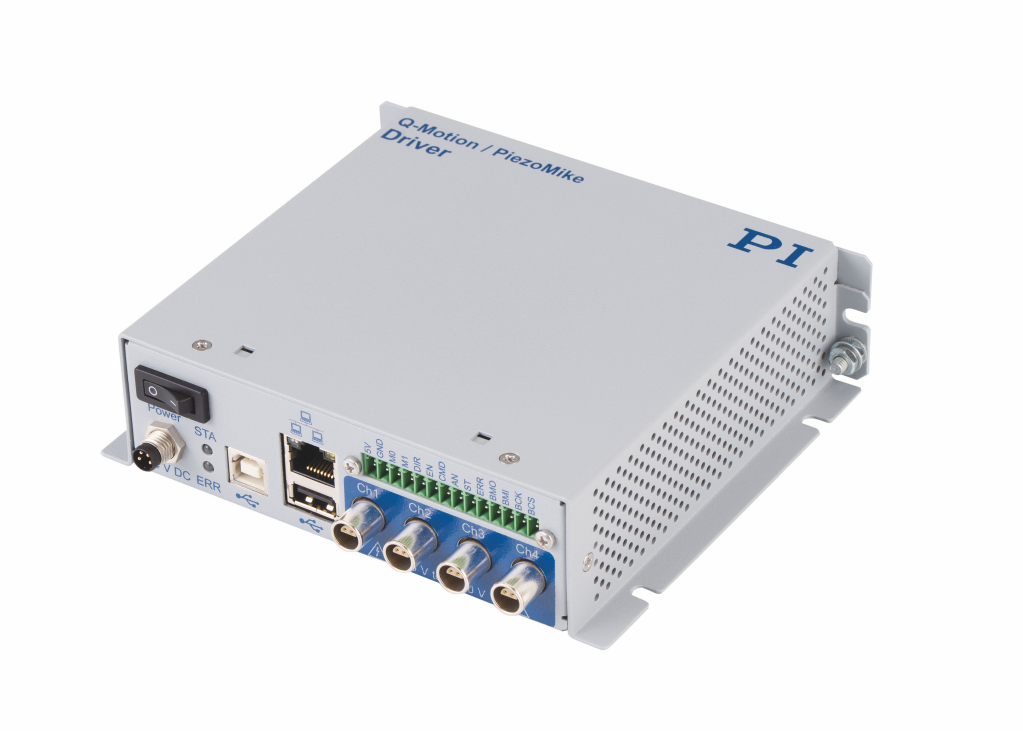


User Manual

E-872.401 DRIVER FOR PIEZO INERTIA DRIVES

Contents

1 Legal Information

Original instructions
First print: 9/16/2020
Document number: PZ279en, ASt

Subject to change. This manual is superseded by any new release. The latest versions of the user manuals are available for download (p. 10) at www.pi.ws.

Publisher:
Physik Instrumente (PI) GmbH & Co. KG
Auf der Roemerstrasse 1
76228 Karlsruhe
Germany
info@pi.de
www.pi.de

Customer service department:
Physik Instrumente (PI) GmbH & Co. KG
Auf der Roemerstrasse 1
76228 Karlsruhe
Germany
service@pi.de
www.pi.de

2 About this Document

2.1 Objective and Target Group

This user manual contains the information needed for the intended use of the E-872.401.

Basic knowledge of closed-loop systems, motion control concepts, and applicable safety measures is assumed.

2.2 Other Applicable Documents

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

Document number	Document type	Product
SM148E	Software Manual	PIMikroMove
SM146E	Software Manual	GCS Array Data Format Description
SM151E	Software Manual	PI GCS 2.0 DLL
SM155E	Software Manual	PI MATLAB Driver GCS 2.0
A000T0028	User Manual	PIUpdateFinder: Updating PI Software
A000T0081	Technical Note	Downloading Manuals from PI: PDF file with links to the manuals for digital electronics and software from PI. Is on the PI software CD.

The latest versions of the user manuals can be downloaded (p. 10) at www.pi.ws.

2.3 Explanation of Symbols

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

2.3.1 Typographic Conventions

Symbol / label	Meaning
1. 2.	Action consisting of one or several steps with strict sequential order
►	Action consisting of one or more steps without relevant sequential order
■	Lists
p. 5	Cross-reference to page 5
RS-232	Label on the product indicating an operating element (example: RS-232 interface socket)
Start > Settings	Menu path in the PC software (example: to open the menu, ***Start*** and ***Settings*** must be clicked successively)
POS?	Command line or a command from PI's General Command Set (GCS) (example: command to get the axis position)
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

2.3.2 Symbols Used

Symbol / Label	Meaning
	General hazard symbol
	Electrical voltage

CAUTION

Dangerous situation

Failure to comply could lead to minor injury.

► Precautionary measures for avoiding the risk.

NOTICE

Dangerous situation

Failure to comply could lead to material damage.

► Precautionary measures for avoiding the risk.

Information

Additional information on the E-872.401 that can affect your application.

2.4 Figures

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

2.5 Downloading Manuals

Information

If a manual is missing or problems occur with downloading:

► Contact our customer service department (p. 128).

Downloading Manuals

1. Open the website **www.pi.ws**.
2. Search the website for the product number (e.g., E-872).
3. Click the corresponding product to open the product detail page.
4. Click the ***Downloads*** tab.

➔ *The manuals are shown under* ***Documentation****. Software manuals are shown under* ***General Software Documentation****.*

5. Click the desired manual and fill out the inquiry form.

➔ *The download link will then be sent to the email address entered.*

3 Safety

3.1 Intended Use

The E-872.401 is a laboratory device as defined by DIN EN 61010-1. It is intended for indoor use and use in an environment that is free of dirt, oil, and lubricants.

In accordance with its design, the E-872.401 is intended for operating positioners with piezo inertia drives (Q-Motion®, PiezoMike). The channels are controlled sequentially. The E-872.401 is intended for open-loop control.

The E-872.401 may not be used for purposes other than those stated in this user manual. The E-872.401 may only be used in compliance with the technical specifications and instructions in this user manual.

3.2 General Safety Instructions

The E-872.401 is built according to state-of-the-art technology and recognized safety standards. Improper use of the E-872.401 may result in personal injury and/or damage to the E-872.401.

- ► Use the E-872.401 only for its intended purpose and if it is in perfect condition.
- ► Read the user manual.
- ► Eliminate any malfunctions that may affect safety immediately.

The operator is responsible for installing and operating the E-872.401 correctly.

- ► Install the E-872.401 near the power adapter so that the power plug can be quickly and easily disconnected from the mains.
- ► Use the components supplied to connect the E-872.401 to the power supply.
- ► If one of the supplied components for connecting to the power supply has to be replaced, use a sufficiently dimensioned component.
- ► Only use cables and connections that comply with local safety regulations.

3.3 Organizational Measures

3.3.1 User Manual

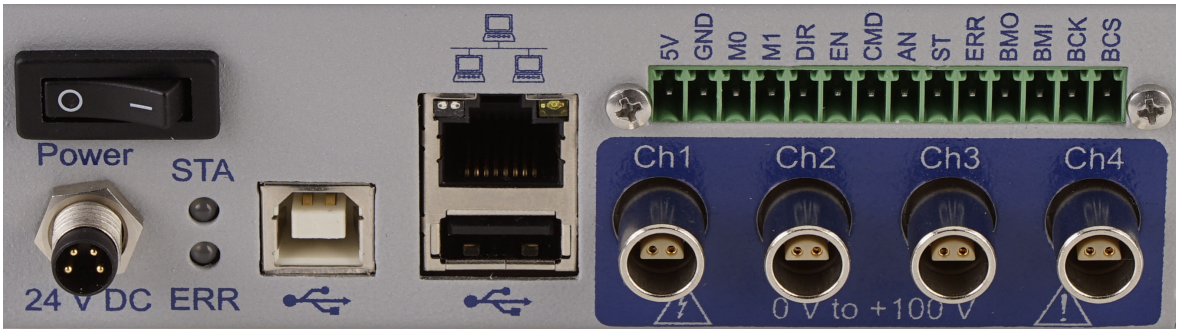
- ► Always keep this user manual available with the E-872.401. The latest versions of the user manuals can be downloaded (p. 10) at www.pi.ws.
- ► Add all information from the manufacturer such as supplements or technical notes to the user manual.
- ► If you give the E-872.401 to a third party, also include this user manual as well as other relevant information provided by the manufacturer.
- ► Only use the device on the basis of the complete user manual. Missing information due to an incomplete user manual can result in minor injury and damage to equipment.
- ► Only install and operate the E-872.401 after you have read and understood this user manual.

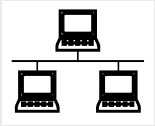
3.3.2 General Personnel Qualification

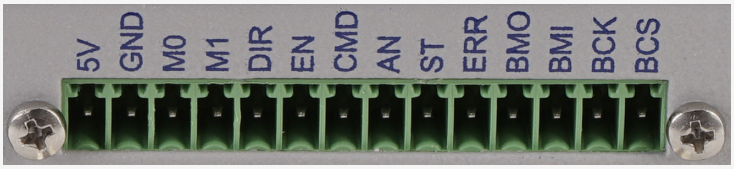
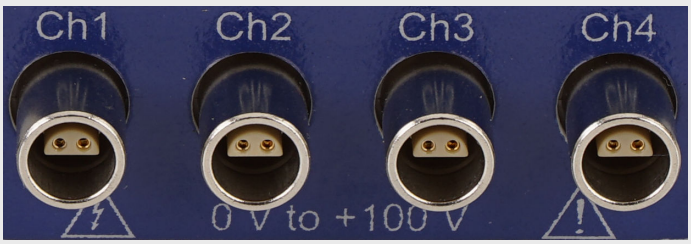
The E-872.401 may only be installed, started up, operated, maintained, and cleaned by authorized and appropriately qualified personnel.

4 Product Description

4.1 Front Panel



Element	Labeling	Type	Function
	Power	Toggle switch	On/Off switch O: E-872.401 switched off —: E-872.401 switched on
	24 V DC	M8 connector, 4-pole (m)	Connector for the supply voltage (p. 132) On delivery, a protective cap is screwed on the connector: 
	STA	LED, green	Device status: Green - lights up continuously: E-872.401 ready for normal operation Green - flashing: E-872.401 in firmware update mode Off: E-872.401 not connected to the supply voltage
	ERR	LED red	Error indicator: On: Error (error code not equal to 0) Off: No error (error code = 0) The error code can be queried with the ERR? command. The query resets the error code to zero and the LED is switched off.
		USB type B	Universal serial bus for connecting to the PC
		RJ45 socket	Ethernet interface for communication via TCP/IP
		USB type A	Connector for digital HID (Human Interface Device), e.g., joystick

Element	Function
	Phoenix Contact MC 1,5/14-GF-3,5-LR I/O connector (p. 131)
	LEMO EPL.0S.303.HLN connectors (p. 130) for positioner

4.2 Type Plate

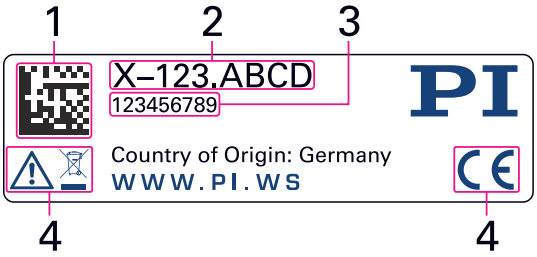


Figure 1: Type plate of the E-872.401

1. Data matrix code (example; contains the serial number)
2. Product number (example)
3. Serial number (example), individual for each E-872.401
   Meaning of the position (counting from the left):
   1 = internal information,
   2 and 3 = year of manufacture,
   4 to 9 = consecutive numbers
4. Warning and conformity symbols (old equipment disposal (p. 129), CE mark (p. 160))

4.3 Scope of Delivery

Product number	Description
E-872.401	Drive electronics according to the order
C-501.24050H	Wide range input power supply 24 V, 50 W
K050B0003	Power supply adapter, barrel connector to M8 (f)
3763	Power cord
MS242EK	Short instructions for digital motor controllers and drivers
000011448	USB cable (type A to B) for connecting to the PC
C-990.CD1	Data storage medium with PC software from PI

4.4 Accessories

The following articles are not in the scope of delivery of the E-872.401 and must be ordered separately, if required.

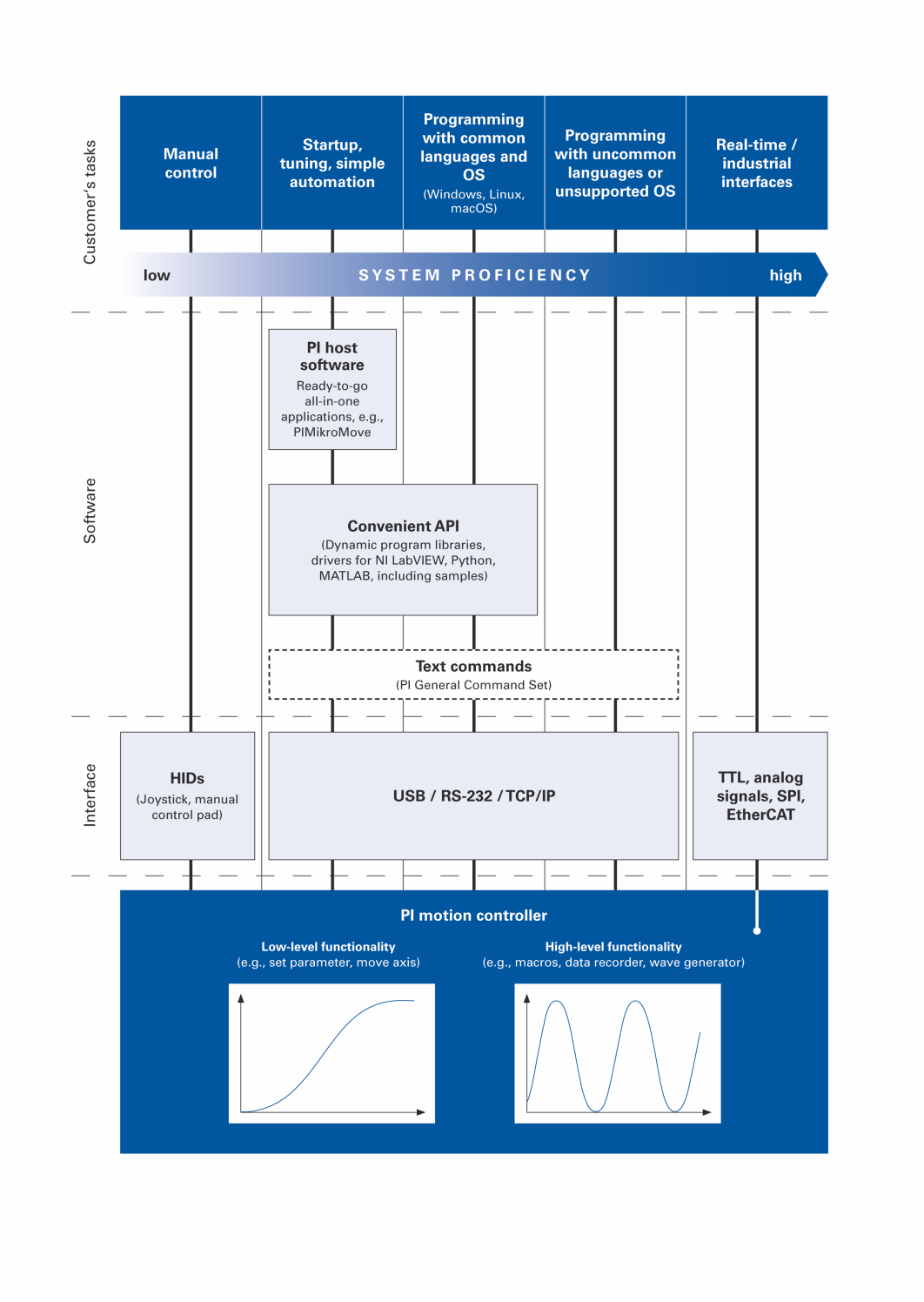
Product number	Description
C-815.553	Straight-through network cable for connecting the PC via a TCP/IP network
C-815.563	Crossover network cable for direct connection to the PC via TCP/IP

To order, contact our customer service department (p. 128).

4.5 Communication Interfaces

4.5.1 Controlling PI Systems

Basically, PI systems can be controlled as follows:



4.5.2 E-872.401 Interfaces

The E-872.401 can be controlled via the following communication interfaces:

- PC interfaces, via software (p. 17) or PI General Command Set:
  - TCP/IP
  - USB
- SPI (reserved for future extensions)
- Analog signals

- Digital Signals: TTL

The interface parameters in the E-872.401's volatile memory can be queried with the IFC? command and changed with IFC command. The IFS? and IFS commands are available for querying and changing the interface parameters in the E-872.401's nonvolatile memory.

The number of available digital input and output lines to and from the E-872.401 can be queried with the TIO? (p. 107) command. The status of the digital input lines can be queried with the DIO? (p. 85) command.

TCP/IP

Interface parameters of the E-872.401 for TCP/IP communication:

Interface property (parameter name)	Factory setting	Note
Default IP address (IPADR)	192.168.0.75	Is not used when an IP address is assigned to the E-872.401 by a DHCP server (IPSTART).
Port for communication with the E-872.401	:50000	Not editable.
Subnet mask (IPMASK)	255.255.255.0	
Start-up behavior for configuring the IP address for TCP/IP communication (IPSTART)	DHCP active	The IP address of the E-872.401 is assigned via DHCP by the default setting of the startup behavior. The default setting of the startup behavior must only be changed if the network devices are to use static addresses instead.
Auto MDI-X	No (MDI)	Not editable. If the E-872.401 is to be connected directly to the PC via TCP/IP, a crossover cable must be used.

4.6 Software Overview

The following table shows the PC software on the data storage device supplied. The specified operating systems stand for the following versions:

- Windows: Versionen 8.1, 10 (32 Bit, 64 Bit)
- Linux: Kernel 2.6, GTK 2.0, from glibc 2.15

Libraries, drivers

PC software	Operating system	Short description	Recommended use
Dynamic program-libraries for GCS	Windows,Linux	Allows software programming of the E-872.401 with programming languages such as C++. The functions in dynamic program libraries are based on the PI General Command Set (GCS).	For users who would like to use a dynamic program library for their application. Is required for PIMikroMove. Is required for NI LabVIEW drivers if communication is to be established via USB (with Linux only via virtual COM port) or a daisy chain network.
NI LabVIEW drivers	Windows,Linux	NI LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments). The NI LabVIEW software from PI is a collection of virtual instrument drivers (VI drivers) for PI controllers. These drivers support GCS commands.	For users who wish to use NI LabVIEW to program their application.
MATLAB drivers	Windows	MATLAB is a development environment and programming language for numerical calculations (must be ordered separately from MathWorks). The PIMATLAB driver consists of a MATLAB class that can be included in any MATLAB script. This class supports the PI General Command Set. The PI MATLAB driver does not require any additional MATLAB toolboxes.	For users who wish to use MATLAB to program their application.
USB driver	Windows	Driver for the USB interface	For users who want to connect the controller to the PC via the USB interface.

User software

PC software	Operating system	Short description	Recommended use
PIMikroMove	Windows	Graphical user interface for Windows, which can be used for controllers from PI: ■ Start the system without programming effort ■ Graphic representation of the motion ■ Macro functionality for storing command sequences on the PC (host macros) ■ Complete environment for command entry 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. No command knowledge is necessary to operate PIMikroMove. A log window showing the commands sent makes it possible to learn how to use the commands.
NI LabVIEW Merge Tool	Windows	The NI LabVIEW Merge Tool allows you to combine product-specific NI LabVIEW drivers from PI with each other.	For users who want to operate several products from PI at the same time while using NI LabVIEW.
PIStages3Editor	Windows	Program opening and editing positioner databases in .db format.	For users who want to deal with the contents of positioner databases more intensively.
PITerminal	Windows,Linux	Simple user interface that can be used for nearly all PI controllers.	For users who want to send GCS commands directly to the controller.
PI Firmware Updater	Windows	Program for updating the firmware of the E-872.401.	For users who want to update the firmware.
PIUpdateFinder	Windows	Checks the PI software installed on the PC. If newer versions of the PC software are available on the PI server, they are offered for download.	For users who want to update the PC software.

4.7 Positioner database

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

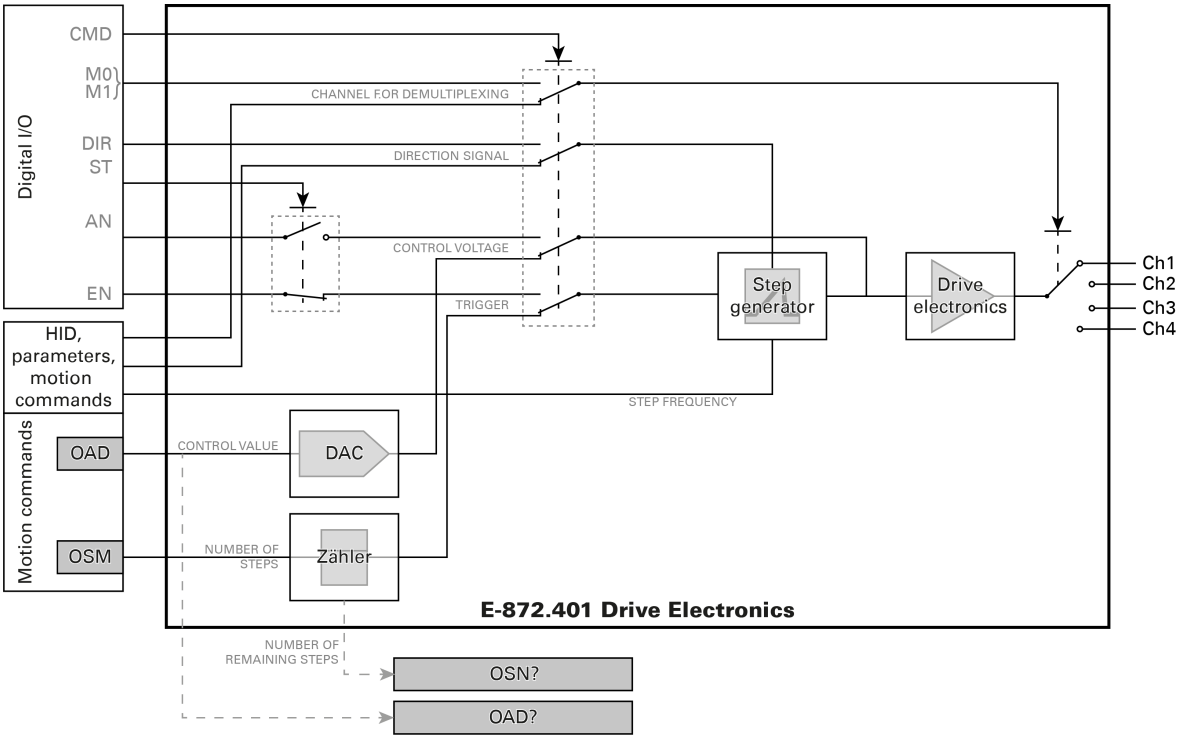
Database file name	Description
PISTAGES3.DB	Delivery includes parameter sets for all standard positioners from PI and PI miCos, and is saved to the PC automatically during installation of the PC software. New parameter sets can be created, edited, and saved.
Product code.db	Includes the parameter set for the custom positioner "product code". In order for the parameter set to be selectable in the PC software, it must be imported (p. 35) into PISTAGES3.DB first.

The positioner database only contains some of the information that is required to operate a positioner with the E-872.401. When the positioner connected to the E-872.401 is equipped with an ID chip: Further information is loaded as parameter values from the ID chip to the volatile memory of the E-872.401 when the E-872.401 is switched on or rebooted.

4.8 Functional Principles of the E-872.401

4.8.1 Block Diagram

The E-872.401 controls the motion of a positioner's logical axis. The following block diagram shows how the E-872.401 generates the piezo voltage for the axis connected:



The E-872.401 supports positioners with Q-Motion® or PiezoMike inertia drive.

4.8.2 Important Firmware Components

The functional units of the E-872.401's firmware are described in the following.

The firmware can be updated with a tool. The current firmware version can be ascertained with the VER? command.

Component	Description
Parameters	Parameters reflect the properties of the positioner connected (e.g., travel range) and specify the behavior of the E-872.401 (e.g., settings for the servo algorithm). The parameters can be divided into the following categories: ■ Protected parameters whose default settings cannot be changed ■ Parameters that must be set by the user to adapt to the application In the case of positioners with ID chip, the values of some parameters are stored on the ID chip. They are loaded to the volatile memory when switching on or rebooting the E-872.401. Command levels determine the write permission for the parameters. The current command level can be queried with the CCL? command and changed with the CCL command. This may require entering a password. The list of parameters available in the E-872.401 can be queried with the HPA? command. Refer to Adapting Settings (p. 69) for more information on parameters
ASCII commands (GCS)	Communication with the E-872.401 can be made with commands (p. 79) from the PI General Command Set (GCS). The GCS syntax version (p. 67) can be queries with the CSV? command. Examples of the use of GCS: ■ Configuring the E-872.401 ■ Setting the operating mode ■ Starting positioner motion ■ Getting system and position values The list of commands available in the E-872.401 can be queried with the HLP? command.
Macros	The E-872.401 can save macros. Command sequences can be defined and stored permanently in the nonvolatile memory of the device via the macro function. A startup macro can be defined that runs each time the E-872.401 is switched on or rebooted. The startup macro simplifies stand-alone operation (operation without a connection to the PC). Refer to Controller Macros (p. 61) for information

Commands

C		Page
CCL	Set Command Level	83
CCL?	Get Command Level	83
CSV?	Get Current Syntax Version	84
H		**Page**
HLP?	Get List Of Available Commands	91
HPA?	Get List Of Available Parameters	92

V		Page
VER?	Get Versions Of Firmware And Drivers	108

Parameters

4.8.3 Commandable Items

The following table contains the elements of the E-872.401 that can be commanded with GCS commands.

Element	Quantity	ID	Description
Drive channels	4	1 to 4	The E-872.401 supports drive channels for positioners with piezo inertia drive. The E-872.401 controls the axis of a positioner connected to it via a drive channel. One axis each is driven by the integrated amplifier. Switching between the channels is done via commands or via the digital input lines. This allows serial control of up to 4 axes with a piezo inertia drive. For information, see: Switching between the Drive Channels (p. 28) Depending on the command mode set, it is possible to command motion in step mode with the OSM command or via HID, or via digital input lines. Motion in linear mode can only be commanded via an analog input signal. For information, see: Triggering Motion (p. 26) The drive channel's identifier can be queried with the SAI? command and the connected positioner type with the CST? command. If the **Stage Name** parameter (0x3C) has the NOSTAGE value, the connected axis is "deactivated". A deactivated axis is not accessible for axis-related commands (e.g., motion commands or position queries). The identifier of a deactivated axis can only be queried with SAI? ALL.
Analog input	1	1	The input signal for analog signals (0 to 4.8 V) is at the E-872.401's I/O socket. The analog input can be used for controlling the connected piezo inertia drive in linear mode. For information, see: Analog Input Signals (p. 54)
Digital inputs	6	1 to 6	1 to 6 identify digital input lines 1 to 6 of the *I/O* socket. For information, see: Digital Input Signals (p. 53)
Digital outputs	1	1	1 identifies digital output line 1 of the *I/O* socket. For information, see: Digital Output Signals (p. 53)
HID (Human Interface Device)	1	1	A digital HID can be connected to the USB socket of the E-872.401. The HID (e.g.,. joystick) is used for HID control of the axes connected to the E-872.401. Information on the HID's axes and buttons can be queried with the HIS? command. For information, see: Controlling with an HID (p. 55)
HID axes	x	1 to x	The number of commandable axes per HID depends on the HID connected. The E-872.401 assigns IDs 1 to x to the HID axes.
HID buttons	x	1 to x	The number of commandable buttons per HID depends on the HID connected.

Element	Quantity	ID	Description
			The E-872.401 assigns IDs 1 to x to the HID buttons.
Overall system	1	1	E-872.401 as an overall system Information on name, serial number, and firmware version of the E-872.401 can be queried with the *IDN? command. The ready state of the E-872.401 can be queried with the #7 command.

Commands

#		Page
#7	Request Controller Ready Status	81
*		Page
***IDN?**	Get Device Information	82
C		Page
CST?	Get Assignment Of Stages To Axes	84
H		Page
HIS?	Get Configuration Of HI Device	90
O		Page
OSM	Open-Loop Step Moving	103
S		Page
SAI?	Get List Of Current Axis Identifiers	104

Parameters

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

4.8.4 Operating Modes

The E-872.401 is a driver for piezo inertia drives without position sensor, i.e., motion is done in open-loop operation.

A distinction between the following modes is made in drive mode:

- Step mode
- Linear mode

Drive mode

Piezo inertia drives are piezo-based drives with virtually unlimited travel range. They use the stick-slip effect (inertia effect) – a cyclical alternation of static and sliding friction between a moving runner and the piezo actuator generated by the piezo element – for a continuous feed of the runner. The operating voltage is therefore output by the E-872.401 as a modified sawtooth signal with a maximum frequency of 25 kHz. The output of one period of the modified sawtooth signal generates one "step" of the runner.

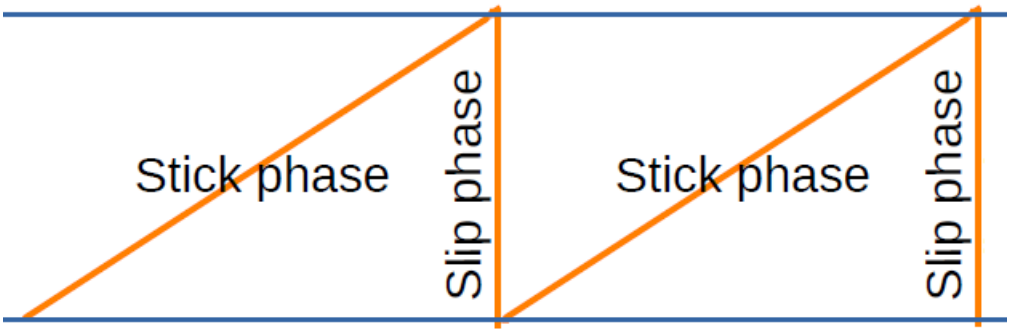


Figure 2: Stick-slip principle of a piezo inertia drive

In the case of a piezo inertia drive, a piezo actuator acts on a moving runner to generate motion. The E-872.401 supports the following drive modes for piezo inertia drives in open-loop operation:

Step mode
The drive electronics of the E-872.401 convert the control value to a modified sawtooth signal with a maximum frequency of 25 kHz and output the corresponding piezo voltage. The piezo voltage generates a cyclic alternation of static and sliding friction between the runner and the piezo actuator and therefore continuous feed of the runner. The output of one period of the modified sawtooth signal generates one "step" for the runner.

The travel range is only limited by the physical limits of the positioner.

Linear mode
The drive electronics of the E-872.401 convert the control value to an analog signal. The output piezo voltage corresponds to 10 times this analog signal. The feed to the runner is generated by the expansion of the piezo actuator caused by the piezo voltage. The piezo actuator achieves its maximum expansion when the E-872.401 outputs the maximum permitted piezo voltage.

The travel range is limited by the maximum expansion of the piezo actuator.

Configuration

The drive mode is selected via the digital input line on pin 9 (ST) of the E-872.401'S I/O socket:

- High: Linear mode
- Low: Step mode

The E-872.401's drive channels are configured for the piezo inertia drive with the following parameters:

- PIShift Upper Supply Voltage (V) (0x1F000000)
- PIShift Lower Supply Voltage (V) (0x1F000100)
- PIShift Forward Current (A) (0x1F000200)
- PIShift Backward Current (A) (0x1F000300)
- PIShift Frequency (Hz) (0x1F000400)
- PIShift Charge Cycle (0x1F000500)
- PIShift Delay (ms) (0x1F000701)

Commands

Parameters

0x1F000000	PIShift Upper Supply Voltage (V)	Maximum output voltage for piezo inertia drives. The value depends on the type of the drive.
0x1F000100	PIShift Lower Supply Voltage (V)	Minimum output voltage for piezo inertia drives. The value depends on the type of the drive.
0x1F000200	PIShift Forward Current (A)	Maximum output current for piezo inertia drives during forward motion. The value depends on the type of the drive.
0x1F000300	PIShift Backward Current (A)	Maximum output current for piezo inertia drives during backward motion. The value depends on the type of the drive.
0x1F000400	PIShift Frequency (Hz)	Frequency of the piezo voltage for open-loop operation of piezo inertia drives. Determines the velocity of a drive in open-loop operation. This parameter must not exceed the value of the parameter 0x9 (Maximum Motor Output).
0x1F000500	PIShift Charge Cycle	Duty cycle of the current source during output of a step. Specified as part of a period which the current source is switched on for. 0 to 1 The value depends on the type of the drive.
0x1F000701	PIShift Delay (ms)	Delay time when switching between two operating modes (e.g., step mode and linear mode). 0 to 2000 [ms]

4.8.5 Triggering Motion

The E-872.401 is a driver for piezo inertia drives without position sensor, i.e., motion is done in open-loop operation.

The piezo inertia drive executes motion in step mode or in linear mode.

Step mode

Depending on the command mode, motion is triggered in step mode via digital input lines or via commands or via HID.

The command mode is selected via the digital line on pin 7 (CMD) of the E-872.401's I/O socket:

- High: Commanding via digital I/O lines
- Low: Commanding via interface: Commands from PC, HID control

Configuring via interface overwrites the settings of the digital inputs and ignores them until the E-872.401 is rebooted.

Linear mode

Motion in linear mode can only be commanded via an analog input signal. The analog input voltage is connected to pin 8 (AN) of the E-872.401's I/O socket. The permissible voltage range amounts to a tenth of the piezo inertia drive's permissible operating voltage range.

Example:

Permissible operating voltage range of the piezo inertia drive: 0 to 48 V

Permissible range of analog input voltage: 0 to 4.8 V

Triggering motion in open-loop operation

Commanding via input lines

Motion in step mode can be commanded via the digital inputs at pins 5 (DIR) and 6 (EN) of the E-872.401's I/O socket:

The direction signal for step mode can be fed via pin 5:

- High: Forward motion
- Low: Backward motion

The drive is activated in step mode via pin 6:

- High: Steps are done; the velocity (i.e., the current step frequency) is specified by the **PIShift Frequency (Hz)** parameter (0x1F000400) and remains constant.
- Low: No motion

Commanding via interface

Motion is triggered in step mode either via commands or via an HID, e.g., a joystick.

Motion commands for an axis are not permitted when HID control is activated for the axis.

Options for commanding via interface:

- Software on the PC (connected via one of the E-872.401's communications interfaces): The number of steps to be made and the direction can be specified in PIMikroMove; the velocity (i.e., the current step frequency) is specified by the **PIShift Frequency (Hz)** parameter and remains constant.
- Commands (via PC or controller macro): Sending the OSM command specifies the number of steps to be made and the direction; the velocity (i.e., the current step frequency) is specified by the **PIShift Frequency (Hz)** parameter and remains constant.
  Example: Sending `OSM 1 200` starts 200 steps forwards, `OSM 1 -550` starts 550 steps backwards.
- HID, e.g., joystick (connected via USB): Displacement of the HID's axis determines the velocity (i.e., the current step frequency) and the direction.

The motion status of the axes connected to the E-872.401 can be queried with the #5 command.

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

- #24, STP: Abrupt stop
- HLT: Gentle stop

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

Motion is triggered by the following commands:

Commands	Description
OSM	Moves an axis by a particular number of steps The number of steps that the axis still must do can be queried with the OSN? command.
OAD	Determines the control value for the specified axis of the E-872.401 and starts motion immediately. In the case of piezo inertia drives, the control value corresponds to the analog output voltage [V]. The current control value can be queried with the OAD? command.

The HID control is configured with and activated by the following commands:

Commands	Description
HIN	Activates or deactivates control of the E-872.401's channels via the HID's axes
HIA	Configures HID control for the E-872.401's channels. HID axes can control the following motion parameters of axes connected to the E-872.401: ■ Velocity of the axis connected to the channel

Commands

#		Page
#24	Stop All Axes	82
#5	Request Motion Status	81
H		**Page**
HIA	Configure Control Done By HID Axis	87
HIN	Set Activation State For HID Control	89
HLT	Halt Motion Smoothly	92
O		**Page**
OAD	Set Open-Loop Control Value (starts motion)	102
OAD?	Get Control Value	103
OSM	Open-Loop Step Moving	103
OSN?	Read Number Steps	103
S		**Page**
STP	Stop All Axes	106

Parameters

0x1F000400	PIShift Frequency (Hz)	Frequency of the piezo voltage for open-loop operation of piezo inertia drives. Determines the velocity of a drive in open-loop operation. This parameter must not exceed the value of the parameter 0x9 (Maximum Motor Output).

4.8.6 Switching Between the Drive Channels

The E-872.401 has 4 drive channels. One axis each is driven by the integrated amplifier of the E-872.401. Switching between the drive channels is done via commands or via the digital input lines. This allows serial control of up to 4 axes with a piezo inertia drive.

The drive channel can be selected in one of the following ways:

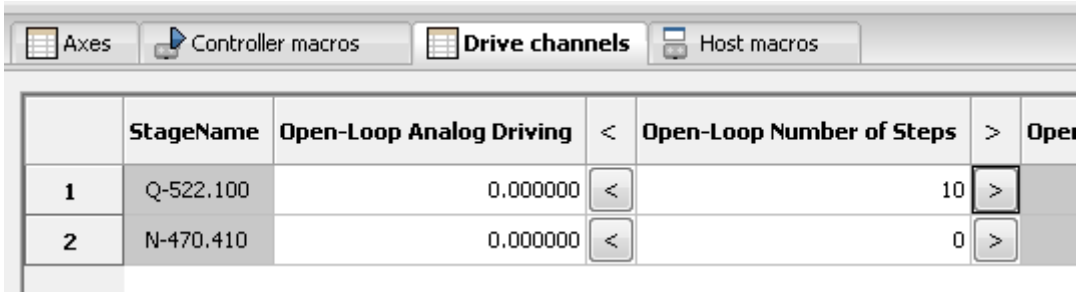
- Via interface: Sending commands or selecting in the PC program
- Via digital inputs: Pin 3 (M0) and pin 4 (M1) of the E-872.401's I/O socket

Configuring via interface overwrites the settings of the digital inputs and ignores them until the E-872.401 is rebooted.

Selecting the drive channel via communication interface

The drive channel is selected by

- Specifying the channel ID when sending commands, e.g., `OSM 1 10` moves the axis connected to drive channel 1 10 steps in a positive direction.
- Triggering motion for a drive channel in the main window of PIMikroMove



Selecting the drive channel via digital inputs

The drive channel is selected via the digital input lines on pins 3 (M0) and 4 (M1) of the E-872.401's I/O socket:

Function	Level of the Input Line	
Activating the drive channel	**M0**	**M1**
1	OFF / LOW	OFF / LOW
2	ON / HIGH	OFF / LOW
3	OFF / LOW	ON / HIGH
4	ON / HIGH	ON / HIGH

Commands

Parameters

4.8.7 Travel Range and Soft Limits

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

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

Settings for Soft Limits

The E-872.401 determines the soft limits using the Maximum Travel In Positive Direction (Phys. Unit) parameter (0x15) and Maximum Travel In Negative Direction (Phys. Unit) parameter (0x30):

- **Maximum Travel In Positive Direction (Phys. Unit)** (0x15) and **Maximum Travel In Negative Direction (Phys. Unit)** (0x30):
  - The limits determine the limits for the permissible travel range.
  - Motion commands are executed only if the commanded position is within these soft limits.
  - The limits always refer to the current zero position.
  - Appropriate values are loaded when the positioner is selected from the positioner database.

Commands

Parameters

5 Unpacking / Transportation

5.1 Unpacking

E-872.401 auspacken

1. Packen Sie den E-872.401 vorsichtig aus.
2. Wenn der E-872.401 mit Schutzkappen auf den Anschlüssen ausgeliefert wurde: Entfernen Sie die Schutzkappen **nicht**.
3. Vergleichen Sie die erhaltene Lieferung mit dem Lieferumfang laut Vertrag und mit dem Lieferschein.
4. Überprüfen Sie den Inhalt auf Anzeichen von Schäden. Bei Schäden oder fehlenden Teilen wenden Sie sich sofort an unseren Kundendienst (p. 128).
5. Bewahren Sie das komplette Verpackungsmaterial auf für den Fall, dass das Produkt zurückgeschickt werden muss.

5.2 Transportation

Preparing the E-872.401 for Transportation

1. Pay attention to the ambient conditions and classifications (p. 126).
2. Pack the E-872.401 in the original packaging.
3. If the E-872.401 is to be sent, use a stable outer box.

6 Installation

6.1 Mounting the E-872.401

The E-872.401 can be used as a benchtop device or mounted on a surface in any orientation or installed in a control cabinet.

Overview

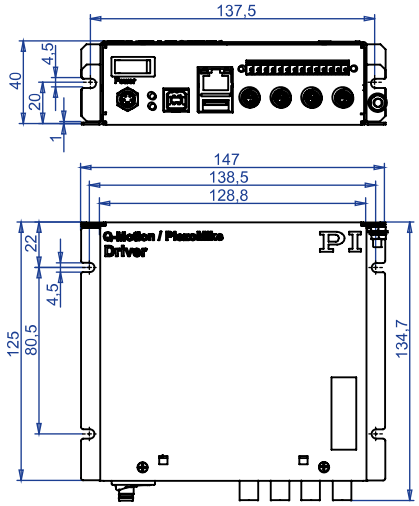


Figure 3: Dimensions and position of the recesses for mounting

Tools and Accessories

- Suitable screws
- Suitable screwdriver

Requirements

✓ You have read and understood the general safety instructions (p. 11).

NOTICE

Heating up of the E-872.401 during operation!
High temperatures can overheat the E-872.401.

► Install the E-872.401 with a gap of at least 10 cm to the top and rear panels and at least 5 cm to its sides. If this is not possible, make sure that the surroundings are cooled sufficiently.

► Ensure sufficient ventilation at the place of installation.

► Keep the ambient temperature at a noncritical level.

Mounting the E-872.401 on a Surface

1. Drill the holes required into the surface.
   The arrangement of the recesses in the mounting rails of the E-872.401 can be seen in the figure.
2. Tighten the E-872.401 to the recesses provided with suitable screws.

6.2 Connecting the E-872.401 to the Protective Earth Conductor

The E-872.401 is not grounded via the voltage connection and must therefore be connected to the protective earth conductor.

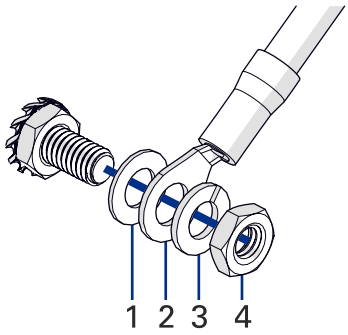


Figure 4: Connecting the protective earth conductor to the threaded bolt

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

Tools and Accessories

- Suitable protective earth conductor with cable lug:
  - Cable cross section ≥ 0.75 mm²
  - Contact resistance < 0.1 ohm at 25 A at all connection points relevant for attaching the protective earth conductor
- Suitable wrench

Requirements

✓ You have read and understood the general safety instructions (p. 11).

✓ The E-872.401 is **not** connected to the power supply.

Information

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

Connecting the E-872.401 to the Protective Earth Conductor

1. If necessary, attach a suitable cable lug to the protective earth conductor.
2. Remove the outer nut and the spring washer from the threaded bolt of the protective earth connector.
3. Push the cable lug of the protective earth conductor and the spring washer onto the threaded bolt.
4. Screw the nut onto the threaded bolt again.

➔ *The cable lug of the protective earth conductor is clamped between the flat washer and the spring washer.*

5. Tighten the nut with at least three turns and a torque of 1.2 Nm to 1.5 Nm.

6.3 Connecting the Power Supply to the E-872.401

Tools and Accessories

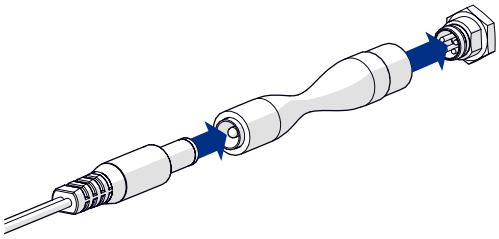
- Included power supply (alternative: Sufficiently rated power supply)
- Adapter included for power supply connection (alternative: Sufficiently rated adapter)
- Included power cord (alternative: Sufficiently rated power cord)

Requirements

✓ The power supply is **not** connected to the power socket via the power cord.
✓ The E-872.401 is installed near the power supply so that the power plug can be quickly and easily disconnected from the mains.

Connect the Power Adapter to the E-872.401

1. If necessary: Remove the protective cap from the E-872.401's voltage connector.
2. Connect the M8 4-pole connector (f) of the adapter to the voltage connector of the E-872.401.
3. Secure the adapter against unintentional removal.
4. Connect the barrel connector on the adapter to the barrel connector socket of the power adapter.



5. Connect the power cord to the power adapter.

6.4 Connecting the Positioner to the E-872.401

Tools and Accessories

- Compatible positioner with Q-Motion® or PiezoMike piezo inertia drive
- If necessary: Suitable adapter cable from PI, available as optional accessory for the positioner
- If the distance between the E-872.401 and the positioner is too long: Suitable drive or extension cable from PI, available as optional accessory for the positioner

Requirements

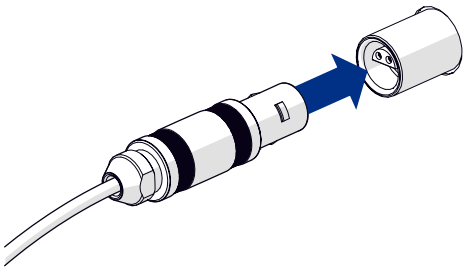
✓ The power adapter is **not** connected to the power socket via the power cord or the E-872.401 is switched off.
✓ You have read and understood the user manual for the positioner to be connected.
✓ You have installed the positioner to be connected according to the instructions in its respective user manual.

NOTICE

Damage if a wrong drive type is connected!
Connecting a positioner with incompatible drive type can cause irreparable damage.
► Connect the E-872.401 only to positioners with Q-Motion® or PiezoMike piezo inertia drive.

Connecting the Positioner to the E-872.401

1. If necessary: Connect a suitable adapter to the positioner's drive connector.
2. Insert the drive connector of the positioner or the adapter into one of the sockets on the drive of the E-872.401.



6.5 Installing the PC Software

6.5.1 Installing the PC Software for the First Time

Tools and Accessories

- PC with Windows or Linux (p. 17) operating system and at least 30 MB free storage space
- Software from PI: On the storage device supplied or available for download from our website www.pi.de.
- Optional for custom positioners: Data storage device or archive file with the following content:
  - *Import PI CustomStage* program
  - Custom positioner database with the parameter set for the positioner

Installing the PC software in Windows

1. Insert the data storage device into the PC or go to the directory where you saved the downloaded software to.
2. Start the install wizard by double-clicking **PISoftwareSuite.exe**.

➔ *The **InstallShield Wizard** window opens for installing the PC software from PI.*

3. Follow the instructions on the screen.
   The PI software suite includes the following components:
   - Driver for use with NI LabVIEW software
   - Dynamic program library for GCS
   - PIMikroMove
   - PC software for updating the firmware of the E-872.401
   - PIUpdateFinder for updating the PC software
   - USB driver

Installing a Custom Positioner Database in Windows

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

1. Insert the data storage device into the PC or go to the directory where you saved the downloaded software to.
2. Run the installation assistant for the custom positioner database by clicking ***Import_PI_CustomStage.exe***.

➔ *The **Import PI Custom Stage** program is run and the parameter set is imported from the custom positioner database into PIStages3.*

3. If a message appears that installation of the custom positioner database failed:
   a) Update the PIStages3 positioner database on your PC (p. 36).
   b) Repeat the installation of the custom positioner database.

Installing the PC Software in Linux

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

6.5.2 Updating the PC Software

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

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

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

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

Updating the PC Software and Positioner Database in Windows

Requirements

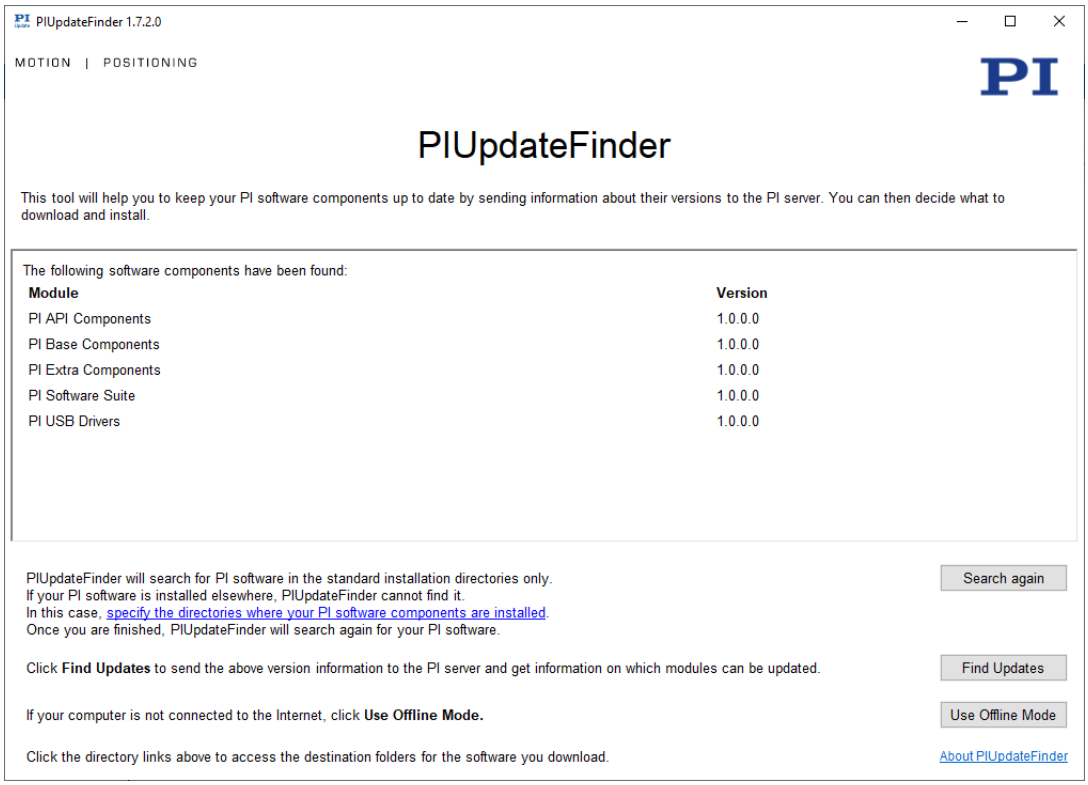
✓ You have installed the PIUpdateFinder on the PC to be updated.

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

Updating the Software on a PC with Internet Connection

1. Start PIUpdateFinder on the PC to be updated.

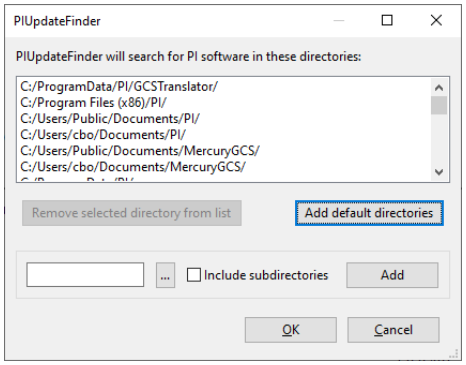
➔ *A table is displayed with information on the PI software installed on your computer.*



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

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

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



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

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

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

d) Click the ***Add*** button.

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

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

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

2. Click the ***Find Updates*** button.

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

PI Update Finder

PI Software Found on Your System

Name	Installed Version	Version on PI Server	Release Notes	Download Link
PI API Components	V1.0.0.0	V1.1.0.0	Release note	PI server
PI Base Components	V1.0.0.0	V1.1.0.0	Release note	PI server
PI Extra Components	V1.0.0.0	V1.1.0.0	Release note	PI server
PI Software Suite	V1.0.0.0	V1.1.0.0	Release note	PI server
PI USB Drivers	V1.0.0.0	V1.1.0.0	Release note	PI server

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

If you have any questions or problems please contact us.

3. To download an update, click ***PI server*** in the ***Download Link*** column.

➔ *You are then asked to specify whether the file is to be downloaded or run directly. Depending on the browser settings, it is possible that the update file is downloaded directly. It is then in the download directory of your computer.*

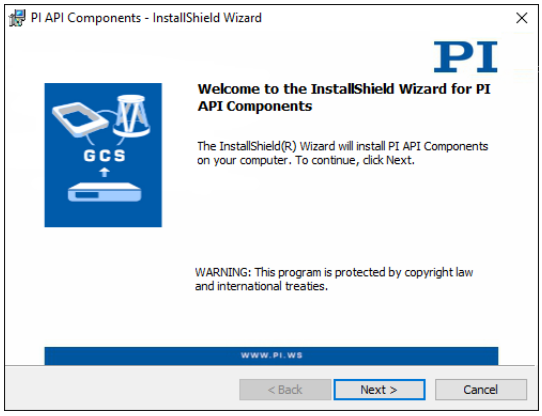
4. Activate the option in the browser query to save the update file to the your computer's hard disk.

➔ *The file is saved to the download directory of your computer. Depending on the browser settings, it is also possible to run the file directly or open it. In this case, continue with step 6.*

5. Install the update:
   - Executable setup files (.exe): Run the downloaded update file.
   - Data archive (.zip): Unpack the archive to a directory on your PC. Open this directory and run the desired setup file (.exe).

   You may have to confirm running the file.

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

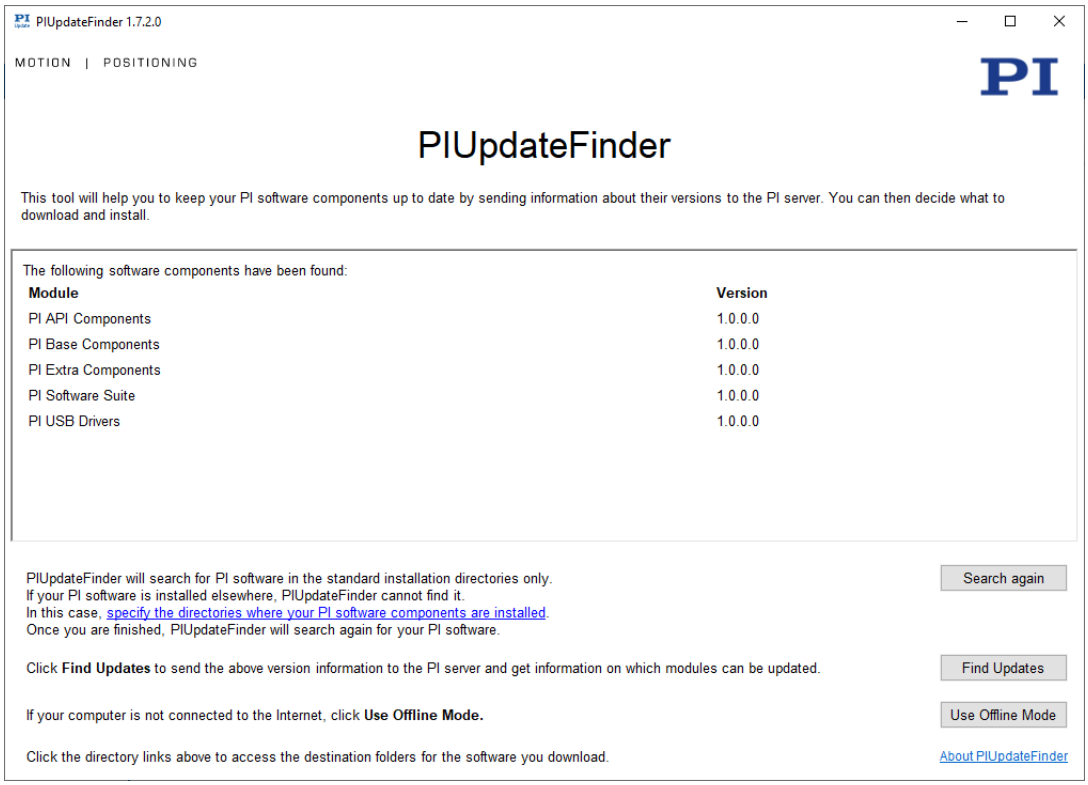


6. Follow the instructions in the InstallShield Wizard.

Updating the Software on a PC without Internet Connection

1. Connect the portable data storage device, e.g., an USB stick, with the PC to be updated.
2. Start PIUpdateFinder on the PC to be updated.

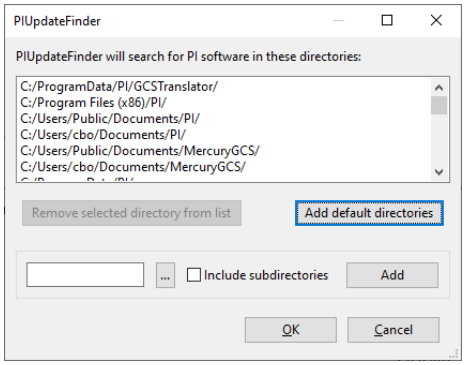
➔ *A table is displayed with information on the PI software installed on your computer.*



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

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

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



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

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

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

d) Click the ***Add*** button.

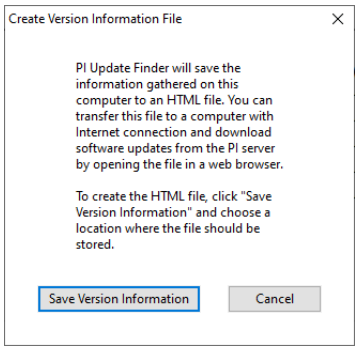
➔ *The directory, and any subdirectories, appear at the end of the list.*

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

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

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

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



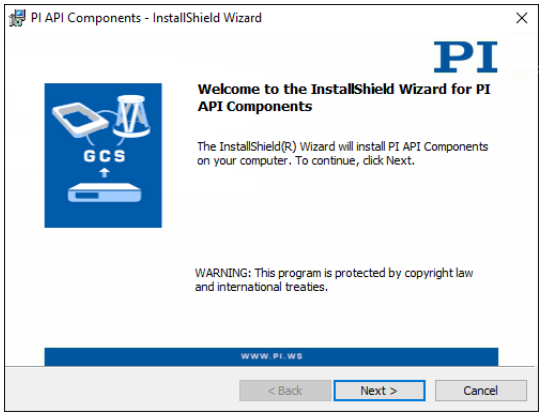
4. Click the ***Save Version Information*** button to save the software version information to a HTML file on your portable data storage device.
5. Transfer the HTML file to a computer connected to the Internet.
6. Open the HTML file in a web browser on the computer connected to the Internet.

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

7. Download your updates and save them to your portable data storage device.
8. Transfer the updates to the PC without Internet connection.
9. Install the updates.
   - Executable setup files (.exe): Run the downloaded update file.
   - Data archive (.zip): Unpack the archive to a directory on your PC. Open this directory and run the desired setup file (.exe).

   You may have to confirm running the file.

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



10. Follow the instructions in the InstallShield Wizard.

Updating the PC Software and Positioner Database in Linux

Requirements

✓ Active connection to the Internet.

✓ You have the access data (user name and password) for the PI software. Information on how to get the access data can be found in the file "C-990.CD1_Releasenews" in the main directory or in the \Manuals folder on the PI software CD.

Updating the PC software in Linux

1. Open the website https://www.physikinstrumente.de/de/produkte/motion-control-software/.
2. Log onto the website:

a) Click ***Login***.
b) Log in with the user name and password for the PI software CD.

3. Scroll down to the ***Downloads*** section.
4. Click the archive file "CD Mirror" or the associated download link.
5. Select the option in the following request to save the file to your PC.
   If you do not specify anything else, the "CD Mirror" archive file is stored in the default download directory of your PC.
6. Unpack the archive file into a separate directory.
7. Go to the ***linux*** subdirectory in the directory with the unpacked files. You will find an additional archive file there.
8. Unpack the archive file in the ***linux*** directory by entering the command tar -xvpf <name of the archive file> on the console.
9. Read the accompanying information on the software update (readme file and/or releasenews file) and decide whether the update makes sense for your application.
   - If no: Stop the update procedure.
   - If yes: Go through the following steps.
10. Log into the PC as superuser (root privileges).
11. Install the update.
12. If you also received an updated PISTAGES3.DB database from our customer service department: Install that update onto the PC.

6.6 Connecting the PC

Communication between the E-872.401 and a PC is required to configure the E-872.401 and to command motion using the GCS commands.

NOTICE

Damage due to noncompliant USB cable

The type A USB connection () of the E-872.401 is only intended for connection to a USB human interface device. If a USB host (e.g., PC) is connected to the type A USB connection of theE-872.401 via a noncompliant USB cable, the E-872.401 or the host could be damaged.

► Use a standard-compliant USB cable only.

► For connecting to a PC, use the type B () or type Mini-B () USB connection only.

Information

The E-872.401's communication interfaces are active at the same time. Commands are executed in the order in which the complete command lines arrive. However, simultaneous use of several communication interfaces can cause problems with the PC software.

► Always only use one interface of the E-872.401.

6.6.1 Connecting the E-872.401 to a PC

Tools and Accessories

- Suitable cable for the selected communication interface (in the scope of delivery (p. 15) or available as optional accessory (p. 15))

Requirements

✓ The PC has a connection available for the selected communication interface.

✓ If necessary: The interface parameters of the PC have been adapted for the E-872.401 (p. 16).

Connecting the E-872.401 to the PC

1. Connect the cable to the selected communication interface of the E-872.401.
2. Connect the cable to an unused port on the PC.

6.6.2 Integrating into E-872.401 a Network

Tools and Accessories

- Suitable network cable (in the scope of delivery (p. 15) or available as optional accessory (p. 15))

Requirements

✓ A network access point is available for the E-872.401.

✓ The default settings of the TCP/IP interface (p. 17) do not collide with the settings of other devices in the network.

Integrating the E-872.401 into a TCP/IP Network

1. Connect the RJ45 socket of the E-872.401 to the network cable.
2. Connect the network cable to the network access point.

7 Startup and Operation

7.1 Switching on the E-872.401

Requirements

✓ You have read and understood the general safety instructions (p. 11).
✓ The E-872.401 has been installed properly (p. 32).

⚠ CAUTION

Risk of electric shock if the protective earth conductor is not connected!
If the protective earth conductor is missing or not properly connected, risk of dangerous touch voltages on the E-872.401 in the event of malfunction or failure of the system. If there are touch voltages, touching the E-872.401 can lead to minor injury due to electric shock.

► Connect the E-872.401 to a protective earth conductor (p. 32) before startup.
► Do **not** remove the protective earth conductor during operation.
► If the protective earth conductor has to be removed temporarily (e.g., for modifications), reconnect the E-872.401 to the protective earth conductor before restarting.

Switching the E-872.401 On

1. Connect the power cord of the power adapter to the power socket.
2. Switch on the E-872.401.
➔ *The E-872.401 boots the firmware and loads information from the nonvolatile memory to the volatile memory.*
3. Wait until the **STA** LED lights up green.
➔ *The information has loaded and the E-872.401 is ready for normal operation.*
4. If the **STA** LED does not light up a few seconds after switching on, contact our customer service department (p. 128).

7.2 Establishing Communication with the PC

The procedure for PIMikroMove is described in the following.
The figures show the procedure for any electronics; the procedure for the E-872.401 corresponds.

7.2.1 Establishing Communication via USB

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

Requirements

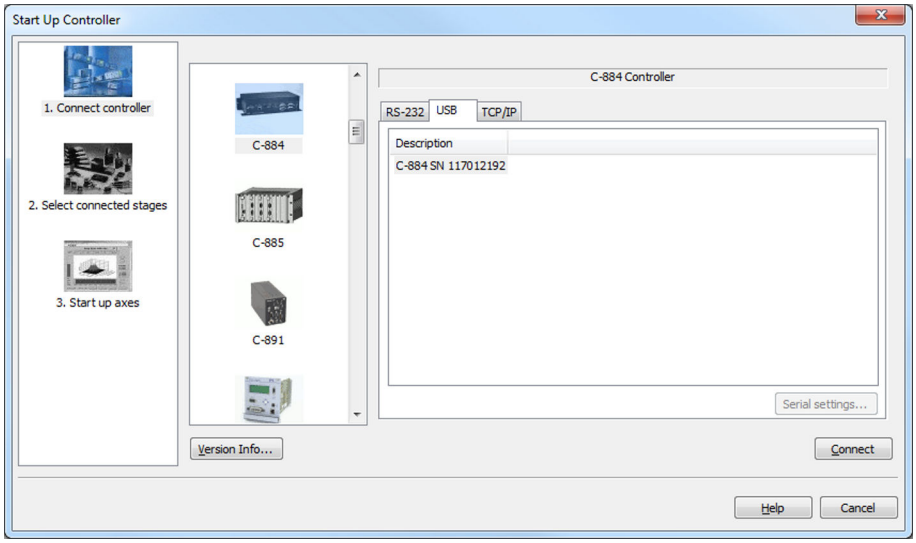
✓ The E-872.401 is connected to the USB interface of the PC (p. 41).
✓ The E-872.401 is switched on (p. 43).
✓ The PC is switched on.
✓ The required software and drivers are installed (p. 35) on the PC.
✓ You have read and understood the manual for the PC software. The links to the software manuals are in the A000T0081 file on the PI software CD.

Establishing Communication via USB

1. Start PIMikroMove.

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

2. If the ***Start up controller*** window does not open automatically, select the ***Connections > New...*** menu item in the main window.
3. Select ***E-872*** in the controller selection field.
4. Select the ***USB*** tab on the right-hand side of the window.



5. Select the connected E-872.401 in the ***USB*** tab.
6. Click ***Connect*** to establish communication.

➔ *If communication could not be established, look for a solution to the problem in the "Troubleshooting (p. 123)" chapter.*

➔ *If communication was established successfully, PIMikroMove guides you through the configuration of the E-872.401 for the connected positioner, refer to "Starting Motion (p. 47)".*

7.2.2 Establishing Communication via TCP/IP

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

- Network with DHCP server: No adjustment of the factory settings of the E-872.401 interface parameters is required. You can begin with setup of the communication (p. 46).
- Network without DHCP server or direct connection of the E-872.401 to the PC's Ethernet socket: it is necessary to adapt the E-872.401's interface parameters (p. 45). Make the necessary adaptations before establishing communication.

Requirements

✓ The E-872.401 is connected (p. 42) to the network or directly to the PC via the RJ45 Ethernet socket.

✓ If several E-872.401s are connected to the same network via their TCP/IP interfaces: You have the serial number of the E-872.401 ready with which the communication is to be established. The serial number can be found on the type plate of the E-872.401 (p. 14).

✓ The E-872.401 is switched on (p. 43).

✓ The PC is switched on.

✓ The required software and drivers are installed (p. 35) on the PC.

✓ You have read and understood the manual for the PC software. The links to the software manuals are in the A000T0081 file on the PI software CD.

Adapting the TCP/IP Interface Parameters of the E-872.401

If you need to adapt the interface parameters of the E-872.401 to use the E-872.401 in a network, proceed as follows.

1. Establish communication between the E-872.401 and the PC via a different interface (e.g., USB (p. 43)).
2. Select the ***E-872.401 > Configure interface*** menu item in the main window of PIMikroMove.

➔ *The **Configure Interface** window opens.*

3. Select the ***TCP/IP*** tab in the ***Stored Settings*** area in the ***Configure Interface*** window.

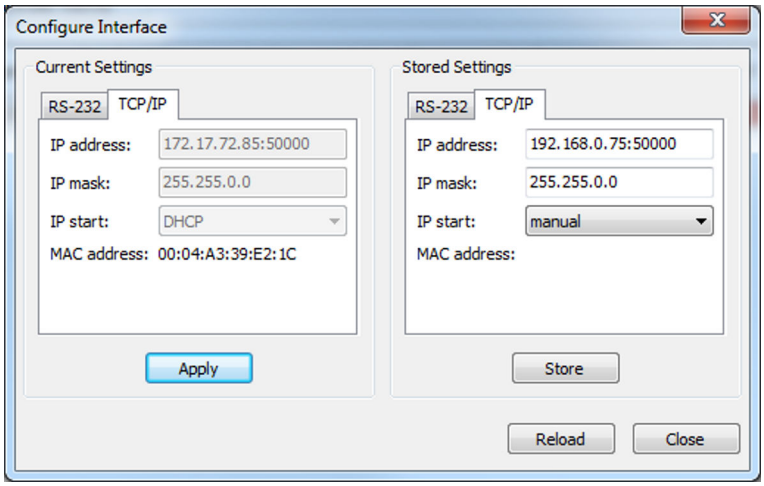


Figure 5: "Configure Interface" window with examples of settings

4. Make the necessary adaptations in the ***TCP/IP*** tab in the ***Stored Settings*** area:
   a) ***IP address*** field: E-872.401's IP address in format xxx.xxx.xxx.xxx:50000
   b) ***IP mask*** mask: Network's subnet mask
   c) ***IP start*** field: E-872.401's startup behavior

      manual: Manually specified, static IP address is used

      DHCP: IP address is assigned automatically by a DHCP server.

5. Save the changed settings to the nonvolatile memory of the E-872.401 by clicking ***Store***.

➔ *The **Store interface settings** dialog opens.*

6. Click ***Store settings*** in the ***Store interface settings*** dialog.

➔ *The dialog closes. The settings were stored in the nonvolatile memory of the E-872.401 .*

7. Close the ***Configure Interface*** window.
8. Close the connection with the E-872.401 by selecting the ***Connections > Close > E-872.401*** menu item in the main window of PIMikroMove.
9. Switch the E-872.401 off and on again via its toggle switch.

⚠ *CAUTION*

Risk of crushing from unexpected motion!

When the communication between the E-872.401 and the PC is established via TCP/IP, the PC software offers all electronics for selection that are available in the same network. After selecting a E-872.401 for the connection, all commands are transmitted to this device. If the wrong device is selected, unexpected motion could be commanded and result in bruising injuries to the operating and maintenance staff of the positioner connected.

► If several E-872.401 entries are displayed in the PC software, make sure that you select the right E-872.401.

Information

Communication via TCP/IP can fail if the network cable was connected to the Ethernet socket of the E-872.401 while the E-872.401 was switched on.

► If communication cannot be established, switch the E-872.401 off, connect the network cable, and switch the E-872.401 on again.

Information

For communication via TCP/IP, the E-872.401 only has one unchangeable port (50000) available that cannot be used for more than one connection at a time.

Establishing Communication via TCP/IP

1. Start PIMikroMove.

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

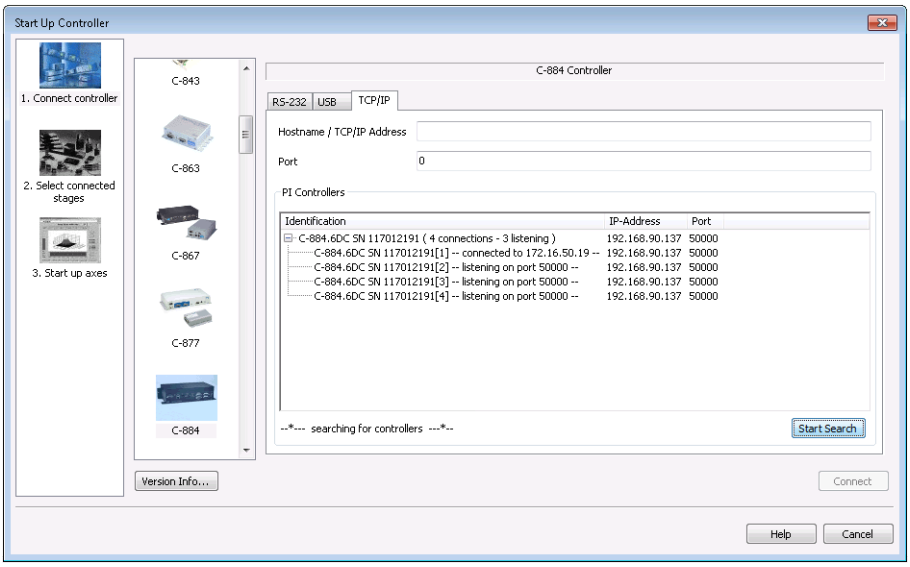
2. If the ***Start up controller*** window does not open automatically, select the ***Connections > New...*** menu item in the main window.
3. Select ***E-872*** in the controller selection field.
4. Select the ***TCP/IP*** tab on the right-hand side of the window.

➔ *The software now searches the network for all controllers of the E-872 type.*

Click ***Start Search*** if the search for E-872 type controllers does not start automatically.

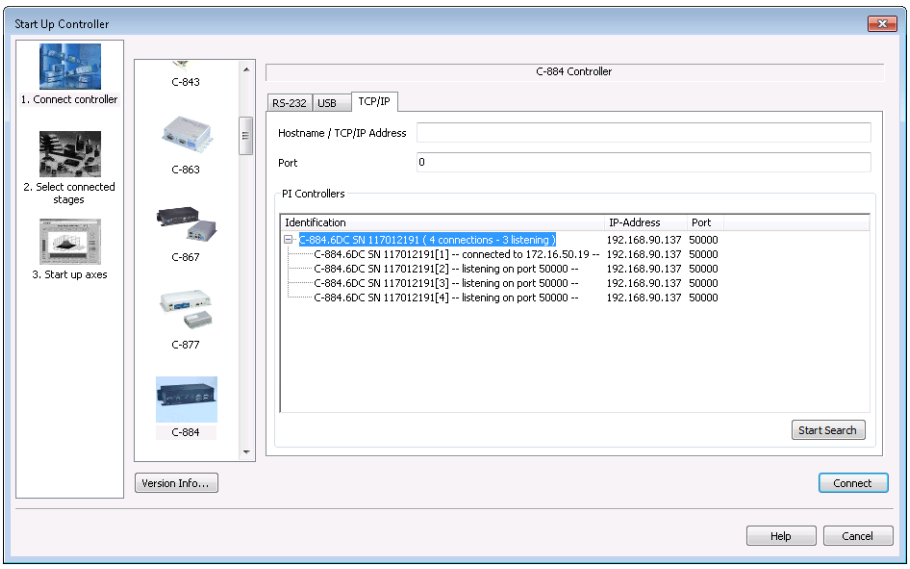
➔ *Searching the network for the E-872 type of controller has started.*

*As long as the search is running, the **Connect** button is deactivated. If the search was successful, all E-872 controllers in the same network are displayed in the **PI Controllers** field.*



5. Click the entry for your E-872.401 found in the list of controllers. This must show the status "listening on port 50000".
   - Do not select a controller that is already connected via TCP/IP (status "connected to ..."). Otherwise, an error message will be displayed as soon as you try to establish communication with this controller.
   - If several entries with the same name are shown, identify your E-872.401 on the basis of its nine-digit serial number.
   - If the E-872.401 is not displayed in the list of the controllers found, check the network settings. Consult your network administrator if necessary.

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



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

➔ *If communication could not be established, look for a solution to the problem in the "Troubleshooting (p. 123)" chapter.*

➔ *If communication was established successfully, PIMikroMove guides you through the configuration of the E-872.401 for the connected positioner, refer to "Starting Motion (p. 47)".*

7.3 Starting Motion

The procedure for PIMikroMove is described in the following.

After communication has been established between the E-872.401 and the PC, PIMikroMove guides you through the configuration of the E-872.401 for the positioner. It is then possible to run the first motion tests of the positioner.

Selection of the configuration steps offered by PIMikroMove is based on evaluation of the following parameter values in the volatile memory of the E-872.401:

- **Stage Name** (0x3C): The value is used by PIMikroMove as criterion for finding a suitable parameter set in the positioner databases.
- **Stage Type** (0x0F000100): The value was loaded from the ID chip of the connected positioner when the E-872.401 was switched on.

Possible configuration steps:

- If the values of the parameters 0x3C and 0x0F000100 are not identical, the ***Stage Type Configuration*** window opens. A corresponding message is displayed when a suitable parameter set is not in the positioner database.
- If the value of parameter 0x0F000100 is empty, e.g., because the positioner does not have an ID chip, the ***Start up controller*** window switches to the ***Select connected stages*** step.
- When the values of the parameters 0x3C and 0x0F000100 are identical, PIMikroMove assumes that all parameters of the E-872.401 have already been adapted to the connected positioner. The ***Start up controller*** window goes directly to the ***Start up axes*** step, where the reference move can be started.

The figures show the procedure for any electronics; the procedure for the E-872.401 corresponds.

Requirements

✓ PIMikroMove is installed on the PC (p. 35).
✓ You have installed the latest version of the PISTAGES3.DB database onto your PC (p. 35).
✓ If PI provided a custom positioner database for your positioner, the dataset was imported into PIStages3 (p. 35).
✓ You have installed the positioner in the same way as it will be used in your application (corresponding load, orientation, and mounting).
✓ You have connected the positioner to the E-872.401 (p. 34).
✓ You have established communication between the E-872.401 and the PC with PIMikroMove (p. 43).

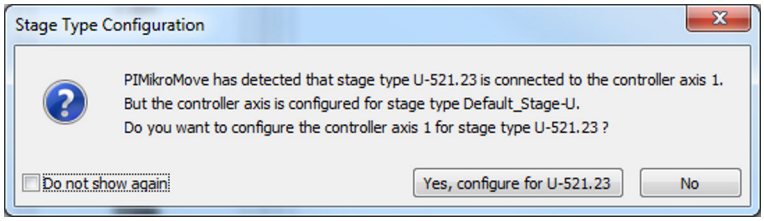
NOTICE

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

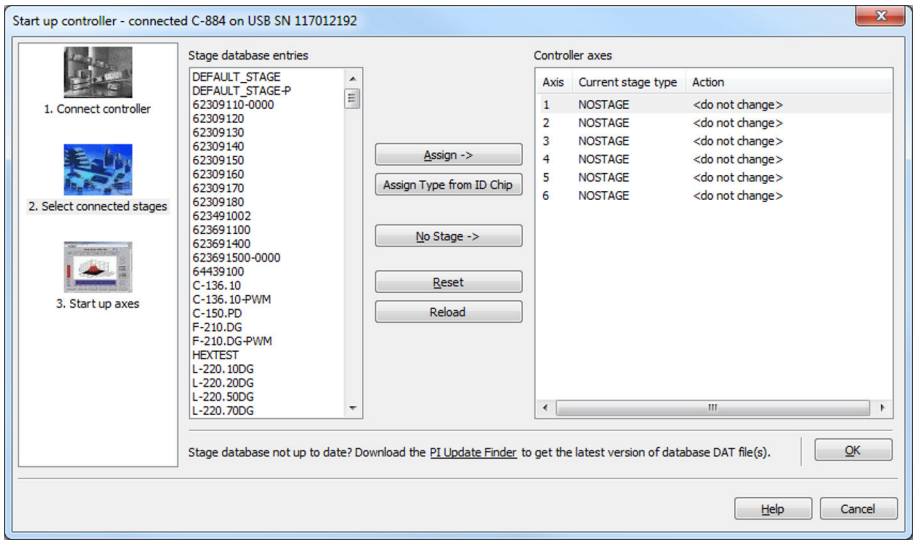
► Make sure that the positioner type selected in the PC software matches the positioner connected.

Starting motion with PIMikroMove

1. Load the parameter set from the positioner database into the E-872.401:
   - When the ***Stage Type Configuration*** dialog has opened: Click the ***Yes, configure for ...*** button to load the matching parameter set from a positioner database into the E-872.401.

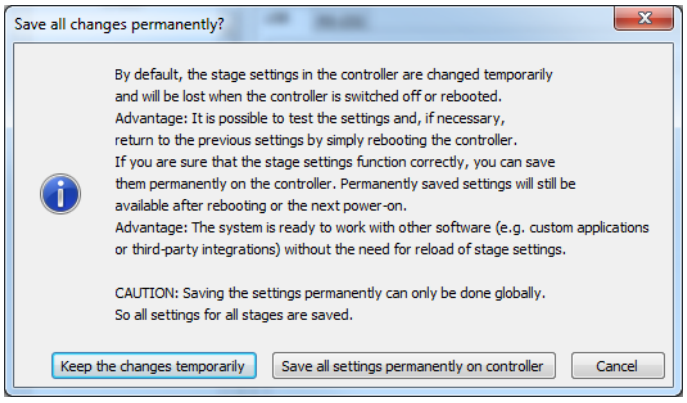


   - If the ***Select connected stages*** step is displayed in the ***Start up controller*** window:



   a) Select the appropriate positioner type: Click either ***Assign Type from ID Chip*** or mark the matching positioner type in the ***Stage database entries*** list and click ***Assign***.
   b) Confirm selection with ***OK*** to load the parameter settings for the selected positioner type from the positioner database into the E-872.401.

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

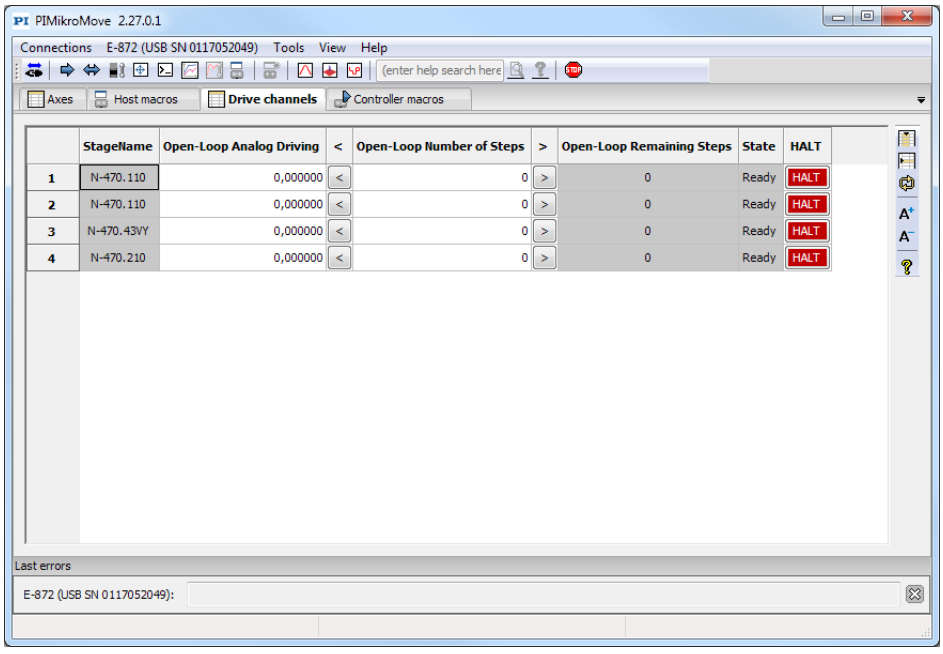


2. Specify how you want to load the parameter settings into the E-872.401 in the ***Save all changes permanently?*** dialog box:
   - Temporary load: Click ***Keep the changes temporarily*** to load the parameter settings into the volatile memory of the E-872.401. The settings are lost when the E-872.401 is switched off or rebooted.
   - Load as default values: Click ***Save all settings permanently on controller*** to load the parameter settings into the nonvolatile memory of the E-872.401. The settings are available immediately after switching on or rebooting the E-872.401 and do not need to be reloaded.

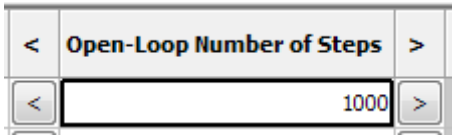
➔ *The **Start up controller** window with the **Start up axes** step is displayed.*

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

➔ *The main window of PIMikroMove opens.*



4. Test the motion several times in step mode:
   a) Enter the number of steps for the channel to move into the ***Open-Loop Number of Steps*** column.
   b) Click one of the **<** or **>** arrows to do the motion in corresponding direction.



5. Run several motion tests in analog mode:

a) Enter a value in volts into the Open-Loop Analog Driving column for the analog output voltage.
b) Confirm with the Enter button.

7.4 Making Data Backups

Saving the Parameter Set to the Positioner Database

1. Click ***Load and Save Parameters -> Save parameters to stage database...*** in the expanded single axis window in PIMikroMove.

➔ *The **Save Parameters as User Stage Type** dialog opens.*

2. Save the changed parameter values as new positioner type in the ***Save Parameters as User Stage Type*** dialog:
   a) Leave the entry in the ***Parameters of axis*** field unchanged.
   b) Enter the name for the new positioner type into the ***Save as*** field.
   c) Click ***OK***.

➔ *The new positioner type was saved to the PISTAGES3.DB database. The display of the connected positioner type was updated in the single axis window and in the main window of PIMikroMove.*

7.4.1 Saving Parameter Values

The E-872.401 is configured via parameters, e.g., for adapting to the connected positioner. The parameter values can be saved to a text file so that they can be restored at a later time.

Information

Changing parameter values can cause undesirable results.

- ► Create a backup copy on the PC before changing the parameter settings of the E-872.401. You can then restore the original settings at any time.
- ► Create an additional backup copy with a new file name each time after optimizing the parameter values or adapting the E-872.401 to a particular positioner.

Parameter values saved in a text file on the PC can be loaded back to the E-872.401 in PIMikroMove or PITerminal. The ***Send file...*** button is available for this purpose in the send command window. Before loading into the E-872.401, the individual lines of the text files must be converted into command lines that contain the corresponding SPA or SEP commands.

Requirements

✓ You have established communication between the E-872.401 and the PC with PIMikroMove (p. 43) or PITerminal.

Saving Parameter Values in a Text File

1. If you are using PIMikroMove, open the window for transmitting commands: Select ***Tools > Command entry*** in the main window or press F4 on the keyboard.

➔ *After communication has been established, the main window is opened in PITerminal automatically and commands can then be sent.*

2. Query the parameter values that you want backup.
   - If you want to save the parameter values from the volatile memory of the E-872.401: Send the SPA? command.
   - If you want to save the parameter values from the nonvolatile memory of the E-872.401: Send the SEP? command.
3. Click the ***Save...*** button.

➔ *The **Save content of terminal as textfile** window opens.*

4. Save the queried parameter values to a text file on your PC in the ***Save content of terminal as textfile*** window.

7.4.2 Saving Controller Macros

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

The procedure for PIMikroMove is described in the following.

Saving Controller Macros on the PC

1. Select the ***Controller macros*** tab in PIMikroMove's main window.
2. Select the macros in the ***Macros on controller*** list that you want to back up to the PC:
   - Click the desired entry in the list to select a macro.
   - To select several macros, hold down the Shift button and click the desired entries in the list.
   - To deselect, click an open area in the list.

➔ *By selecting one or more macros, the*  *(**Save selected macros to PC**) button becomes active.*

3. Save the selected macros on the PC:
   a) Click the  button to open a directory selection window.
   b) Select the directory on the PC where you want to save the macros.
   c) Click ***Save***.

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

Loading Controller Macros from the PC to the E-872.401

1. Select the ***Controller macros*** tab in PIMikroMove's main window.
2. Load macros from the PC to the E-872.401:
   a) Click the  button to open a file selection window.
   b) Select the text files (<macro name>.txt) in the file selection window whose contents you want to load as a macro from the PC to the E-872.401.
   c) Click ***Open***.

➔ *For each selected text file (<macro name>.txt), the content is loaded as a macro <macro name> into the E-872.401.*

8 E-872.401 Functions

8.1 Protective Functions of the E-872.401

The E-872.401 has functions that are intended to protect it against damage.

8.1.1 Protecting Against Overheating

If a certain internal temperature (70°C) is reached, the E-872.401 reacts as follows to protect the system against damage:

- The control value is set to zero for the axis concerned.
- Error code 603 is output.

Then restore the operational readiness (p. 52) for the E-872.401.

8.1.2 Restoring Operational Readiness

Restoring the E-872.401's Operational Readiness

1. Send the ERR? command to read out the error code.

➔ ERR? *resets the error code to zero during the query.*

2. Check your system and make sure that the following points are fulfilled:
   - The axis can be moved without danger.
   - The E-872.401 has not overheated.

8.1.3 Commands

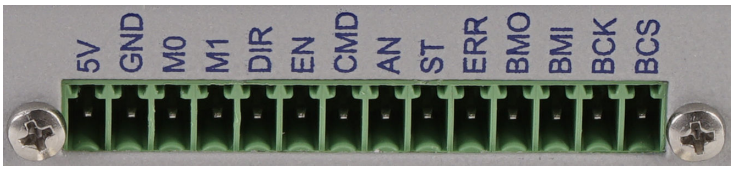
E		Page
ERR?	Get Error Number	85

8.1.4 Parameters

8.2 Digital Inputs and Outputs

The E-872.401's digital inputs and outputs are at the device's I/O connector, Phoenix Contact MC 1,5/14-GF-3,5-LR. The number of available input and output lines to the E-872.401 can be queried with the TIO? command.

Overview



The E-872.401's I/O connector with pin **M0**, **M1**, **DIR**, **EN**, **CMD**, and **ST** for digital input signals and pin **ERR** for digital output signals

8.2.1 Connecting the Digital Outputs

Tools and Accessories

- Suitable Phoenix Contact MC 1,5/14-ST-3,5 plug, with cable
- Device to be triggered having digital input for TTL signals

Information

Digital output signals are available on pin 10 of the E-872.401's I/O socket.

Connecting a Device to be Triggered

1. Connect a suitable device to the E-872.401's I/O socket with a Phoenix Contact MC 1,5/14-ST-3,5 plug.

8.2.2 Connecting the Digital Inputs

Tools and Accessories

- Suitable signal source
- Suitable Phoenix Contact MC 1,5/14-ST-3,5 plug, with cable

Information

Digital TTL input signals can be fed to the E-872.401 via pins 3, 4, 5, 6, 7, and 9 of the I/O socket.

Connecting a Digital Signal Source

1. Connect a suitable signal source to the E-872.401's I/O socket with a Phoenix Contact MC 1,5/14-ST-3,5 plug.

8.2.3 Digital Output Signals

The number of output lines available on the E-872.401 can be queried with the TIO? command.

External devices can be triggered via the digital outputs of the E-872.401.

The E-872.401's I/O socket has 1 digital output (ID 1) with the following function:

ID	Pin	Labeling	Function
1	10	ERR	Error signal ■ High: Error (error code ≠ 0) ■ Low: No error

8.2.4 Digital Input Signals

The number of input lines available on the E-872.401 can be queried with the TIO? command. The status of the digital input lines can be queried with the DIO? command.

The E-872.401's I/O socket has 6 digital output lines (IDs 1 to 6) with the following functions:

ID	Pin	Labeling	Function
1	3	M0	Selecting the drive channel See "Switching Between the Drive Channels" (p. 28)
2	4	M1	
3	5	DIR	Direction signal for step mode ■ High: Forward motion ■ Low: Backward motion See "Triggering Motion" (p. 26)
4	6	EN	Activating the drive in step mode See "Triggering Motion" (p. 26)
5	7	CMD	Selecting the command mode ■ High: Commanding via digital I/O lines ■ Low: Commanding via PC or HID See "Triggering Motion" (p. 26)
6	9	ST	Selecting the drive mode ■ High: Linear mode ■ Low: Step mode See "Drive Modes for Piezo Inertia Drives" (p. 24)

8.2.5 Commands

D		Page
DIO?	Get Digital Input Lines	85
T		**Page**
TIO?	Tell Digital I/O Lines	107

8.2.6 Parameters

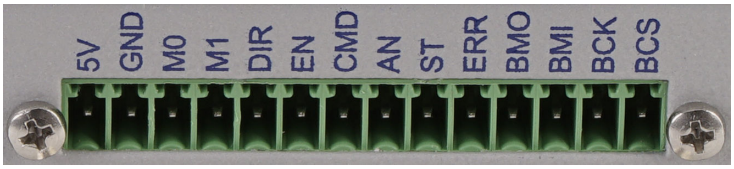
8.3 Analog Input Signals

The E-872.401's analog input is available on the device's I/O socket, Phoenix Contact MC 1,5/14-GF-3,5-LR:

Pin	Labeling	Function
8	AN	Analog input voltage for linear mode (0 - 4.8 V) $V_{Ch} = V_{AN} \times 10$ (V_{Ch} = output voltage of the channel, V_{AN} = analog input voltage)

8.3.1 Connecting Analog Signal Sources

Overview



E-872.401's I/O socket with **AN** pin for analog input signals

Tools and Accessories

- Suitable signal source
- Suitable Phoenix Contact MC 1,5/14-ST-3,5 plug, with cable

Connecting an Analog Signal Source

1. Connect a suitable signal source to the E-872.401's I/O socket with a Phoenix Contact MC 1,5/14-ST-3,5 plug.

8.3.2 Commands

8.3.3 Parameters

8.4 Controlling with HID

HID (Human Interface Device) denotes an input or output device connected to the E-872.401 and is intended to be operated manually. Joysticks and gamepads are typical HIDs.

HID control means controlling motion variables of a positioner's axis connected to the E-872.401 by displacing an axis of the HID.

8.4.1 How it Works

An HID's axis can control the following motion variables of a positioner's axis connected to the E-872.401:

- Velocity of the axis connected: Product from the lookup table value that corresponds to the current displacement of the HID's axis as well as the currently valid maximum velocity of the axis connected. The currently valid maximum velocity is specified by one of the following sources:
  - Maximum Motor Output parameter (0x9)
  - Displacement of an HID's axis

8.4.2 HID Control Configuration

Control of the axis/axes connected to the E-872.401 by the HID's axes is configured via the HIA command. The current HID control configuration can be queried with the HIA? command. The direction of motion of HID-controlled axes can be inverted via the **Invert Direction Of Motion For Joystick-Controlled Axis?** parameter (0x61).

Assigning a lookup table to an HID axis is done with the HDT command. The current lookup table assignments to HID axes can be queried with the HDT? command. The values in the lookup table are factors that are applied to the motion parameter to be controlled during HID

control. The E-872.401's firmware gives a choice of two predefined lookup table types (linear and parabolic) and allows four customer-specific lookup tables to be filled with individual values. Lookup tables can be filled with values with the HIT command. HIT? queries the values of the points in the lookup tables.

Use the HIN command to activate/deactivate HID control of the axes connected to the E-872.401. The HIN? command queries the activation state of HID control. When HID control is deactivated, the target position is set to the current position of the controlled axis.

The HIS? command queries the properties of the HID's operating elements. The current status of the HID buttons can be queried with HIB? and the current displacement of HID axes with HIE?.

8.4.3 Programming HID Control

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

8.4.4 Connecting an HID

Overview

E-872.401's connector for a manually operated device such as a joystick or gamepad

Tools and Accessories

- Suitable HID with type A USB connector such as a joystick or gamepad

Connecting an HID to the E-872.401

1. Connect the HID to the E-872.401's type A USB connector.

First of all, we recommend testing the HID's operating elements after connecting it to the E-872.401. If the response behavior of the HID's axes does not meet your requirements, it is possible to calibrate its axes.

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

The procedure for PIMikroMove is described in the following.

8.4.5 Testing the HID and Calibrating the Axes

The positioner does not need to be connected to the E-872.401 for testing the HID and calibrating its axes.

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

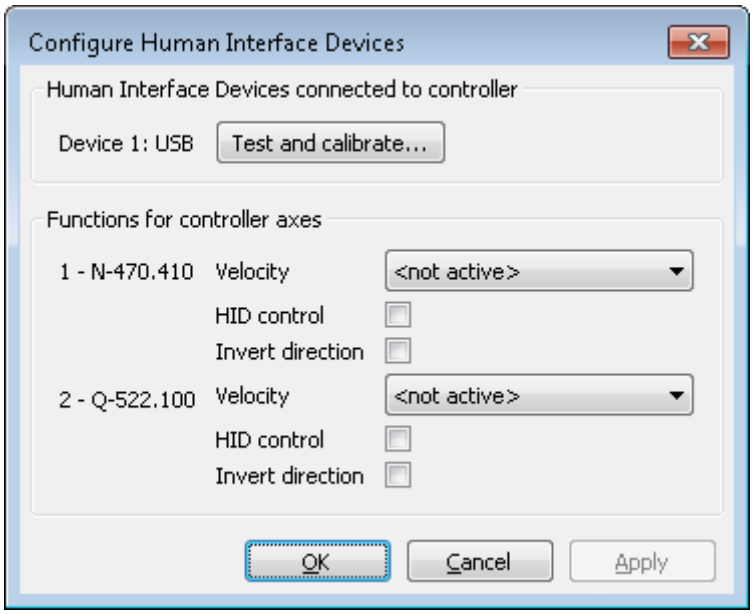
Requirements

✓ PIMikroMove is installed on the PC (p. 35).
✓ PIMikroMove has established communication between the E-872.401 and the PC (p. 43).
✓ The E-872.401 has been connected to the HID (p. 56).

Testing the HID

1. Open the window for configuring HID control via the ***E-872.401 > Configure controller HIDevice(s)…*** menu item in PIMikroMove's main window.

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



2. Open the window for testing and calibrating the HID by clicking ***Test and calibrate...***.

➔ *The **Test and Calibrate HIDevice** window opens.*

3. Select the ***Test device*** tab in the ***Test and Calibrate HIDevice*** window and test the HID's operating elements:
   a) Move the HID's axes and at the same time, watch the status indicators in the ***State of axes*** area.
   b) Press the HID's buttons and at the same time, watch the status indicators in the ***State of buttons*** area.
   c) Enter various values into the fields in the ***State of LEDs*** area (if any) and at the same time, watch the behavior of the corresponding operating elements on the HID.

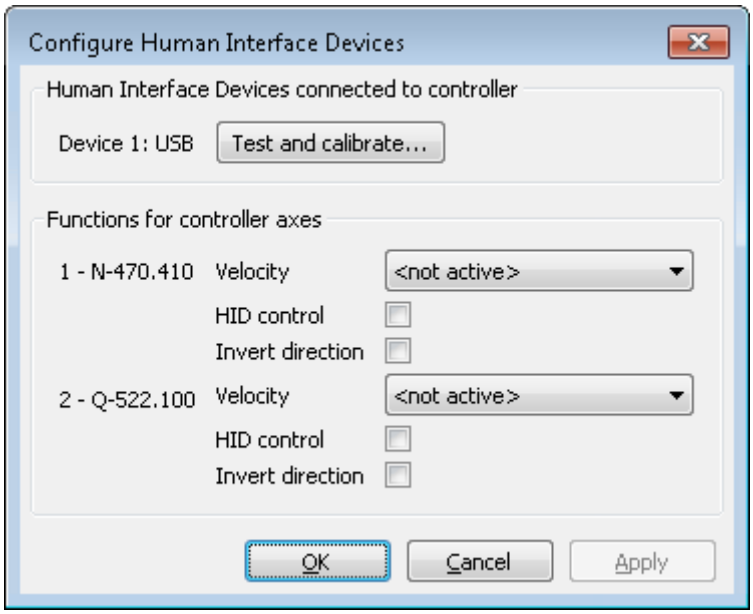
   ➔ *Displacement of the HID's axes is shown in the **State of axes** area. Any of the HID'S buttons pressed are displayed green in the **State of buttons** area.*

4. Depending on what is next, do the following:
   - If you want to calibrate the HID's axes directly afterwards, proceed as described below (p. 57).
   - If you want to set up and activate HID control for the E-872.401 directly afterwards, close the ***Test and Calibrate HIDevice*** window with ***Close*** and continue as described in "Setting up and Activating HID Control (p. 59)".
   - If you do not want to make any further settings at this point, close ***Test and Calibrate HIDevice*** with ***Close*** and the ***Configure Human Interface Devices*** window with ***OK***.

Calibrating HID Axes

1. If necessary, open the window for configuring HID control via the ***E-872.401 > Configure controller HIDevice(s)...*** menu item in PIMikroMove's main window.

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

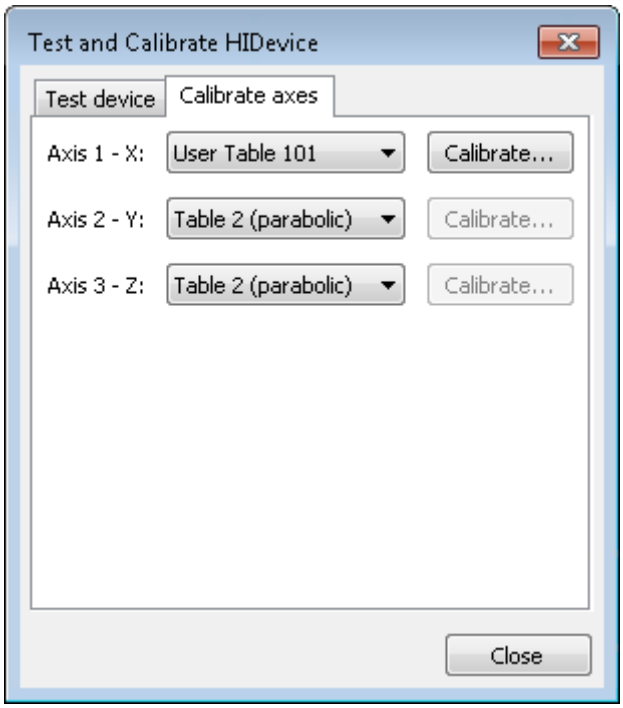


2. If necessary, open the window for calibrating the HID and click the ***Test and calibrate...*** button.

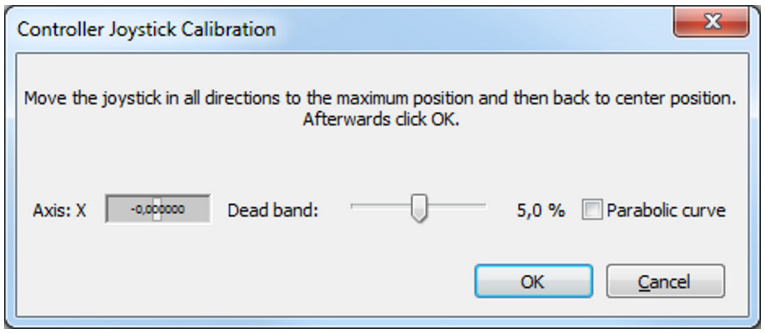
➔ *The **Test and Calibrate HIDevice** window opens.*

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

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



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



b) Move the HID's axes to all extreme positions. The custom lookup table values are determined in this way.
c) Let go of the axis.
d) If you want to change the neutral area of the axis (i.e., the area around the center position of the axis where no change in the controlled motion variable is triggered), set the ***Dead band*** slider accordingly.
e) If the values in the user-defined lookup table are to describe a parabolic waveform, click the ***Parabolic curve*** checkbox.
f) Click ***OK*** in the ***Controller Joystick Calibration*** window to copy the appropriate values from the lookup table.

➔ *The lookup table values are written to the E-872.401's volatile memory. The writing progress is indicated in a separate window. The window for the writing process and the Controller Joystick Calibration window automatically close after the writing process has finished.*

6. Close the ***Test and Calibrate HIDevice*** window with ***Close***.
7. Depending on what is next, do the following:
   - If you want to set up and activate HID control for the E-872.401 directly afterwards, continue as described in "Setting up and Activating HID Control (p. 59)".
   - If you want to save the lookup table assignments to the HID's axes and the content of user-defined lookup table to the E-872.401's nonvolatile memory directly afterwards, close the ***Configure Human Interface Devices*** window with ***OK*** and continue as described in "Saving the Configuration of HID Control Permanently (p. 61)".
   - If you do not want to make any further settings at this point, close the ***Configure Human Interface Devices*** window with ***OK***.

8.4.6 Setting Up and Activating the HID Control

The following motion variables for the axes connected to the E-872.401 can be controlled by HID:

- Velocity - Velocity for motion of the axis

Before activating HID control, the following steps are recommended:

- Testing the HID
- Calibrating the HID axes

See "Testing the HID and Calibrating the Axes (p. 56)" for a description of these steps.

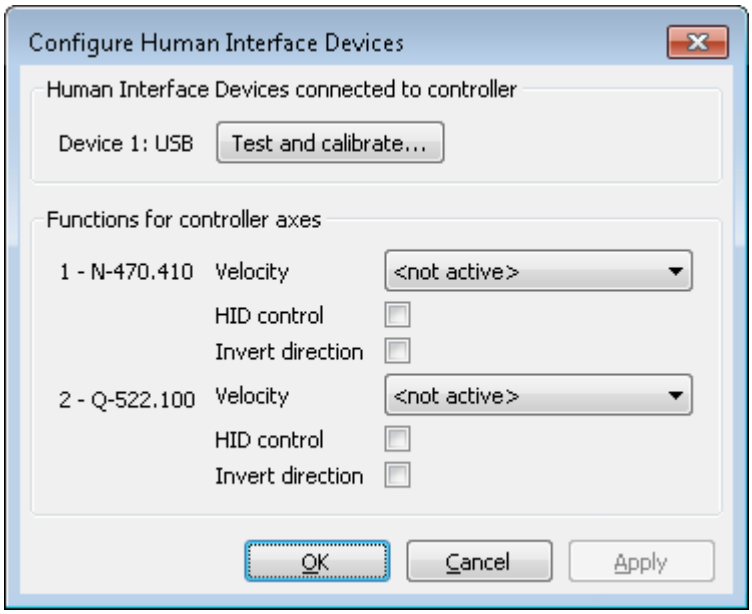
Requirements

✓ You have connected the HID to the E-872.401 (p. 56).
✓ All devices are still ready for operation.

Setting Up and Activating HID Control

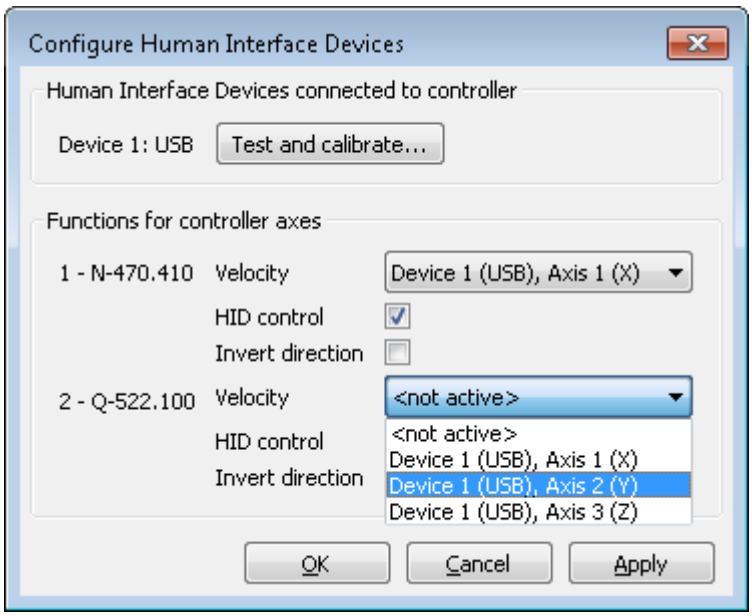
1. If necessary, open the window for configuring HID control via the ***E-872.401 > Configure controller HIDevice(s)...*** menu item in PIMikroMove's main window.

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



2. Make the following respective settings for the E-872.401's axes displayed in the ***Functions for controller axes*** area:
   a) Select the HID's axis in the corresponding field that is to be used for the motion variable to be controlled.
   b) Activate HID control by clicking the ***HID control*** checkbox.
   c) If the direction of motion is to be inverted during HID control, click the ***Invert direction*** checkbox.

   ➔ *In the example illustrated, the E-872.401's X axis is connected to channel 1 and its Y axis is connected to channel 2 of HID 1 (USB joystick) and HID control is activated.*



3. Click the ***Apply*** button in the ***Configure Human Interface Devices*** window to activate the settings.
4. Send the settings for setting up HID control to the E-872.401 by clicking the ***OK*** button.

➔ *The **Configure Human Interface Devices** window closes.*

5. Make sure that servo mode for the E-872.401's axes is switched on in PIMikroMove (e.g., by clicking the ***Servo*** checkbox in the ***Axes*** tab in PIMikroMove's main window).

➔ *The E-872.401's axes can now be controlled by the HID according to the settings made.*

6. If you want to save the new settings for HID control to the E-872.401's nonvolatile memory, continue as described in "Saving the Configuration of HID Control Permanently (p. 61)".

Saving the Configuration of the HID Control Permanently

Select the ***E-872.401 > Save parameters to non-volatile memory*** menu item in PIMikroMove's main window.

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

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

8.4.7 Commands

H		Page
HDT	Set HID Default Lookup Table	85
HDT?	Get HID Default Lookup Table	86
HIA	Configure Control Done By HID Axis	87
HIA?	Get Configuration Of Control Done By HID Axis	88
HIB?	Get State Of HID Button	88
HIE?	Get Deflection Of HID Axis	89
HIN	Set Activation State For HID Control	89
HIN?	Get Activation State Of HID Control	89
HIS?	Get Configuration Of HI Device	90
HIT	Fill HID Lookup Table	91
HIT?	Get HID Lookup Table Values	91

8.4.8 Parameters

0x9	Maximum Motor Output	Maximum control value for driving an axis respectively a channel. See the response to HPA? for possible values
0x61	Invert Direction Of Motion For Joystick-Controlled Axis?	Inverts the direction of motion for HID-controlled axes. 0 Direction of motion not inverted (default) 1 Direction of motion inverted

8.5 Controller Macros

The E-872.401 can save and process command sequences as macros.

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

- Several macros can be stored at the same time.
- Any macro can be defined as the startup macro. The startup macro runs each time the E-872.401 is switched on or rebooted.
- Processing a macro and stopping a macro can be linked to conditions. In this way, loops can be realized as well.
- Macros can call up themselves or other macros at several nesting levels.
- Variables can be set for the macro and in the macro itself and used in different operations.
- Input signals can be evaluated for conditions and variables.

Working with Macros

- The E-872.401 can save up to 32 macros simultaneously.
- Up to 10 nesting levels are possible in macros.
- Local and global variables can be used in macros.
- A macro is overwritten if a macro with the same name is rerecorded.
- For working with controller macros, it is recommended to use the ***Controller macros*** tab in PIMikroMove. There you can conveniently record, start, and manage controller macros.
- ThePITerminal or the PIMikroMove's ***Command entry*** window can be used for entering commands, e.g., for starting macros.

GCS commands in macros

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

- RBT for rebooting the E-872.401
- MAC BEG and MAC END for macro recording
- MAC DEL for deleting a macro

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

8.5.1 Recording Macros

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

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

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

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

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

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

- If you are working in PIMikroMove in the ***Controller macros*** tab: Click the ***Send macro to controller*** symbol and enter the macro name into a separate dialog window.

➔ *The macro has been stored in the nonvolatile memory of the E-872.401.*

4. If you want to check in PITerminal or in the PIMikroMove's ***Command entry*** window whether the macro was recorded correctly:
   a) Query which macros are saved in the E-872.401 by sending the MAC? command.
   b) Query the content of "macro name" with the MAC? macro name command.
5. If you want to check PIMikroMove's ***Controller macros*** tab to see whether the macro was recorded correctly:
   a) Click the ***Read list of macros from controller*** icon.
   b) Mark the macro to be checked in the list on the left-hand side and click the ***Load selected macro from controller*** icon.

Example macro: Move the axes one after the other

Axes 1 and 2 are to move 5000 steps respectively in a positive direction one after the other. A macro is recorded for this purpose. The macro starts the motion and waits until the axis has completed motion.

Record the macro by sending:

```
MAC BEG macro1
OSM 1 5000
WAC OSN? 1 = 0
OSM 2 5000
WAC OSN? 2 = 0
MAC END
```

8.5.2 Running the Macro

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

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

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

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

Variables can be used in macros. Setting is done via the VAR command and querying variable values via VAR?. Returns on query commands can be copied into variables with CPY.

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

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

   Specifying "string" is optional and stands for the values of local variables. The values only need to be specified when the macro contains corresponding local variables. The sequence of the values in the input must correspond to the numbering of the appropriate

local variables, starting with the value of the local variable 1. The individual values must be separated from each other by spaces.

3. If you want to check that the macro is running:
   - Query whether a macro is running on the controller by sending the #8 command.
   - Query the name of the macro currently running on the controller by sending the RMC? command.

8.5.3 Stopping the Macros

Stopping the macro can be linked to a condition with the MEX command. The command must be included in the macro.

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

8.5.4 Configuring a Startup Macro

Any macro can be defined as the startup macro. The startup macro runs each time the E-872.401 is switched on or rebooted.

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

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

8.5.5 Deleting Macros

A macro cannot be deleted while it is running.

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

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

8.5.6 Saving and Loading Macros

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

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

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

Saving Controller Macros on the PC

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

➔ *The **Save selected macros to PC** button becomes active when selecting one or more macros.*

3. Save the selected macros on the PC:
   a) Click the  button to open a directory selection window.
   b) Select the directory on the PC where you want to save the macros.
   c) Click ***Save***.

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

Loading Controller Macros from the PC to the E-872.401

1. Select the ***Controller macros*** tab in PIMikroMove's main window.
2. Load macros from the PC to the E-872.401:
   a) Click the  button to open a file selection window.
   b) Select the text files (<macro name>.txt) in the file selection window whose contents you want to load as a macro from the PC to the E-872.401.
   c) Click ***Open***.

➔ *For each selected text file (<macro name>.txt), the content is loaded as a macro <macro name> into the E-872.401.*

8.5.7 Example Macros

Example macro: Move the axes back and forth one after the other
The MOVWAI macro moves axes 1 and 2 one after the other. It moves the respective axes and waits until motion is completed.

The MOVLR macro calls the MOVWAI macro, first for motion in a positive direction, then for motion in a negative direction. The MOVLR macro uses a variable where the number of motion steps is defined. This variable is passed to the MOVWAI macro when called. Create the variable either in the MOVLR macro or each time the E-872.401 is rebooted.

Creating a variable:
```
VAR STEPS 5000
```

MOVLR macro:
```
MAC START MOVWAI ${STEPS}
MAC START MOVWAI -${STEPS}
```

MOVWAI macro:
```
OSM 1 $1
WAC OSN? 1 = 0
OSM 2 $1
WAC OSN? 2 = 0
```

Calling:
```
MAC START MOVLR
```

Example macro: Activate HID control for axis
The following macro activates HID control for axis 2.

Command	Explanation
`HIN 2 0`	Deactivate HID control for axis 2
`HIA 2 0 0 0`	Delete current configuration of HID control for axis 2
`HIA 2 3 1 2`	Set HID configuration for axis 2: Control velocity (3) via HID 1, HID axis 2
`SPA 2 0x61 0`	Set the value of the parameter 0x61 (invert direction of motion?) for axis 2 to 0 (do not invert)
`HIN 2 1`	Activate HID control for axis 2

8.5.8 Commands

#		Page
#24	Stop All Axes	82
#8	Query If Macro Is Running	82
C		**Page**
CPY	Copy Into Variable	83
D		**Page**
DEL	Delay The Command Interpreter	84
J		**Page**
JRC	Jump Relatively Depending On Condition	98
M		**Page**
MAC BEG	Call Macro Function: BEG	99
MAC DEF	Call Macro Function: DEF	99
MAC DEF?	Call Macro Function: DEF?	99
MAC DEL	Call Macro Function: DEL	99
MAC END	Call Macro Function: END	99
MAC ERR?	Call Macro Function: ERR?	100
MAC NSTART	Call Macro Function: NSTART	100
MAC START	Call Macro Function: START	100
MAC?	List Macros	101
MEX	Stop Macro Execution Due To Condition	102
R		**Page**
RBT	Reboot System	104
RMC?	List Running Macros	104
S		**Page**
STP	Stop All Axes	106
V		**Page**
VAR	Set Variable Value	107
VAR?	Get Variable Values	107

W		Page
WAC	Wait For Condition	108

8.5.9 Parameters

0x72	Ignore Macro Error?	Ignore macro error? 0 Stop macro on error (default) 1 Ignore error

8.6 Working with GCS Commands

8.6.1 GCS Command Syntax

Notation

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

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

Syntax

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

Exceptions:

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

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

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

Exception:

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

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

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

Sending a Command

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

1. Send
   `MOV␣1␣10.0␤`

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

Sending the Command with Several Arguments

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

Information

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

1. Send
   `MOV␣1␣17.3␣2␣2.05␤`

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

Sending commands without arguments

The position of all axes is to be queried.

1. Send
   `POS?␤`

➔ *Outputs the position of all axes. The response syntax is as follows:*
`{[<Argument>[{␣<Argument>}]"="]<Wert>␣␤}`
`[<Argument>[{␣<Argument>}]"="]<Wert>␤` *(for the last line)*

8.6.2 Variables

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

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

The following conventions apply to variable names:

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

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

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

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

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

8.7 Adapting Settings

The properties of the E-872.401 and the mechanics connected are stored in the E-872.401 as parameter values.

The parameters can be divided into the following categories:

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

Write permission for the parameters is determined by command levels.

Each parameter is in the E-872.401's volatile and nonvolatile memory. The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the E-872.401. The values in the volatile memory determine the current behavior of the system.

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

Use the HPA? command to query the parameters available in the E-872.401.

NOTICE

Unsuitable parameter settings!

The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the E-872.401 and take effect immediately. Unsuitable parameter settings can cause damage to the mechanics.

► Change parameter values only after careful consideration.

► Save the current parameter values to the PC before you make changes in the nonvolatile memory.

8.7.1 Parameter Commands

The following general commands are available for changing parameters:

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

8.7.2 Saving Parameter Values in a Text File

Overview

The E-872.401 is configured via parameters, e.g., for adapting to the mechanics. Changing parameter values can cause undesirable results.

- ► Create a backup copy on the PC before changing the parameter settings of the E-872.401. You can then restore the original settings at any time.
- ► Create an additional backup copy with a new file name each time after optimizing the parameter values or adapting the E-872.401 to specific mechanics.

Parameter values saved in a text file on the PC can be loaded back to the E-872.401 in PIMikroMove or PITerminal. The ***Send file...*** button is available for this purpose in the send command window. Before loading into the E-872.401, the individual lines of the text files must be converted into command lines that contain the corresponding SPA or SEP commands.

Requirements

✓ You have established communication between the E-872.401 and the PC (p. 43) with PIMikroMove or PITerminal.

Saving Parameter Values in a Text File

1. If you are using PIMikroMove, open the window for transmitting commands: Select ***Tools > Command entry*** in the main window or press F4 on the keyboard.
   After communication has been established, the main window is opened PITerminal automatically and commands can be sent.
2. Query the parameter values that you want to create a backup.
   - If you want to save the parameter values from the volatile memory of the E-872.401: Send the SPA? command.
   - If you want to save the parameter values from the nonvolatile memory of the E-872.401: Send the SEP? command.
3. Click the ***Save...*** button.
4. The ***Save content of terminal as textfile*** window opens.
5. Save the queried parameter values to a text file on your PC in the ***Save content of terminal as textfile*** window.

8.7.3 Changing Parameter Values

Overview

The following procedure is generally recommended for changing parameter values:

1. Change the parameter values in the volatile memory.
2. Check whether the E-872.401 works correctly with the changed parameter values.
   If so:
   - Write the changed parameter values into the nonvolatile memory.

   If not:
   - Change and check the parameter values in the volatile memory again.

Requirements

✓ If you want to change parameter values in the E-872.401's nonvolatile memory: You have saved the E-872.401's parameter values in a text file on the PC (p. 69).

✓ You have established communication between the E-872.401 and the PC (p. 43) with PIMikroMove or PITerminal.

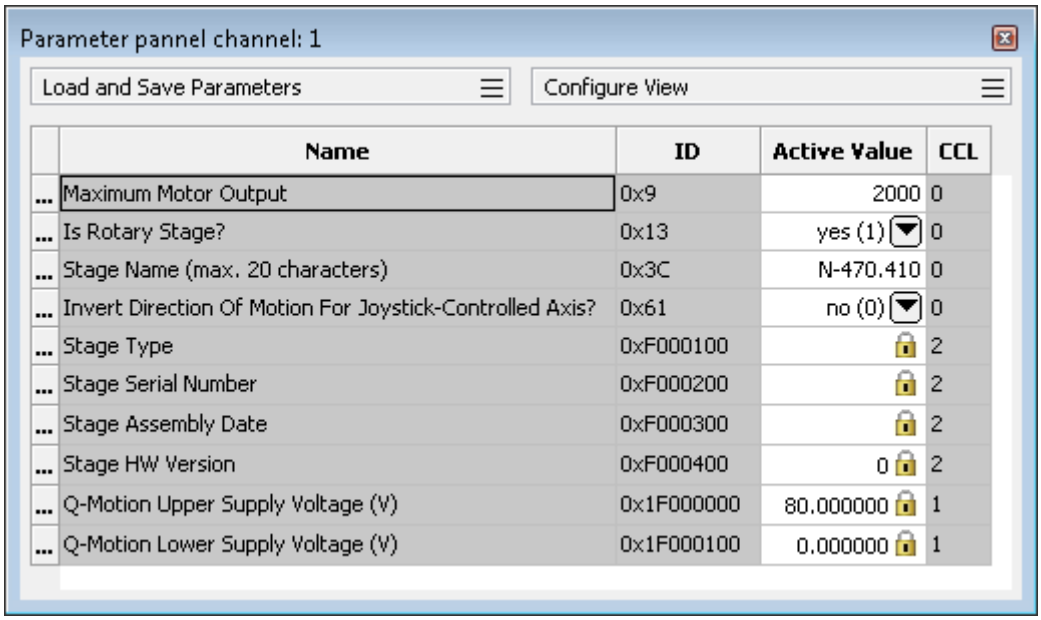
Information

Write access for the parameters of the E-872.401 is defined by command levels. After the controller is switched on or rebooted, the active command level is always 0. On command levels > 1, write access is only available to PI service personnel.

Changing Parameter Values

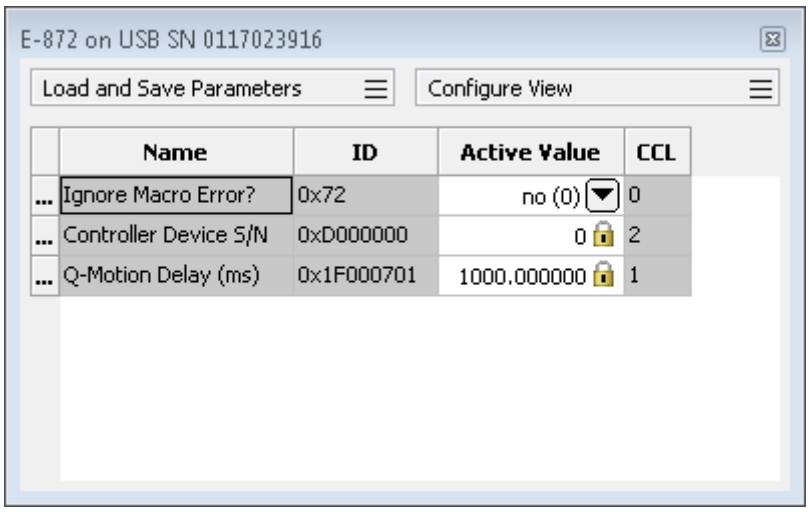
1. If you want to change the axis-related parameters of the E-872.401:
   a) Open the parameter window for the connected positioner in the PIMikroMove's main window by clicking the corresponding line in the ***Drive channels*** tab with the right mouse button and selecting ***Show drive channel parameter window*** in the context menu.
   b) If the parameter to be modified is not included in the list on the right-hand side of the window, click ***Configure View > Select parameters...*** and add it to the list. You can also display certain groups of parameters or all axis-related parameters.

➔ *The list of channel-related parameters is displayed.*



2. If you want to change the system-related parameters of the E-872.401:
   a) Open the window for the system-related parameters of the E-872.401 in the main window of PIMikroMove by selecting ***E-872.401 > Show system parameters*** in the menu.

➔ *The list of system-related parameters is displayed.*



3. If you want to change the parameter values in the E-872.401's **volatile memory**, you have the following options:

a) Type the new parameter values into the corresponding input field in the ***Active Value*** column and press the enter key on the PC's keyboard or click the mouse button outside of the input field.

➔ *The modified parameter value is transferred to the E-872.401's volatile memory.*

b) Click ***Load and Save Parameters -> Load all startup parameters of the axis / system from controller***.

➔ *The values of all channel-related / system-related parameters are loaded from the E-872.401's nonvolatile memory.*

c) Click ***Load and Save Parameters > Load parameters from stage database...*** in the parameter window.

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

➔ *A selected parameter set for channel A is loaded from the positioner database.*

4. If you want to change parameter values in the E-872.401's **nonvolatile memory**, you have the following options:

a) Type the new parameter value into the corresponding input field in the list's ***Startup Value*** column and press the enter key on the PC's keyboard or click the mouse button outside of the input field.

➔ *The changed parameter value is transferred to the E-872.401's volatile memory.*

b) Click ***Load and Save Parameters -> Save all currently active axis / system parameters as startup parameters to controller***.

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

➔ *The values of all channel-related / system-related parameters are written from the E-872.401's volatile memory to the nonvolatile memory.*

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

8.7.4 Creating or Changing Parameter Sets for Positioners

Overview

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

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

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

- You can create a new positioner type easily by changing an existing parameter set in PIMikroMove and saving it under a new name.
- You can open and edit the positioner database directly with the PIStages3Editor.

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

Requirements

✓ You have installed the latest version of the PISTAGES3.DB database onto your PC.

✓ If PI provided a custom positioner database for your positioner, the dataset was imported into PIStages3.

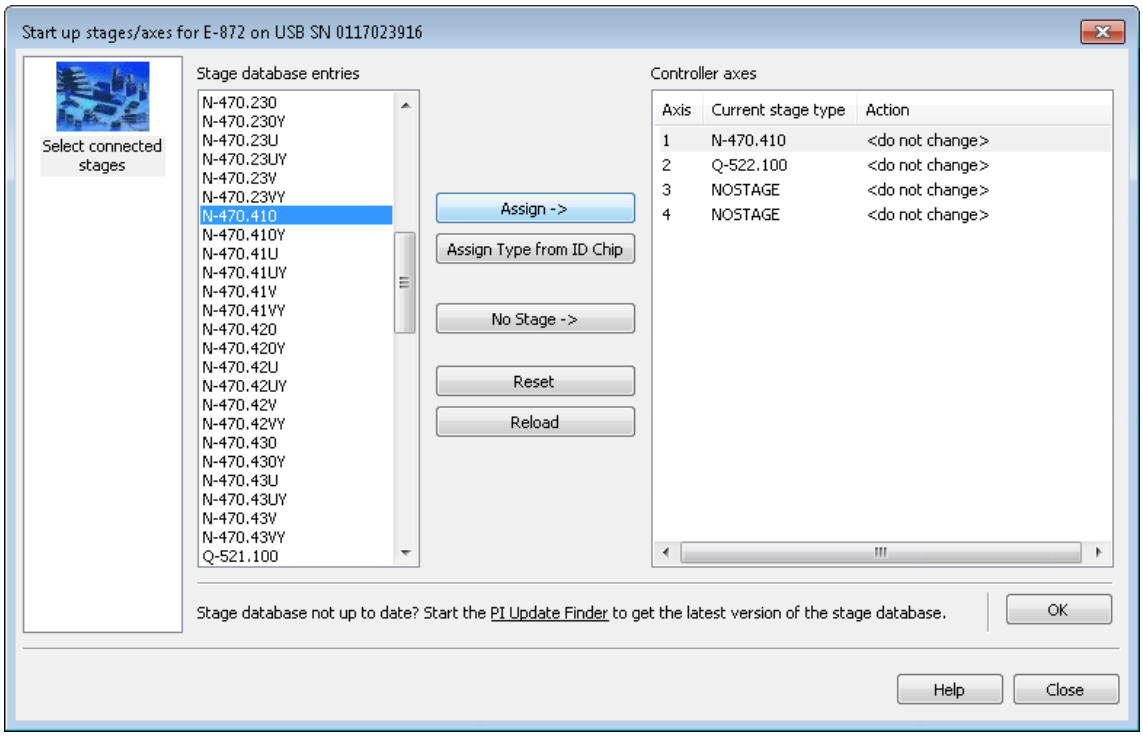
✓ PIMikroMove has established communication between the E-872.401 and the PC.

Creating a Parameter Set for Positioners

1. Select the ***E-872.401 > Select connected stages...*** menu item in the main window of PIMikroMove.

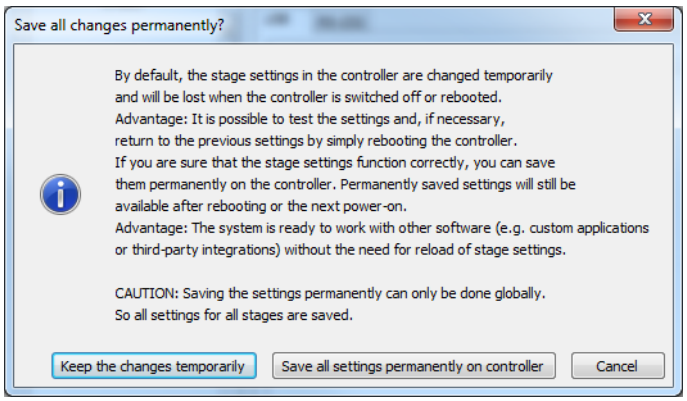
➔ *The **Start up stages/axes for E-872.401** window opens and the **Select connected stages** step is active.*

2. Select an appropriate type of positioner during the ***Select connected stages*** step:
   a) Click ***Assign Type from ID Chip***.
   or
   a) Highlight the positioner in the ***Stage database entries*** list.
   b) Click ***Assign***.



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

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

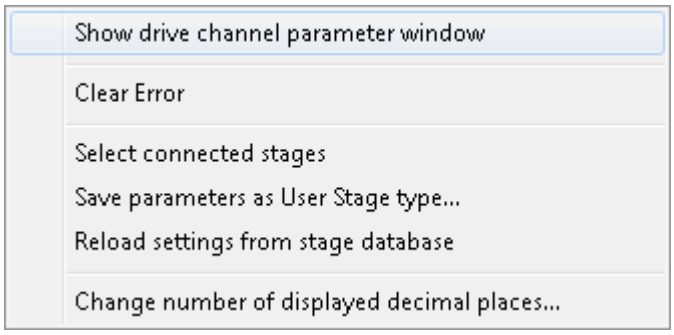


3. Click ***Keep the changes temporarily*** in the ***Save all changes permanently*** dialog to load the parameter settings into the volatile memory of the E-872.401.

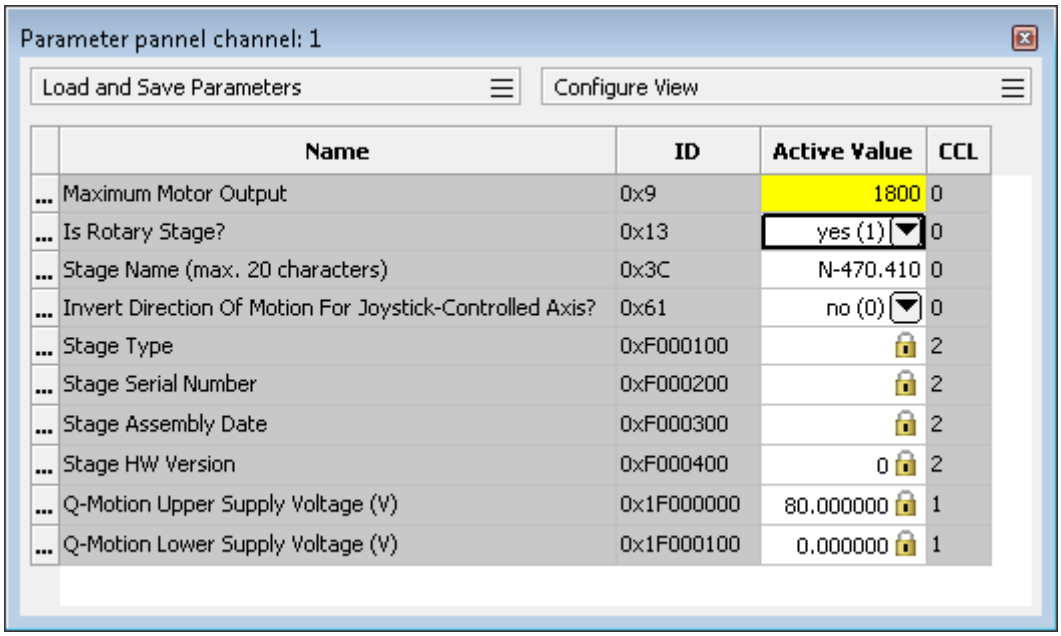
➔ *The **Start up stages/axes** window changes to the **Start up axes** step.*

4. Click ***Close*** to close the ***Start up stages/axes*** window.

5. Open the parameter window for the selected positioner type in the PIMikroMove's main window by clicking the corresponding line in the ***Drive channels*** tab and selecting the ***Show drive channel parameter window*** in the context menu.

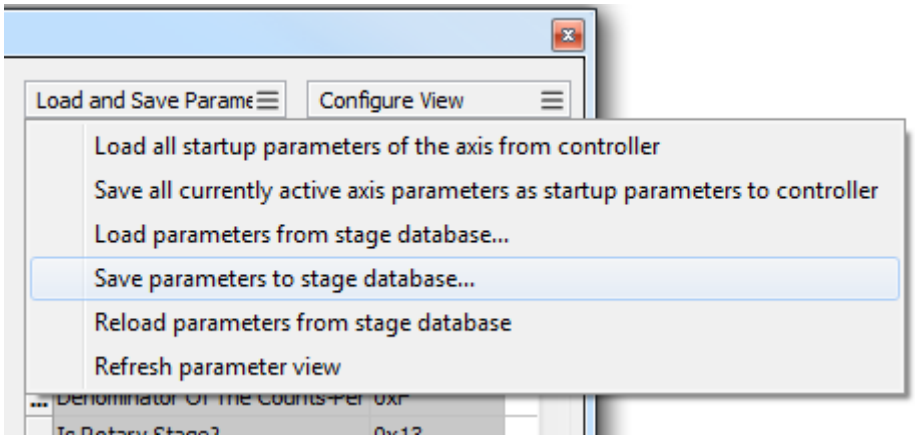


6. Enter new values for the parameters to be changed:



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

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



8. The ***Save Parameters as User Stage Type*** dialog opens.
9. Save the changed parameter values as new positioner type in ***Dialog Save Parameters as User Stage Type***:
   a) Leave the entry in the ***Parameters of axis*** field unchanged.
   b) Enter the name for the new positioner type into the ***Save as*** field.
   c) Click ***OK***.

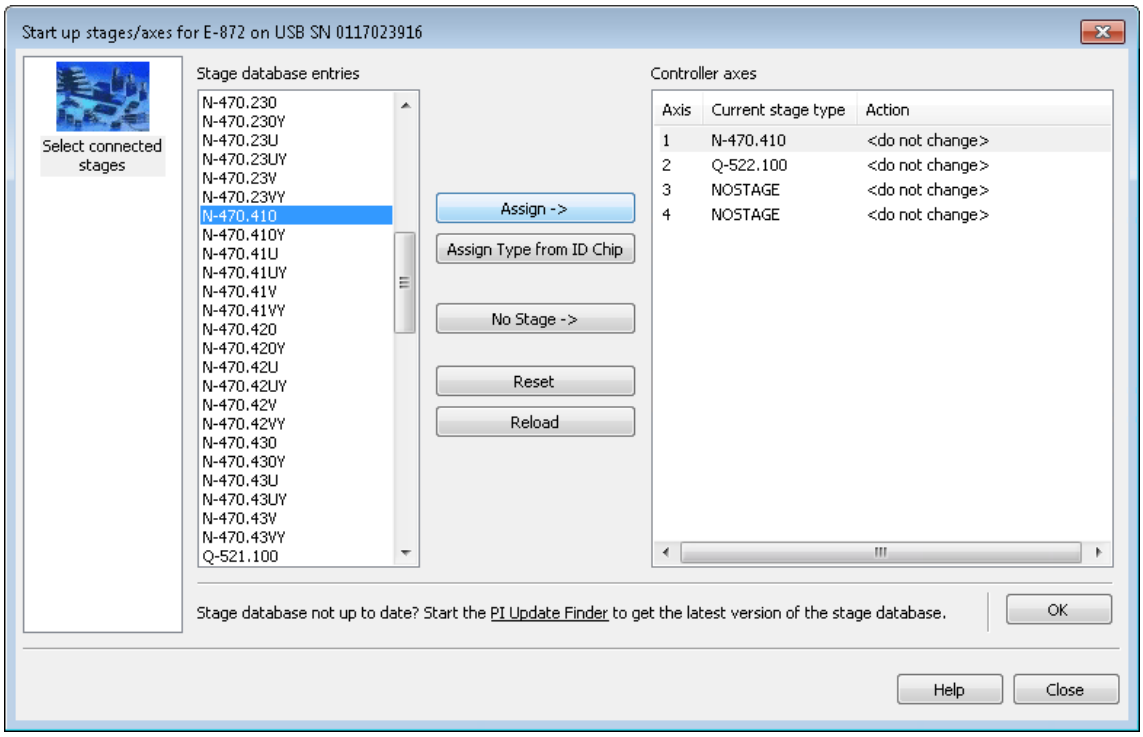
➔ *The new positioner type was saved in the positioner database. The displayed positioner type was updated in PIMikroMove. The new positioner type is also available immediately for selection in the* ***Select connected stages*** *step.*

Changing a Positioner's Parameter Set

1. Select the ***E-872.401 > Select connected stages...*** menu item in the main window of PIMikroMove.

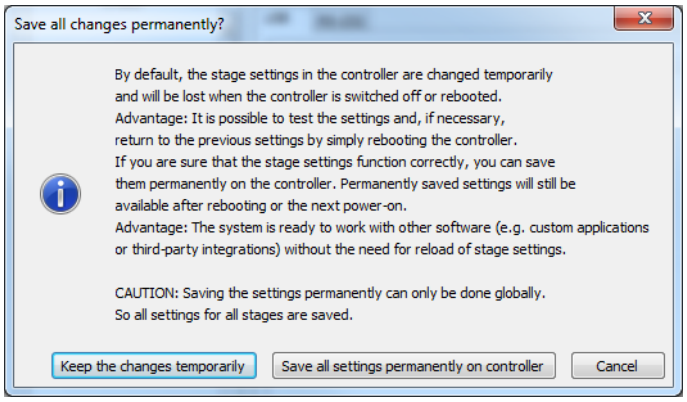
➔ *The* ***Start up stages/axes for E-872.401*** *window opens and the* ***Select connected stages*** *step is active.*

2. Select a positioner type that you created during the ***Select connected stages*** as described above:
   a) Click ***Assign Type from ID Chip***.
   or
   a) Highlight the positioner in the ***Stage database entries*** list.
   b) Click ***Assign***.



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

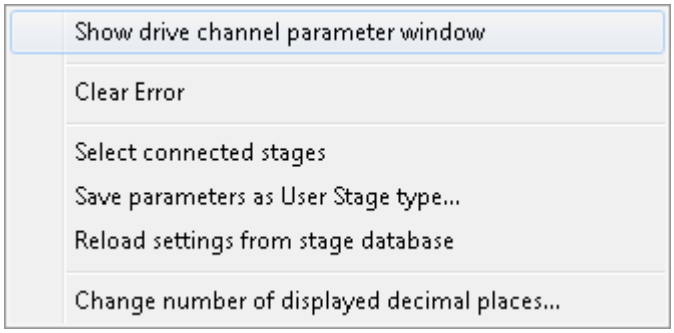
➔ *The **Save all changes permanently?** dialog is opened.*



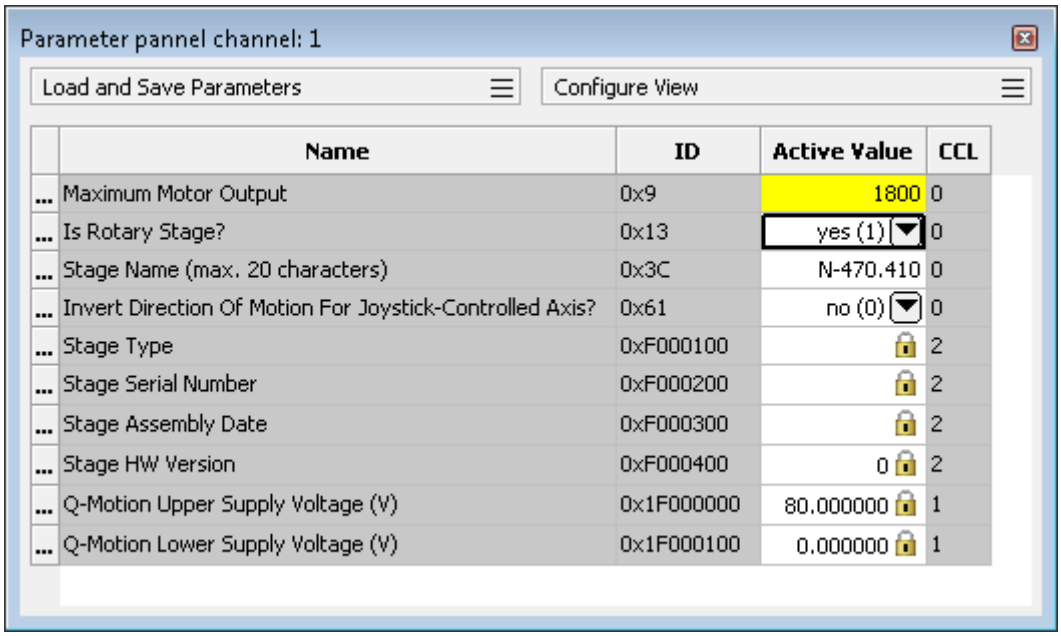
3. Click ***Keep the changes temporarily*** in the ***Save all changes permanently*** dialog to load the parameter settings into the volatile memory of the E-872.401.

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

4. Click ***Close*** to close the ***Start up stages/axes*** window.
5. Open the parameter window for the selected positioner type in the PIMikroMove's main window by clicking the corresponding line in the ***Drive channels*** tab and selecting the ***Show drive channel parameter window*** in the context menu.

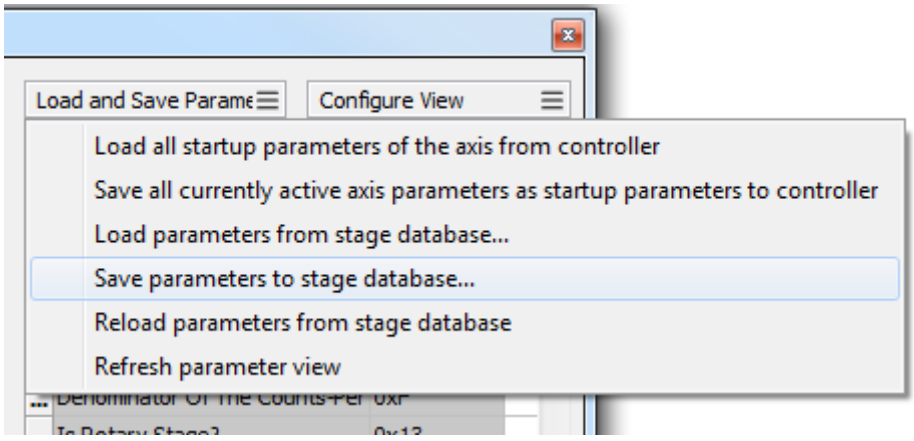


6. Enter new values for the parameters to be changed:



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

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



8. The ***Save Parameters as User Stage Type*** dialog opens.
9. Save the modified parameter values of the positioner type in the ***Save Parameters as User Stage Type*** dialog:
   a) Leave the entry in the ***Parameters of axis*** field unchanged.
   b) Leave the entry in the ***Save as*** field unchanged.
   c) Click ***OK***.
   d) Click ***Change settings*** in the ***Stage type already defined*** dialog. The ***Save Parameters as User Stage Type*** dialog closes automatically after a short time.

➔ *The parameter values of the positioner type were updated in the positioner database and in the main window of PIMikroMove.*

9 Command Reference

VAR?	Get Variable Values	107
VER?	Get Versions Of Firmware And Drivers	108
W		**Page**
WAC	Wait For Condition	108
WPA	Save Parameters To Nonvolatile Memory	109

#5 Request Motion Status

Used in: Triggering motion in open-loop operation (27)

Description: Queries the motion status of the axes.

Format: **#5**

#5 corresponds to the ENQ control character in ISO/IEC 6429.

Response: <MotionStatus>

<MotionStatus> motion status (HEX)

<MotionStatus> is bit-mapped. Each axis corresponds to one bit and the bit value corresponds to the axis number, e.g.,

Response	Description
1	Axis 1 is moving
2	Axis 2 is moving
5	Axis 3 and 1 are moving

#7 Request Controller Ready Status

Used in: Commandable Items (22)

Description: Queries the controller's ready state.

The controller is ready as soon as a new command can be executed.

Format: **#7**

#7 corresponds to the ᴮE_L control character in ISO/IEC 6429.

Response: <ReadyStatus>

<ReadyStatus> ready state (HEX)

Possible answers:

Response from controller	Character in ISO/IEC 8859-1	Description
0xB1	±	Controller is ready
0xB0	°	Controller is not ready

Troubleshooting: The response characters may be displayed differently in other character sets.

#8 Query If Macro Is Running

Used in: (63)

Description: Tests if a macro is running on the controller.

Format: **#8**

#8 corresponds to the BS control character in ISO/IEC 6429.

Response: <MacroRunning>

<MacroRunning> Macro is running (UINT)

<MacroRunning>	Description
1	A macro is running.
0	No macro is running.

#24 Stop All Axes

Used in: (64), Triggering motion in open-loop operation (27)

Description: Stops all axes abruptly.

Stops all motion started by motion commands (e.g., MOV, MVR, MVE, STE, SMO), referencing commands (FNL, FPL, FRF), and macros.

Also stops the macro.

Sets the error code to 10.

After the axes are stopped, their target positions are set to their current positions.

This command is identical in function to STP (p. 106) except only one character is sent via the interface. Therefore, #24 can also be used when the controller is doing time-consuming tasks.

Format: **#24**

#24 corresponds to the CAN control character in ISO/IEC 6429.

*IDN? Get Device Identification

Used in: Commandable Items (22)

Description: Queries the device identification string.

Format: **`*IDN?`**

Response: <DeviceInformation> LF

<DeviceInformation> Device identification string with controller name, serial number, and firmware version. (STRING)

CCL Set Command Level

Used in: (21), Parameter Commands (69)

Description: Changes the active command level.

The command level determines the availability of commands and write access to system parameters.

Command level: 0 - standard; 1 - expert mode; > 1 - PI service mode

Command levels > 1 cannot be activated; they are intended for PI service personnel. Contact the customer service department if there are problems with the parameters for command level 2 or higher (p. 128).

The active command level is always 0 when the electronics are switched on or rebooted.

The HPA? (p. 92) command lists the parameters and includes information on which command level allows write access to them.

Format: `CCL␣<Level>[␣<PSWD>]`

Arguments:

<Level>	Controller command level (UINT)
<PSWD>	Password for changing to the corresponding command level Password for changing to command level 1: advanced

Troubleshooting: Wrong password

CCL? Get Command Level

Used in: (21), Parameter Commands (69)

Description: Queries the active command level.

Format: `CCL?`

Response: <Level> LF

<Level>	Command level (UINT)

CPY Copy Into Variable

Used in: Running the Macro (63)

Description: Copies the response to a command into a variable.

Format: `CPY␣<Variable>␣<CMD?>`

Arguments:

<Variable>	Name of the variable to be copied to
<CMD?>	Query command that responds with a single value

CST? Get Assignment Of Stages To Axes

Used in: Commandable Items (22)

Description: Queries the name of the positioner type that is configured for the specified axis.

The positioner name is read from parameter 0x3C (p. 112). If the parameter has the value "NOSTAGE", the axis is deactivated.

Format: `CST? [{<AxisID>}]`

Arguments:

<AxisID>	Axis ID

Response: {<AxisID>=<String>LF}

<AxisID>	Axis ID
<String>	Name of the positioner type (STRING)

CSV? Get Current Syntax Version

Used in: (21)

Description: Get current GCS syntax version used in the firmware.

Format: `CSV?`

Response: <SyntaxVersion>

<SyntaxVersion>	GCS syntax version (STRING)

<SyntaxVersion>	Description
1.0	GCS syntax version 1.0
2.0	GCS syntax version 2.0

DEL Delay The Command Interpreter

Used in: Running the Macro (63)

Description: Delays running the macro for a fixed period of time.

DEL can only be used in macros.

Note: Do not mix up DEL (delayed) with MAC DEL (p. 99) (deletes macros).

Format: `DEL␣<uint>`

Arguments:

<uint>	Delay time span in milliseconds [UINT]

DIO? Get Digital Input Lines

Used in: Digital Input Signals (53)

Description: Queries the status of a digital input line.

Use TIO? (p. 107) to query the number of available digital I/O lines.

Format: **DIO?[{␣<DIOID>}]**

Arguments:

<DIOID>	Digital input line of the electronics

The state of all digital input lines is queried and output as bit-mapped hexadecimal number if no arguments are specified.

Response: {<DIOID>=<InputOn>LF}

<DIOID>	Digital input line of the electronics
<InputOn>	Digital input line state (HEX)

ERR? Get Error Number

Used in: Restoring the E-872.401's Operational Readiness (52)

Description: Queries the error code of the last error that occurred and resets the error code to 0.

Only the last error is buffered; therefore, in the case of a problem, ERR? should be called after each command.

Format: **ERR?**

Response: <ErrorNumber>

<ErrorNumber>	Code for the last error (INT)

Troubleshooting: Communication breakdown

HDT Set HID Default Lookup Table

Used in: HID Control Configuration (55)

Description: Assigns a lookup table to the specified axis of the specified HID.

Format: **HDT{␣<HIDeviceID>␣<HIDeviceAxis>␣<HIDTableID>}**

Arguments:

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceAxis>	HID axis
<HIDTableID>	Lookup table identifier

HDT? Get HID Default Lookup Table

Used in: HID Control Configuration (55)

Description: Queries the currently assigned lookup table for the specified axis of an HID.

Format: `HDT?[{␣<HIDeviceID>␣<HIDeviceAxis>}]`

Arguments:

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceAxis>	HID axis

Response: {<HIDeviceID>␣<HIDeviceAxis>=<HIDTableID>LF}

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceAxis>	HID axis
<HIDTableID>	Lookup table identifier

HIA Configure Control Done By HID Axis

Used in: HID Control Configuration (55), Triggering motion in open-loop operation (27)

Description: Configures control of electronics' axes via HID axes (HID control).

Assigns an HID axis to the specified motion variable.

The configuration of the HID control is saved only to the volatile memory (RAM) of the electronics.

HID control may not be activated for the corresponding axis.

Motion variables that can be controlled by HIDs (depending on the controller; use the HIA? (p. 88) command to query the current configuration of the HID controlled motion variables):

<MotionParam>	Description
0	Deletes the current configuration of the HID control. Can be sent from <HIDeviceID> and <HIDeviceAxis> without specification.
1	Absolute target position The lookup table value corresponding to the current displacement of the human interface device axis is mapped to the travel range of the E-872.401 axis to be controlled. The travel range limits are specified by the values of parameters 0x30 and 0x15 can be queried with TMN? and TMX?.
2	Relative target position Each pulse received (if applicable: Each mechanical detent) triggers relative motion over the distance set with the SST command. No lookup tables are used to control the relative target position.
3	Velocity of the axis Product of the lookup table value corresponding to the current displacement of the HID axis and the currently valid maximum velocity of the controller axis.
4	Maximum velocity of the axis Product of the lookup table value corresponding to the current displacement of the HID axis and the currently valid maximum velocity of the controller axis. Motion variable 4 can only be assigned to an HID axis when motion variable 3 has been assigned to a different HID axis.

Format: `HIA{␣<AxisID>␣<MotionParam>␣<HIDeviceID>␣<HIDeviceAxis>}`

Arguments:

<AxisID>	Axis ID
<MotionParam>	Axis motion variable
<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceAxis>	HID axis

Troubleshooting: <MotionParam> has the value zero, i.e., function to be controlled has not been selected for the axis
<HIDeviceID> has the value zero, i.e., HID has not been selected
<HIDeviceAxis> has the value zero, i.e., HID's axis has not been selected for HID control

HIA? Get Configuration Of Control Done By HID Axis

Used in: HID Control Configuration (55)

Description: Queries an HID's axis assigned to the specified motion variable of the specified motion variable.

Format: `HIA?[{␣<AxisID>␣<MotionParam>}]`

Arguments:

<AxisID>	Axis ID
<MotionParam>	Axis motion variable

Response: {<AxisID>␣<MotionParam>=<HIDeviceID>␣<HIDeviceAxis>LF}

<AxisID>	Axis ID
<MotionParam>	Axis motion variable
<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceAxis>	HID axis

HIB? Get State Of HID Button

Used in: HID Control Configuration (55)

Description: Queries the current state of the specified button of the specified HID.

Format: `HIB?[{␣<HIDeviceID>␣<HIDeviceButton>}]`

Arguments:

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceButton>	HID button

Response: {<HIDeviceID>␣<HIDeviceButton>=<HIDButtonState>LF}

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceButton>	HID button
<HIDButtonState>	Button status (INT)

Any possible values of <HIDButtonState> depend on the button type. The value range can be queried with HIS? (p. 90).

HIE? Get Deflection Of HID Axis

Used in: HID Control Configuration (55)

Description: Queries the current displacement of the specified axis of the specified HID.

Format: `HIE?[{␣<HIDeviceID>␣<HIDeviceAxis>}]`

Arguments:

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceAxis>	HID axis

Response: {<HIDeviceID>␣<HIDeviceAxis>=<HIDDeflection> LF }

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDeviceAxis>	HID axis
<HIDDeflection>	Displacement of the HID's axis (FLOAT)

<HIDDeflection> is a value between -1.0 and 1.0. A value almost equal to 0.0 corresponds to the axis' center position, -1.0 respectively 1.0 of the maximum displacement in a negative respectively positive direction.

HIN Set Activation State For HID Control

Used in: HID Control Configuration (55), Triggering motion in open-loop operation (27)

Description: Sets the status of HID control for the specified axis.

Format: `HIN{␣<AxisID>␣<HIDControlState>}`

Arguments:

<AxisID>	Axis ID
<HIDControlState>	HID control's activation state (BOOL)

Troubleshooting: Illegal axis identifier

HID control is not suitably configured (p. 87)

HIN? Get Activation State Of HID Control

Used in: HID Control Configuration (55)

Description: Queries status of HID control for the specified axis.

Format: `HIN?[{␣<AxisID>}]`

Arguments:

<AxisID>	Axis ID

Response: {<AxisID>=<HIDControlState> LF }

<AxisID>	Axis ID
<HIDControlState>	HID control's activation state (BOOL)

HIS? Get Configuration Of HI Device

Used in: Commandable Items (22), HID Control Configuration (55)

Description: Queries the specified property for the specified operating element of an HID.

Format: `HIS?[{␣<HIDeviceID>␣<HIDItemID>␣<HIDPropID>}]`

Arguments:

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDItemID>	HID operating element
<HIDPropID>	Property of the operating element

If no arguments are specified, information is queried on the supported operating elements of all HIDs.

Response: {<HIDeviceID>␣<HIDItemID>␣<HIDPropID>=<HIDPropValue>LF}

<HIDeviceID>	Identifier of an HID connected to the electronics
<HIDItemID>	HID operating element
<HIDPropID>	Property of the operating element
<HIDPropValue>	The property of the operating element is set to this value (STRING)

Possible values:

<HIDPropID>	Description
	Axis_<x>: HID axis, e.g., joystick axis or continuous slider, <x> indicates the identifier
	Button_<x>: HID button, <x> indicates the identifier
	Led_<x>: Output unit, e.g., LED or vibration motor on/off time, <x> indicates the identifier
2	Operating element status, e.g., axis displacement or LED activation status (FLOAT)
3	Name of the operating element (STRING)
4	HID Name (STRING)
5	Smallest possible value for the status of a "Button"- or "Led"-type operating element (INT)
6	Largest possible value for the status of a "Button"- or "Led"-type operating element (INT)

HIT Fill HID Lookup Table

Used in: HID Control Configuration (55)

Description: Fills the specified lookup table with values.

HIT can only be used to fill user-defined tables. Tables with identifier ≤ 100 are predefined and write-protected.

The first point of a lookup table corresponds to the maximum axis displacement of the HID in the negative direction; the 256th point corresponds to the maximum displacement in the positive direction. The values for points 1 to maximally 127 have a negative sign by default, while the remaining values have a positive sign. The sign of the values determines the direction of motion of the HID-controlled axis. Parameter 0x61 (p. 112) can be used to reverse the assigned direction that is specified by the values in the lookup table for an HID-controlled axis.

The HDT (p. 85) command assigns the lookup tables to HID axes.

Format: `HIT{␣<HIDTableID>␣<HIDTableAddr>␣<HIDTableValue>}`

Arguments:

<HIDTableID>	Lookup table identifier
<HIDTableAddr>	Index of a point in the lookup table
<HIDTableValue>	Value of the point with the index <HIDTableAddr> (FLOAT, -1.0...1.0)

HIT? Get HID Lookup Table Values

Used in: HID Control Configuration (55)

Description: Queries the values of the specified points in the specified lookup table.

Format: `HIT?[␣<StartPoint>[␣<NumberOfPoints>[{␣<HIDTableID>}]]]`

Arguments:

<StartPoint>	Index of the first point that is queried
<NumberOfPoints>	Number of points to be queried per lookup table
<HIDTableID>	Lookup table identifier

Response: (Data in GCS array format)

HLP? Get List Of Available Commands

Used in: (21)

Description: Lists a help string which contains all commands available.

Format: `HLP?`

Response: (List of available commands)

HLT Halt Motion Smoothly

Used in: Triggering motion in open-loop operation (27)

Description: Stops motion for the specified axis while considering the maximum set deceleration.

Sets the error code to 10.

Does not apply to trajectories: HLT also triggers an abrupt stop of motion when a trajectory is being followed.

Format: `HLT[{ <AxisID>}]`

Arguments:

<AxisID>	Axis ID

Troubleshooting: Illegal axis identifier

HPA? Get List Of Available Parameters

Used in: (21), Adapting Settings (69), Parameter Commands (69)

Description: Shows a help text that contains all available parameters with a short description.

Format: `HPA?`

Response: List of available parameters in the format:

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

<PamID>	Parameter ID
<CmdLevel>	Command level for write access to the parameter
<MaxItem>	Maximum number of elements of the same type that are affected by the parameter
<DataType>	Data type of the parameter value
<FunctionGroupDescription>	Name of the function group which the parameter belongs to
<ParameterDescription>	Name of the parameter
<PossibleValue>	Possible value
<ValueDescription>	Value description

HPV? Get Parameter Value Description

Description: Displays a help text that contains possible parameter values.

Format: **HPV?**

Response:

#Possible␣parameter␣values␣are: LF

{<PamID>␣<ItemID>=<ListType>[{→|<PossibleValue>=<ValueDescription>}] LF }

#CCL␣levels␣are: LF

{<PamID>␣<ItemID>=<CmdLevel> LF }

end of help

<PamID>	Parameter ID
<ItemID>	Element of the electronics <ItemID> = 0: Description applies to all elements
<ListType>	Value list type <ListType> = 0: Parameter does not apply to this element <ListType> = 1: List of possible values <ListType> = 2: Minimum and maximum value
<PossibleValue>	Possible value
<ValueDescription>	Value description
<CmdLevel>	Command level for write access to the parameter

IFC Set Interface Parameters Temporarily

Used in: E-872.401 Interfaces (16)

Description: Configures the interface parameters in the volatile memory.

The changed interface parameters are active immediately. The PC's interface configuration may also have to be changed and the connection to the electronics re-established.

The configuration of the interface parameters is saved only to the volatile memory (RAM) of the electronics. Use IFS (p. 96) to change interface parameters in the nonvolatile memory.

Format: `IFC{␣<InterfacePam>␣<PamValue>}`

Arguments:

<InterfacePam>	Interface parameters (STRING)
<PamValue>	Parameter value

Possible values

<InterfacePam>	<PamValue>	Description
RSBAUD	9600, 19200, 38400, 57600 or 115200	Baud rate for the RS-232 interface
IPADR	<UINT>.<UINT>.<UINT>.<UINT>:50000	IP address and port for the TCP/IP interface Port 50000 cannot be changed. IPADR is only used when IPSTART = 0.
IPSTART	0, 1	<PamValue> = 0: The IP address defined by IPADR is used <PamValue> = 1: DHCP is used (default)
IPMASK	<UINT>.<UINT>.<UINT>.<UINT>	TCP/IP interface subnet mask
IPGTWAY	<UINT>.<UINT>.<UINT>.<UINT>	Default gateway for TCP/IP communication

IFC? Get Current Interface Parameters

Used in: E-872.401 Interfaces (16)

Description: Queries the interface parameter values in the volatile memory.

Format: `IFC?[{␣<InterfacePam>}]`

Arguments: <InterfacePam> Interface parameters (STRING)

Response: {<InterfacePam>=<PamValue>LF}

<InterfacePam> Interface parameters (STRING)

<PamValue> Parameter value

in the volatile memory

Possible values for <InterfacePam>:

<InterfacePam>	Description
RSBAUD	Baud rate for the RS-232 interface
IPADR	IP address and port for the TCP/IP interface
IPSTART	Startup behavior (DHCP status) of the TCP/IP interface
IPMASK	TCP/IP interface subnet mask
MACADR	Mac address (unique network hardware address)
IPGTWAY	Default gateway for TCP/IP communication

IFS Set Interface Parameters As Default Values

Used in: E-872.401 Interfaces (16)

Description: Configures the interface parameters in the nonvolatile memory.

The changed interface parameters are active after the next reboot. The PC's interface configuration may also have to be changed.

Notice: Note that the number of write cycles in the nonvolatile memory is limited. Therefore, save to the nonvolatile memory only when necessary.

Use IFC (p. 94) to change the interface parameters in the volatile memory (RAM).

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

Arguments:

<Pswd> Password for writing to the nonvolatile memory

Default value is "100".

<InterfacePam> Interface parameters (STRING)

<PamValue> Parameter value

Possible values

<InterfacePam>	<PamValue>	Description
RSBAUD	9600, 19200, 38400, 57600 or 115200	Baud rate for the RS-232 interface
IPADR	<UINT>.<UINT>.<UINT>.<UINT>:50000	IP address and port for the TCP/IP interface Port 50000 cannot be changed. IPADR is only used when IPSTART = 0.
IPSTART	0, 1	<PamValue> = 0: The IP address defined by IPADR is used <PamValue> = 1: DHCP is used (default)
IPMASK	<UINT>.<UINT>.<UINT>.<UINT>	TCP/IP interface subnet mask
IPGTWAY	<UINT>.<UINT>.<UINT>.<UINT>	Default gateway for TCP/IP communication

IFS? Get Interface Parameters As Default Values

Used in: E-872.401 Interfaces (16)

Description: Queries the values of the interface parameters in the nonvolatile memory.

Format: `IFS?[{␣<InterfacePam>}]`

Arguments:

<InterfacePam>	Interface parameters (STRING)

Response: {<InterfacePam>=<PamValue>LF}

<InterfacePam>	Interface parameters (STRING)
<PamValue>	Parameter value

in the nonvolatile memory

Possible values for <InterfacePam>:

<InterfacePam>	Description
RSBAUD	Baud rate for the RS-232 interface
IPADR	IP address and port for the TCP/IP interface
IPSTART	Startup behavior (DHCP status) of the TCP/IP interface
IPMASK	TCP/IP interface subnet mask
MACADR	Mac address (unique network hardware address)
IPGTWAY	Default gateway for TCP/IP communication

JRC Jump Relatively Depending On Condition

Used in: Running the Macro (63)

Description: Jumps relative to a specified number of program lines within a macro.

Jumps irrespective of a specified condition.

Can only be used in macros.

Possible relational operators:

<OP>	Description
=	Equal
!=	Not equal
<=	Smaller than or equal
<	Smaller than
>=	Larger than or equal
>	Larger than

Format: `JRC␣<Jump>␣<CMD?>␣<OP>␣<Value>`

Arguments:

<Jump>	Size of relative jump
<CMD?>	Query command that responds with a single value
<OP>	Relational operator
<Value>	Relational value with <CMD?>

Troubleshooting: Wrong jump target (<Jump>) specified

Wrong relational operator (<OP>) specified

MAC Call Macro Function

Description: Calls a macro function.

Possible macro functions are described separately:

- MAC BEG (p. 99)
- MAC DEF (p. 99)
- MAC DEF? (p. 99)
- MAC DEL (p. 99)
- MAC END (p. 99)
- MAC ERR? (p. 100)
- MAC NSTART (p. 100)
- MAC START (p. 100)

Format: `MAC␣<Keyword>{␣<Parameter>}`

Arguments:

<Keyword>	Macro function called
<Parameter>	Function-dependent parameters

Troubleshooting: Macro recording is active

MAC BEG Call Macro Function: BEG

Used in: (62)

Description: Start recording a macro.

Recording is stopped by MAC END (p. 99).

Format: `MAC␣BEG␣<MacroName>`

Arguments: <MacroName> Macro name

MAC DEF Call Macro Function: DEF

Used in: (64)

Description: Specifies a macro as startup macro.

Format: `MAC␣DEF␣<MacroName>`

Arguments: <MacroName> Name of the macro that is specified as startup macro

A startup macro is not used when no arguments are specified.

MAC DEF? Call Macro Function: DEF?

Used in: (64)

Description: Queries the name of the startup macro.

Format: `MAC␣DEF?`

Response: [<MacroName>] LF

<MacroName> Startup macro name

MAC DEL Call Macro Function: DEL

Used in: (64)

Description: Deletes the specified macro.

Format: `MAC␣DEL␣<MacroName>`

Arguments: <MacroName> Macro name to be deleted

MAC END Call Macro Function: END

Used in: (62)

Description: Ends macro recording.

Format: `MAC␣END`

MAC ERR? Call Macro Function: ERR?

Used in: (64)

Description: Reports the last error that occurred while the macro was running.

Format: `MAC␣ERR?`

Response: <MacroName>␣<uint1>=<uint2>"<"<CMD>">"

<MacroName>	Name of the macro
<uint1>	Line in the micro where the error occurred
<uint2>	Error code
<CMD>	Bad command

MAC NSTART Call Macro Function: NSTART

Used in: (63)

Description: Runs the specified macro several times.

The macro is restarted when the last macro recording has completed, until <uint> has been reached.

Format: `MAC␣NSTART␣<MacroName>␣<uint>[␣<String1>[␣<String2>[␣<String3>[␣<String4>]]]]`

Arguments:

<MacroName>	Macro name
<uint>	Number of runs
<String1...4>	Local variables 1 to 4

Troubleshooting: No local variables specified although local variables are used in the macro

MAC START Call Macro Function: START

Used in: (63)

Description: Runs the specified macro.

Format: `MAC␣START␣<MacroName>[␣<String1>[␣<String2>[␣<String3>[␣<String4>]]]]`

Arguments:

<MacroName>	Macro name
<String1...4>	Local variables 1 to 4

Troubleshooting: No local variables specified although local variables are used in the macro

MAC? List Macros

Used in: (62)

Description:	Lists macros or content of a specified macro.	
Format:	`MAC?[␣<MacroName>]`	
Arguments:	<MacroName>	Name of a macro
	If no arguments are specified, the names of all saved macros are listed.	
Response:	<String>	
	<String>	Content of the macro or list of the names of all saved macros
Troubleshooting:	Wrong macro name (<MacroName>)	

MAN? Get Help String For Command

Description:	Shows a help text for a command.	
Format:	`MAN?␣<CMD>`	
Arguments:	<CMD>	Command that the help text is to be shown for
Response:	<String>	
	<String>	Help text

MEX Stop Macro Execution Due To Condition

Used in: Stopping the Macros (64)

Description: Stops the macro due to a specified condition.

If the parser encounters this command, the condition is checked. If the condition is fulfilled at a later time, it is ignored by the parser.

Can only be used in macros.

Possible relational operators:

<OP>	Description
=	Equal
!=	Not equal
<=	Smaller than or equal
<	Smaller than
>=	Larger than or equal
>	Larger than

Format: `MEX␣<CMD?>␣<OP>␣<Value>`

Arguments:

<CMD?>	Query command that responds with a single value
<OP>	Relational operator
<Value>	Relational value with <CMD?>

OAD Set Open-Loop Control Value (starts motion)

Used in: Triggering motion in open-loop operation (27)

Description: Sets the control value for the specified axis to the specified value.

Starts motion immediately.

In the case of piezo inertia drives, the control value corresponds to the analog output voltage [V].

Format: `OAD{␣<AxisID>␣<Value>}`

Arguments:

<AxisID>	Axis ID
<ControlValue>	Control value that is to be set for the axis

Troubleshooting: Illegal axis identifier

HID control is enabled

OAD? Get Control Value

Used in: Triggering motion in open-loop operation (27)

Description: Queries the control value of the specified axis.

In the case of piezo inertia drives, the control value corresponds to the analog output voltage [V].

Format: `OAD?[{␣<AxisID>}]`

Arguments:

<AxisID>	Axis ID

The value for all axes will be queried if no arguments are specified.

Response: {<AxisID>=<ControlValue>LF}

<AxisID>	Axis ID
<ControlValue>	Axis control value (FLOAT)

Troubleshooting: Illegal axis identifier

OSM Open-Loop Step Moving

Used in: Commandable Items (22), Triggering motion in open-loop operation (27)

Description: Moves the specified axis by the specified number of steps.

The velocity in open-loop step mode is controlled via the step frequency (parameter 0x1F000400).

Format: `OSM{␣<AxisID>␣<Value>}`

Arguments:

<AxisID>	Axis ID
<Value>	Number of steps to be made

Troubleshooting: Illegal axis identifier

HID control is enabled

OSN? Read Number Steps

Used in: Triggering motion in open-loop operation (27)

Description: Queries the number of steps that still have to be performed by the specified axis.

Format: `OSN?[{␣<AxisID>}]`

Arguments:

<AxisID>	Axis ID

The value for all axes will be queried if no arguments are specified.

Response: {<AxisID>=<uint>LF}

<AxisID>	Axis ID
<uint>	Number of steps still to be made

Troubleshooting: Illegal axis identifier

RBT Reboot System

Used in: Controller Macros (61)

Description: Restarts the electronics.

The electronics behave in the same way after restarting as they do after switching on.

RBT cannot be used in macros.

Format: **RBT**

RMC? List Running Macros

Used in: (63)

Description: Queries all macros currently running.

Format: **RMC?**

Response: {[<MacroName>]LF}

<MacroName> Macro name

SAI? Get List Of Current Axis Identifiers

Used in: Commandable Items (22)

Description: Queries the axis identifiers.

ALL ensures that the response also includes axes that are deactivated for electronics that permit axes to be deactivated.

Format: **SAI?[␣ALL]**

Arguments:		
	[␣ALL]	Ensures that electronics that allow axes to be deactivated (parameter 0x3C (p. 112) = "NOSTAGE") also allow deactivated axes to be queried.

Response:		
	{<AxisID>LF}	
	<AxisID>	Axis ID

SEP Set Non-Volatile Memory Parameters

Used in: Parameter Commands (69)

Description: Sets a parameter in the nonvolatile memory to a specific value.
Up to four parameters can be set per command.
Wrong values can lead to faulty operation or damage to the hardware.

Format: `SEP␣<Pswd>{␣<ItemID>␣<PamID>␣<PamValue>}`

Arguments:

<Pswd>	Password for writing to the nonvolatile memory
<ItemID>	Element of the electronics
<PamID>	Parameter ID
<PamValue>	Parameter value

Troubleshooting: Impermissible element identifier
Wrong parameter ID
Wrong password

SEP? Get Non-Volatile Memory Parameters

Used in: Parameter Commands (69), Saving Parameter Values in a Text File (70)

Description: Queries the value of a parameter in the nonvolatile memory.
Up to four parameters can be queried per command.

Format: `SEP?[{␣<ItemID>␣<PamID>}]`

Arguments:

<ItemID>	Element of the electronics
<PamID>	Parameter ID

Response: {<ItemID>␣<PamID>=<PamValue>LF}

<ItemID>	Element of the electronics
<PamID>	Parameter ID
<PamValue>	Parameter value

Troubleshooting: Impermissible element identifier
Wrong parameter ID

SPA Set Volatile Memory Parameters

Used in: Parameter Commands (69)

Description: Sets a parameter in the volatile memory to a specific value.

Up to four parameters can be set per command.

Wrong values can lead to faulty operation or damage to the hardware.

Format: **SPA{␣<ItemID>␣<PamID>␣<PamValue>}**

Arguments:

<ItemID>	Element of the electronics
<PamID>	Parameter ID
<PamValue>	Parameter value

Troubleshooting: Impermissible element identifier

Wrong parameter ID

SPA? Get Volatile Memory Parameters

Used in: Parameter Commands (69), Saving Parameter Values in a Text File (70)

Description: Queries the value of a parameter in the volatile memory.

Up to four parameters can be queried per command.

Format: **SPA?[{␣<ItemID>␣<PamID>}]**

Arguments:

<ItemID>	Element of the electronics
<PamID>	Parameter ID

Response: {<ItemID>␣<PamID>=<PamValue>LF}

<ItemID>	Element of the electronics
<PamID>	Parameter ID
<PamValue>	Parameter value

Troubleshooting: Impermissible element identifier

Wrong parameter ID

STP Stop All Axes

Used in: (64), Triggering motion in open-loop operation (27)

Description: Stops all axes abruptly.

Stops all motion started by motion commands (e.g., MOV, MVR, MVE, STE, SMO), referencing commands (FNL, FPL, FRF), and macros.

Also stops the macro.

Sets the error code to 10.

After the axes are stopped, their target positions are set to their current positions.

Format: **STP**

TIO? Tell Digital I/O Lines

Used in: Digital Input Signals (53), Digital Inputs and Outputs (52), Digital Output Signals (53)

Description: Queries the number of available digital I/O lines.

Format: `TIO?`

Response: I=<uint1>␣LF
O=<uint2>LF

<uint1> Number of digital input lines (UINT)
<uint2> Number of digital output lines (UINT)

TVI? Tell Valid Character Set For Axis Identifiers

Description: Queries permissible characters for axis identifiers.

Format: `TVI?`

Response: <String>

<String> Characters that are permitted for use in axis identifiers

VAR Set Variable Value

Used in: Running the Macro (63)

Description: Sets a variable to a specific value.

Local variables can only be set in macros.
The variable is only in volatile memory (RAM).

Format: `VAR␣<Variable>␣<String>`

Arguments: <Variable> Name of the variable whose value is set
<String> Value, that the variable is set to
Can be specified directly or via the value of a variable.

VAR? Get Variable Values

Used in: Running the Macro (63)

Description: Queries the value of a variable.

Local variables can only be queries when a macro is running that contains local variables.

Format: `VAR?[{␣<Variable>}]`

Arguments: <Variable> Name of the variable being queried
All variables are queried if no arguments are specified.

Response: {<Variable>=<String>LF}

<Variable> Name of the variable
<String> Value of the variable

VER? Get Versions Of Firmware And Drivers

Used in: Important Firmware Components (20)

Description: Queries the version numbers of the firmware.

VER? also queries the version numbers of further components such as drivers and libraries.

Format: **VER?**

Response: {<string1>:␣<string2>LF}

<string1> Name of the component

<string2> Version information and optional specifications

WAC Wait For Condition

Used in: Running the Macro (63)

Description: Waits until a condition is met.

WAC compares a specified value with a queried value according to a specified rule.

Can only be used in macros.

Possible relational operators:

<OP>	Description
=	Equal
!=	Not equal
<=	Smaller than or equal
<	Smaller than
>=	Larger than or equal
>	Larger than

Format: **WAC␣<CMD?>␣<OP>␣<Value>**

Arguments:

<CMD?>	Query command that responds with a single value
<OP>	Relational operator
<Value>	Relational value with <CMD?>

WPA Save Parameters To Non-Volatile Memory

Used in: Parameter Commands (69)

Description: Writes the value of a parameter from the volatile memory (RAM) to the nonvolatile memory.

WPA can also save parameter-independent settings. The used password determines what is saved with WPA:

<Pswd>	Description
100	Saves current valid values of all parameters and the current valid settings for HDT (p. 85), HIA (p. 87), and HIT (p. 91).
101	Saves the currently valid values of all parameters.
HID	Saves the current valid settings for HDT (p. 85), HIA (p. 87), and HIT (p. 91).

Wrong values can lead to faulty operation or damage to the hardware.

Notice: Note that the number of write cycles in the nonvolatile memory is limited. Therefore, save to the nonvolatile memory only when necessary.

Format: `WPA␣<Pswd>[{␣<ItemID>␣<PamID>}]`

Arguments:

<Pswd>	Password for writing to the nonvolatile memory
<ItemID>	Element of the electronics
<PamID>	Parameter ID

The <ItemID> and <PamID> specifications are not supported by all electronics.

Troubleshooting:

Impermissible element identifier

Wrong parameter ID

Wrong password

10 Parameter Reference

0x9	Maximum Motor Output	Maximum control value for driving an axis respectively a channel. See the response to HPA? for possible values
0x13	Is Rotary Stage?	Is this a rotation stage? Is not evaluated by the electronics but instead by the PC software. 0 Not a rotation stage 1 Rotation stage
0x3C	Stage Name	Positioner name. Default value: NOSTAGE String up to 20 characters
0x61	Invert Direction Of Motion For Joystick-Controlled Axis?	Inverts the direction of motion for HID-controlled axes. 0 Direction of motion not inverted (default) 1 Direction of motion inverted
0x72	Ignore Macro Error?	Ignore macro error? 0 Stop macro on error (default) 1 Ignore error
0xD000000	Controller Device S/N	Serial number of the electronics.
0xF000100	Stage Type	Mechanics type. x-xxx default positioners x-xxxKxxx customized positioners
0xF000200	Stage Serial Number	Serial number of the mechanics. 9-digit number
0xF000300	Stage Assembly Date	Manufacturing date of the mechanics. Date in DDMMYY format
0xF000400	Stage HW Version	Version number of the mechanics hardware.
0x1F000000	PIShift Upper Supply Voltage (V)	Maximum output voltage for piezo inertia drives. The value depends on the type of the drive.
0x1F000100	PIShift Lower Supply Voltage (V)	Minimum output voltage for piezo inertia drives. The value depends on the type of the drive.
0x1F000200	PIShift Forward Current (A)	Maximum output current for piezo inertia drives during forward motion. The value depends on the type of the drive.
0x1F000300	PIShift Backward Current (A)	Maximum output current for piezo inertia drives during backward motion. The value depends on the type of the drive.
0x1F000400	PIShift Frequency (Hz)	Frequency of the piezo voltage for open-loop operation of piezo inertia drives. Determines the velocity of a drive in open-loop operation. This parameter must not exceed the value of the parameter 0x9 (Maximum Motor Output).

0x1F000500	PIShift Charge Cycle	Duty cycle of the current source during output of a step. Specified as part of a period which the current source is switched on for. 0 to 1 The value depends on the type of the drive.
0x1F000701	PIShift Delay (ms)	Delay time when switching between two operating modes (e.g., step mode and linear mode). 0 to 2000 [ms]

0x9 Maximum Motor Output

Used in: How it Works (55)

Description:	Maximum control value for driving an axis respectively a channel.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	See the response to HPA? for possible values

0x13 Is Rotary Stage?

Description:	Is this a rotation stage? Is not evaluated by the electronics but instead by the PC software.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Not a rotation stage 1 Rotation stage

0x3C Stage Name

Used in: Commandable Items (22)

Description:	Positioner name. Default value: NOSTAGE
Data type	CHAR
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	String up to 20 characters

0x61 Invert Direction Of Motion For Joystick-Controlled Axis?

Used in: HID Control Configuration (55)

Description:	Inverts the direction of motion for HID-controlled axes.
Data type	INT
Command level	0
Item type	Axis
Source of data	Positioner database
Possible values	0 Direction of motion not inverted (default) 1 Direction of motion inverted

0x72 Ignore Macro Error?

Used in: (63)

Description:	Ignore macro error?
Data type	INT
Command level	0
Item type	System
Source of data	PC software commands (SPA (p. 106), SEP (p. 105)) or operating elements
Possible values	0 Stop macro on error (default) 1 Ignore error

0xD000000 Controller Device S/N

Description:	Serial number of the electronics.
Data type	CHAR
Command level	2
Item type	System

0xF000100 Stage Type

Description:	Mechanics type.
Data type	CHAR
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics
Possible values	x-xxx default positioners x-xxxKxxx customized positioners

0xF000200 Stage Serial Number

Description:	Serial number of the mechanics.
Data type	CHAR
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics
Possible values	9-digit number

0xF000300 Stage Assembly Date

Description:	Manufacturing date of the mechanics.
Data type	CHAR
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics
Possible values	Date in DDMMYY format

0xF000400 Stage HW Version

Description:	Version number of the mechanics hardware.
Data type	INT
Command level	2
Item type	Axis
Source of data	ID chip of the mechanics

0x1F000000 PIShift Upper Supply Voltage (V)

Used in: (25)

Description:	Maximum output voltage for piezo inertia drives.
Data type	FLOAT
Command level	1
Item type	Axis
Source of data	Positioner database
Possible values	The value depends on the type of the drive.

0x1F000100 PIShift Lower Supply Voltage (V)

Used in: (25)

Description:	Minimum output voltage for piezo inertia drives.
Data type	FLOAT
Command level	1
Item type	Axis
Source of data	Positioner database
Possible values	The value depends on the type of the drive.

0x1F000200 PIShift Forward Current (A)

Used in: (25)

Description:	Maximum output current for piezo inertia drives during forward motion.
Data type	FLOAT
Command level	1
Item type	Axis
Source of data	Positioner database
Possible values	The value depends on the type of the drive.

0x1F000300 PIShift Backward Current (A)

Used in: (25)

Description:	Maximum output current for piezo inertia drives during backward motion.
Data type	FLOAT
Command level	1
Item type	Axis
Source of data	Positioner database
Possible values	The value depends on the type of the drive.

0x1F000400 PIShift Frequency (Hz)

Used in: (25), OSM Open-Loop Step Moving (103), Triggering motion in open-loop operation (27)

Description:	Frequency of the piezo voltage for open-loop operation of piezo inertia drives. Determines the velocity of a drive in open-loop operation.
Data type	FLOAT
Command level	1
Item type	Axis
Source of data	Positioner database
Possible values	This parameter must not exceed the value of the parameter 0x9 (p. 111) (**Maximum Motor Output**).

0x1F000500 PIShift Charge Cycle

Used in: (25)

Description:	Duty cycle of the current source during output of a step. Specified as part of a period which the current source is switched on for.
Data type	FLOAT
Command level	1
Item type	Axis
Source of data	Positioner database
Possible values	0 to 1 The value depends on the type of the drive.

0x1F000701 PIShift Delay (ms)

Used in: (25)

Description:	Delay time when switching between two operating modes (e.g., step mode and linear mode).
Data type	FLOAT
Command level	1
Item type	Axis
Source of data	Positioner database
Possible values	0 to 2000 [ms]

11 Maintenance

11.1 Cleaning

Requirements

✓ You have disconnected the E-872.401 from the power supply.

Auxiliary Materials Required

- Soft, lint-free cloth
- Mild cleaning agent or disinfectant

If you have any questions on the auxiliary materials recommended for the E-872.401, contact our customer service department (p. 128).

NOTICE

Short circuits or flashovers!

The E-872.401 contains electrostatically sensitive devices that can be damaged by short circuits or flashovers when cleaning fluids penetrate the housing.

► Before cleaning, disconnect the E-872.401 from the power supply.

► Prevent cleaning fluid from penetrating the case.

Cleaning the E-872.401

1. Dampen the cloth with the cleaning agent or disinfectant.
2. Carefully wipe the surfaces of the E-872.401.

11.2 Updating the Firmware

The following describes the procedure for updating the E-872.401's firmware.

The figures show the procedure for any electronics; the procedure for the E-872.401 corresponds.

Requirements

✓ You have connected the E-872.401 to the PC via the communication interface (p. 41).

✓ "PI Firmware Updater" is installed on the PC (p. 35).

✓ You have copied the new firmware file, which you have received from our customer service department, to a directory on the PC.

✓ You have read and understood the documentation that you received from our customer service department together with the new firmware. You have learned from the documentation whether new parameters are introduced with the firmware update or the memory management of the E-872.401 changes.

✓ You have saved the parameter values of the E-872.401 in a text file on the PC (p. 50).

✓ You have saved the controller macros of the E-872.401 in files on the PC (p. 51).

✓ You have established communication between the E-872.401 and the PC with PIMikroMove® or PITerminal (p. 43).

NOTICE

Malfunction due to faulty firmware update!

A faulty or incomplete update of the E-872.401's firmware may mean that the E-872.401 can only be made operational again by the PI customer service department.

- ► Update the E-872.401's firmware only after consulting our customer service department. If possible, ask our customer service department to do the firmware update for you.
- ► Before updating the firmware, make sure that you have received a suitable firmware from our customer service department and have stored it at a location that is accessible to the update program.

Information

The status LED of the E-872.401 flashes as long as the E-872.401 is in firmware update mode. The E-872.401 exits the firmware update mode only when it **reboots** after the firmware was **successfully** updated. If the firmware update was unsuccessful or aborted, the E-872.401 remains in the firmware update mode after a reboot.

If the status LED lights up continuously although the E-872.401 was restarted after updating the firmware:

- ► Repeat the firmware update.
- ► If the update of the firmware fails, contact our customer service department (p. 128).

Updating the Firmware of the E-872.401

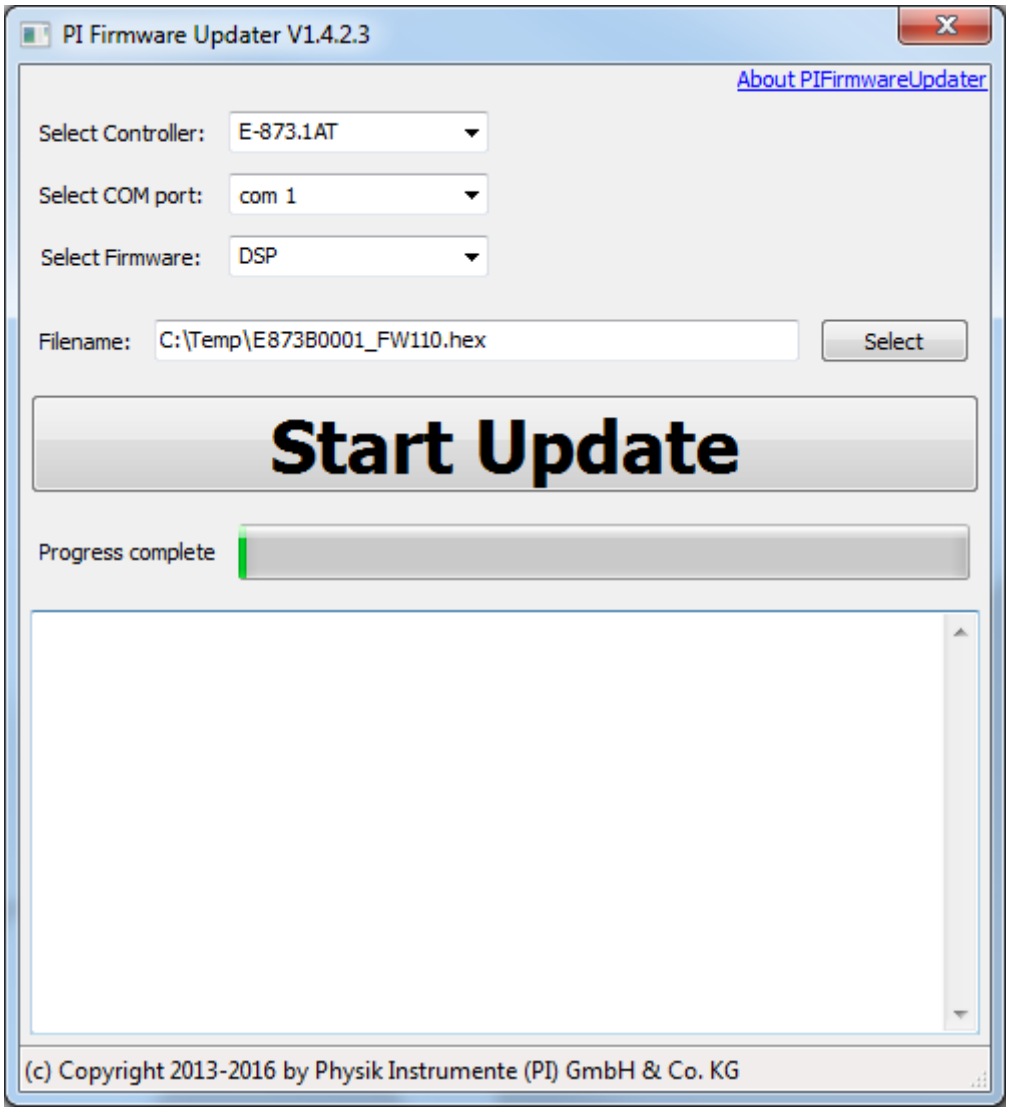
1. Activate the firmware update mode in PIMikroMove or PITerminal:
   a) Select ***Tools > Command entry*** in the PIMikroMove's main window or press F4 on the keyboard.
   b) Send the following commands successively:
   ```
   ZZZ 100 Flash
   rbt
   ```

➔ *The E-872.401 reboots and restarts in firmware update mode. The status LED of the E-872.401 flashes as long as the E-872.401 is in firmware update mode.*

2. Close PIMikroMove respectively PITerminal.
3. Run "PI Firmware Updater" on the PC.

➔ *The **PI Firmware Updater** window opens.*

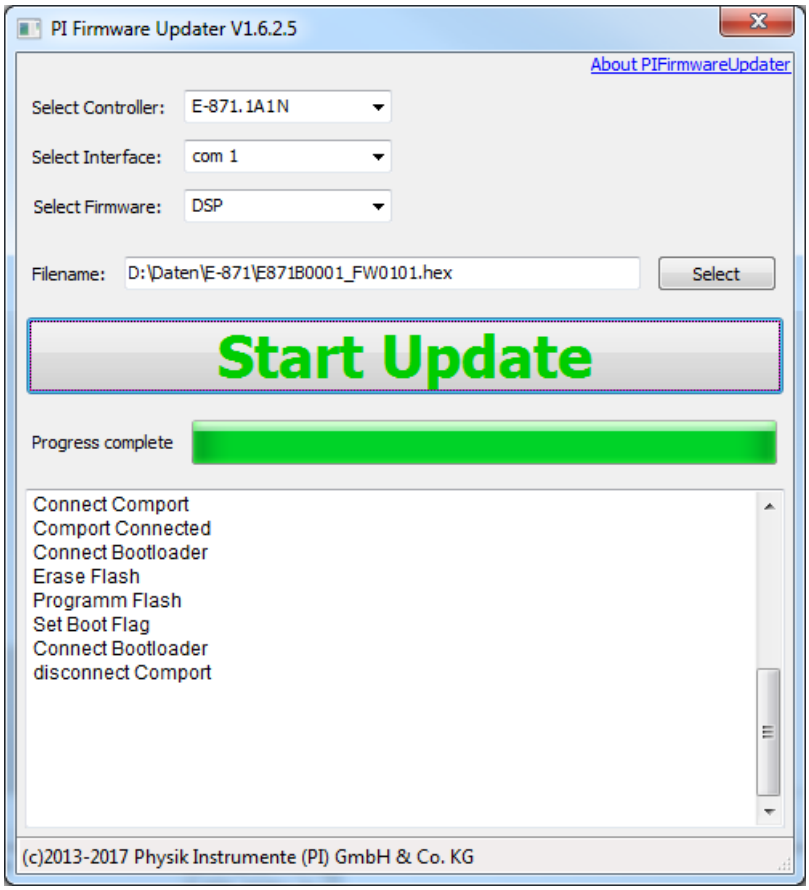
4. Set the following in the selection fields:
   - Select the entry for your controller model in the ***Select Controller*** field: E-872.401.
   - Select the COM port of the PC that is connected to the E-872.401 in the ***Select COM port*** field.
   - If necessary: Select "DSP" (= Digital Signal Processor) in the ***Select Firmware*** field.
5. Select the new firmware file:
   a) Click the ***Select*** button.
   b) Go to the directory in the file selection window where you stored the firmware file.
   c) Double-click the new firmware file (.hex extension) to enter the file path in the ***Filename*** field.



6. Start the firmware update by clicking on the ***Start Update*** button.

➔ *The firmware of the E-872.401 is updated. The update progress is displayed in the message list and by the progress bar.*

➔ *The update was successful when the* `disconnect Comport` *message appears as the last entry in the message list.*



7. Close the "PI Firmware Updater" by clicking the cross in the top right corner of the window.
8. Switch the E-872.401 off and on again via its toggle switch.

➔ *If the firmware update was successful, the E-872.401 exits the firmware update mode and the STA LED lights up green.*

9. If new parameters were added during updating of the firmware or the memory management of the E-872.401 was changed: Initialize the E-872.401 (p. 121).

Initializing the E-872.401 after a Firmware Update

1. Make sure that the current parameter values and controller macros of the E-872.401 have been saved on the PC.
   The initialization of the E-872.401 resets all parameters to their factory settings and deletes all controller macros. Consequently, parameter values and controller macros that are not saved are lost during the initialization process.
2. On the PC, start PITerminal or PIMikroMove, connect to the E-872.401, and, if necessary, open the window to send commands.
   Initialize the E-872.401 by sending the following commands successively:
   ```
   ZZZ 100 parameter
   ZZZ 100 macros
   ```

➔ *After successful initialization, the controller issues a corresponding message.*

3. Adapt the parameter values of the E-872.401:
   - Reset the parameters that were already present prior to the firmware update to the saved values from the text file.
   - Set the parameters that were introduced with the firmware update to the appropriate values.

4. If you have saved controller macros on the PC: Load the controller macros into the E-872.401, see "Loading Controller Macros from the PC into the E-872.401 (p. 51)".

12 Troubleshooting

The positioner does not move	
Cable not connected correctly	▶ Check the cable connections.
Unsuitable positioner cable used. Interference with the signal transmission between the positioner and E-872.401 can occur when an unsuitable cable is used.	▶ Only use genuine PI parts when connecting the positioner to the E-872.401. ▶ If you need extension cables, contact our customer service department (p. 128).
Positioner or drive cable is defective	▶ If available, replace the defective positioner with another one and test the new combination. ▶ If available, replace the defective drive cable with another one and test the new combination.
Positioner was connected to the switched-on E-872.401 The sensor electronics in the positioner was not initialized and the sensor's ID chip was not read out.	▶ Switch the E-872.401 off and on again, or reboot the E-872.401 with the RBT command or with the corresponding functions of the PC software.
Incorrect axis or channel commanded	▶ Make sure that the correct axis or channel identifier is used and that the positioner is connected correctly.
Incorrect configuration	▶ Check the parameter settings of the E-872.401 with the SPA? (p. 106) (volatile memory) and the SEP? (p. 105) commands (nonvolatile memory), and make the necessary corrections, refer to adapting settings (p. 69).
Incorrect command or incorrect syntax	▶ Send the ERR? command and check the error code that is returned.
HID control active Motion commands are not permitted when HID control is activated for the axis or the channel.	▶ Deactivate HID control (p. 55).
Positioner performs unintentional motion	
Control device is not connected, but HID control is activated in the E-872.401	▶ Activate HID control (p. 55) only when a control device is actually connected to the E-872.401.
HID axis is not calibrated	▶ Calibrate the axis of the control device (p. 57).
Startup macro is run	▶ Check whether a macro is specified as the start-up macro (p. 64) and cancel selection of the startup macro if necessary.
Communication between the E-872.401 and the PC not functioning	
Wrong communication cable used	▶ Use a straight-through cable for TCP/IP connections to a network. ▶ Use a crossover network cable for TCP/IP connections direct to the PC. ▶ Use a null-modem cable for RS-232 connections. ▶ Use a standard-compliant USB cable (type A to type B or type A to type Mini-B) for USB connections
Communication cable defective	▶ Replace the communication cable.

TCP/IP interface not configured correctly	▶ Connect the controller to the network **before** you switch it on. Restart the E-872.401 if necessary. ▶ Check the network settings. ▶ Make sure that the network is not blocked for unknown devices. ▶ Make sure that you have selected the correct E-872.401 when establishing communication. ▶ If you cannot solve the problems, consult your network administrator if necessary.
Another program is accessing the interface	▶ Close the other program.
The start procedure of the E-872.401's firmware has not finished yet	▶ Wait until the corresponding LED indicates operational readiness after switching on or rebooting the E-872.401. ▶ Try to establish communication.
Problems with special software	▶ Check whether the system works with other software, e.g., a terminal program, or a development environment; for this purpose, enter the `*IDN?` or `HLP?` command using that software. Make sure that you end commands with an LF (line feed); a command is only executed when an LF is received.
E-872.401 does not send an error code in the case of incorrect system behavior	
The error code was already queried by a different instance In the case of simultaneous access to the E-872.401 by several instances, the error code is only returned to the first instance that sent the `ERR?` command. The error code is reset to 0 during the query.	▶ If possible, access the E-872.401 with one instance only. ▶ Check whether the error code is queried regularly in the background by a macro or script or PC software (e.g., PIMikroMove).
LEDs do not light up even though the E-872.401 is switched on	
E-872.401 not connected to the power supply or the the power cord is defective.	▶ Switch off the E-872.401. ▶ Make sure that the E-872.401 is connected to the power supply and the power cord is not defective. ▶ Switch on the E-872.401.

If the problem is not listed in the table or cannot be solved as described, contact our customer service department (p. 128).

13 Technical Data

13.1 Specifications

	E-872.401
Function	Driver electronics for Q-Motion® positioners and PiezoMike linear actuators; benchtop device
Drive type	Piezo inertia drive
Channels	4 scalable to 64 channels via external module (on request)
Supported functions	Full-step mode, linear mode (analog control)

Amplifier	**E-872.401**
Amplifier channels	1 drive of one axis at a time, the switching between the drive channels is done via commands
Output voltage	0 to 100 V
Peak power	30 W
Output current / channel (<5 ms)	±650 mA

Interfaces and operation	E-872.401
Communication interfaces	USB, Ethernet
Actuator connection	4 x LEMO connector, 3-pin
Digital inputs	TTL inputs for commanding and configuration
Digital output	Error status
Command set	PI General Command Set (GCS)
User software	PIMikroMove
Application programming interfaces	API for C / C++ / C# / VB.NET / MATLAB / Python, drivers for NI LabVIEW
Display and indicators	LED indicator for status and operation
Manual control	Joystick via USB

Miscellaneous	E-872.401
Operating temperature range	0 to 50 °C
Overtemperature protection	Deactivation at 75 °C
Dimensions	147 mm × 125 mm × 40 mm
Mass	0.46 kg
Operating voltage	24 V (power adapter in the scope of delivery)
Max. power consumption	35 W

13.2 Maximum Ratings

The E-872.401 is designed for the following operating data:

Maximum operating voltage	Operating frequency	Maximum power consumption
24 V	DC	35 W

13.3 Ambient Conditions and Classifications

The following ambient conditions and classifications for the E-872.401 must be observed:

Area of application	For indoor use only
Maximum altitude	2000 m above msl
Air pressure	1100 hPa to 0.1 hPa
Relative humidity	Max. 80 % for temperatures to 31 °C, decreasing linearly to 50 % at 40 °C
Storage temperature	0 °C to 70 °C
Transport temperature	-25 °C to +85 °C
Overvoltage category	II
Protection class	I
Degree of pollution	2
Degree of protection according to IEC 60529	IP20

13.4 Dimensions

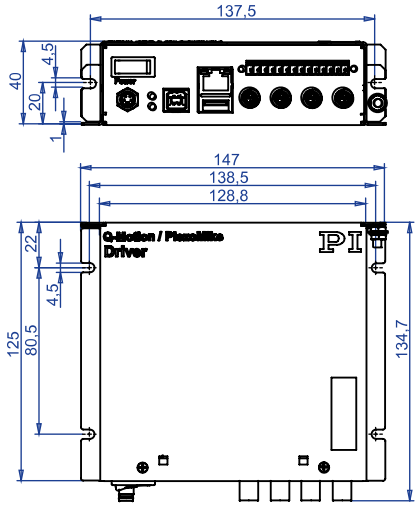


Figure 6: Dimensions of the E-872.401

Dimensions in mm. Note that the decimal places are separated by a comma in the drawings.

14 Customer Service Department

For enquiries and orders, contact your PI representative or send us an email.

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

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

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

Customer service address:

Physik Instrumente (PI) GmbH & Co. KG
Auf der Roemerstrasse 1
76228 Karlsruhe
Germany

service@pi.de
www.pi.de

15 Old Equipment Disposal

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

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

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

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

Physik Instrumente (PI) GmbH & Co. KG
Auf der Roemerstrasse 1
76228 Karlsruhe
Germany

info@pi.de
www.pi.de

16 Appendix

16.1 Pin Assignment

16.1.1 Axis Connector

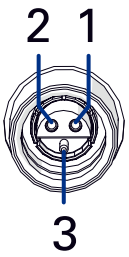


Figure 7: Drive connection: LEMO EPL.0S.303.HLN

Pin	Signal	Function
1	M+	Output: Motor voltage +
2	M-	Output: Motor voltage -
3	ID	Bidirectional: Data line for ID chip

16.1.2 I/O Connector

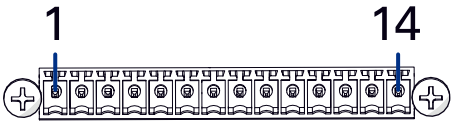


Figure 8: I/O connection: Phoenix Contact MC 1,5/14-GF-3,5-LR

Pin	Function
1 (**5V**)	+5 V DC, max. 50 mA (output)
2 (**GND**)	GND
3 (**M0**)	Selection of the drive channel, least significant bit (TTL, input)
4 (**M1**)	Selection of the drive channel, most significant bit (TTL, input)
5 (**DIR**)	Direction signal for step mode (TTL, input) High: Forward motion Low: Backward motion
6 (**EN**)	Activation of the drive in step mode (TTL, input)
7 (**CMD**)	Selection of command mode (TTL, input) High: Commanding via digital I/O lines Low: Commanding via PC or HID
8 (**AN**)	Analog input voltage for linear mode (0 - 4.8 V, input) $V_{Ch} = V_{AN} \times 10$ (V_{Ch} = output voltage of the channel, V_{AN} = analog input voltage)
9 (**ST**)	Selection of the drive mode (TTL, input) High: Linear mode Low: Step mode
10 (**ERR**)	Error signal (TTL, output) High: Error (error code ≠ 0) Low: No error
11 (**BMO**)	Reserved
12 (**BMI**)	Reserved
13 (**BCK**)	Reserved
14 (**BCS**)	Reserved

The matching plug connector **MC 1,5/14-ST-3,5** is **not** included in the E-872.401's scope of delivery.

16.1.3 Power Adapter Connector

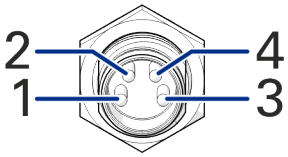


Figure 9: Voltage connection: M8 4-pin (m)

Pin	Signal	Function
1	GND	Ground
2	GND	Ground
3	V+	Input: Power supply, +24 V
4	V+	Input: Power supply, +24 V

16.2 GCS Error Codes

The error codes listed here are those of the PI General Command Set. As such, some may be not relevant to your controller and will simply never occur.

Controller errors		
0	PI_CNTR_NO_ERROR	No error
1	PI_CNTR_PARAM_SYNTAX	Parameter syntax error
2	PI_CNTR_UNKNOWN_COMMAND	Unknown command
3	PI_CNTR_COMMAND_TOO_LONG	Command length out of limits or command buffer overrun
4	PI_CNTR_SCAN_ERROR	Error while scanning
5	PI_CNTR_MOVE_WITHOUT_REF_OR_NO_SERVO	Unallowable move attempted on unreferenced axis, or move attempted with servo off
6	PI_CNTR_INVALID_SGA_PARAM	Parameter for SGA not valid
7	PI_CNTR_POS_OUT_OF_LIMITS	Position out of limits
8	PI_CNTR_VEL_OUT_OF_LIMITS	Velocity out of limits
9	PI_CNTR_SET_PIVOT_NOT_POSSIBLE	Attempt to set pivot point while U,V and W not all 0
10	PI_CNTR_STOP	Controller was stopped by command
11	PI_CNTR_SST_OR_SCAN_RANGE	Parameter for SST or for one of the embedded scan algorithms out of range
12	PI_CNTR_INVALID_SCAN_AXES	Invalid axis combination for fast scan
13	PI_CNTR_INVALID_NAV_PARAM	Parameter for NAV out of range
14	PI_CNTR_INVALID_ANALOG_INPUT	Invalid analog channel
15	PI_CNTR_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
16	PI_CNTR_INVALID_STAGE_NAME	Unknown stage name
17	PI_CNTR_PARAM_OUT_OF_RANGE	Parameter out of range
18	PI_CNTR_INVALID_MACRO_NAME	Invalid macro name
19	PI_CNTR_MACRO_RECORD	Error while recording macro

Controller errors		
20	PI_CNTR_MACRO_NOT_FOUND	Macro not found
21	PI_CNTR_AXIS_HAS_NO_BRAKE	Axis has no brake
22	PI_CNTR_DOUBLE_AXIS	Axis identifier specified more than once
23	PI_CNTR_ILLEGAL_AXIS	Illegal axis
24	PI_CNTR_PARAM_NR	Incorrect number of parameters
25	PI_CNTR_INVALID_REAL_NR	Invalid floating point number
26	PI_CNTR_MISSING_PARAM	Parameter missing
27	PI_CNTR_SOFT_LIMIT_OUT_OF_RANGE	Soft limit out of range
28	PI_CNTR_NO_MANUAL_PAD	No manual pad found
29	PI_CNTR_NO_JUMP	No more step-response values
30	PI_CNTR_INVALID_JUMP	No step-response values recorded
31	PI_CNTR_AXIS_HAS_NO_REFERENCE	Axis has no reference sensor
32	PI_CNTR_STAGE_HAS_NO_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

Controller errors		
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave
42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter) missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error
49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	MOV! motion still in progress
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data Record Table does not exist

Controller errors		
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).

Controller errors		
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly
77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_SERVO_ON	Open-loop commands (SVA, SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM
81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command
82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
90	PI_CNTR_DIO_AND_TRACING_NOT_POSSIBLE	No tracing possible while digital IOs are used on this HW revision. Reconnect to switch operation mode.
91	PI_CNTR_COLLISION	Move not possible, would cause collision
92	PI_CNTR_SLAVE_NOT_FAST_ENOUGH	Stage is not capable of following the master. Check the gear ratio.
93	PI_CNTR_CMD_NOT_ALLOWED_WHILE_AXIS_IN_MOTION	This command is not allowed while the affected axis or its master is in motion.

Controller errors		
94	PI_CNTR_OPEN_LOOP_JOYSTICK_ENABLED	Servo cannot be switched on when open-loop joystick control is activated.
95	PI_CNTR_INVALID_SERVO_STATE_FOR_PARAMETER	This parameter cannot be changed in current servo mode.
96	PI_CNTR_UNKNOWN_STAGE_NAME	Unknown stage name
97	PI_CNTR_INVALID_VALUE_LENGTH	Invalid length of value (too much characters)
98	PI_CNTR_AUTOZERO_FAILED	AutoZero procedure was not successful
99	PI_CNTR_SENSOR_VOLTAGE_OFF	Sensor voltage is off
100	PI_LABVIEW_ERROR	PI driver for use with NI LabVIEW reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on
206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error
210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller

Controller errors		
212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken
216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, some controllers need CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop
218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATION	Position was calculated during MOV motion
220	PI_CNTR_INTERPOLATION_FIFO_UNDERRUN	FIFO buffer underrun during interpolation
221	PI_CNTR_INTERPOLATION_FIFO_OVERFLOW	FIFO buffer overflow during interpolation
230	PI_CNTR_INVALID_HANDLE	Invalid handle
231	PI_CNTR_NO_BIOS_FOUND	No bios found
232	PI_CNTR_SAVE_SYS_CFG_FAILED	Save system configuration failed
233	PI_CNTR_LOAD_SYS_CFG_FAILED	Load system configuration failed
301	PI_CNTR_SEND_BUFFER_OVERFLOW	Send buffer overflow
302	PI_CNTR_VOLTAGE_OUT_OF_LIMITS	Voltage out of limits
303	PI_CNTR_OPEN_LOOP_MOTION_SET_WHEN_SERVO_ON	Open-loop motion attempted when servo ON

Controller errors		
304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long
305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus
307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro
310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined
402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
500	PI_CNTR_EMERGENCY_STOP_BUTTON_ACTIVATED	The \"red knob\" is still set and disables system
501	PI_CNTR_EMERGENCY_STOP_BUTTON_WAS_ACTIVATED	The \"red knob\" was activated and still disables system - reanimation required

Controller errors		
502	PI_CNTR_REDUNDANCY_LIMIT_EXCEEDED	Position consistency check failed
503	PI_CNTR_COLLISION_SWITCH_ACTIVATED	Hardware collision sensor(s) are activated
504	PI_CNTR_FOLLOWING_ERROR	Strut following error occurred, e.g. caused by overload or encoder failure
505	PI_CNTR_SENSOR_SIGNAL_INVALID	One sensor signal is not valid
506	PI_CNTR_SERVO_LOOP_UNSTABLE	Servo loop was unstable due to wrong parameter setting and switched off to avoid damage.
507	PI_CNTR_LOST_SPI_SLAVE_CONNECTION	Digital connection to external SPI slave device is lost
508	PI_CNTR_MOVE_ATTEMPT_NOT_PERMITTED	Move attempt not permitted due to customer or limit settings
509	PI_CNTR_TRIGGER_EMERGENCY_STOP	Emergency stop caused by trigger input
530	PI_CNTR_NODE_DOES_NOT_EXIST	A command refers to a node that does not exist
531	PI_CNTR_PARENT_NODE_DOES_NOT_EXIST	A command refers to a node that has no parent node
532	PI_CNTR_NODE_IN_USE	Attempt to delete a node that is in use
533	PI_CNTR_NODE_DEFINITION_IS_CYCLIC	Definition of a node is cyclic
536	PI_CNTR_HEXAPOD_IN_MOTION	Transformation cannot be defined as long as Hexapod is in motion
537	PI_CNTR_TRANSFORMATION_TYPE_NOT_SUPPORTED	Transformation node cannot be activated
539	PI_CNTR_NODE_PARENT_IDENTICAL_TO_CHILD	A node cannot be linked to itself
540	PI_CNTR_NODE_DEFINITION_INCONSISTENT	Node definition is erroneous or not complete (replace or delete it)
542	PI_CNTR_NODES_NOT_IN_SAME_CHAIN	The nodes are not part of the same chain

Controller errors		
543	PI_CNTR_NODE_MEMORY_FULL	Unused nodes must be deleted before new nodes can be stored
544	PI_CNTR_PIVOT_POINT_FEATURE_NOT_SUPPORTED	With some transformations pivot point usage is not supported
545	PI_CNTR_SOFTLIMITS_INVALID	Soft limits invalid due to changes in coordinate system
546	PI_CNTR_CS_WRITE_PROTECTED	Coordinate system is write protected
547	PI_CNTR_CS_CONTENT_FROM_CONFIG_FILE	Coordinate system cannot be changed because its content is loaded from a configuration file
548	PI_CNTR_CS_CANNOT_BE_LINKED	Coordinate system may not be linked
549	PI_CNTR_KSB_CS_ROTATION_ONLY	A KSB-type coordinate system can only be rotated by multiples of 90 degrees
551	PI_CNTR_CS_DATA_CANNOT_BE_QUERIED	This query is not supported for this coordinate system type
552	PI_CNTR_CS_COMBINATION_DOES_NOT_EXIST	This combination of work-and-tool coordinate systems does not exist
553	PI_CNTR_CS_COMBINATION_INVALID	The combination must consist of one work and one tool coordinate system
554	PI_CNTR_CS_TYPE_DOES_NOT_EXIST	This coordinate system type does not exist
555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
556	PI_CNTR_CS_TYPE_NOT_ACTIVATED	No coordinate system of this type is activated
557	PI_CNTR_CS_NAME_INVALID	Name of coordinate system is invalid
558	PI_CNTR_CS_GENERAL_FILE_MISSING	File with stored CS systems is missing or erroneous
559	PI_CNTR_CS_LEVELING_FILE_MISSING	File with leveling CS is missing or erroneous
601	PI_CNTR_NOT_ENOUGH_MEMORY	not enough memory

Controller errors		
602	PI_CNTR_HW_VOLTAGE_ERROR	hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	hardware temperature out of range
604	PI_CNTR_POSITION_ERROR_TOO_HIGH	Position error of any axis in the system is too high
606	PI_CNTR_INPUT_OUT_OF_RANGE	Maximum value of input signal has been exceeded
607	PI_CNTR_NO_INTEGER	Value is not integer
608	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_RUNNING	Fast alignment process cannot be paused because it is not running
609	PI_CNTR_FAST_ALIGNMENT_PROCESS_IS_NOT_PAUSED	Fast alignment process cannot be restarted/ resumed because it is not paused
650	PI_CNTR_UNABLE_TO_SET_PARAM_WITH_SPA	Parameter could not be set with SPA - SEP needed?
651	PI_CNTR_PHASE_FINDING_ERROR	Phase finding error
652	PI_CNTR_SENSOR_SETUP_ERROR	Sensor setup error
653	PI_CNTR_SENSOR_COMM_ERROR	Sensor communication error
654	PI_CNTR_MOTOR_AMPLIFIER_ERROR	Motor amplifier error
655	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_I2T	Overcurrent protection triggered by I2T-module
656	PI_CNTR_OVER_CURR_PROTEC_TRIGGERED_BY_AMP_MODULE	Overcurrent protection triggered by amplifier module
657	PI_CNTR_SAFETY_STOP_TRIGGERED	Safety stop triggered
658	PI_SENSOR_OFF	Sensor off?
659	PI_CNTR_PARAM_CONFLICT	Parameter could not be set. Conflict with another parameter.

Controller errors		
700	PI_CNTR_COMMAND_NOT_ALLOWED_IN_EXTERNAL_MODE	Command not allowed in external mode
710	PI_CNTR_EXTERNAL_MODE_ERROR	External mode communication error
715	PI_CNTR_INVALID_MODE_OF_OPERATION	Invalid mode of operation
716	PI_CNTR_FIRMWARE_STOPPED_BY_CMD	Firmware stopped by command (#27)
717	PI_CNTR_EXTERNAL_MODE_DRIVER_MISSING	External mode driver missing
718	PI_CNTR_CONFIGURATION_FAILURE_EXTERNAL_MODE	Missing or incorrect configuration of external mode
719	PI_CNTR_EXTERNAL_MODE_CYCLETIME_INVALID	External mode cycletime invalid
720	PI_CNTR_BRAKE_ACTIVATED	Brake is activated
725	PI_CNTR_DRIVE_STATE_TRANSITION_ERROR	Drive state transition error
731	PI_CNTR_SURFACEDETECTION_RUNNING	Command not allowed while surface detection is running
732	PI_CNTR_SURFACEDETECTION_FAILED	Last surface detection failed
733	PI_CNTR_FIELDBUS_IS_ACTIVE	Fieldbus is active and is blocking GCS control commands
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004

Controller errors		
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
1006	PI_CNTR_INVALID_IDENTIFIER	Invalid identifier (invalid special characters, ...)
1007	PI_CNTR_UNKNOWN_VARIABLE_OR_ARGUMENT	Variable or argument not defined
1008	PI_CNTR_RUNNING_MACRO	Controller is (already) running a macro
1009	PI_CNTR_MACRO_INVALID_OPERATOR	Invalid or missing operator for condition. Check necessary spaces around operator.
1010	PI_CNTR_MACRO_NO_ANSWER	No response was received while executing WAC/MEX/JRC/...
1011	PI_CMD_NOT_VALID_IN_MACRO_MODE	Command not valid during macro execution
1012	PI_CNTR_ERROR_IN_MACRO	Error occured during macro execution
1024	PI_CNTR_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
1025	PI_CNTR_MAX_MOTOR_OUTPUT_REACHED	Maximum motor output reached
1063	PI_CNTR_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
1064	PI_CNTR_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
1065	PI_CNTR_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
1066	PI_CNTR_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
1071	PI_CNTR_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
1072	PI_CNTR_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
1073	PI_CNTR_PROFILE_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
1090	PI_CNTR_TOO_MANY_TCP_CONNECTIONS_OPEN	There are too many open tcpip connections

Controller errors		
2000	PI_CNTR_ALREADY_HAS_SERIAL_NUMBER	Controller already has a serial number
4000	PI_CNTR_SECTOR_ERASE_FAILED	Sector erase failed
4001	PI_CNTR_FLASH_PROGRAM_FAILED	Flash program failed
4002	PI_CNTR_FLASH_READ_FAILED	Flash read failed
4003	PI_CNTR_HW_MATCHCODE_ERROR	HW match code missing/invalid
4004	PI_CNTR_FW_MATCHCODE_ERROR	FW match code missing/invalid
4005	PI_CNTR_HW_VERSION_ERROR	HW version missing/invalid
4006	PI_CNTR_FW_VERSION_ERROR	FW version missing/invalid
4007	PI_CNTR_FW_UPDATE_ERROR	FW update failed
4008	PI_CNTR_FW_CRC_PAR_ERROR	FW Parameter CRC wrong
4009	PI_CNTR_FW_CRC_FW_ERROR	FW CRC wrong
5000	PI_CNTR_INVALID_PCC_SCAN_DATA	PicoCompensation scan data is not valid
5001	PI_CNTR_PCC_SCAN_RUNNING	PicoCompensation is running, some actions can not be executed during scanning/recording
5002	PI_CNTR_INVALID_PCC_AXIS	Given axis cannot be defined as PPC axis
5003	PI_CNTR_PCC_SCAN_OUT_OF_RANGE	Defined scan area is larger than the travel range
5004	PI_CNTR_PCC_TYPE_NOT_EXISTING	Given PicoCompensation type is not defined
5005	PI_CNTR_PCC_PAM_ERROR	PicoCompensation parameter error
5006	PI_CNTR_PCC_TABLE_ARRAY_TOO_LARGE	PicoCompensation table is larger than maximum table length
5100	PI_CNTR_NEXLINE_ERROR	Common error in NEXLINE® firmware module

Controller errors		
5101	PI_CNTR_CHANNEL_ALREADY_USED	Output channel for NEXLINE® can not be redefined for other usage
5102	PI_CNTR_NEXLINE_TABLE_TOO_SMALL	Memory for NEXLINE® signals is too small
5103	PI_CNTR_RNP_WITH_SERVO_ON	RNP can not be executed if axis is in closed loop
5104	PI_CNTR_RNP_NEEDED	Relax procedure (RNP) needed
5200	PI_CNTR_AXIS_NOT_CONFIGURED	Axis must be configured for this action
5300	PI_CNTR_FREQU_ANALYSIS_FAILED	Frequency analysis failed
5301	PI_CNTR_FREQU_ANALYSIS_RUNNING	Another frequency analysis is running
6000	PI_CNTR_SENSOR_ABS_INVALID_VALUE	Invalid preset value of absolute sensor
6001	PI_CNTR_SENSOR_ABS_WRITE_ERROR	Error while writing to sensor
6002	PI_CNTR_SENSOR_ABS_READ_ERROR	Error while reading from sensor
6003	PI_CNTR_SENSOR_ABS_CRC_ERROR	Checksum error of absolute sensor
6004	PI_CNTR_SENSOR_ABS_ERROR	General error of absolute sensor
6005	PI_CNTR_SENSOR_ABS_OVERFLOW	Overflow of absolute sensor position

Interface errors		
0	COM_NO_ERROR	No error occurred during function call
-1	COM_ERROR	Error during com operation (could not be specified)
-2	SEND_ERROR	Error while sending data
-3	REC_ERROR	Error while receiving data
-4	NOT_CONNECTED_ERROR	Not connected (no port with given ID open)
-5	COM_BUFFER_OVERFLOW	Buffer overflow
-6	CONNECTION_FAILED	Error while opening port
-7	COM_TIMEOUT	Timeout error
-8	COM_MULTILINE_RESPONSE	There are more lines waiting in buffer
-9	COM_INVALID_ID	There is no interface or DLL handle with the given ID
-10	COM_NOTIFY_EVENT_ERROR	Event/message for notification could not be opened
-11	COM_NOT_IMPLEMENTED	Function not supported by this interface type
-12	COM_ECHO_ERROR	Error while sending "echoed" data
-13	COM_GPIB_EDVR	IEEE488: System error
-14	COM_GPIB_ECIC	IEEE488: Function requires GPIB board to be CIC
-15	COM_GPIB_ENOL	IEEE488: Write function detected no listeners
-16	COM_GPIB_EADR	IEEE488: Interface board not addressed correctly
-17	COM_GPIB_EARG	IEEE488: Invalid argument to function call
-18	COM_GPIB_ESAC	IEEE488: Function requires GPIB board to be SAC
-19	COM_GPIB_EABO	IEEE488: I/O operation aborted
-20	COM_GPIB_ENEB	IEEE488: Interface board not found
-21	COM_GPIB_EDMA	IEEE488: Error performing DMA

Interface errors		
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended operation
-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost
-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full
-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits
-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)
-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources
-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate

Interface errors		
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device
-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present
-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error
-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port: was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again
-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer

Interface errors		
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)
-62	COM_SOCKET_HOST_NOT_FOUND	Host not found
-63	COM_DEVICE_CONNECTED	Device already connected

DLL errors		
-1001	PI_UNKNOWN_AXIS_IDENTIFIER	Unknown axis identifier
-1002	PI_NR_NAV_OUT_OF_RANGE	Number for NAV out of range--must be in [1,10000]
-1003	PI_INVALID_SGA	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	PI_UNEXPECTED_RESPONSE	Controller sent unexpected response
-1005	PI_NO_MANUAL_PAD	No manual control pad installed, calls to SMA and related commands are not allowed
-1006	PI_INVALID_MANUAL_PAD_KNOB	Invalid number for manual control pad knob
-1007	PI_INVALID_MANUAL_PAD_AXIS	Axis not currently controlled by a manual control pad
-1008	PI_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g., reference move, fast scan algorithm)
-1009	PI_THREAD_ERROR	Internal error--could not start thread
-1010	PI_IN_MACRO_MODE	Controller is (already) in macro mode--command not valid in macro mode
-1011	PI_NOT_IN_MACRO_MODE	Controller not in macro mode--command not valid unless macro mode active
-1012	PI_MACRO_FILE_ERROR	Could not open file to write or read macro
-1013	PI_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
-1014	PI_MACRO_EDITOR_ERROR	Internal error in macro editor
-1015	PI_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	PI_AXIS_ALREADY_EXISTS	Axis identifier is already in use by a connected stage
-1017	PI_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
-1018	PI_COM_ARRAY_ERROR	Could not access array data in COM server
-1019	PI_COM_ARRAY_RANGE_ERROR	Range of array does not fit the number of parameters

DLL errors		
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?
-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error: position error too large, servo is switched off automatically
-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed
-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup
-1032	PI_USER_CANCELLED	The user cancelled a dialog
-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL
-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter file open. Close it before opening a new file

DLL errors		
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid
-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis
-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started
-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC
-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication with OPM (Optical Power Meter), maybe no OPM connected

DLL errors		
-1057	PI_WAVE_EDITOR_WRONG_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

DLL errors		
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.
-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by another function. Please try again later.
-1077	PI_PARAM_DAT_FILE_INVALID_VERSION	Version of parameter DAT file does not match the required version. Current files are available at www.pi.ws.
-1078	PI_CANNOT_WRITE_TO_PARAM_DAT_FILE	Cannot write to parameter DAT file to store user defined stage type.
-1079	PI_CANNOT_CREATE_PARAM_DAT_FILE	Cannot create parameter DAT file to store user defined stage type.
-1080	PI_PARAM_DAT_FILE_INVALID_REVISION	Parameter DAT file does not have correct revision.
-1081	PI_USERSTAGES_DAT_FILE_INVALID_REVISION	User stages DAT file does not have correct revision.
-1082	PI_SOFTWARE_TIMEOUT	Timeout Error. Some lengthy operation did not finish within expected time.
-1083	PI_WRONG_DATA_TYPE	A function argument has an unexpected data type.
-1084	PI_DIFFERENT_ARRAY_SIZES	Length of data arrays is different.
-1085	PI_PARAM_NOT_FOUND_IN_PARAM_DAT_FILE	Parameter value not found in parameter DAT file.
-1086	PI_MACRO_RECORDING_NOT_ALLOWED_IN_THIS_MODE	Macro recording is not allowed in this mode of operation.
-1087	PI_USER_CANCELLED_COMMAND	Command cancelled by user input.
-1088	PI_TOO_FEW_GCS_DATA	Controller sent too few GCS data sets
-1089	PI_TOO_MANY_GCS_DATA	Controller sent too many GCS data sets
-1090	PI_GCS_DATA_READ_ERROR	Communication error while reading GCS data

DLL errors		
-1091	PI_WRONG_NUMBER_OF_INPUT_ARGUMENTS	Wrong number of input arguments.
-1092	PI_FAILED_TO_CHANGE_CCL_LEVEL	Change of command level has failed.
-1093	PI_FAILED_TO_SWITCH_OFF_SERVO	Switching off the servo mode has failed.
-1094	PI_FAILED_TO_SET_SINGLE_PARAMETER_WHILE_PERFORMING_CST	A parameter could not be set while performing CST: CST was not performed (parameters remain unchanged).
-1095	PI_ERROR_CONTROLLER_REBOOT	Connection could not be reestablished after reboot.
-1096	PI_ERROR_AT_QHPA	Sending HPA? or receiving the response has failed.
-1097	PI_QHPA_NONCOMPLIANT_WITH_GCS	HPA? response does not comply with GCS2 syntax.
-1098	PI_FAILED_TO_READ_QSPA	Response to SPA? could not be received.
-1099	PI_PAM_FILE_WRONG_VERSION	Version of PAM file cannot be handled (too old or too new)
-1100	PI_PAM_FILE_INVALID_FORMAT	PAM file does not contain required data in PAM-file format
-1101	PI_INCOMPLETE_INFORMATION	Information does not contain all required data
-1102	PI_NO_VALUE_AVAILABLE	No value for parameter available
-1103	PI_NO_PAM_FILE_OPEN	No PAM file is open
-1104	PI_INVALID_VALUE	Invalid value
-1105	PI_UNKNOWN_PARAMETER	Unknown parameter
-1106	PI_RESPONSE_TO_QSEP_FAILED	Response to SEP? could not be received.
-1107	PI_RESPONSE_TO_QSPA_FAILED	Response to SPA? could not be received.
-1108	PI_ERROR_IN_CST_VALIDATION	Error while performing CST: One or more parameters were not set correctly.

DLL errors		
-1109	PI_ERROR_PAM_FILE_HAS_DUPLICATE_ENTRY_WITH_DIFFERENT_VALUES	PAM file has duplicate entry with different values.
-1110	PI_ERROR_FILE_NO_SIGNATURE	File has no signature
-1111	PI_ERROR_FILE_INVALID_SIGNATURE	File has invalid signature
-10000	PI_PARAMETER_DB_INVALID_STAGE_TYPE_FORMAT	PI stage database: String containing stage type and description has invalid format.
-10001	PI_PARAMETER_DB_SYSTEM_NOT_AVAILABLE	PI stage database: Database does not contain the selected stage type for the connected controller.
-10002	PI_PARAMETER_DB_FAILED_TO_ESTABLISH_CONNECTION	PI stage database: Establishing the connection has failed.
-10003	PI_PARAMETER_DB_COMMUNICATION_ERROR	PI stage database: Communication was interrupted (e.g. because database was deleted).
-10004	PI_PARAMETER_DB_ERROR_WHILE_QUERYING_PARAMETERS	PI stage database: Querying data failed.
-10005	PI_PARAMETER_DB_SYSTEM_ALREADY_EXISTS	PI stage database: System already exists. Rename stage and try again.
-10006	PI_PARAMETER_DB_QHPA_CONTANS_UNKNOWN_PAM_IDS	PI stage database: Response to HPA? contains unknown parameter IDs.
-10007	PI_PARAMETER_DB_AND_QHPA_ARE_INCONSISTENT	PI stage database: Inconsistency between database and response to HPA?.
-10008	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_ADDED	PI stage database: Stage has not been added.
-10009	PI_PARAMETER_DB_SYSTEM_COULD_NOT_BE_REMOVED	PI stage database: Stage has not been removed.
-10010	PI_PARAMETER_DB_CONTROLLER_DB_PARAMETERS_MISMATCH	Controller does not support all stage parameters stored in PI stage database. No parameters were set.
-10011	PI_PARAMETER_DB_DATABASE_IS_OUTDATED	The version of PISTAGES3.DB stage database is out of date. Please update via PIUpdateFinder. No parameters were set.

DLL errors		
-10012	PI_PARAMETER_DB_AND_HPA_MISMATCH_STRICT	Mismatch between number of parameters present in stage database and available in controller interface. No parameters were set.
-10013	PI_PARAMETER_DB_AND_HPA_MISMATCH_LOOSE	Mismatch between number of parameters present in stage database and available in controller interface. Some parameters were ignored.
-10014	PI_PARAMETER_DB_FAILED_TO_SET_PARAMETERS_CORRECTLY	One or more parameters could not be set correctly on the controller.
-10015	PI_PARAMETER_DB_MISSING_PARAMETER_DEFINITIONS_IN_DATABASE	One or more parameter definitions are not present in stage database. Please update PISTAGES3.DB via PIUpdateFinder. Missing parameters were ignored.

17 EU Declaration of Conformity

An EU Declaration of Conformity was issued for the E-872.401 in accordance with the following European directives:

- Low Voltage Directive
- EMC Directive
- RoHS Directive

The standards applied for certifying conformity are listed below.

- Safety (Low Voltage Directive): EN 61010-1
- EMC: EN 61326-1
- RoHS: EN 50581 or EN IEC 63000

Glossary

Daisy chain
Wiring diagram by which one controller is connected to the next in sequence (series connection principle). The first controller is connected directly to the PC. The additional controllers are always connected to the ones that precede them so that a chain is formed. The signal to and from a controller goes to the PC via the previous controllers.

Dynamics profile
The dynamic profile includes the target position, speed, and acceleration of the axis calculated by the profile generator of the electronics for each point in time of motion. The calculated values are called "commanded values".

GCS
Abbreviation for "General Command Set", the command set for electronics from PI. Piezo and servo controllers can be operated together with minimal programming effort thanks to GCS.

HID
Abbreviation for "Human Interface Device". HID refers to an input or output device that is connected to the electronics and intended for manual operation. Depending on the electronics, the connection can be made via USB, analog or digital interfaces. Joysticks and gamepads are typical HIDs.

Nonvolatile memory
Electronics read-only memory. The default values for the parameters are loaded from the nonvolatile memory into the volatile memory when the electronics are started. The parameter values in the nonvolatile memory are also referred to as "Startup Values" in the PC software from PI.

Volatile memory
Electronics main memory. Parameters are stored in the volatile memory when the device is switched on. The parameter values in the volatile memory determine the current behavior of the system. The parameter values in the volatile memory are also referred to as "Active Values" in the PC software from PI.