PZ116E User Manual

E-816 Computer Interface and Command Interpreter

Submodule for Piezo Controller

Release: 1.3.1 Date: 2021-09-22

This document describes the following product(s):

- E-816.00
  Computer Interface and Command Interpreter Submodule (firmware version 3.20 and newer) for Piezo Controllers

Physik Instrumente (PI) GmbH & Co. KG, Auf der Roemerstrasse 1, 76228 Karlsruhe, Germany
Phone +49 721 4846-0, Fax +49 721 4846-1019, Email info@pi.ws, www.pi.ws

First printing 2021-09-22
Document Number PZ116E Eco, BRo, Release 1.3.1
E-816_User_PZ116E131.doc

Related Documents

All PI devices and programs mentioned in this documentation are described in separate manuals.
You can download the manuals mentioned below summarized in zip archives from our website. To download, use the links "E-816" and "PI software" in the file "A000T0081-Downloading Manuals from PI.pdf", which you will find in the \Manuals folder on the PI software CD.
In general, the latest versions of our manuals are available for download on our website (www.pi.ws).

Manuals for Software:

PI GCS 2.0 DLL: Manual SM151E

GCS Driver for Use with NI LabVIEW Software: Manual SM158E

PIMikroMove: Manual SM148E

PI Update Finder: Manual A000T0028

Manuals for Controller Hardware

E-621.CR User Manual, PZ160E

E-621.SR, .LR User Manual, PZ115E

E-625.CR User Manual, PZ166E

E-625.SR, .LR User Manual, PZ167E

E-665 User Manual, PZ127E

E-801 User Manual, PZ117E

E-802 User Manual, PZ150E

Table of Contents

1. Introduction

The E-816 is a submodule that plugs into a piezo controller main board. Typically, it comes preinstalled as part of a pre-calibrated, ready-to-use system. The E-816 provides ASCII interfaces over which the device on which the submodule is installed can be controlled by a host computer.

- 24-bit A/D and 20-bit D/A resolution
- RS-232 interface and USB 2.0 (full speed/low speed) (not with some older hardware revisions)
- I^2C bus for interlinking up to 12 units in a network
- Wave table output for periodic motion
- Triggered motion (with firmware revision 3.20 and newer)
- Macro feature (with firmware revision 3.20 and newer)
- Individual calibration registers for offset and gain of piezo amplifier and sensor
- Analog input channels for monitoring actual piezo voltage and sensor position
- PI General Command Set

1.1. Intended Use

The E-816 is designed as a plug-in module for analog piezo control electronics, e.g. E-621, E-625 or E-665. It makes possible controlling the analog piezo control electronics via PC interface (USB, RS-232), macro programming and wave table output. The current voltage and position values can be queried over the PC interface.

The E-816 may only be used for applications suitable according to the device specifications. Operation other than instructed in this User Manual may affect the safeguards provided. Operate the E-816 only when you have read the operating instruction. Keep the instruction readily available close to the device in a safe place. When the instruction is lost or has become unusable, ask the manufacturer for a new copy. Add all information given by the manufacturer to the instruction, e.g. supplements or Technical Notes. Carefully read also the user manuals and/or technical notes of all other components involved, such as analog piezo control electronics or software.

The E-816 is a laboratory apparatus as defined by DIN EN 61010. It meets the following minimum specifications for safe operation:

- Indoor use only
- Altitude up to 2000 m
- Temperature range 5°C to 40°C
- Max. relative humidity 80% for temperatures up to 31°C, decreasing linearly to 50% relative humidity at 40°C

- Line voltage fluctuations not greater than ±10% of the line voltage
- Transient overvoltages as typical for public power supply
  Note: The nominal level of the transient overvoltage is the standing surge voltage according to the overvoltage category II (IEC 60364-4-443).
- Degree of pollution: 2

1.2. Safety Precautions

Install and operate the E-816 Computer Interface and Command Interpreter Submodule only when you have read the operating instruction. Keep the instruction readily available close to the device in a safe place. When the instruction is lost or has become unusable, ask the manufacturer for a new copy. Add all information given by the manufacturer to the instruction, e.g. supplements or Technical Notes.

CAUTION

E-816s are fully calibrated together with the other components of the system before being shipped from PI. It is not necessary for customers to recalibrate them.

Calibration should only be done after consultation with PI, otherwise the internal configuration data may be destroyed by erroneous operation.

Be careful when changing calibration parameters using the SPA command. If the current RAM values set with SPA are incompatible, the system may malfunction. Be sure that you have entered correct parameter settings before using the WPA command to save the current settings to non-volatile memory where they become the new power-on defaults.

CAUTION

Thermally stable systems have the best performance. For a thermally stable system, power on the piezo control electronics in which the E-816 is installed at least one hour before you start working with it.

CAUTION

If the piezo positioner starts oscillating (humming noise):

In closed-loop operation, switch off the servo immediately. The load and / or the dynamics of operation probably differ too much from the setup for which the system was calibrated.

In open-loop operation, stop the motion immediately. Do not operate the piezo positioner at its resonant frequency because the notch filter(s) may be deactivated in open-loop operation.

Otherwise the piezo positioner could be irreparable damaged.

CAUTION

Units with firmware revision older than 2.11 cannot be updated as described in “Firmware Updates” on p. 22. Contact PI if you want to update such units.

Use only suitable firmware files for firmware updates. The most significant digit of the current firmware revision must correspond to that in the name of the update hex file:

Units with firmware **2**.11 or newer require hex files with revision **2**.12 or newer.

Units with firmware **3**.20 or newer require hex files with revision **3**.21 or newer.

Do not use the USB interface for firmware updates.

1.3. Software Description

Operating system details:

- "Windows" stands for versions 8.1, 10 (32-bit, 64-bit)
- "Linux" stands for kernel 4.15.0; glibc: 2.23; GTK2: 2.24.30

The following table shows a selection of PI software suitable for the E-816.

For more information see the corresponding software manuals.

Software Tool	Supported Operating System	Short Description	Recommended for
Dynamic program library for GCS PI GCS 2.0 DLL	Windows, Linux	Allows software programming for the E-816 with programming languages such as C++. The functions in the dynamic program library are based on the PI General Command Set (GCS).	For users who would like to use a dynamic program library for their application. Is required for PIMikroMove. Is required for the NI LabVIEW drivers.
Driver for use with NI LabVIEW software	Windows	NI LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments). The driver library is a collection of virtual instrument drivers for PI electronics. The drivers support the PI General Command Set.	For users who want to use NI LabVIEW to program their application.

Software Tool	Supported Operating System	Short Description	Recommended for
PIMikroMove	Windows	Graphical user interface for Windows with which the E-816 and other controllers from PI can be used: ▪ The system can be started without programming effort ▪ Graphic representation of the motion ▪ Macro functionality for storing command sequences on the PC (host macros) ▪ Complete environment for command entry, for trying out different commands No command knowledge is necessary to operate PIMikroMove.	For users who want to do simple automation tasks or test their equipment before or instead of programming an application. A log window showing the commands sent makes it possible to learn how to use the commands.
PITerminal	Windows, Linux	Terminal program that can be used for nearly all PI controllers (refer to the description of the Command Entry window in the PIMikroMove user manual).	For users who want to send GCS commands directly to the E-816.
Firmware Update Wizard	Windows	The Firmware Update Wizard guides you through the update of the firmware for the hardware modules of your E-816 system.	Users who want to update the firmware.

2. First Steps

2.1. Installing the Software on the Host PC

Windows operating systems:

1 Start the installation wizard by double-clicking PISoftwareSuite.exe in the installation directory (root directory on the PI software CD).
The InstallShield Wizard window opens for installing the PC software.

2 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-816

− PI Update Finder for updating the PC software

− USB driver

Linux operating systems:

1 Unpack the tar archive from the /Linux directory on the PI software CD to a directory on your PC.

2 Open a terminal and go to the directory where you unpacked the tar archive.

3 Log in as superuser (root privileges).

4 Enter ./INSTALL to start the installation. Pay attention to lower and upper case when entering commands.

5 Follow the instructions on the screen.

For an overview over the host software provided see "Software Description" on p. 6.

2.2. Quick Start

CAUTION

Thermally stable systems have the best performance. For a thermally stable system, power on the piezo control electronics in which the E-816 is installed at least one hour before you start working with it.

CAUTION

If the piezo positioner starts oscillating (humming noise):

In closed-loop operation, switch off the servo immediately. The load and / or the dynamics of operation probably differ too much from the setup for which the system was calibrated.

In open-loop operation, stop the motion immediately. Do not operate the piezo positioner at its resonant frequency because the notch filter(s) may be deactivated in open-loop operation.

Otherwise the piezo positioner could be irreparable damaged.

Note

The following E-816 factory defaults are valid for the first start-up, unless agreed otherwise before delivery:

- Number of readings to use for an average: 32
  Can be changed using the AVG command (p. 38)
- Channel name (= axis identifier): “A” for single units (e.g. if you ordered a single E-625 or E-665 piezo controller); “B” up to “M” for multi-axis units (e.g. multiple E-621 modules on one chassis)
  Can be changed using the SCH command (p. 48)
- Data rate: 115,200 baud
  Can be changed in the range of 9,600 to 115,200 baud using the BDR command (p. 38)

Values set with AVG, SCH and BDR can be saved to non-volatile memory where they become the new power-on defaults. See the WPA command description (p. 54) for details.

The following instructions illustrate the first steps with the E-816 using PIMikroMove because that software offers a convenient user interface for operating the system. See the PIMikroMove Manual for the complete, more-detailed software description and "Installing the Software on the Host PC" (p. 7) for how to install the program.

In the example below, two networked E-816s are to be used. They are installed in two single-axis E-665s which are to be interconnected via an E-665.CN network cable and both drive single-axis piezo positioners. The channel names are B and V. V is to be the master unit, i.e. it must be directly connected to the host PC.

Proceed as follows to take the system into operation:

1. Set up the unit in which the E-816 is installed (e.g. E-621, E-625 or E-665) as described in the “Starting Operation” or “Quick Start” section of the corresponding User Manual.

   In particular, be sure the unit is configured to allow computer-controlled operation. Furthermore, the operating elements on the piezo control electronics should be set to open-loop operation (servo OFF; e.g. by the corresponding DIP switch or the SERVO ON/OFF toggle switch) to give the E-816 complete control over the servo mode selection. See “Control Modes” on p. 12 and the User Manual of the piezo control electronics for details.
   In the example, the ANALOG/DIGITAL toggle switches on both E-665s must be set to “DIGITAL”, and the SERVO ON/OFF toggle switches must be set to “OFF”.

2. If you are planning to run networked E-816s, prepare the system for networking:

   A unique channel name (= axis identifier) must be assigned to each of the units. E-816s delivered together installed in the same chassis (e.g. E-621 modules) come preconfigured with unique channel names starting with “B”, but the names of E-816s installed in stand-alone devices such as E-625s or E-665 by default are all set to “A” and need to be changed.

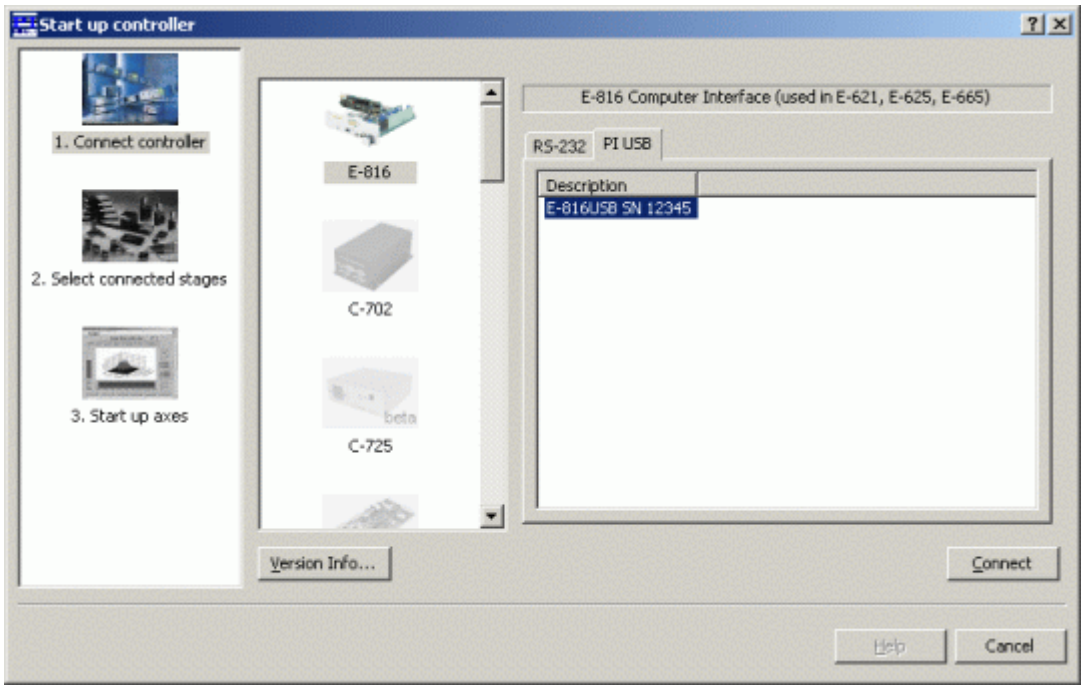
   Follow the instructions in “Setting Channel Names” on p. 28 to change the channel name of an E-816, and see “Checking Connection and Master Unit” on p. 29 for how to check the settings. In the example, “B” and “V” were assigned to the E-816s.

   If all channel names were adapted, interconnect the units which are to be networked as described in “Interlinking Multiple E-816s” on p. 27. After interconnecting all units, power-cycle them.

3. Connect the unit to the host PC using the included null-modem cable or USB cable. With a network of multiple E-816s, the device which is to be the master unit must be connected to the host PC that way (V in this example).

   The USB driver is provided on the PI software CD and installed with the installation wizard (see p. 7).

4. Start PIMikroMove on the host PC. It opens with the *Start up controller* window



5. Establish a connection from PIMikroMove to the E-816 (single unit or master unit of a network). This can be done via the USB or RS-232 interface, using the corresponding tab card in the *Start up controller* window (tab card selection depends on the current hardware connection). See "Communication" on p. 24 for details.

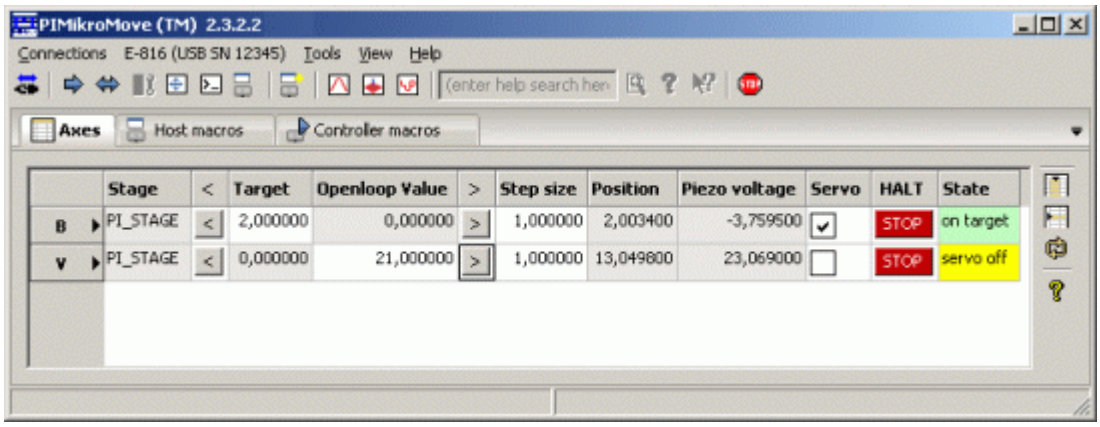
   When the connection is established successfully, the PIMikroMove main window will open. Note that you can configure the appearance of the *Axes* tab card, e.g. hide or show certain columns. See Section 2.3.1 in the PIMikroMove User Manual for details

6. Make some test moves with the individual axes using the controls on the *Axes* tab card:

   The first moves should be made in open-loop operation (*Servo* boxes must be unchecked). With the E-816, open-loop commanding means to give the desired output voltage values for the axes. Command an output voltage of 0 V by entering 0 in the *Openloop Value* field of an axis and pressing Enter on your keyboard. Then enter a new value of about 10% of the axis voltage range and press Enter. Increment the *Openloop Value* this way by steps of a suitable size, up to the upper voltage limit of the axis (see datasheet or measurement protocol of the mechanics), and then reduce it in an analogous manner to the lower voltage range limit of the axis. In doing so, observe the position display for the axis (in the *Position* field) and the current output voltage for the corresponding piezo actuator in the positioner (in the *Piezo Voltage* field). The output voltage values should follow the commanded values. At 0 V output voltage, the current position value of the axis should be approximately 0 µm, but due to the calibration settings of the system, the axis position can differ from 0 by about 20% of the axis travel range.

   You can also use the < and > buttons to decrement / increment the commanded value by the value given in the *Step size* field (1.0 V in the figure below).

If open-loop operation is successful, you can switch to closed-loop operation by checking the *Servo* boxes on the *Axes* tab card. In closed-loop operation, target positions for the axes are commanded. Enter the target position in the *Target* field and press Enter on your keyboard, or use the arrow buttons to decrement / increment the target position by the *Step size* value. The axis position (*Position* field) should correspond to the commanded value.



3. Operation

3.1. Axis Identifiers

E-816 submodules can be used as single devices or interlinked on the embedded I^2C bus as multi-axis system. Irrespective of the current usage, several commands require an axis identifier to address a certain unit (in this document, the terms “axis identifier” and “channel name” are used interchangeably). When multiple units are connected via the I^2C bus, arriving commands are forwarded to the proper unit according to its axis identifier.

Possible axis identifiers are A to Z. By default, the following axis identifiers are preset:

- “A” for single units (e.g. if you ordered a single E-625 or E-665 piezo controller)
- “B” up to “M” for multi-axis units (e.g. multiple E-621 modules on one chassis)

You can ask with the SAI? command (p. 48) for the axis identifiers of all E-816s on the I^2C bus. The axis identifier of the master unit (i.e. of the unit which is connected with the computer via the serial communications null-modem or USB cable) can be changed using the SCH command (p. 48).

To facilitate networking, E-816s behave as follows:

- The master unit is not only reachable with its identifier assigned with SCH but always also with ”A”
- If units whose axis identifier is set to “A” with the SCH command are connected as slave on the I^2C bus, they are not accessible since commands addressed to “A” will always be executed by the master unit

For more information, see “Networking on I^2C Bus” on p. 26 and “GCS Commands” on p. 33.

3.2. Modes of Operation

Control modes: The piezo channel (amplifier) of the control electronics can be operated in either analog mode or computer-controlled mode. The active mode determines the applicable control sources for the output voltage. See "Control Modes" below and the User Manual of the analog piezo control electronics (e.g. E-621, E-625 or E-665) for more information.

Servo modes: The current servo mode determines if a motion axis is driven in open-loop (servo OFF) or closed-loop (servo ON) operation. In closed-loop operation a servo loop participates in the generation of the control value for the piezo channel. The servo loop thus maintains the current axis position, based on a given target position and the position feedback of the corresponding sensor channel. See "Servo Modes (ON / OFF)" below and the User Manual of the analog piezo control electronics (e.g. E-621, E-625 or E-665) for more information.

The individual control and servo modes can be combined arbitrarily.

3.2.1. Control Modes

The current control mode of a piezo channel determines the applicable control sources for the output voltage and hence for the axis motion:

- **Analog mode:**
  The output voltage depends on analog control input and—if present—DC offset applied to the channel (e.g. a voltage applied to the socket labeled CONTROL IN or ANALOG IN; if present, setting of the DC-OFFSET knob).
  Move commands (received via interface or from a running macro), trigger input or wave table output are ignored.

  Analog control input can be a computer-generated analog signal (e.g. from a DAQ board). You can use PI's drivers for NI LabVIEW from the PI software CD to generate that analog signal.
  Note: After installing the drivers, it is necessary to activate the analog functionality, see the Analog_Readme.txt file in the installation directory (C:\ProgramData\PI\LabVIEW).

- **Computer-controlled mode:**
  The E-816 controls the generation of the output voltage. Target values for the axis motion can be given by move commands (received via interface or from a running macro), trigger input or wave table output. Analog control input voltage and DC offset settings are ignored.
  See "Applicable Control Sources" on p. 14 for details.

Switching between analog and computer-controlled mode can be done by DIP switches (E-621, E-625) or by the ANALOG/ DIGITAL toggle switch (E-665) on the front panel of the piezo control electronics in which the E-816 submodule is integrated. See the User Manual of your piezo control electronics for more information.

Notes

In analog mode, the E-816 accepts all commands just as in computer-controlled mode. The only difference between the modes is the control source selection of the channel.

3.2.2. Servo Modes (ON / OFF)

The current servo mode determines if a motion axis is driven in open-loop (servo OFF) or closed-loop (servo ON) operation:

- **Closed-loop operation:**
  Any control input (analog, DC offset, E-816 input like move commands and wave table output) is interpreted as target position. Based on this target position and on the position feedback of the corresponding sensor channel, a servo loop generates the control value for the piezo channel. The servo loop thus maintains the axis position. It is not located on the E-816 but in the analog part of the piezo control electronics (see the corresponding User Manuals of your system for more information). The servo loop applies a proportional-integral (P-I) servo-controller, a notch filter and a slew rate limiter.

- **Open-loop operation:**
  Any control input is interpreted as piezo voltage target. Open-loop operation omits the servo loop, and the control input directly controls the output voltage of the piezo channel.

The servo mode can be set as follows:

- Using DIP switch 3 (E-621, E-625) or the SERVO ON/OFF toggle switch (E-665) on the piezo control electronics, with E-621 also with pin 28a on the 32-pin main connector

- Via axis-specific SVO commands (p. 51) sent over the communications interface or received from a macro running on the E-816
  Using the SVO? command (p. 52), you can check the last sent SVO settings on a per-axis basis

Notes

To give the E-816 complete control over the servo mode selection, the operating elements on the piezo control electronics must be set to open-loop operation (servo OFF; e.g. by the corresponding DIP switch or the SERVO ON/OFF toggle switch). See the User Manual of your piezo control electronics for more information.

SVO? does not report the hardware settings for the servo mode but only the last sent SVO settings.

Closed-loop operation can be activated using a start-up macro. See “Start-Up Macro” on p. 20 for more information.

The example below illustrates the interpretation of the control input as piezo voltage target or target position, depending on the current servo mode. The E-816 is in analog mode in this example, i.e. controlled by analog input. The analog input signal remains unchanged while switching from open-loop to closed-loop operation, and the voltages and positions are read out via VOL? and POS? commands over the computer interface.

	Closed-loop operation (servo ON)	Open-loop operation (servo OFF)
Output voltage	78.78 V	86.46 V
Position	85.993 µm	92.886 µm

3.3. How to Command Axis Motion

Notes

With the E-816 there are no software limits on the commanded voltages or positions. If a voltage or position beyond the travel range of the connected axis is sent, it will be accepted, but the axis will move only to the limit position. No error is set, and the overflow condition may not go on.

The VOL? and POS? commands (p. 53 and 47) are used for reading current voltages and positions. The values reported by VOL? and POS? are the actual output voltages and axis positions, respectively.

3.3.1. Applicable Control Sources

The applicable control sources for the axis motion depend on the current control mode of a piezo channel (analog or computer-controlled mode, see "Control Modes" on p. 12 for details).

In analog mode, the axis motion is commanded by analog control input and DC offset applied to the channel. For the applicable voltage range of the input signal see the User Manual of the piezo control electronics (E-621, E-625 or E-665; analog mode may be referred to as "analog operation" there).

In computer-controlled mode, the axis motion can be commanded by multiple sources:

- Move commands (received via interface or from a running macro):
  SVA and SVR in open-loop operation; MOV and MVR in closed-loop operation (see below for examples and "Servo Modes (ON / OFF)" on p. 12 for more information)

- Trigger input which causes relative steps of a predefined size. Triggered motion is enabled/disabled by the MVT command (p. 45). The step size is given by a parameter 11 which can be set with the SPA command (p. 49). This functionality is available with firmware revision 3.20 and newer

- Wave table output for periodic motion (see "Working with the Wave Table" on p. 16 for more information and examples)

Move commands are not accepted when the wave table output is running (WTO) or when triggered motion is enabled (MVT). When the wave table output is running, triggered motion cannot be enabled and vice versa.

The macro feature allows defining command sequences and storing them permanently in non-volatile memory in the controller. Macros can run in analog and computer-controlled mode but move commands from macros are only accepted in computer-controlled mode (i.e. when the E-816 controls the generation of the output voltage). See "Working with E-816 Macros" p. 18 for more information.

3.3.2. Command Examples for Computer-Controlled Mode

The following examples can be used in a terminal, e.g. in the *Command Entry* window of PIMikroMove or in the PI Terminal.

Example 1:

An E-621 piezo control electronics is connected to a single-axis piezo positioner with sensor and 50 µm travel range. The axis is moved in open-loop operation.

Command	Response	Function
ERR?		Clear (master unit) error flag
SVO A 0		Set servo control off (open-loop operation) for axis A; there is no correction of drift or other effects. Make sure that the servo is not switched on by DIP switch 3 on the front panel or by pin 28a on the 32-pin main connector.
SVA A 80		Move axis A by setting the corresponding output voltage to 80 volts.
VOL? A	79.9947	Query the current output voltage which drives axis A. This command reads the A/D converter and delivers the current output. Note the difference to the SVA? command, which reports the last *commanded* value.
SVA A 150		Set the output voltage for axis A to 150.00 volts (this is over the limit)
ERR?	0	No error is set (master unit)
OVF?	0	Overflow is off
SVA? A	150	Query axis A last commanded voltage
VOL? A	110.34	Query the voltage which is actually output for axis A

Example 2:

An E-621 piezo control electronics is connected to a single-axis piezo positioner with sensor and 50 µm travel range. The axis is moved in closed-loop operation to three different absolute positions, one after the other. Then a relative motion is done. After each move, the current position is queried.

Command	Response	Function
SVO A 1		Set servo control on (closed-loop operation) for axis A; this also writes the current axis position to the target register, to avoid jumps of the mechanics.
MOV A 30.5		Command axis A to 30.5 µm
POS? A	30.4902	Query current position of axis A
MOV A 20		Command axis A to 20 µm
POS? A	19.8516	Query current position of axis A
MOV A 35		Command axis A to 35 µm
POS? A	35.0243	Query current position of axis A
MVR A -1		Command axis A to 34 µm (-1 relative to last target position)
POS? A	34.0248	Query current position of axis A

3.4. Working with the Wave Table

3.4.1. Basic Wave Table Operation

In computer-controlled operation, each E-816 can be controlled by an internal "wave table" which outputs user-specified target points. These target points are interpreted as positions or voltages, depending on the current servo mode selection. The wave table feature is especially important in dynamic applications which require periodic motion of the axes. Programmable trigger input facilitates synchronization of external events.

With firmware revision 3.20 and newer, the wave table provides 256 points (64 points with former revisions).

Wave table operation is as follows:

1. Define the wave table content for an axis point by point using the SWT command (p. 53). SWT saves the points automatically to non-volatile memory
2. Start the wave table output and hence the axis motion using the WTO command (p. 54). You can choose between immediate start and start by trigger input, and give the number of points to be output; see description of WTO command for details.
3. Stop the wave table output with WTO or #24 (p. 37)

With the SWT? command (p. 53) you can check the wave table content point by point.

NOTE

With PIMikroMove there is no need to learn the corresponding commands. PIMikroMove provides the *Wave Table Editor* with which you can define the wave table content and dialog windows where you can start and stop the wave table output. See the PIMikroMove manual for more information.

If you want to start wave table output by trigger input, make sure that the piezo control electronics is configured to accept trigger input, and that a suitable trigger signal is available (min. trigger pulse width = 200 µs; max. trigger frequency = 400 Hz; triggering when changing from low to high; detecting delay can be up to 5 µs):

- E-621, E-625: check DIP switch settings, connect suitable signal to ANALOG IN/WTT or ANALOG INPUT SMB socket
- E-665: connect suitable signal to I/O Connector (sub-D9f on rear panel)

For more information, see the User Manual for the piezo control electronics and “Trigger (Digital) Input” on p. 57.

As long as the wave table output is running, move commands are not accepted, and triggered motion cannot be enabled. If the E-816 is in analog mode, wave table output is ignored. See “Applicable Control Sources” on p. 14 for more information.

3.4.2. Special Output Configuration

Firmware revision 3.20 and newer provides parameter 12 for special configuration of the wave table output. You can change this parameter with SPA (p. 49) and save the value with WPA (p. 54). The value of parameter 12 is bit-coded as follows:

	Bit 1 “TrigOnce”	Bit 0 “En”
Description	If set, every trigger pulse received starts one wave table output cycle, i.e. all points set by the last WTO command are output once.	If set, the last saved status of WTO and SVO is recovered after power-on or reset
Default setting	0 = Normal wave trigger mode, i.e. one point is output per trigger pulse	0 = Status recovery disabled

If you want to start the output of a certain number of wave table points by one single trigger pulse (the first received), proceed as follows:

1. Enable bit 1 of parameter 12 by sending
   SPA *A* 12 2
   where *A* is the axis identifier
2. Determine the number of points to be output and the amount of time the output of each point takes:
   Send
   WTO *A n x.x*
   where *A* is the axis identifier, *n* is the number of wave-table points to use for output (must be > 0), and *x.x* is the amount of time in milliseconds (must be > 0)
3. Start a wave table output cycle by applying a trigger pulse which means that the number of wave table points given by the WTO command in step 2 will be output. Every further trigger pulse starts a new output cycle. Output can be stopped by a WTO *A* 0 command or by #24.

If you want to save this special configuration to non-volatile memory to make it the power-on default, proceed as follows:

1. Enable bit 0 of parameter 12 by sending
   SPA *A* 12 1
   where *A* is the axis identifier
2. Save the current settings to non-volatile memory by sending
   WPA 100
   This command saves the SPA settings and also the current WTO and servo (SVO) settings. Since bit 0 of parameter 12 is now enabled, the saved WTO and SVO settings will be recovered on power-on or reset. This means that starting the output of a certain number of wave table points by one single trigger pulse (the first received) is the new default behaviour.

   Note that WPA 100 saves also macros and the current values of other parameters settable by the SPA, AVG, BDR, DCO and SCH commands

If you want to change the output configuration back to the factory default, disabled bit 0 and bit 1 of parameter 12 using SPA and save the settings with WPA 100.

3.5. Working with E-816 Macros

With firmware revision 3.20 and newer, the E-816 supports macro functionality. The macro feature allows defining command sequences and storing them permanently in non-volatile memory in the E-816. Each defined macro can be called up by its own user-defined name. In addition, it is possible to define a macro that will be executed automatically every time the E-816 is started, making possible stand-alone operation without a host computer. See the subsections below and the MAC command description (p. 42) for more details and examples. MAC commands are executed by the master unit only (unit connected with RS-232 or USB cable to host computer) and cannot be addressed to a slave unit.

When a macro was created, has been changed, deleted or (re-)defined as start-up macro using the appropriate MAC command, you should afterwards save the modifications to E-816s nonvolatile memory using the WPA command (p. 54; saves also the current values of parameters settable by the SPA, AVG, BDR, DCO and SCH commands). Otherwise all changes are discarded when powering down or restarting the E-816.

NOTES

PIMikroMove offers a comfortable macro editor on the *Controller macros* tab card.

Furthermore, PIMikroMove offers the "Host macro" feature which makes it possible to save macros on the host PC. With the Host macro feature, you can also program conditions and loops in macros which is not directly supported by the E-816. See the PIMikroMove manual for more information regarding Host macros.

3.5.1. Defining Macros

To define a macro command sequence on an E-816, first make it the master unit, i.e. connect it to the host PC directly using the RS-232 or USB interface. Then activate macro recording mode with the command MAC BEG <macroname> where <macroname> is a user-settable name with a maximum of 8 characters. While in macro recording mode, commands are not executed but stored in macro storage. Recording mode is exited by the MAC END command. The maximum number of macros to be stored on the E-816 is 16.

If there is not enough memory, the whole macro you typed in is not stored, and there is no warning about this! You can use the ERR? command (p. 41) to check if there is an appropriate error code not until you typed MAC END.

In macro recording mode, the macro is written to volatile memory only. To save it to non-volatile memory, you must send WPA 100 after you have exited the macro recording mode with MAC END.

Although macros can only be recorded on the master unit, they can include commands for slave units (see “Command Survey” on p. 35 for a list of commands valid for slave units). When working with macros in a network, the master unit should not be changed. If macros are present on multiple controllers in the network, it is recommended to make a note since MAC? and MAC DEF? are answered only by the master unit.

A macro can start another macro without any nesting level limitation. A macro can call itself to form an infinite loop.

During macro recording no macro execution is allowed.

A macro can be overwritten by a macro with the same name.

A running macro sends no responses to any interface. This means questioning commands are allowed in macros but not answered and therefore useless.

Example 1: This simple macro initializes a network of three E-816s (A, B, C) and performs moves forward and backward. Note that with the E-816 only one axis can be addressed per command.

`MAC BEG macro1`	Start recording macro "macro1"
`SVO A 1`	Enable servo control mode for axis A (the master unit)
`SVO B 1`	Enable servo control mode for axis B (slave unit)
`SVO C 1`	Enable servo control mode for axis C (slave unit)
`DCO A 1`	Enable drift compensation mode for axis A. The E-816 now keeps track of proper motion.
`DCO B 1`	Enable drift compensation mode for axis B
`DCO C 1`	Enable drift compensation mode for axis C
`MOV A 10`	Move axis A to position 10 µm.
`MOV B 10`	Move axis B to position 10 µm.
`MOV C 10`	Move axis C to position 10 µm.
`DEL 1000`	Delay of approx. 1000 ms.
`MOV A 0`	Move axis A to its home.
`MOV B 0`	Move axis B to its home
`MOV C 0`	Move axis C to its home
`DEL 1000`	This delay only matters if the macro is run several times in a loop.
`MAC END`	Exit the macro recording mode. The macro now is complete.

Example 2: Macro1 now is split up in two separate macros (macro2 and macro3) because it is not necessary to initialize the E-816s every time motion is to be performed. When you start macro3 this implicates that macro2 is executed 10 times.

Write macro2:

```
MAC BEG macro2
MOV A 10
MOV B 10
MOV C 10
DEL 1000
MOV A 0
MOV B 0
MOV C 0
DEL 1000
MAC END
```

Write macro3:

```
MAC BEG macro3
SVO A 1
```

```
SVO B 1
SVO C 1
DCO A 1
DCO B 1
DCO C 1
MAC NSTART macro2 10
MAC END
```

3.5.2. Starting Macro Execution

A defined macro can be run on the master unit by the command MAC START <macroname> where <macroname> is the name that was given to the macro to be run.

To run a macro multiple times on the master unit, call it with MAC NSTART <macroname> n where n gives the number of times the macro is to be run (1 to 65535).

Macro execution will be stopped if there is an error in the macro—either syntax error or illegal command.

Move commands from a running macro are not accepted when wave table output is running or triggered motion is enabled. If the E-816 is in analog mode, move commands from a running macro are ignored. See “Control Modes” on p. 12 for details.

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

Macro execution can be stopped with #24 (p. 37).

During macro execution, no other macro operation is allowed (e.g. creating or deleting macros is not possible).

You can query with #8 (p. 37) if a macro is currently running on the controller. Keep in mind that in a network, #8 is only answered by the master unit. It is not possible to query if macros are running on slave units.

3.5.3. Start-Up Macro

With MAC DEF <macroname> used on the master unit, it is possible to set the specified macro as start-up macro. This macro will be automatically executed on every power-on or reset of the E-816.

When a start-up macro is defined, after power-on or reset the controller waits 1.2 s before the macro is executed to avoid timing problems.

Commands in a macro running on a slave unit in a network can be addressed to axes other than this slave. If start-up macros running on multiple units in a network depend on each other, it may be necessary to include delays in the macros (with the DEL command) to avoid timing problems.

Example:

```
MAC BEG init
SVO A 1
SVO B 1
SVO C 1
DCO A 1
DCO B 1
DCO C 1
MAC END
```

This macro switches the E-816s in the network to closed-loop operation and enables drift compensation mode.

To ask for the current start-up macro setting, send

```
MAC DEF?
```

To undo the current start-up macro selection, send

```
MAC DEF
```

i.e. omit <macroname>.

Note that if you change the content of the start-up macro, you have to reactivate its start-up functionality afterwards by typing MAC DEF <macroname> again.

Deleting the current start-up macro with MAC DEL <macroname> also deletes the start-up macro selection.

When a macro has changed, was created, has been deleted or (re-)defined as start-up macro using the appropriate MAC command, you should afterwards save the modifications to nonvolatile memory using the WPA command (p. 54). Otherwise all changes are discarded when powering down or restarting the E-816.

3.6. Drift Compensation

Drift compensation is provided to eliminate drift in the digital-analog converters on the E-816. Activating it with the DCO command (p. 39) will avoid unwanted change in displacement over time in static operation. It should be deactivated during dynamic operation. Drift compensation works by eliminating unjustified, slow changes in the E-816 analog-side target output based on the position, piezo voltage and digital target.

Drift compensation mode is disregarded during wave-table output. If DCO was on, after the wave table output has finished, it again becomes active. This can cause slight motion as the digital and analog positions adjust. Therefore, if it is important to hold the position at the end of wave-table motion, deactivate DCO before starting.

3.7. Updates

3.7.1. Software Updates

Updating the PC software in Windows

- Use the PI Update Finder: Follow the instructions in the PI Update Finder manual (A000T0028).

Updating the PC software in Linux

1 Open the website https://www.physikinstrumente.com/en/products/motion-control-software/

2 Click Login.

3 Log in with the user name and password for the PI software CD.

Information on the access data is in the "C-990.CD1_Releasenews.pdf" in the root directory on the PI software CD.

4 Scroll down to Downloads.

5 Click the archive file "CD Mirror" or the associated download link.

6 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.

7 Unpack the archive file into a separate installation directory.

8 Go to the linux subdirectory in the directory with the unpacked files.

9 Unpack the archive file in the linux directory by entering the command tar -xvpf <name of the archive file> on the console.

10 Read the accompanying information on the software update (readme file and/or "C-990.CD1_Releasenews.pdf") and decide whether the update makes sense for your application.

- If no: Stop the update procedure.
- If yes: Go through the following steps.

11 Log into the PC as superuser (root privileges).

12 Install the update.

3.7.2. Firmware Updates

If your E-816 is directly connected to the host PC (i.e. is the master unit), its firmware revision can be identified in the answer of the *IDN? command (p. 37).

CAUTION

Units with firmware revision older than 2.11 cannot be updated as described here. Contact PI if you want to update such units.

Use only suitable firmware files for firmware updates. The most significant digit of the current firmware revision must correspond to that in the name of the update hex file:

Units with firmware **2**.11 or newer require hex files with revision **2**.12 or newer.

Units with firmware **3**.20 or newer require hex files with revision **3**.21 or newer.

Do not use the USB interface for firmware updates.

Firmware updates for the master unit can be made by running the Firmware Update Wizard on the host computer using the RS-232 interface.

The Firmware Update Wizard is provided on the PI software CD and installed with the installation wizard (see p. 7).

Figure 1: The Firmware Update Wizard Start Screen

In the *Select PI controller* window, select *Auto detection* and press the *Connect...* button. The Firmware Update Wizard guides you through the firmware update of your E-816.

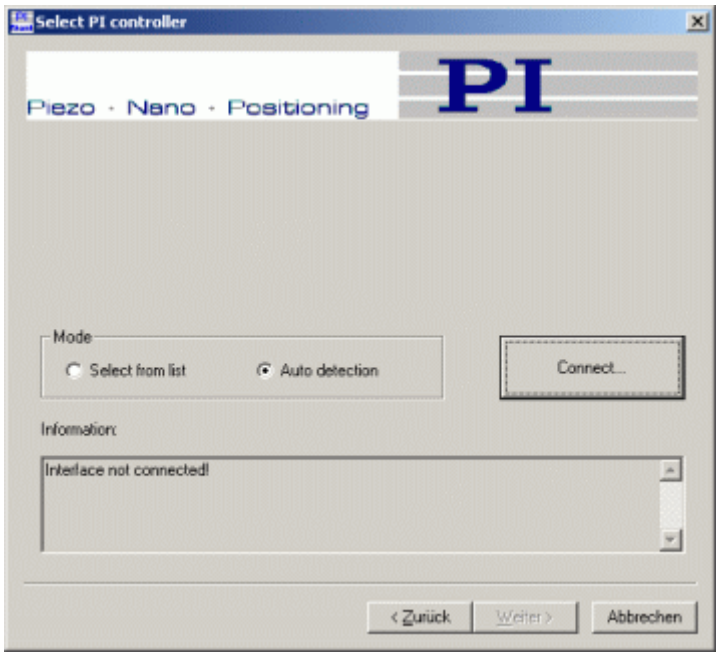


Figure 2: Select controller to be connected to Firmware Update Wizard

4. Communication

4.1. Interfaces Available

The E-816 can be controlled from a host computer (not included) with ASCII commands sent via:

- RS-232 serial connection
- USB connection (USB 2.0, full speed/low speed), available with the latest hardware revision

Both interfaces can be active simultaneously. The commands from the interfaces are queued in the order the completed command lines are received.

Notes

Older hardware revisions are not equipped with USB interface.

The USB driver is provided on the PI software CD and installed with the installation wizard (see p. 7).

Using multiple interfaces simultaneously may cause problems with the host software.

When controlling the E-816, timing problems can occur if several commands are run in rapid sequence, resulting in lost commands. To prevent such communication errors, it is recommended that you include a certain wait time between the different commands, depending on the command to be executed. This is especially true for commands that need a certain execution time inside the E-816 module, like MOV, MVR, SPA, SVA, SVR, RST, WPA, SWT and WTO.

When a start-up macro is defined, after power-on or reset the controller waits 1.2 s before the macro is executed to avoid timing problems.

4.2. RS-232 Serial Communication

The serial communications port is accessed via a sub-D 9m "RS-232" connector on the piezo control electronics in which the E-816 is integrated. Use the null-modem cable that comes with the piezo control electronics to connect the device to the host PC; if the PC has only one COM port, it is probably COM 1.

The serial port on the E-816 is preset to the following parameters:

115,200 baud, 8 data bits, 1 stop bit, no parity, RTS/CTS

In the connection dialog of the host software (e.g. PIMikroMove, PITerminal, or drivers for use with NI LabView), you make the settings on the host PC side. Select the correct PC COM port and make sure that the baud rate (and other settings) in the dialog match those of the E-816. Otherwise no communication can be established.

In a network of multiple E-816s, the unit connected directly to the host PC via the RS-232 interface is the master unit. All other units are slaves (see “Networking on I^2C Bus” on p. 26 for more information). If you are networking multiple E-816s which have different baud rate settings, you must set the baud rate in the host

software to be the same as the baud rate of the current master unit. Otherwise all communication will fail.

The currently active baud rate of a unit directly connected to the PC via the RS-232 interface can be changed as follows:

1. Use the BDR command (p. 38) in the command entry facility of the program, e.g. by sending BDR 57.6 (notation is mandatory—typing e.g. 57600 will not work!)

2. Check with BDR? (p. 39) if the setting was accepted, i.e. if the response corresponds to the value set with BDR before

3. Save the setting to nonvolatile memory by sending WPA 100 (p. 54). Otherwise it will not take effect

4. To make the value from nonvolatile memory the current setting, reset the E-816 with the RST command (p. 47)

5. Close the connection in the program

6. Open the connection again with the baud rate you set with BDR (in the example 57.6).

4.3. USB Communication

The USB interface is accessed via a mini USB A socket on the piezo control electronics in which the E-816 is integrated. Use the USB cable (USB-A (m)/USB Mini-B (m)) that comes with the piezo control electronics to connect the E-816 to the host PC.

The USB driver is provided on the PI software CD and installed with the installation wizard (see p. 7). Note that installation files for USB drivers are also located on the CD in the directory \SingleSetups.

In the host software (e.g., PIMikroMove, PITerminal, or drivers for use with NI LabView), you see all E-816s which are connected to the USB sockets of the host PC. In the *Start up controller* window of PIMikroMove, for example, the present devices are listed on the *PI USB* tab card. Click on the E-816 to which you want to connect. Then press the *Connect* button to establish the connection.

NOTES

With USB connections, communication cannot be maintained after the E-816 is power-cycled or reset. The connection must then be closed and reopened.

Do not use the USB interface for firmware updates.

4.4. Networking on I²C Bus

If you want to use multiple E-816s in one system, interlinking them on the embedded I²C bus is the best solution. The benefits are:

- Only one host-computer serial port is necessary for accessing all the E-816s.
- Support for up to 12 E-816s on one I²C bus (the number of devices that can be connected to the bus is limited by the bus capacitance limit of 400 pF and the 7-bit slave address space)

4.4.1. Overview

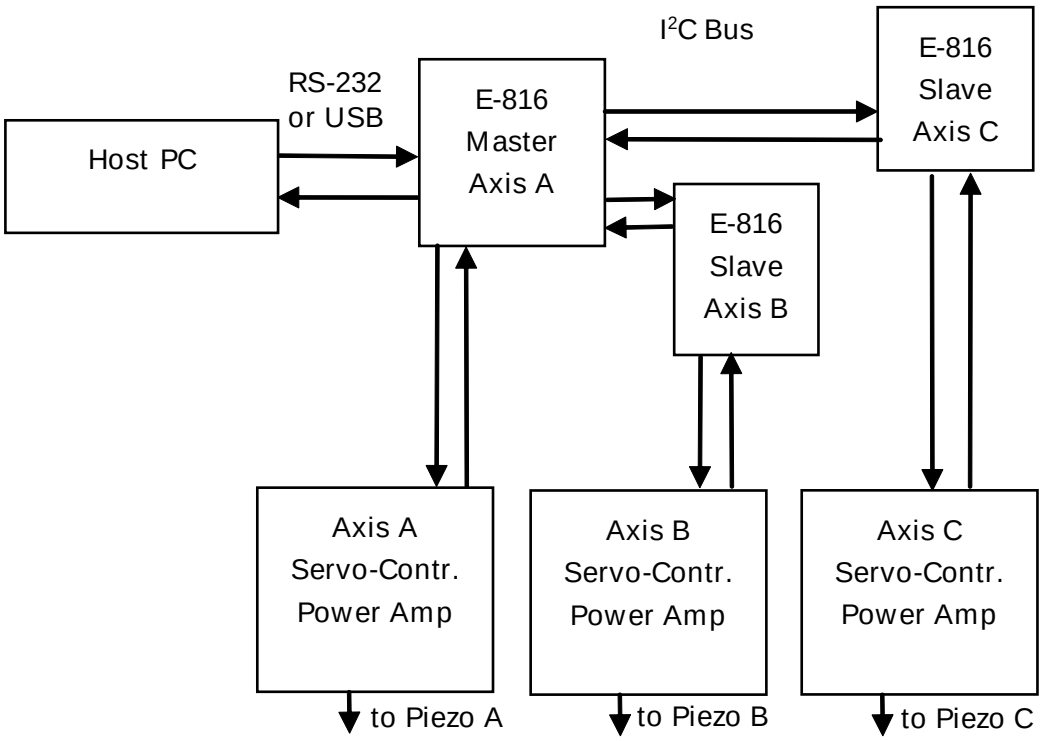


Fig. 1: System overview

Master Unit: the unit which is connected with computer via the serial communications null-modem or USB cable. The master unit is always addressable with the channel name “A”.

Slave Unit: a unit which is connected with the master unit via the I²C bus but with no direct connection to the computer. A few commands cannot be addressed to a slave unit (see p. 35 for a list of valid commands), so it may occasionally be necessary to make the unit the master (move the RS-232 or USB cable from one unit to another and power-cycle all E-816s) before it can execute those commands.

Channel or Axis Name or **Identifier:** When multiple units are connected via the I²C bus, arriving commands are forwarded to the proper unit according to its channel name. In this document, the terms “channel” and “axis” are used interchangeably, as are “name” and “identifier”.

4.4.2. Interlinking Multiple E-816s

Interlink all units to be networked before you power them up. Otherwise it may be necessary to power-cycle all units for proper detection of the master unit.

Interlinking multiple E-816s depends on the piezo control electronics in which the E-816 is integrated:

- With E-621 controller modules, the I^2C bus is implemented on the 32-pin main connector. If E-621s come installed in compatible PI chassis (E-500.621 or E-501.621), networking of multiple E-621s with each other requires no additional wiring.

- To network E-625s and E-665s, a way of busing the required network lines is required. This can be done with E-665.CN cables (not included), which have piggyback sub-D connectors allowing more than one cable to be plugged into a single E-625 or E-665. Because of the bus restrictions mentioned above, at most three such cables can be used.

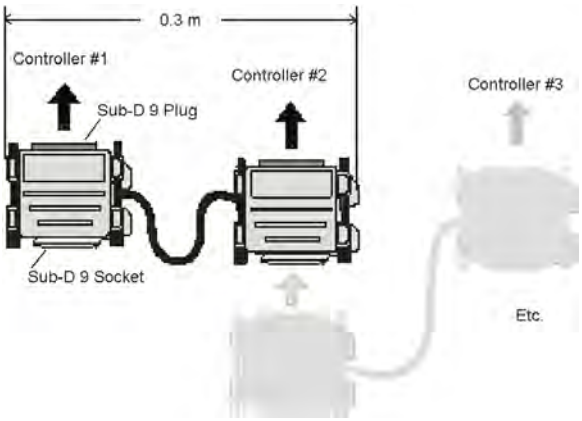


Fig. 2: E-665.CN networking cable

For more information regarding interlinking multiple units see the User Manual of your piezo control electronics.

How to change the master unit:

1. Remove the RS-232 or USB cable from the present master
2. Connect the RS-232 or USB cable to the new master
3. Power-cycle all E-816s (reset with RST is not sufficient to change the master!)

As long as the new master has the same baud rate setting as the former master, with an RS-232 connection the master can be changed while a connection to the host software is enabled. With USB connections, communication cannot be maintained after the E-816 is power-cycled or reset. The connection must therefore be closed and reopened after changing the master.

4.4.3. Master/Slave Command Processing

If, for example, the computer sends the following command to the master unit:

MOV C 5.0

The master unit will process it as follows:

1. If the master unit has been named channel “C”, it will process the command and issue the response, if any.
2. Otherwise, the master unit looks for the unit with channel name ”C". If it finds one, it forwards the command to that unit. If it does not find one, it sets an error status bit
3. The slave unit with channel name ”C” receives the command from the master, processes it, and if necessary, returns a response.
4. The master unit receives the response (if any) from the slave unit and forwards it to the host computer.

Note: The master unit is always reachable with channel name ”A”. That means that if the computer sends a command like MOV A 5.0, the command will always be processed by the master unit, even if there is a slave unit set up with channel name “A”.

4.4.4. Setting Channel Names

To set the channel name (= axis identifier) of an E-816, proceed as follows:

1. Make that E-816 the master unit: connect the E-816 to the computer with the serial communications null-modem or USB cable and power-cycle it.
2. Start the PITerminal software which is provided on the PI software CD (see “Installing the Software on the Host PC” on p. 7 for how to install PITerminal)
3. Press the *Connect* button and establish communications in the *Connect* dialog, either via the *RS-232* or the *PI USB* tab card, depending on the physical connection to the host PC. When using the *RS-232* tab card, be sure to select the correct COM port and interface settings (default: 115,200 baud, RTS/CTS, 8 data bits, 1 stop bit, no parity)
4. Use the SCH command (p. 48) to assign the desired channel name to the E-816:
   Send: SCH *x*
   with *x* the new channel name.
   Note that “A” must not be assigned to a unit which will later be accessed as a slave.
5. Use the WPA command (p. 54) to write the setting to the E-816 EEPROM.
   Send: WPA 100

   Example:

   You receive an E-816 from PI and want to set its channel name to ”E”. Connect the device to the computer, start PITerminal and send the following commands:

   SCH?

   Response: A

   SCH E

   SCH?

   Response: E

   WPA 100

Change the channel name to a unique value for every device to be networked via the I^2C bus.

4.4.5. Checking Connection and Master Unit

If you are networking multiple E-816s, you can get the axis names of all connected units with the SAI? command (p. 48) and determine which is the master unit by asking its axis name with the SCH? command (p. 49). If these commands time out, there is probably a problem with the connection to the master unit. Check the baud rate settings as explained under the BDR command (p. 39).

Example:

SAI?

Response: BCF

SCH?

Response: C

The responses indicate that 3 devices with channel names B, C and F are connected on the I^2C bus. The master unit is C, meaning that it is addressable either as C or A.

5. Calibration

One of the key benefits of the E-816 is the calibration parameter structure. After the device has completed the calibration runs in our lab, absolute positions and voltages can be commanded. The calibration data is stored in non-volatile ROM. Write access is protected by password.

As long as the system configuration has not changed or readjusted, calibration data is valid and does not need to be modified. Note that the values of the E-816 calibration parameters operate “on top of” the hardware calibration settings of the potentiometers on the servo-control and amplifier boards. If the latter are adjusted, the corresponding E-816 parameters must be reset.

The E-816 has 10 calibration parameters, as shown in the E-816 block diagram below. Six parameters are for internal use and will be set by PI only (shown with dotted lines). For safety sake, we grant customers read access only. Four calibration parameters—Ksen, Osen, Kpzt, Opzt—can be changed by the customer, based on the servo-controller, positioner and amplifier used.

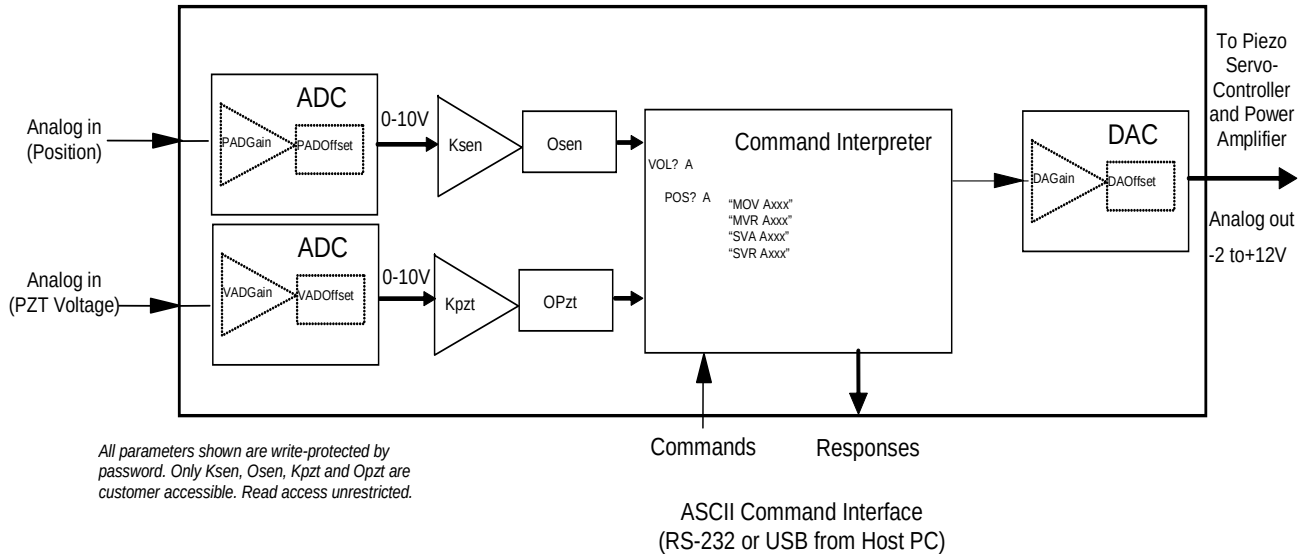


Fig. 3: E-816 block diagram and calibration parameter

CAUTION

E-816s are fully calibrated together with the other components of the system before being shipped from PI. It is not necessary for customers to recalibrate them.

Calibration should only be done after consultation with PI, otherwise the internal configuration data may be destroyed by erroneous operation.

Be careful when changing calibration parameters using the SPA command. If the current RAM values set with SPA are incompatible, the system may malfunction. Be sure that you have entered correct parameter settings before using the WPA command to save the current settings to non-volatile memory where they become the new power-on defaults.

All positioner-specific data are stored in non-volatile memory (flash ROM) and a copy is made in RAM at power-up. The SPA and SPA? commands (p. 49 and p. 50) can be used to access the RAM copy. These commands have the following format:

SPA *A n x.x*LF
SPA? *A n*LF

where:

SPA[?] command mnemonic
A axis identifier (= channel name)
n parameter ID
x.x parameter value, floating point
LF LineFeed (Char #10).

Example:

Command: SPA? A 1

Report: -3.6427

The SPA command saves the parameters in RAM only. To save these and other currently valid parameters to non-volatile memory (flash ROM), where they become the power-on defaults, you must use the WPA command (p. 54). Parameter changes not saved with WPA will be lost when the E-816 is powered off or reset.

ID	*Name*	*Type*	*Parameter Description*
1*	VADGain	Float	Gain of the A/D converter which measures the piezo voltage*****
2*	VADOffset	Float	Offset of the A/D converter which measures the piezo voltage*****
3*	PADGain	Float	Gain of the A/D converter which measures the sensor voltage*****
4*	PADOffset	Float	Offset of the A/D converter which measures the sensor voltage*****
5*	DAGain	Float	Gain of the D/A converter*****
6*	DAOffset	Float	Offset of the D/A converter*****
7	Ksen	Float	Sensor coefficient K_s in µm/V (when sensor output changes by 1V, the position change of positioner is K_s µm)
8	Osen	Float	Sensor offset O_s in µm (when sensor output is 0V, the actual position of positioner is O_s µm)
9	Kpzt	Float	Piezo voltage amplifier coefficient K_{pzt} in V/V (when D/A converter output changes by 1V, the piezo voltage at amplifier output changes by K_{pzt} V)
10	Opzt	Float	Piezo voltage amplifier offset O_{pzt} in V (when D/A converter output is 0V, the piezo voltage at amplifier output is O_{pzt}

* Parameters 1-6 are for PI use only, normally no write access for customer

Since the *E-816* is designed as a plug-in module for piezo control electronics and can be used with different combinations of high-voltage and low-voltage piezo amplifiers and position servo-controllers, the last 4 parameters (Ksen, Osen, Kpzt, Opzt) must be set to values appropriate for the piezo positioner(s) and amplifier(s) installed. If the E-816 was ordered together with the amplifier plug-in module, servo-controller and piezo positioner, these 4 parameters will have been set at the factory before shipment.

If, however, the E-816 is being used with separately purchased PI positioners, the above 4 parameters must be determined and set. These parameters must also be reset if the hardware calibration elements on the servo-controller (e.g. E-802 submodule), sensor readout (e.g. E-801 submodule) or power amplifier (e.g. E-621) are moved. The formulas for calculating these parameters are:

$Ksen = (P_{10} – P_0) / 10.0$	$Kpzt = (V_{10} – V_0) / 10.0$
$Osen = P_0$	$Opzt = V_0$

Where:

P_{10} is the actual positioner position when sensor monitor outputs 10 V
P_0 is the actual positioner position when sensor monitor outputs 0V
V_{10} is the actual piezo voltage when the input to the piezo amplifier is 10 V in open-loop mode (servo OFF)
V_0 is the actual piezo voltage when the input to the piezo amplifier is 0 V in open-loop mode (servo OFF).

Example 1:

A positioner has sensor monitor output of 0-10 V, the positioner travel is to be 0 µm to approx. 50 µm, the piezo voltage is to be 0 and 100 V when analog input is 0 V and 10 V, respectively, in open–loop mode. The settings must then be:

Ksen = (50.0 - 0.0) / 10.0 = 5.0
Osen = 0.0
Kpzt = (100.0 - 0.0) / 10.0 = 10.0
Opzt = 0.0

Example 2:

The piezo voltage in open-loop mode is to be -0.5 and 100.5 V when the control input is 0 V and 10 V respectively. Then:

Ksen = (25 - (-25)) / 10.0 = 5.0
Osen = -25
Kpzt = (100.5 - (-0.5)) / 10.0 = 10.1
Opzt = -0.5

Example 3:

A positioner has sensor monitor output of 0 to 10 V, the nominal positioner extension is 0 µm to 15 µm when the piezo voltage is 0 to 100 V. However, after being calibrated with an external standard, the real piezo extension is found to be 0 µm to 14.5 µm and the piezo voltage to be 0 to 98 V. Then

Ksen = (14.5 - 0) / 10.0 = 1.45
Osen = 0
Kpzt = (98 - 0) / 10.0 = 9.8
Opzt = 0

6. GCS Commands

Commands are used to set operating modes, transfer motion parameters and query system and motion values. Because of the variety of functions and parameters, a sequence of commands is usually needed to achieve a specific system action.

6.1. Format

6.1.1. GSC Syntax

Commands are transmitted as ASCII characters in the following format (for examples see Section 3.3.2 on p. 15):

CMD SP X SP sV.V LF

where

CMD command mnemonic (token)

SP space (char #32).

X axis designation (A , B , C, ...),

s sign (positive values can be transmitted without sign)

V.V parameter, values are floats or integers, depending on the command.

LF LineFeed (Char #10).

Optionally, the second space can be omitted. The space and LineFeed are not marked explicitly in all the examples in this manual. Current firmware versions also accept CR (carriage return, ASCII 13) as command termination character.

Example:

Send: MOV A 10.0

Move axis A to position 10.0.

Response Format:

Some commands deliver a report (response). Responses always consist of a single line, unlike those from many other PI controllers. They are made up of ASCII characters and have one of the following formats:

sV.V LF

sV LF

where:

s sign (positive values are transmitted without sign)

V.V or **V** result, representation of double-precision or integer values, depending on the command

LF LineFeed (Char #10).

Example:

Send: POS? A

Report: 1.0006

6.1.2. Commands with Floating Point Arguments

Some commands accept parameters in floating point format. For these parameters the following syntax is possible:

sv LF

sv.v LF

sv.vEsxx LF

where:

s sign (positive values can be without sign)

v ASCII digit or digits (will be converted into floating point internally by firmware)

v.v the decimal separator must be a ".", not ","

E exponent separator

sxx 2-digit exponent with optional sign

LF LineFeed (Char #10).

In reports, floating point values are always represented as follows:

sv.vvvv LF

where:

s sign (positive values transmitted without sign)

v.vvvv always transmitted with 4 digits after decimal point (internal calculations use double precision)

LF LineFeed (Char #10).

6.1.3. Limitations for GCS Commands

More than one command mnemonic per line is not allowed.

The number of characters per line is limited to 25 bytes (1 character = 1 byte).

Only one axis can be addressed per command. With the SPA and SPA? commands, only one parameter for the given axis can be accessed per command.

Responses always consist of a single line, unlike those from many other PI controllers.

When controlling the E-816, timing problems can occur if several commands are run in rapid sequence, resulting in lost commands. To prevent such communication errors, it is recommended that you include a certain wait time between the different commands, depending on the command to be executed. This is especially true for commands that need a certain execution time inside the E-816 module, like MOV, MVR, SPA, SVA, SVR, RST, WPA, SWT and WTO.

6.2. Command Survey

Command	Parameter	Description	Page
The commands in this section are executed by the master unit only (unit connected with RS-232 or USB cable to host computer) and cannot be addressed to a slave.			
ERR?		Get master unit error code	41
I2C?		Get status of I^2C bus	42
*IDN?		Get master unit version information	37
SSN?	*A*	Get master unit serial number	50
#24		Stop wave table output, macros and delay on master unit	37
BDR	*N*	Set master unit baudrate	38
BDR?		Get master unit baudrate	39
AVG	*N*	Set number of samples to use for averages on master unit	38
AVG?		Get number of samples being used for averages on master unit	38
SCH	*c*	Set channel name (= axis identifier) of master unit	48
SCH?		Get channel name (= axis identifier) of master unit	49
SAI?		Get names assigned to all connected (networked) axes	48
MAC	BEG[START/ NSTART/DEL [*.*]/ DEF/END/ DEF?/FREE?]	Calls macro function on master unit BEG *name*: Start recording macro called *name* START *name*: Start macro called *name* NSTART *name* n: Execute macro *name* n times, n should be in the range from 1 to 65535 DEL *name*: Delete macro called *name* DEF *name*: Set *name* as start-up macro END: End macro recording DEF?: Ask name of start-up macro FREE?: Get free memory to store additional macros	42
MAC?	[*name*]	List the content of one macro or the names of all macros on the master E-816	43
#8		Test if a macro is running on master unit	37
DEL	*n*	Delay the controller for n milliseconds, recommended for usage in macro operation	40
WPA	*password*	Write all master unit parameters to master unit flash ROM **	54
RST		Reset the master unit	47
HLP?		Get online help list; currently no information is available	41

Command	Parameter	Description	Page
The following commands are forwarded by the master unit to the unit assigned to control the specified axis for execution.			
DCO	*A n*	Set D/A converter drift compensation on or off	39
DCO?	*A*	Get D/A converter drift compensation setting	39
MOV	*A x.x*	Move the given axis to absolute position	44
MVR	*A x.x*	Move the given axis relative to current position	45
MOV?	*A*	Read the last commanded position of the given axis	44
SVA	*A x.x*	Set the given axis to absolute piezo voltage	51
SVR	*A x.x*	Change the given axis piezo voltage relative to current value	52
SVA?	*A*	Read the last commanded piezo voltage of the given axis	51
MVT	*A n*	Set “move triggered” mode on/off for the given axis	45
MVT?	*A*	Get current “move triggered” mode of the given axis	46
DIP?	*A*	Ask if a digital pulse was detected since the last call of DIP? for the given axis	40
POS?	*A*	Read the actual position of the given axis	47
VOL?	*A*	Read the actual piezo voltage of the given axis	53
OVF?	*A*	Get overflow status of the given axis	47
ONT?	*A*	Get on-target status of the given axis	46

Command	Parameter	Description	Page
SVO	*A n*	Set servo-ON/OFF status of the given axis	51
SVO?	*A*	Get servo-ON/OFF status of the given axis	52
SWT	*A n x.x*	Set wave table data	53
SWT?	*A n*	Get wave table data	53
WTO	*A n x.x*	set wave table output	54
SPA	A *n x.x*	Set specified parameter of the specified axis*	49
SPA?	A *n*	Get specified parameter of the specified axis*	50

*** Parameters 1-6 are for PI use only, no write access for customer**

**** Password protected, must use "WPA 100" to write the parameters to flash ROM.**

6.3. Command Reference (Alphabetical)

#8 (Macro running?)

Command Type:	Report Command
Description:	Query if a macro is running on the master unit by sending the single ASCII character 8. Note that it is not possible to query if macros are running on slave units in a network. Available with firmware revision 3.20 and newer.
Format:	#8 (single ASCII character number 8)
Arguments:	none
Response:	0 (ASCII character 48) no macro is running 1 (ASCII character 49) a macro is running

#24 (Stop all axes)

Command Type:	Move Command
Description:	Stop macro execution (MAC), wave table output (WTO) and delay (DEL) on the master unit. Sets error code to 10. Available with firmware revision 3.20 and newer.
Format:	#24 (single ASCII character number 24)
Arguments:	none
Response:	none

*IDN? (Get Identity Number) *IDN?

Command Type:	Report Command
Description:	Reports master unit device identity and firmware number
Format:	*IDN?
Arguments:	none
Response:	One-line string, terminated by LF.

AVG (Set Average Times) AVG

Command Type:	Configuration Command
Description:	Sets the number of samples to be used when calculating averages on the master unit. Larger values mean more stable output, but slower measurement speed. The default is 32. This command only changes the setting in RAM; the new setting will be lost when the unit is powered down or reset (RST command) unless the RAM settings are written to ROM with WPA 100.
Format:	AVG *n*
Arguments:	*n*, the number of samples. It must be one of following values: 1, 2, 4, 8, 16, 32 or 64; sending other values can have unpredictable results and may not set the error flag.
Response:	None

AVG? (Read Average Times) AVG?

Command Type:	Report Command
Description:	Reports the current average setting of the master unit.
Format:	AVG?
Response:	The number of samples used to calculate averages on the master unit. Will be 1, 2, 4, 8, 16, 32 or 64.

BDR x.x (Set Baud Rate) BDR

Command Type:	System Configuration Command
Description:	Set master unit RS-232 communications baud rate.
Format:	BDR *x.x*
Arguments:	*x.x,* the desired baud rate in thousands (must be one of 9.6, 19.2, 38.4, 57.6, 115.2).
Response:	None
Note:	This command only changes the setting in RAM; the new setting will be lost when the unit is powered down or reset (RST command) unless the RAM settings are written to ROM with WPA 100. The new baud rate does not take effect before the next power on or RST. (So to take effect at all, it must be saved to ROM!) Incorrect entries (such as 56) have unpredictable results and may not set an error status. Check RAM setting with BDR? before attempting to use it.
Example:	The current power-on baud rate is 9.6, the desired baud rate is 115.2. Start the terminal program at 9.6, type in the following commands: BDR 115.2 BDR? Response: 115.2

WPA 100

RST

Now the device runs at a baud rate of 115.2. You must now close the connection in the host software and establish a new connection with 115.2.

If you forget the current ROM baud rate and cannot communicate with the device, try different baud rates in the host software. Verify the baud rate selection with the *IDN? command—if the E-816 does not respond to this command, try another baud rate.

BDR? (Get Baud Rate) BDR?

Command Type:	Report Command
Description:	Get the master unit RAM baud rate setting for RS-232 communication.
Format:	BDR?
Arguments:	None.
Response:	Can be 9.6, 19.2, 38.4, 57.6, 115.2
Note:	This is the setting that will be written to flash ROM if a WPA command is executed. The baud rate in use is the one that was in ROM at the time of the last power-on.

DCO (Set Drift Compensation Mode) DCO

Command Type:	Configuration Command
Description:	Sets the drift compensation for the digital-analog converter ON or OFF. Drift compensation avoids unwanted changes in displacement over time but is recommended for static operation only. For a detailed description see p. 21. Note that drift compensation is not performed during wave table output, even if DCO is set to 1. This command only changes the setting in RAM; the new setting will be lost when the unit is powered down or reset (RST command) unless the RAM settings are written to ROM with WPA 100.
Format:	DCO *A n*
Arguments:	A: Axis identifier. *n*: 1 or 0. 1 for ON, 0 for OFF.
Response:	none

DCO? (Get Drift Compensation Mode) DCO?

Command Type:	Report Command
Description:	Reports the Drift Compensation Mode setting
Format:	DCO? *A*
Arguments:	*A*: Axis identifier.

Response: "1" for ON, "0" for OFF

DEL (Set Delay) DEL

Command Type: Configuration Command

Description: Delays the master unit for *n* milliseconds, recommended for usage in macro operation (MAC, p. 42)
Available with firmware revision 3.20 and newer.

Can be aborted with the #24 command.

Format: DEL *n*

Arguments: *n*: delay time in ms, maximum delay time is 13.107 s

Response: none

DIP? (Ask if Digital Pulse was Detected) DIP?

Command Type: Report Command

Description: Reports if a digital pulse was detected for the given axis since the last call of DIP?. After this command, the E-816 resets the pulse flag.
Available with firmware revision 3.20 and newer.

Format: DIP? *A*

Parameters: *A*: Axis identifier

Response: A=1: digital input has detected pulse
A=0: digital input has no pulse detected

Note: For meaningful results, make sure that the piezo control electronics in which the E-816 is installed is configured properly:

- E-621, E-625: check DIP switch settings, connect suitable signal to ANALOG IN/WTT or ANALOG INPUT SMB socket
- E-665: connect suitable signal to I/O Connector (sub-D9f on rear panel)

For more information, see the User Manual for the piezo control electronics and "Trigger (Digital) Input" on p. 57. Note that the trigger input line may also be referred to as "Wave table trigger input".

ERR? (Get Error Message) ERR?

Command Type: Report Command

Description: Reports error code of the last occurred error on the master unit and reset the error to 0. Only the last error is buffered. Therefore you should call ERR? after each command.

Format: ERR?

Arguments: none

Response: Error Codes

Code	*Meaning*
0	No error
1	Parameter syntax error
2	Unknown command
3	Command length out of limits or command buffer overrun
5	Unallowable move attempted on unreferenced axis, or move attempted with servo off
10	Controller was stopped by command
15	Invalid axis identifier
17	Parameter out of range
20	Macro not found
54	Unknown parameter
56	Password invalid
60	Protected Param: current Command Level (CCL) too low
73	Motion commands are not allowed when wave generator is active
79	Open-loop commands (SVA, SVR) are not allowed when servo is on
89	Command not allowed in current motion mode
210	Illegal file name (must be 8-0 format)
232	Save system configuration failed
233	Load system configuration failed
306	Error on I2C bus
309	Insufficient space to store macro
405	Wave parameter out of range

HLP? (Get Online HELP List) HLP?

Command Type: Report Command

Description: Reports device's online help list for the master unit. This command is provided for compatibility reasons since the current E-816 firmware is not yet able to generate a help string.
Available with firmware revision 3.20 and newer.

Format: HLP?

Parameters: none

Response: String "No help available"

I2C? (Get I2C Status) I2C?

Command Type: Report Command

Description: Reports the status of the I^2C bus that connects networked controllers. Status codes are cleared after they are reported. Can only be used with the master unit.

Format: I2C?

Arguments: None

Response: Status codes of I^2C-Bus are reported as bit-mapped hex values. I^2C bus status code bits:

bit 0 (LSB):	CHK_SEN0 timeout
bit 1:	CHK_PEN0 timeout
bit 2:	CHK_RSEN0 timeout
bit 3:	CHK_RW0 timeout
bit 4:	CHK_BF0 timeout
bit 5:	CHK_BF1 timeout
bit 6:	CHK_ACK0 timeout
bit 7: (MSB)	SLAVE_BUSY timeout

MAC (Macro Operation) MAC

Command Type: Macro Command

Description: Calls macro function for master unit. Macro function can be "BEG", "DEF", "DEF?", "DEL", "END", "FREE?", "NSTART" or "START" Available with firmware revision 3.20 and newer.

When you start macro recording with MAC BEG, all commands input is considered as macro content. After MAC END is detected, the controller exits the macro recording mode and normal operation can continue.

Note that a macro will not be saved to E-816 nonvolatile memory until WPA 100 was sent (see p. 54). WPA 100 is also required if an existing macro has been changed, was deleted or defined as start-up macro. Changes not saved with WPA are only present in RAM and will be lost when the E-816 is powered off.

Use #8 (p. 37) to check if a macro is running on the master unit.

Use #24 (p. 37) to stop a running macro on the master unit.

Use ERR? (p. 41) to check macro operation on the master unit.

See Section 0 on p. 17 for further details.

Format: MAC BEG[START/NSTART/DEL [*.*]/DEF/END/DEF?/FREE?]

Arguments: Macro functions:

BEG *name*: Start recording of the macro called *name*.

DEF *name*: Set *name* as default or start-up macro. This macro will be executed at startup. After power-on or reset the controller waits 1.2 s before the start-up macro is executed to avoid timing

problems. Use #24 to stop start-up macro execution on the master unit. If *name* is "" or NULL, no macro is running at startup.

DEF?: Ask *name* of default or start-up macro.

DEL *name*: Delete macro called *name*. MAC DEL *.* deletes all macros.

END: End macro recording.

FREE?: Get free memory to store additional macros, in number <uint> of characters.

NSTART *name n*: Execute macro *name n* times (*n* should be in the range from 1 to 65535; use #24 (p. 37) to stop execution)

START *name*: Start macro called *name* (use #24 (p. 37) to stop execution).

name: A macro name may consist of 1 to 8 characters.

n: number of repetitions, integer

Response:

MAC BEG: none
MAC DEF: none
MAC DEF? : <name> of default or start-up macro (this macro is currently defined as default macro and executed at startup), if no start-up macro is defined, the response is LF
MAC DEL: none
MAC END: none
MAC FREE?: <uint> is the number of characters that can be saved (free memory)
MAC NSTART: none
MAC START: none

Problem Solver:

MAC BEG will fail if the controller is already recording a macro

MAC END will fail if the controller is not recording a macro

No macro was recorded since there is not enough memory, check with ERR? (response: 309) and MAC FREE?

Controller does not execute commands:
A macro is recorded (controller in macro recording mode)—stop recording with MAC END, than the controller exits macro recording mode.
E-816 is not in computer-controlled mode
Move commands are not accepted when wave table output is running or triggered motion is enabled

MAC? (List macro) MAC?

Command Type: Macro Command

Description: Lists the content of one macro (with *name* parameter) or the names of all macros on the master unit.
Available with firmware revision 3.20 and newer.

See Section 0 on p. 17 for further details.

Format: MAC? [*name*]

Arguments: *name*: name of a macro

Response:	When a macro called *name* exists, the content of the macros is returned as multi-line response (see p. 33), otherwise the response is LF If *name* is "" or NULL, the names of all macros available on the master unit are listed.

MOV (Move Axis Absolute) MOV

Command Type:	Move Command
Description:	Move the specified axis to the commanded absolute position. Before using this command, please make sure the E-816 is in computer-controlled mode and has servo-control mode set ON. See “Modes of Operation” on p. 12 for more information. Move commands like MOV are not accepted when the wave table output is running (WTO) or when triggered motion is enabled (MVT).
Format:	MOV *A x.x*
Arguments:	*A*: axis identifier; *x.x*: the commanded value, in µm. Both parameters are required.

MOV? (Read Commanded Position) MOV?

Command Type:	Report Command
Description:	Reports the commanded position of the specified axis
Format:	MOV? *A*
Arguments:	*A:* axis identifier. The parameter is required.
Response:	Reports the last commanded position of the specified axis. The value returned is floating point in µm.

MVR (Move Axis Relative) MVR

Command Type:	Move Command
	Description: increment/decrement the position of the specified axis by the commanded value.
	Before using this command, please make sure the E-816 is in computer-controlled mode and has servo-control mode set ON. See "Modes of Operation" on p. 12 for more information.
	Move commands like MVR are not accepted when the wave table output is running (WTO) or when triggered motion is enabled (MVT).
Format:	MVR *A x.x*
Arguments:	*A* : axis identifier,
	x.x: the commanded (relative) value, in µm.
	Both parameters are required.
Response:	none

MVT (Set Move Triggered Mode) MVT

Command Type:	Configuration Command
Description:	Sets the "move triggered" mode ON or OFF. If this mode is enabled for an axis, every trigger pulse received causes a relative step. The step size is given by parameter 11 which can be set with the SPA command. Available with firmware revision 3.20 and newer.
Format:	MVT *A n*
Parameters:	*A*: axis identifier,
	n: 1 or 0. 1 for ON, 0 for OFF.
Response:	none
Notes:	Before you enable the "move triggered" mode, make sure the E-816 is in computer-controlled mode (see "Control Modes" on p. 12). Furthermore, the piezo control electronics must be configured to accept trigger input, and a suitable trigger signal must be available (min. trigger pulse width = 200 µs; max. trigger frequency = 400 Hz; triggering when changing from low to high; detecting delay can be up to 5 µs):
	➢ E-621, E-625: check DIP switch settings, connect suitable signal to ANALOG IN/WTT or ANALOG INPUT SMB socket
	➢ E-665: connect suitable signal to I/O Connector (sub-D9f on rear panel)
	For more information, see the User Manual for the piezo control electronics and "Trigger (Digital) Input" on p. 57. Note that the trigger input line may also be referred to as "Wave table trigger input".

The target will be updated after every trigger pulse as follows:
New target = last target + step size
Depending on the current servo mode, target and step size values are interpreted as position (closed-loop mode) or voltage (open-loop mode). There is no range check for the target values resulting from the given step size. See “How to Command Axis Motion” on p. 14 for further information.

The setting made with MVT is lost upon reset or when the device is powered down. Default setting is 0.

Triggered motion cannot be enabled as long as the wave table output is running. When triggered motion is enabled, move commands (e.g. SVA, MOV) are not accepted and wave table output (WTO) cannot be started.

Examples:

```
SVO A 0
SVA A 0
SPA A 11 2.0
MVT A 1
-- trigger
MVT A 0

SVO A 1
MOV A 0
SPA A 11 -1.0
MVT A 1
-- trigger
MVT A 0
```

MVT? (Get Move Triggered Mode) MVT?

Command Type: Report Command

Description: Reports the current “move triggered” mode of the given axis. Available with firmware revision 3.20 and newer.

Format: MVT? *A*

Arguments: *A*: Axis identifier.

Response: “1” for “move triggered” mode ON, “0” for OFF

ONT? (Get On Target Status) ONT?

Command Type: Report Command

Description: Report the on-target status of the given axis as reported by the servo-controller (e.g. the E-802).

Format: ONT? *A*

Parameter: *A:* Axis identifier.

Response: Replies 1 when the given axis is on target, 0 otherwise.

Note**:** For servo-controllers that do not provide an on-target signal, the return value is meaningless. See the servo-controller (submodule) manual for more information.

OVF? (Get Overflow Status)	OVF?

Command Type:	Report Command
Description:	Reports the status of the overflow signal of the given axis.
Format:	OVF? *A*
Arguments:	*A:* Axis identifier.
Response:	Reply 1 if overflow occurred for the given axis, 0 otherwise. See the hardware manual for the board on which the E-816 is installed for information on when the overflow signal is on.

POS? (Read Real Position)	POS?

Command Type:	Report Command
Description:	Reports the current, actual position of the specified axis, as measured by the sensor and digitized by the ADC. The difference between the POS? and MOV? commands is that POS? gives the real position measured by sensor, while MOV? gives the last commanded position (see MOV). The reported value is an averaged result. The number of values used for the average can be changed with the AVG *n* command.
Format:	POS? *A*
Arguments:	*A* : axis identifier. This parameter is required.
Response:	The current position of the specified axis. The value is floating point, in µm.

RST (Reset the Master Unit)	RST

Command Type:	Configuration Command
Description:	Reset the master unit. Is ignored in macros. With USB connections, communication cannot be maintained after the E-816 is reset. The connection must then be closed and reopened. Reset with RST is not sufficient to change the master. The master unit remains master after a reset, even if the communications cable is pulled before the unit is ready, so changing masters requires power-cycling.
Format:	RST
Arguments:	*none*
Response:	none
NOTE	**Wait about 10s to let the unit get ready for next command**

SAI? (Get Axis Identifier) SAI?

Command Type:	Report Command
Description:	Get channel (axis) names of all connected devices. SAI? is answered by the master unit.
Format:	SAI?
Arguments:	None
Response:	String with the channel names of all devices in the I^2C network.
Example:	You send: SAI? Response: BCF means 3 devices with channel name 'B', 'C' and 'F' are connected on the I^2C bus.

SCH (Set Channel Name) SCH

Command Type:	System Configuration Command
Description:	Set the channel name (axis identifier) of the E-816. Although SCH can only be used with the master unit, the axis identifier set is that to be used to address this unit, whether it is connected as the master or as a slave. This command only changes the setting in RAM; the new setting will be lost when the unit is powered down or reset (RST command) unless the RAM settings are written to ROM with WPA 100 (p. 54).
Format:	SCH *c*
Arguments:	*c*, the channel name of the device, can be any letter from A to Z.
Note:	**If the channel name is set to A, the unit cannot be addressed as slave.** The master unit can always be addressed using A; commands with A as axis identifier will always be executed by the master unit. An "SCH A" command thus serves only to delete an unknown or unwanted setting that may have been made earlier.
Response:	None
Example:	You received three E-816s from PI and want to interlink them via I^2C bus. The following steps should be performed: Step 1: plug the RS-232 cable or USB cable from the host into the first unit, power-cycle the units, establish a connection in the software (e.g. PITerminal), then set desired channel name by command: SCH B WPA 100 Step 2: plug the RS-232 cable or USB cable into the second unit, power-cycle the units, establish a connection in the software, then set desired channel name by command: SCH C WPA 100

Step 3: plug the RS-232 cable or USB cable into the third unit, power-cycle the units, establish a connection in the software, then set desired channel name by command:

SCH D

WPA 100

Step 4: make sure all three units are connected to the I^2C bus (e.g. E-621s plugged into the same chassis), and plug the RS-232 cable or USB cable into the unit which is to be the master (with firmware revision 2.xx it may be necessary to connect the cable to the *first* unit to have access to all units). Power-cycle the units. Type in the following command:

SAI?

Response: BCD

The device should reply BCD indicating that all 3 units are accessible via the one RS-232 or USB port with the respective axis names.

The above procedure also saves these settings to non-volatile memory (flash ROM) in each unit, so they will be available after subsequent power-ons.

SCH? (Get Channel Name)	**SCH?**
Command Type:	Report Command
Description:	Report the channel name (= axis identifier) of the master unit.
Format:	SCH?
Arguments:	None
Response:	Channel name of the master unit, will be an upper-case letter from A to X

SPA (Set Parameter)	**SPA**
Command Type:	Configuration Command
Description:	Set specified parameter for given axis. With firmware revisions 2.xx, SPA is only available for the master unit. With firmware revision 3.20 and newer it can also be used with slave units. The parameters to be set with SPA can be: ▪ Calibration parameters: Each axis has 10 calibration parameters. See p. 31 in Section "Calibration" for the definitions of the different parameters. **NOTE: parameters with IDs 1-6 are reserved for PI use only; customers have no write access to these parameters.** Normally, the system is fully calibrated and tested before it leaves the factory. ▪ Configuration parameters for triggered motion and wave table output: With firmware rev. 3.20 and newer, the

additional parameters 11 and 12 are available. See below for details.

This command saves the parameters in RAM only. **To save these and other currently valid parameters to flash ROM, where they become the power-on defaults, you must use the WPA command. Parameter changes not saved with WPA will be lost when the E-816 is powered off or reset.**

Format: SPA *A n x.x*

Arguments: *A*: axis identifier; with firmware revisions 2.xx it must be either 'A' or the channel name of the master axis

n: the ID of the parameter and
x.x: the value of the parameter in floating point:

IDs in the range of 7 to 10 refer to calibration settings, see Section "Calibration" on p. 30 for possible values
ID = 11 gives the step size for triggered motion, the value is interpreted as µm in closed-loop operation or as volts in open-loop operation
ID = 12 configures wave table operation, the value is bit-coded as follows (see "Special Output Configuration" on p. 17 for details):

	Bit 1 "TrigOnce"	Bit 0 "En"
Description	If set, every trigger pulse received starts one wave table output cycle, i.e. all points set by the last WTO command are output once.	If set, the last saved status of WTO and SVO is recovered after power-on or reset
Default setting	0 = Normal wave trigger mode, i.e. one point is output per trigger pulse	0 = Status recovery disabled

SPA? (Get Parameter) SPA?

Command Type: Report Command

Description: Read specified parameter for given axis.
With firmware revisions 2.xx, SPA? is only available for the master unit. With firmware revision 3.20 and newer it can also be used with slave units.

Format: SPA? *A n*

Arguments: *A*: axis identifier; with firmware revisions 2.xx it must be either 'A' or the channel name of the master axis

n: the ID of the parameter, should be in range of 1 to 12 (see SPA description for details).

Response: Value of the specified parameter in floating point.

SSN? (Get Serial Number) *IDN?

Command Type: Report Command

Description: Reports serial number of master unit

Format: SSN? *A*

Arguments:	*A* : axis identifier of master unit
Response:	One-line string with device serial number, terminated by LF.

SVA (Set Piezo Voltage Absolute) SVA

Command Type:	Move Command
Description:	Set the piezo voltage of the specified axis to the commanded value. Before using this command, please make sure the E-816 is in computer-controlled mode and has servo-control mode set OFF. See “Modes of Operation” on p. 12 for more information. Move commands like SVA are not accepted when the wave table output is running (WTO) or when triggered motion is enabled (MVT).
Format:	SVA *A x.x*
Arguments:	*A* : axis identifier *x.x*: the commanded piezo output voltage value in volts. Both parameters are required.
Response:	none

SVA? (Get the Commanded Piezo Voltage) SVA?

Command Type:	Report Command
Description:	Reports the commanded piezo voltage of the specified axis
Format:	SVA? *A*
Arguments:	*A* : axis identifier
Response:	Reports the last commanded piezo voltage of the specified axis.

SVO (Set Servo-Control Mode) SVO

Command Type:	Configuration Command
Description:	Set the servo-control mode of the given axis. To give the SVO command complete control over the servo mode selection, the operating elements on the piezo control electronics must be set to open-loop operation (servo OFF; e.g. by the corresponding DIP switch or the SERVO ON/OFF toggle switch). See the User Manual of your piezo control electronics and “Modes of Operation” on p. 12 for more information.
Format:	SVO *A n*
Arguments:	*A*: Axis identifier. *n:* 1 Servo-control mode, 1 for set to ON, 0 for set to OFF.
Response:	None

SVO? (Get Servo-Control Mode) SVO?

Command Type:	Report Command
Description:	Reports the last sent SVO settings of the given axis. SVO? does not report the hardware settings for the servo mode. WPA saves the current SVO settings. But to make them the new power-on defaults, you must set parameter 12 with SPA to the corresponding value (see SPA, p. 49). With the default setting of parameter 12, the last saved SVO settings are not recovered on power-on or reset.
Format:	SVO? *A*
Arguments:	*A:* Axis identifier.
Response:	"1" for servo-control mode ON, "0" for OFF

SVR (Set Piezo Voltage Relative) SVR

Command Type:	Move Command
Description:	Increase/decrease piezo voltage of the specified axis by the commanded value. Before using this command, please make sure the E-816 is in computer-controlled mode and has servo-control mode set OFF. See "Modes of Operation" on p. 12 for more information. Move commands like SVR are not accepted when the wave table output is running (WTO) or when triggered motion is enabled (MVT).
Format:	SVR *A x.x*
Arguments:	*A*: axis identifier *x.x*: the commanded (relative) piezo output voltage value in volts. Both parameters are required.
Response:	none

SWT (Set Wave Table Data) SWT

Command Type:	Configuration Command
Description:	Sets a wave-table data point. The data is automatically stored in non-volatile memory and can be reused after next power-up. The first time data is written to the wave table, it is recommended to define all points. Afterwards, it may be sufficient to define certain points.
Format:	SWT *A n x.x*
Arguments:	*A*: Axis identifier. *n*: the index of the wave table data point, must be in 0-63. With firmware revision 3.20 and newer, 256 points are available (index is 0 to 255). *x.x* is the wave-table data point value, the value is interpreted as µm in closed-loop operation or as volts in open-loop operation
Response:	0 if successful, 1 otherwise.
Example:	see WTO command

SWT? (Get Wave Table Data) SWT

Command Type:	Report Command
Description:	Gets a wave-table data point.
Format:	SWT? *A n*
Arguments:	*A*: Axis identifier. *n*: the index of the wave-table data point, must be in 0 to 63. With firmware revision 3.20 and newer, 256 points are available (index is 0 to 255).
Response:	Value of the wave-table data point
Example:	see WTO command

VOL? (Read Real Piezo Voltage) VOL?

Command Type:	Report Command
Description:	Reports the current, actual piezo voltage of the specified axis, as measured by the A/D converter. The difference between the VOL? and SVA? commands is that VOL? gives the current piezo voltage measured by the ADC, while SVA? gives the last commanded voltage (e. g. from SVA command).
Format:	VOL? *A*
Arguments:	*A* : axis identifier. This parameter is required.
Response:	The current piezo voltage of the specified axis. The value returned is in floating point, in volts.

WPA (Write Parameters) WPA

Command Type:	Configuration Command
Description:	Master unit only: the current values of parameters settable by SPA, AVG, BDR, DCO and SCH are written to nonvolatile memory (EEPROM), where they become the new power-on defaults. Furthermore, WPA saves macros, and is also required if an existing macro has been changed, was deleted or defined as start-up macro (see MAC command). Note that the volatile (RAM) value of BDR does not go into effect until after it is written to ROM and the system reset, so the RAM value may differ from the current operating value. WPA also saves the current WTO and SVO settings. But to make them the new power-on defaults, you must set parameter 12 with SPA to the corresponding value (see "Special Output Configuration" on p. 17 for details). With the default setting of parameter 12, the last saved WTO and SVO settings are not recovered on power-on or reset.
Format:	WPA *xxx*
Arguments:	*xxx*: password (the password is **100**)
Response:	none
Problem Solver:	Incorrect password
NOTE	**If the current RAM values are incompatible, the system may malfunction. Be sure that you have entered the correct parameter settings before using this command.**

WTO (Enable/Disable Wave Table Output) WTO

Command Type:	Move Command
Description:	Sets the wave-table output mode for the given axis. Before starting wave table output, please make sure the E-816 is in computer-controlled mode. See "Modes of Operation" on p. 12 for more information. Wave table output cannot be enabled as long as triggered motion is enabled (MVT). When wave table output is running, move commands (e.g. SVA, MOV) are not accepted and triggered motion (MVT) cannot be enabled. During wave-table output drift compensation (see DCO) is not carried out even if set to 1. During wave-table output it is possible to change values in the wave table, with changes taking effect when the next point is output. WPA saves the current WTO settings. But to make them the new power-on defaults, you must set parameter 12 with SPA to the corresponding value (see "Special Output Configuration" on p. 17 for details). With the default setting of parameter 12, the last saved WTO settings are not recovered on power-on or reset. See "Working with the Wave Table" on p. 16 for more information.
Format:	WTO *A n x.x*

Arguments: *A*: Axis identifier.

n: the number of wave-table points to use for output. If n = 0, wave-table output is disabled, otherwise wave-table data points from index 0 to n-1 are specified for output.

x.x: if nonzero, output of the points specified by n will be started immediately and each point will be output for the amount of time specified by x.x in milliseconds. Output will roll over from point n-1 to 0 and continue until stopped by a WTO *A* 0 command or by #24.

If *x.x* is omitted or 0, one wave-table point is output each time an external trigger signal is received (default setting, for configuration option see “Special Output Configuration” on p. 17).
Make sure that the piezo control electronics is configured to accept trigger input, and that a suitable trigger signal is available (min. trigger pulse width = 200 µs; max. trigger frequency = 400 Hz; triggering when changing from low to high; detecting delay can be up to 5 µs)):

- E-621, E-625: check DIP switch settings, connect suitable signal to ANALOG IN/WTT or ANALOG INPUT SMB socket
- E-665: connect suitable signal to I/O Connector (sub-D9f on rear panel)

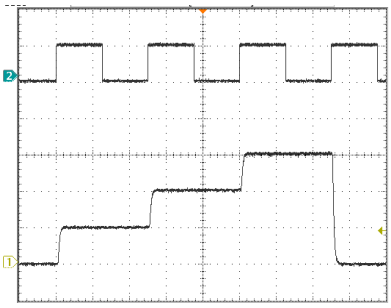
For more information, see the User Manual for the piezo control electronics and “Trigger (Digital) Input” on p. 57. Note that the trigger input line may also be referred to as “Wave table trigger input”.

Response: None

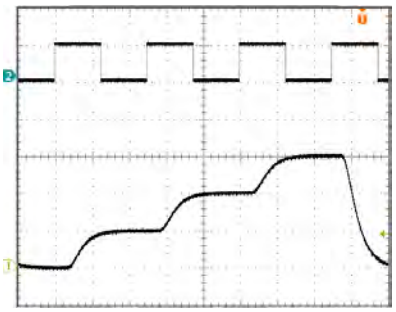
Example: **Servo Off, travel range of axis C =5 µm**

Trigger signal

Piezo motion



10Hz



100Hz

Note: if the WTO command is entered with servo mode commanded OFF (open-loop operation), the amplitudes are interpreted as voltages, otherwise (in closed-loop operation) the amplitudes are interpreted as positions (e.g. µm).

Command	Function
SVO A 0	Servo OFF
SWT A 0 0	Store 0 V as point 0
SWT A 1 20	Store 20 V as point 1
SWT A 2 40	Store 40 V as point 2
SWT A 3 60	Store 60 V as point 3
WTO A 4	Start output of the first four wave table points, one point is output per trigger signal (with default setting of parameter 12)

7. Internal Connections

7.1.1. Pin Assignments

The E-816 submodule plugs into two accurately placed sockets on a PCB, *j1* and *j2,* having the following pinout:

J1

PIN	
1	Analog Power Supply, +15V (5 mA)
2	Analog Power Supply, -15V (5 mA)
3	Analog GND
4	Digital Power Supply, +5V (60 mA)
5	Digital GND
6	RS232-RX
7	RS232-TX
8	RS232-RTS
9	RS232-CTS
10	Reserved

J2

PIN	
1	IN: Sensor Voltage
2	Analog GND
3	OUT: Target Voltage
4	IN: On-target signal
5	IN: Overflow signal
6	OUT: Servo ON/OFF control signal
7	IN: Piezo voltage / 100 (i.e. approx. 0 to 1 V)
8	I2C-SCL
9	I2C-SDA
10	Trigger input (starts wave table output or triggered motion; active HIGH LOW = 0 to 0.5 V; HIGH = 3.0 to 5.0 V, maximum 10 V)

J3

Service connector, leave unconnected.

7.1.2. Target (Analog) Output

Resolution	20 bit
Output Range	0 to 10 V
Update Rate	500 Hz
Precision	±1 mV w/o drift compensation

7.1.3. Sensor (Analog) Input

Resolution	24 bit
Input Range	0 to 10 V
Update Rate	6.4 kHz
Noise	2.5 ppm

7.1.4. Piezo Voltage (Analog) Input

Resolution piezo voltage	10 bit
Input Range	-0.25 V to 1.25 V
Noise	1 LSB rms

7.1.5. Trigger (Digital) Input

Signal level:	Active HIGH
LOW:	0 to 0.5 V
HIGH:	3.0 to 5.0 V (maximum 10 V)
max. freq.	400 Hz.
min. width:	200 µs

7.1.6. Other Digital I/O

TTL

8. Customer Service

For inquiries and orders, contact your PI sales engineer or send us an email (mailto:service@pi.de).

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

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

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

9. Old Equipment Disposal

In accordance with EU directive 2002 / 96 / EC (WEEE), as of 13 August 2005, electrical and electronic equipment may not be disposed of in the member states of the EU mixed with other wastes.

To meet the manufacturer's product responsibility with regard to this product, Physik Instrumente (PI) GmbH & Co. KG will ensure environmentally correct disposal of old PI equipment that was first put into circulation after 13 August 2005, free of charge.

If you have such old equipment from PI, you can send it to the following address postage-free:

Physik Instrumente (PI) GmbH & Co. KG

Auf der Römerstr. 1

76228 Karlsruhe, Germany