Power plug	100 to 240 V AC	50 to 60 Hz	6 A

Output on:	Maximum output voltage	Maximum output current	Maximum output frequency
M12 sockets	24 V DC	∑ max. 5 A	⎓

13.1.4 Ambient Conditions and Classifications

The following ambient conditions and classifications must be observed for the C-887:

Area of application	For indoor use only
Maximum altitude	2000 m
Relative humidity	Highest relative humidity 80 % for temperatures up to 31 °C Decreasing linearly to 50 % relative humidity 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
Measurement category	I
Degree of protection according to IEC 60529	IP20

13.1.5 Specifications of the F-206.iiU and F-206.VVU Options

The photometer cards described in the following are available as optional accessories for the C-887.11 (p. 25).

	F-206.iiU	**F-206.VVU**
2-channel photometer card	for the IR range	for the visible range
Optical performance range	5 nW to 10 mW	5 nW to 10 mW
Analog input voltage range	0 to 10 V	0 to 10 V
A/D resolution	16 bit	16 bit
Sampling rate	10 kHz	10 kHz
Bandwidth	2 kHz (optical input) 10 kHz (voltage input)	2 kHz (optical input) 10 kHz (voltage input)
Max. sensitivity at	1550 nm	880 nm
40 % sensitivity at	850 / 1680 nm	480 / 1040 nm

Signal of the optical input of an F-206.iiU photometer card with a 2 kHz measuring signal

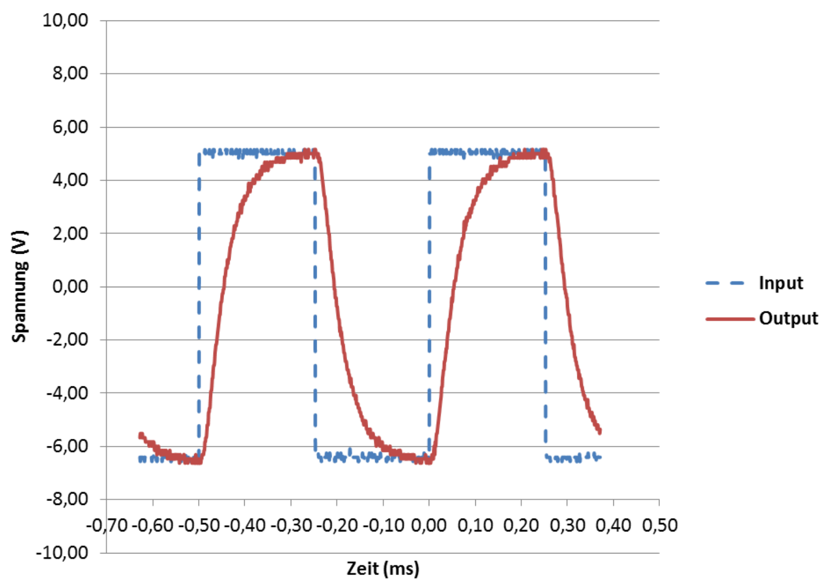


Figure 15: Input stands for the measuring signal (2 kHz), output stands for the signal of the optical input of the photometer card

Spectral sensitivity of the F-206.VVU photometer card

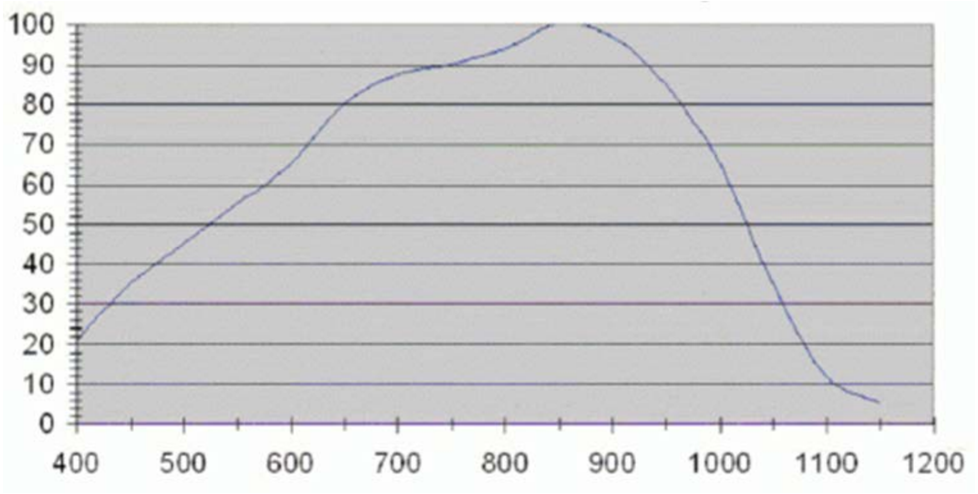


Figure 16: Relative spectral sensitivity of an F-206.VVU photometer card over the wavelength

Spectral sensitivity of the F-206.iiU photometer card

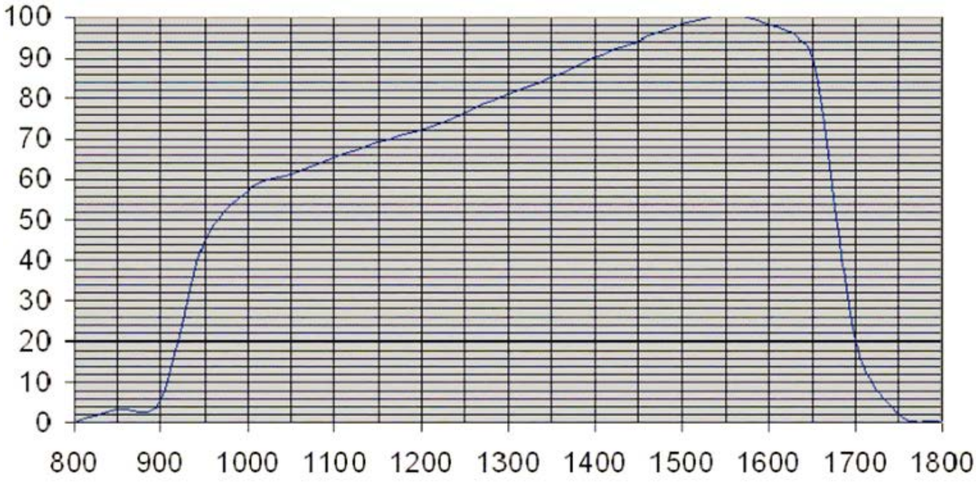


Figure 17: Relative spectral sensitivity of an F-206.iiU photometer card over the wavelength

Optical/analog/digital conversion

The photometer cards convert the optical signals fed in via FC connections into analog signals for processing by the C-887. The following holds true:

- The converters of the optical signals do not have any connection to the ISA bus via which the photometer card is connected to the C-887.

- The analog signals resulting from the conversion are fed into the C-887 by each being conveyed through an analog output of the photometer card to an analog input of the photometer card and transmitted from there via the ISA bus.

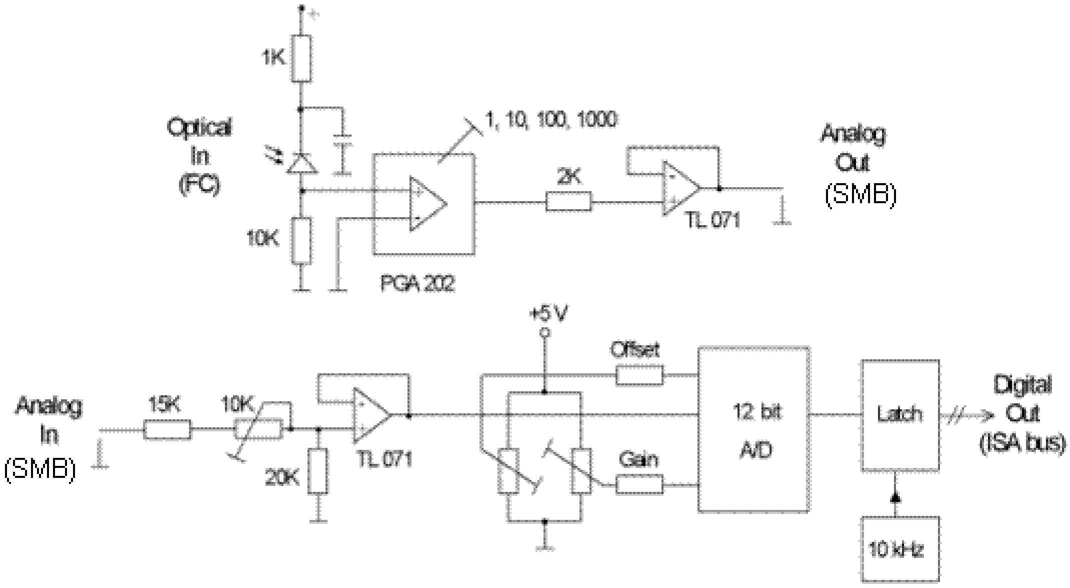


Figure 18: Block diagram of a photometer card channel

13.1.6 Specifications of the F-206.NCU Option

The F-206.NCU option described in the following is available for the C-887.11 (p. 25) and comprises the following components:

- E-760.3S0 controller card for installation in the C-887.11
- P-611.3SF NanoCube® stage

	E-760.3S0 controller card
Axes	3
Sensor	
Servo characteristics	PI (analog) + notch filter
Sensor type	SGS
Sensor channels	3
Amplifier	
Control input voltage	-2 to +12 V
Output voltage range	-20 to +120 V
Amplifier channels	3
Peak output power / channel, max.	9 W
Average output power / channel, max.	1 W
Peak current / channel, <5 ms	90 mA
Average current / channel, >5 ms	8 mA
Current limitation	Short-circuit-proof
Interface and operation	
Communication interfaces	Internal controller bus (ISA)
Piezo / sensor connection	Sub-D socket 25-pin
Analog input	RJ45 socket

	P-611.3SF NanoCube®
Active axes	K, L, M
Motion and positioning	
Integrated sensor	SGS
Open-loop travel, -20 to +120 V	120 µm / axis
Closed-loop travel	100 µm / axis
Open-loop resolution	0.2 nm
Closed-loop resolution	1 nm
Linearity	0.1 %
Repeatability	<10 nm
Pitch in K, L	± 5 µrad
Tilting θ_K (motion in M)	±10 µrad
Yaw in K	±20 µrad
Yaw in L	±10 µrad
Tilting θ_L (motion in M)	±10 µrad
Mechanical properties	
Stiffness	0.3 N/µm
Resonant frequency, unloaded, K/L/M	350/220/250 Hz
Resonant frequency, loaded, K/L/M, 30 g	270/185/230 Hz
Resonant frequency, loaded, K/L/M, 100 g	180/135/200 Hz
Pressure/tensile loading capacity in the direction of motion	+15 / -10 N
Load capacity	15 N
Miscellaneous	
Operating temperature range	-20 to 80°C
Material	Aluminum, steel
Dimensions	44 x 50 x 44.2 mm
Mass	0.32 kg
Cable length	1.5 m
Sensor connection	Sub-D connector 25-pin
Voltage connection	Sub-D connector 25-pin

13.2 System Requirements

The following system requirements must be met to operate the Hexapod system:

- The Hexapod of the Hexapod system to whose configuration data the controller is adapted (see sticker on the controller)
- In the case of a Hexapod without permanently installed cables: cable set of the Hexapod system
- When the C-887 is to be supplied with commands via a PC or a network:
  - PC with Windows (XP, Vista, 7) or Linux operating system
  - Communication interface to the PC:
    free COM port on the PC
    or
    Ethernet connection in the PC and, if necessary, a free access point in the network to which the PC is connected
  - RS-232 or network cable to connect the controller with the PC or with the network
  - C-887 CD with PC software
- When the user interface of the C-887 is to be worked with:
  - Keyboard with USB connection, included in the scope of delivery (p. 24)
  - Monitor with VGA connection

13.3 Dimensions

13.3.1 C-887.11

Dimensions in mm. Note that the decimal places are separated by a comma in the drawings.

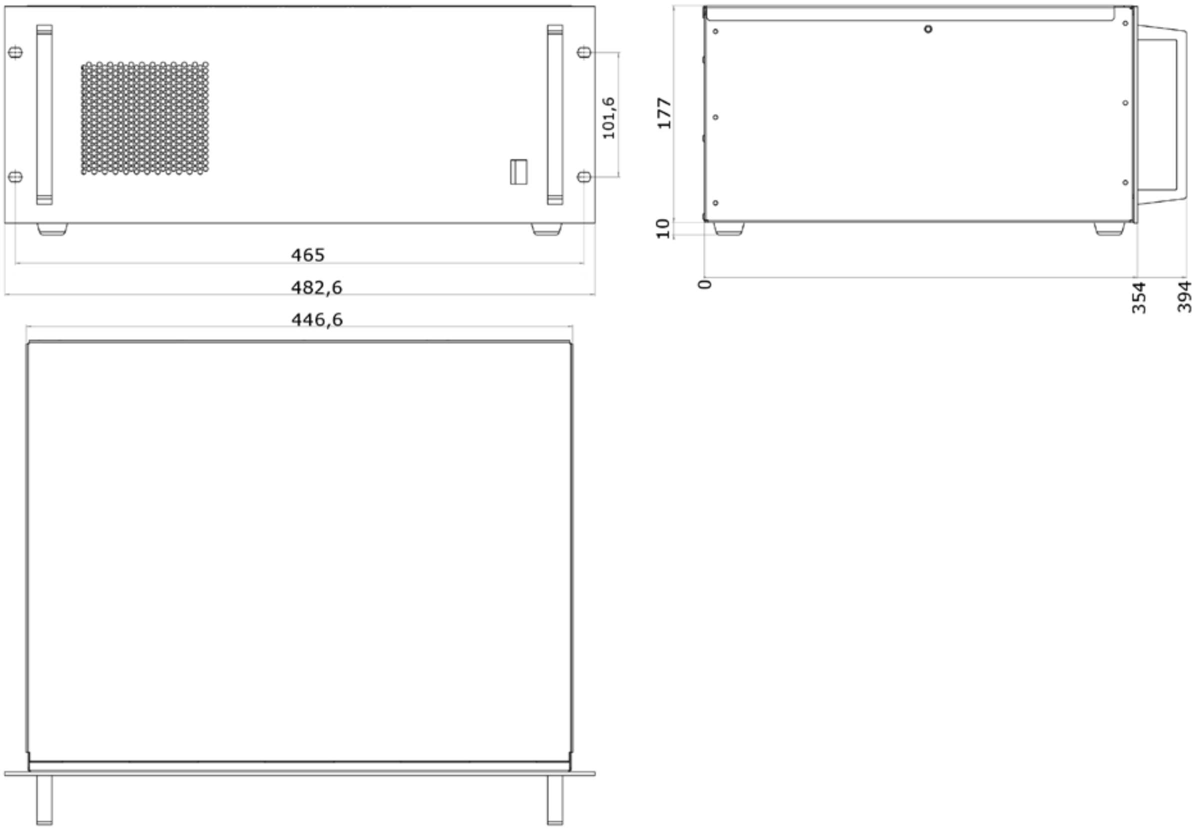


Figure 19: C-887.11

13.3.2 C-887.21

Dimensions in mm. Note that the decimal places are separated by a comma in the drawings.

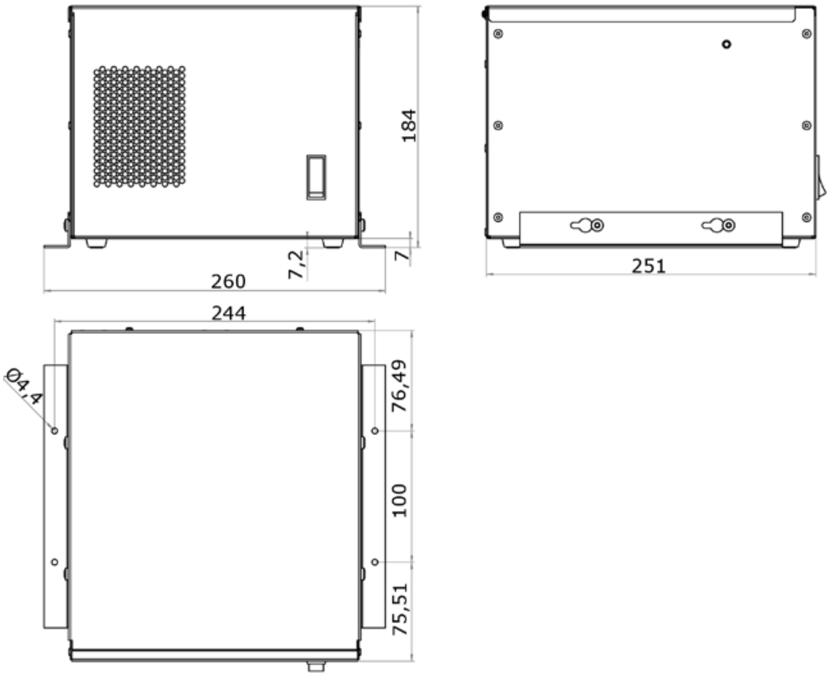


Figure 20: C-887.21

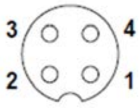
13.4 Pin Assignment

13.4.1 2x 24 VDC Power Out

The two **2x 24 VDC Power Out** connections on the rear panel of the C-887 can be used for the following purposes with any allocation:

- Power source of the Hexapod
- Only when using a cable set with line driver boxes: power source for line driver box; for further information, see the user manual of the Hexapod

A-coded, 4-pin M12 socket, female

Pin	Function	
1	GND	
2	GND	
3	Output: 24 V DC	
4	Output: 24 V DC	

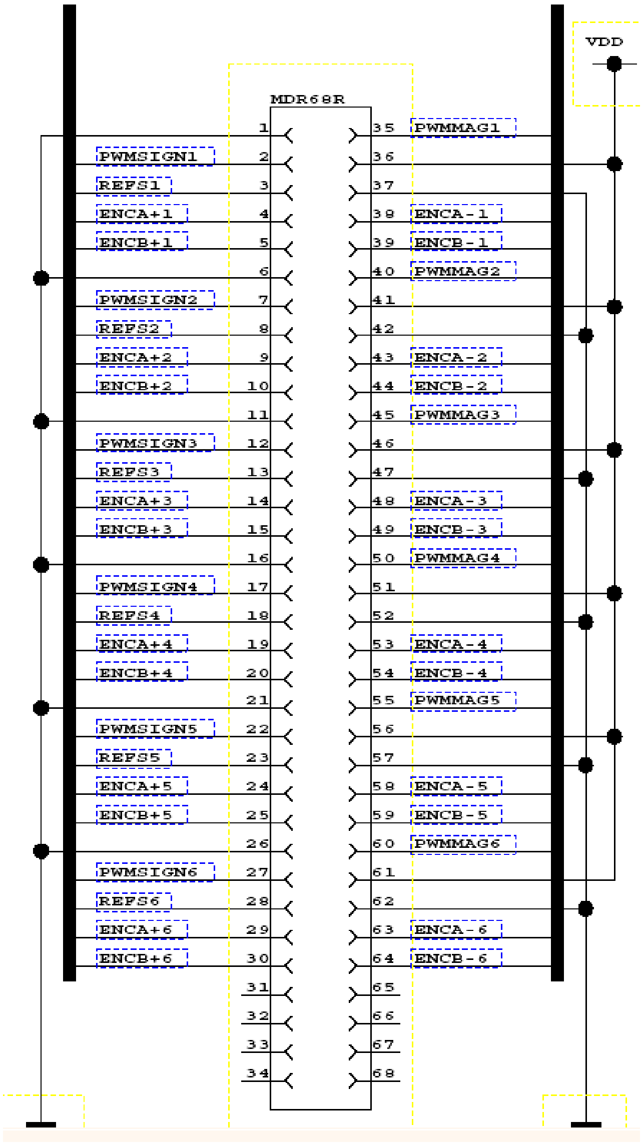
13.4.2 Hexapod

Data transmission between the Hexapod and the Hexapod controller

MDR68 socket

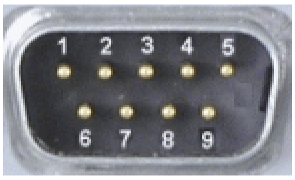
Function	Socket
All signals: TTL	

Pin assignment



13.4.3 RS-232

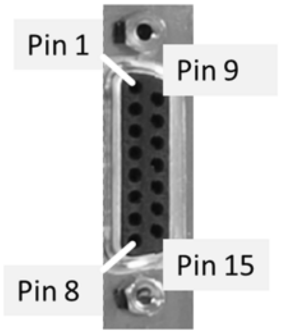
9-pin sub-D panel plug, male

Pin	Function	
1	Not connected	
2	RxD, data input	
3	TxD, data output	
4	Not connected	
5	DGND (digital mass)	
6	Not connected	
7	RTS, hardware handshake output	
8	CTS, hardware handshake input	
9	Not connected	

13.4.4 A and B - only C-887.11

The rear panel of the C-887.11 has **A** and **B** connections for optionally available stages with DC motors (p. 25).

15-pin sub-D socket, female

Pin	Function	
1	Output: +5 V for motor brake	
2	Reserved	
3	Output: MAGN (motor PWM, TTL signal)	
4	Output: +5 V, for encoder	
5	Reserved	
6	Reserved	
7	Input: encoder: A (-)	
8	Input: encoder: B (-)	
9	Reserved	
10	PWM GND	
11	Output: SIGN (direction of rotation of the motor, TTL signal)	
12	Reserved	
13	Input: REFS (reference switch, TTL signal)	
14	Input: encoder: A (+) / ENCA	
15	Input: encoder: B (+) / ENCB	

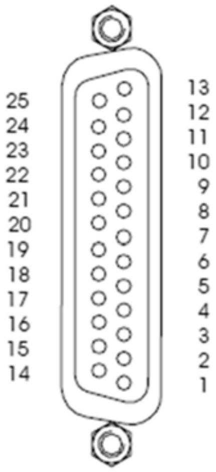
13.4.5 Nanocube - only C-887.11

If the E-760 controller card of the F-206.NCU option is installed in the C-887.11 (p. 45), the C-887.11 will have the following connections on the rear panel:

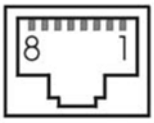
- Sub-D socket 25-pin, female for the connection of the P-611.3SF NanoCube®
- RJ45 socket for the connection of an analog input signal

The E-760 controller card is labeled with **Nanocube**.

25-pin sub-D socket, female

Pin	Function	
1	Input: B31	
2	Input: REF3	
3	Input: B21	
4	Input: REF2	
5	Input: B11	
6	Input: REF1	
7	Not connected	
8	Not connected	
9	Not connected	
10	AGND	
11	Output: PZT3 (-20 to 120 V)	
12	Output: PZT2 (-20 to 120 V)	
13	Output: PZT1 (-20 to 120 V)	
14	Input: B32	
15	AGND	
16	Input: B22	
17	AGND	
18	Input: B12	
19	AGND	
20	Not connected	
21	Not connected	
22	Reserved	
23	AGND	
24	AGND	
25	AGND	

RJ45 socket for the connection of an analog input signal

Pin	Function	
1	GND	
2	Reserved	
3	Reserved	
4	Reserved	
5	Reserved	
6	Reserved	
7	Analog input, ±5 V, resolution: 14 bit	
8	AGND	

14 Old Equipment Disposal

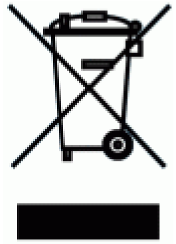
Since 13 August 2005, in accordance with the EU directive 2002/96/EC (WEEE), electrical and electronic equipment can no longer be disposed of in the member states of the EU with other wastes.

When disposing of your old equipment, observe the international, national and local rules and regulations.

To meet the manufacturer's product responsibility with regard to this product, Physik Instrumente (PI) GmbH & Co. KG ensures environmentally correct disposal of old PI equipment that was first put into circulation after 13 August 2005, free of charge.

If you have old PI equipment, you can send it postage-free to the following address:

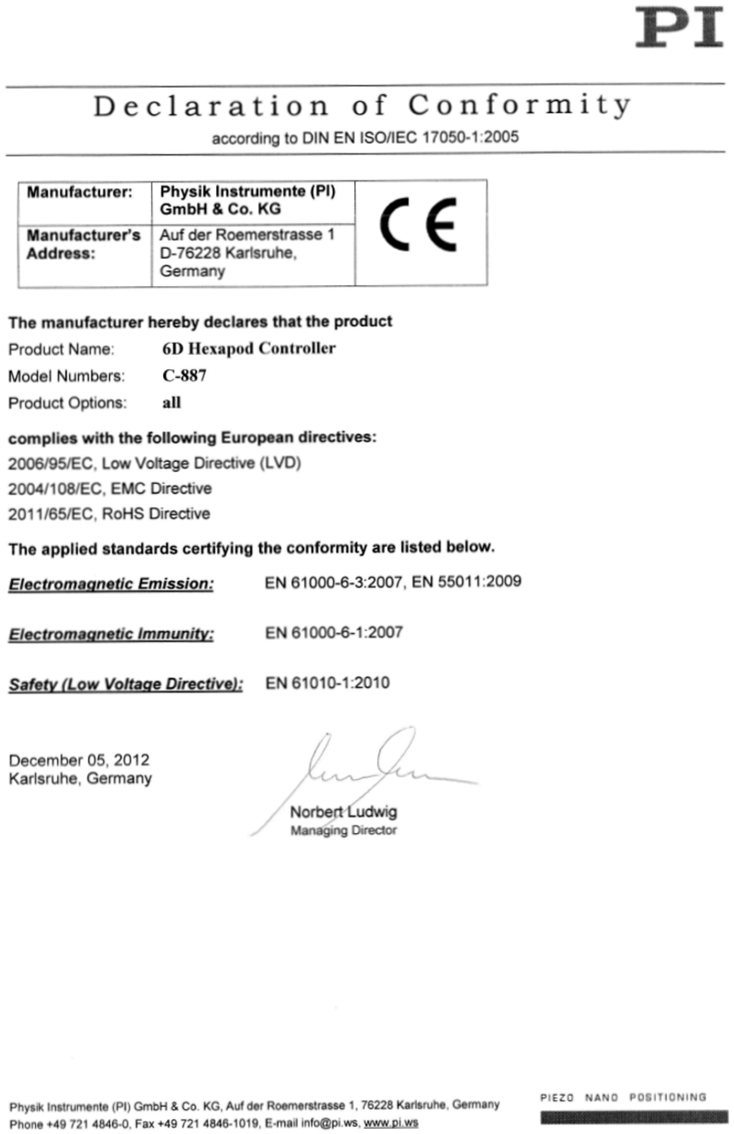
Physik Instrumente (PI) GmbH & Co. KG
Auf der Römerstr. 1
D-76228 Karlsruhe, Germany



15 Appendix

In this Chapter

15.1 EC Declaration of Conformity



PI

Declaration of Conformity

according to DIN EN ISO/IEC 17050-1:2005

Manufacturer:	**Physik Instrumente (PI) GmbH & Co. KG**	CE
Manufacturer's Address:	Auf der Roemerstrasse 1 D-76228 Karlsruhe, Germany	

The manufacturer hereby declares that the product

Product Name:	**6D Hexapod Controller**
Model Numbers:	**C-887**
Product Options:	**all**

complies with the following European directives:

2006/95/EC, Low Voltage Directive (LVD)
2004/108/EC, EMC Directive
2011/65/EC, RoHS Directive

The applied standards certifying the conformity are listed below.

Electromagnetic Emission:	EN 61000-6-3:2007, EN 55011:2009
Electromagnetic Immunity:	EN 61000-6-1:2007
Safety (Low Voltage Directive):	EN 61010-1:2010

December 05, 2012
Karlsruhe, Germany

Norbert Ludwig
Managing Director

Physik Instrumente (PI) GmbH & Co. KG, Auf der Roemerstrasse 1, 76228 Karlsruhe, Germany
Phone +49 721 4846-0, Fax +49 721 4846-1019, E-mail info@pi.ws, www.pi.ws

PIEZO NANO POSITIONING

15.2 GNU General Public License

The software included in this product contains copyrighted software that is licensed under the GNU General Public License (the GPL), or the GNU Lesser General Public License (the LGPL).

For more information about the GPL, you can access the GPL website (http://www.gnu.org/licenses/gpl-3.0.en.html) and download a copy of the GPL (http://www.gnu.org/licenses/gpl.txt).

For more information about the LGPL, you can access the GPL website (http://www.gnu.org/licenses/gpl-3.0.en.html) and download a copy of the LGPL (http://www.gnu.org/licenses/lgpl.txt).

The GPL'ed source code used falls into a number of categories:

1. The compiler toolchain used to build all the software
2. The Linux kernel, with modifications by PI
3. Other third party software (under GPL or LGPL)