Piezo · Nano · Positioning

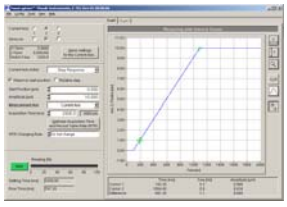
PZ 164E User Manual

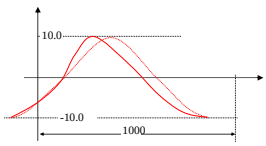
E-761 Digital Piezo Controller PCI Board

Release: 1.1.0 Date: 11 April 2008

This document describes the following product:

- E-761.3CD
  3-Channel Digital Piezo Controller PCI Board
- E-761.3CT
  3-Channel Digital Piezo Controller PCI Board, Trigger Output Bracket





Auf der Römerstr. 1 · 76228 Karlsruhe, Germany
Tel. +49-721-4846-0 · Fax: +49-721-4846-299
info@pi.ws · www.pi.ws

Declaration of Conformity

according to ISO / IEC Guide 22 and EN 45014

Manufacturer:	**Physik Instrumente (PI) GmbH & Co. KG**	CE
Manufacturer´s Address:	Auf der Römerstrasse 1 D-76228 Karlsruhe, Germany	

The manufacturer hereby declares that the product

Product Name: **3-Channel Digital Piezo Controller PCI Board**

Model Numbers: **E-761**

Product Options: **all model types**

complies with the following European directives:

2006/95/EC, Low voltage directive (LVD)
2004/108/EC, EMC-Directive

The applied standards certifying the conformity are listed below.

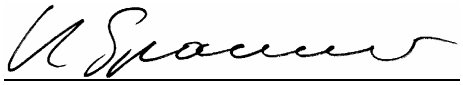
Electromagnetic Emission: EN 61000-6-3, EN 55011

Electromagnetic Immunity: EN 61000-6-1
EN 61000-4-2 (class C)

Safety (Low Voltage Directive) : EN 61010-1

Electrical equipment which is intended to be integrated in other electrical equipment, only conforms to the cited EMC Standards and normative documents, if the user ensures a compliant connection when implementing the total system. Possible necessary measures are installation of the component in a suitable shielded enclosure and usage of suitable connectors.

April 07, 2008
Karlsruhe, Germany



Dr. Karl Spanner
President

First printing 11 April 2008
Document Number PZ 164E, BRo, Release 1.1.0
E-761_User_PZ164E.doc

About this Document

Users of this Manual

This manual is designed to help the reader to install and operate the E-761 Digital Piezo Controller PCI Board. It assumes that the reader has a fundamental understanding of basic servo systems, as well as motion control concepts and applicable safety procedures.
The manual describes the physical specifications and dimensions of the E-761 Digital Piezo Controller PCI Board as well as the software and hardware installation procedures which are required to put the associated motion system into operation.
A listing of the terms which are used in the E-761 documentation is provided on p. 146 at the end of this manual.
This document is available as PDF file on the product CD. Updated releases are available via FTP or email: contact your Physik Instrumente sales engineer or write info@pi.ws.

Conventions

The notes used in this manual have the following meanings:

WARNING

Calls attention to a procedure, practice or condition which, if not correctly performed or adhered to, could result in injury or death.

DANGER

Indicates the presence of high voltage (> 50 V). Calls attention to a procedure, practice or condition which, if not correctly performed or adhered to, could result in injury or death.

CAUTION

Calls attention to a procedure, practice, or condition which, if not correctly performed or adhered to, could result in damage to equipment.

NOTE

Provides additional information or application hints.

Related Documents

The stage and the software tools which might be delivered with the E-761 Digital Piezo Controller PCI Board are described in their own manuals. All documents are available as PDF files. Updated releases are available via FTP or email: contact your Physik Instrumente sales engineer or write info@pi.ws.

NanoCapture_SM071E
E761_GCSLabVIEW_PZ157E
E-761_GCSDLL_PZ163E
GCSData_User_SM146E
PZTControl_PZ162E

Table of Contents

1 Introduction

1.1 Product Description

- For Piezo NanoPositioners with Capacitive Feedback Sensors
- 32-Bit Digital Filters
- 24-Bit DAC resolution
- Polynomial Linearization
- Coordinate Transformation for Parallel Kinematics / Parallel Metrology Systems
- AutoCalibrate Function for Easy Controller/Stage Interchangeability
- Extensive Software Support, i.e *NanoCapture*™ Software for System Optimization, PI General Command Set DLL and LabView drivers
- Fully Programmable Low-Pass Sensor Filter
- Fully Programmable Notch Filters on Servo-Loop Output
- Automatic Zero Adjust after Every Power-Up
- Built-In Wave Generators with Trigger Output Option, Ideal for Multi-Axis Scanning Applications
- Analog Control Input

E-761 digital piezo controllers provide advanced digital control technology in a cost-effective package. The sophisticated controllers in PCI plug-in board format allow for coordinated control of piezo nanopositioning systems ("PZT stages") with up to 3 logical axes. Based on powerful 32-bit DSPs, the E-761s include integrated, low-noise power amplifiers for piezo actuators. High-resolution capacitive position sensors are supported by special excitation/read-out electronics.

The E-761s digital processing includes coordinate transformation that supports advanced nanopositioning stages with parallel kinematics and parallel metrology, such as PI's P-560 or P-528 series stages. These systems are far superior to those based on a number of individual, independent single-axis modules.

Digital piezo controllers and PI nanopositioning stages with ID-chips can be used in any combination due to the controllers' AutoCalibration function. Individual stage data and optimized servo-control parameters are stored in the ID-chip, which is read automatically by the digital controller.

E-761-controlled nanopositioning systems provide outstanding linearity, achieved by digital polynomial linearization. The linearization can improve linearity to 0.001% over the full travel range.

Analog input can be handled as control input—the applied voltage may be connected with one ore more axes and will be interpreted as target value. It is also possible to use the analog input for triggering tasks.

1.2 Prescribed Use

Based on their design and realization, E-761 digital piezo controller boards are intended to drive capacitive loads, in the present case, piezoceramic actuators. E-761s must not be used for applications other than stated in this manual, especially not for driving ohmic (resistive) or inductive loads. E-761s can be operated in closed-loop mode using capacitive position sensors. Appropriate sensors are provided by PI and integrated in the mechanics according to the mechanics product specifications.

The E-761 may only be used for suitable applications according to the specifications of the board. The provided protection may be impaired if you do not follow the operating instructions of the manufacturer. Respect the safety instructions given in this User Manual.

The E-761 is intended to be installed in a PC with PCI-Bus with Windows operating system (2000, XP, Vista) or Linux (kernel 2.6, GTK 2.0, glibc 2.4). Two adjacent free slots are required to install an E-761 due to the board dimensions. Trigger output is provided via an additional trigger output bracket which requires another free slot. The operator is responsible for the intended installation of the E-761 and any additional equipment and for the training of the users of the board.

The verification of the technical specifications by the manufacturer does not imply the validation of complete applications. In fact the operator is responsible for the process validation and the appropriate releases.

E-761s conform to Measurement Category I (CAT I) and may not be used for Measurement Categories II, III or IV.

The E-761 is a laboratory apparatus in terms of DIN EN 61010. Safe operation is provided for under normal ambient conditions:

- 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

1.3 Safety Instructions

Carefully read also the documentation of the included software components and of the mechanics used.

Ignoring the warning notices in the instructions can cause bodily injury of the user or damage to equipment or loss of warranty.

Note that the E-761 does not contain any user serviceable parts.

CAUTION

Install and operate the E-761 digital piezo controller board and the trigger output bracket 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.

DANGER

The E-761 digital piezo controller board and the trigger output bracket do not contain any user-serviceable parts. Never re-assemble the board and the trigger output bracket. Hazardous voltage can be present on the E-761 board components.

DANGER—HAZARDOUS VOLTAGE

The amplifiers used by the E-761 are high-voltage amplifiers capable of generating high output currents. They may cause serious or even lethal injury if used improperly. Working with high-voltage amplifiers requires adequately trained operating personnel. Strictly observe the following:

→ Never touch any part that might be connected to the high-voltage output. Especially do not touch any pins and conductors on the board! Hold the E-761 board only by the edges.

→ Do not touch the pins of the Sub-D mix connector when the LED is turned on. The high voltage output of the board is active as long as the LED is turned on—normally until the PC is shut down. The output values depend on the last commanded targets (positions or voltages), even if you have quit the terminal or the program from which the targets were commanded—up to 120 V can be present on the Sub-D mix connector of the board.

→ When providing custom made adapters / connectors, take appropriate safety measures due to the high voltage output which will be present on the hardware!

→ When the E-761 is not used or is transported, put the protective cap on the Sub-D Mix connector of the board.

→ Do not operate the E-761 board when the PC case is open.

DANGER

Before opening the host PC, remove the line-voltage connection.

WARNING—BURNING HAZARD

Do not touch the aluminum heat sink of the E-761 amplifiers immediately after board operation (see Fig. 17 on p. 49). The temperature can be up to 60 °C.

CAUTION

The E-761 digital piezo controller PCI board and the trigger output bracket are ESD-sensitive (electrostatic discharge sensitive) devices. Observe all precautions against static charge buildup before handling these devices.

Avoid touching circuit components, pins and PCB traces. Discharge any static charge you may have on your body by briefly touching a conductive, grounded object before you touch any electronic assembly. Pose PCBs only on conductive surfaces, such as ESD-safe transport containers (envelopes, foam). Electronic subassemblies must always be kept and transported/shipped in conductive packaging.

CAUTION

Make sure that no conductive particles of any kind (metallic dust or shavings, broken pencil leads, loose screws) get on the board, on the PCB of the trigger bracket or into the PC.

Make sure that no electrical contact between the various add-on cards in the PC is possible. Do not remove the protective cover from the E-761 board (see Section 4.1 on p. 49 for board layout).

CAUTION

Screw down the E-761 board in the PC case during installation. Use of additional fastening of the board inside the PC case is recommended. In addition, removing the board temporarily when the PC is to be transported is recommended to avoid damage to the board, the PCI slot or to the PC motherboard.

CAUTION

Thermally stable systems have the best performance. For a thermally stable system, power on the PC which contains the E-761 at least one hour before you start working with it.

WARNING

All motion of the connected mechanical stages is software controlled, and software may fail.

CAUTION

The high voltage output of the E-761 will be deactivated if the internal voltage and / or the board temperature are out of range (LED is turned off). In this case communication with the board is still possible, but move commands will not be executed.

CAUTION

Most piezo actuators that can be connected to the E-761 can be destroyed by uncontrolled oscillation near the mechanical resonant frequency. If the piezo stage starts oscillating (humming noise):

→ In closed-loop operation, switch off the servo immediately. Adjust the servo parameters (notch filter frequency, servo-loop P-term (loop gain), servo-loop I-term (time constant), servo-loop slew rate; see "Servo-Controller Dynamic Calibration" on p. 62 and the NanoCapture™ manual for more information).

→ In open-loop operation, stop the axis motion immediately. Do not operate the piezo stage at its resonant frequency because the notch filters are not active in open-loop operation. You can measure the resonant frequency using NanoCapture™; see "Servo-Controller Dynamic Calibration" on p. 62 and the NanoCapture™ manual for more information.

1.4 Model Survey

The following models are available:

E-761.3CD 3-Channel Digital Piezo Controller PCI Board, basic version

Fig. 1: E-761 digital piezo controller board

E-761.3CT E-761.3CD with trigger output bracket which makes the digital output lines (see p. 147 and p. 146) and the syncronization lines (see p. 37) available outside of the PC. The trigger output works in conjunction with the wave generator usage (see p. 52).

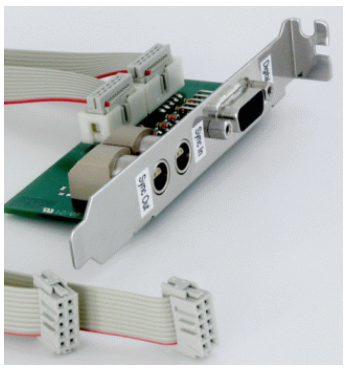


Fig. 2: Trigger output bracket, comes with E-761.3CT and can also be ordered separately as E-761.00T

1.5 Unpacking

CAUTION

The E-761 digital piezo controller PCI board and the trigger output bracket are ESD-sensitive (electrostatic discharge sensitive) devices. Observe all precautions against static charge buildup before handling these devices.

Avoid touching circuit components, pins and PCB traces. Discharge any static charge you may have on your body by briefly touching a conductive, grounded object before you touch any electronic assembly. Pose PCBs only on conductive surfaces, such as ESD-safe transport containers (envelopes, foam). Electronic subassemblies must always be kept and transported/shipped in conductive packaging.

Unpack the E-761 digital piezo controller PCI board with care. Compare the contents against the items covered by the contract and against the packing list.

The following components are always included:

- E-761 digital piezo controller PCI board
- E-761 CD containing host software (see "Software Description", p. 14) and manuals as PDF files
- E-761 User Manual (this document, PZ 164E)
- K040B0077 Analog input cable with Lemo 7-pin connector, open end
- 000011581 PC Y-power cable
- K010B0029 Ribbon cable with connectors 10-pin:
  1 x with E-761.3CD, for synchronization of multiple E-761 boards
  2 x with E-761.3CT, connect the E-761 PCI board to the trigger output bracket

E-761.3CT only:

- E761B0006 Trigger output bracket (with small PCB)
- K030B0296 External sync cable for synchronization of multiple E-761 boards

Inspect the contents for signs of damage. If parts are missing or you notice signs of damage, contact PI immediately.

Save all packing materials in case the product need be shipped again.

If controller and mechanics were ordered together, make sure a label with the serial number(s) of the mechanics is affixed to the protective cover of the controller board.

1.6 Additional Components

The trigger output bracket which is included in E-761.3CT can also be ordered separately as accessory, order number is E-761.00T. It makes the three digital output lines and the synchronization lines of the E-761 available outside of the PC, see p. 147 for more information.

Using the digital output lines requires firmware revision 2.0.1.0 or newer. Trigger output works in conjunction with the wave generator usage (see p. 52).

Contact your PI Sales Engineer or write info@pi.ws for trigger output brackets and firmware upgrades.

1.7 Software Description

The following host software with appropriate documentation is included on the E-761 CD.

NOTES

Operating system details:

→ "Windows" stands for 2000, XP and Vista

→ "Linux" stands for kernel 2.6, GTK 2.0, glibc 2.4

With Windows Vista, the E-761 host software must always be started with the "Run as administrator" option. To do this, click on the Start menu entry or the executable file of the appropriate program with the right mouse button and select the "Run as administrator" entry from the context menu.

Users who have already installed E-761 hardware driver and software: With release 2.0.0 or newer of the E-761 CD, a new hardware driver for the E-761 board is provided ("PI E761 Driver"). When using this driver, revision 4.0 or newer of the GCS library must be installed. You should run the installation procedure as described on p. 16 to make sure that all components are updated.

Software Tool	Operating System Require-ments	Short Description	Recommended for
GCS Library	Windows, Linux	Allows program access to the E-761 from languages like C++. The functions in the library are based on the PI General Command Set (GCS). Windows operating systems: E7XX_GCS_DLL; Linux operating systems: libpi_e7xx_gcs.so.*x.x.x* and libpi_e7xx_gcs-*x.x.x*.a where *x.x.x* gives the version of the library	Recommended for customers who want to use a library for their applications. Using the functions `E761_GetDirectPosition` and `E761_SetDirectTarget` of the E761 GCS library, direct access to the E-761 RAM is possible to get position / set control values every servo loop cycle (speed limitations may be given by your system). The dynamic version of the library is needed by the LabVIEW driver set and by NanoCapture™.
LabVIEW drivers	Windows, Linux	LabVIEW is a software tool (available separately from National Instruments) for data acquisition and process control. The E-712 LabVIEW software consists of a collection of virtual instrument (VI) drivers for the E-712 controller. This driver set supports the PI General Command Set (GCS).	Users who want to use LabVIEW for programming their application

Software Tool	Operating System Require-ments	Short Description	Recommended for
NanoCapture™	Windows, Beta version for Linux	A powerful graphical user interface which gives easy access to step response measurement or waveform motion, but also provides access to the advanced functionality of the controller. Users do not have to know any commands to work with NanoCapture™. Included are: Open- and closed-loop graphic motion display, Bode plots, dynamics tuning, controller configuration with all parameter types, AutoZero adjustment, wave editor, command entry	Users who want to test the equipment before or instead of programming an application and who want to optimize the servo-loop behavior and other controller parameters.
PZTControl™	Windows	A graphical user interface which is recommended for users who are already familiar with the GCS command set or for programmers which want to use the Windows DLL. PZTControl™ makes it possible to test the DLL functions in a convenient way. Included are: Command entry, control and display of positioning tasks, interactive access to important GCS commands	Users who want to test the equipment before or instead of programming an application and who want to learn how to use the commands.
E-761 Terminal	Windows, Linux	A simple terminal program	Users who want to send the commands of the PI General Command Set (GCS) directly.
Firmware Update Wizard	Windows	The Firmware Update Wizard guides you through the update of the firmware of your E-761 board.	Users who want to update the firmware.
E761 FW Update	Linux	In the /linux directory of the E-761 CD, enter './pi_e761_fw_update --help' for detailed information. The pi_e761_fw_update script guides you through the update of the firmware of your E-761 board.	Users who want to update the firmware.

2 Operation

If PI has been given sufficient information about your application and/or the E-761 is ordered together with the PZT stages, it will be calibrated at the factory. To put such an E-761 Digital Piezo Controller PCI Board into operation, carry out the following steps:

- Install device driver and software packages (see p. 16), E-761 board and—if available—the trigger output bracket in the PC (see p. 17)
- Connect the stage(s) (p. 20).
- Start the software (p. 22)
- Write a backup file for the controller parameters (p. 23)
- Check the servo-loop state (p. 23)
- Perform an AutoZero procedure (p. 24)
- Perform first Motions and Measurements (p. 26)

When you have passed the first steps with the controller, you might want to use some more advanced functionality:

- Customize the system (set controller parameters), short description on p. 32
- Use the Analog Input Socket (p. 33)
- Use the Wave Generator and its trigger output options (p. 35)

If you want to update or upgrade your system, you can

- Replace stages with ID-Chip (p. 35)
- Install multiple boards in one PC (p. 37)
- Update host software and/or firmware (p. 38)

2.1 Installation

2.1.1 Install the Software on Host PC

For a short description of the available host software see Section 1.7 on p. 14.

NOTE

Users who have already installed E-761 hardware driver and software:

→ With release 2.0.0 or newer of the E-761 CD, a new hardware driver for the E-761 board is provided ("PI E761 Driver"). When using this driver, revision 4.0 or newer of the GCS library must be installed. You should run the installation procedure as described below to make sure that all components are updated.

→ Windows 2000: During the update, the "Found New Hardware Wizard" may open with the "Driver Files Search Results" window, saying that it found also other drivers that are suitable for the device. In this case, check the "Install one of the other drivers" checkbox and click "Next >". In the "Driver File Found" window, select the "PI E761 Driver", Provider: PI Karlsruhe. Do **not** select "E761", Provider: Jungo LTD because this is the old driver.

Windows operating systems:

1. Be sure to log in with administrator privileges and insert the E-761 CD in the host PC.
2. If the Setup Wizard does not start automatically, start it from the root directory of the CD with the  icon.
   Note for Windows Vista: If the Setup Wizard starts automatically, cancel it. In the root directory of the CD, click on the setup.exe file with the right mouse button and select the "Run as administrator" entry from the context menu.
3. Follow the on-screen instructions. You can choose between “typical” and “custom” installation. Typical components are hardware driver, LabView drivers, GCS DLL, NanoCapture™, PZTControl™ and the Firmware Update Wizard. “Typical” is recommended.
4. Reboot the PC.

Linux operating systems:

1. Insert the E-761 CD in the host PC.
2. Open a terminal and go to the /linux directory on the E-761 CD.
3. Log in as superuser (root).
4. Start the install script with ./INSTALL
   Keep in mind the case sensitivity of Linux when typing the command.
5. Follow the on-screen instructions. You can choose the individual components to install.

If the installation fails, make sure you have installed the kernel header files for your kernel.

2.1.2 Install the E-761 Hardware in Host PC

CAUTION

Before installing the E-761 board and the trigger output bracket, discharge your body by briefly touching a conductive, grounded object (metallic water pipe, radiator or other metallic grounded object).

DANGER

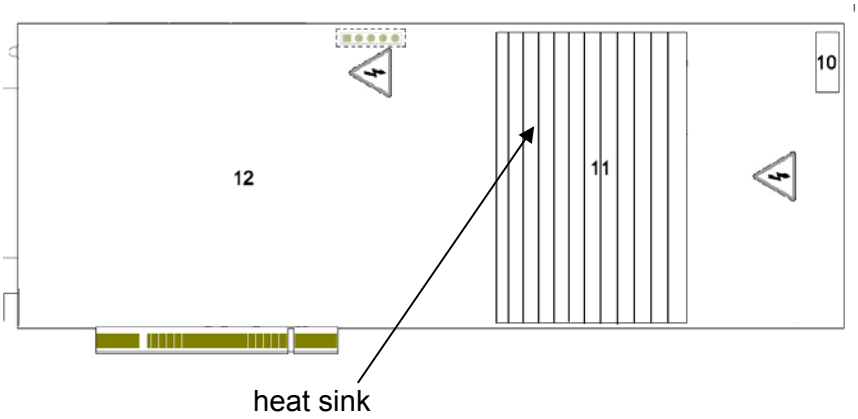
Before opening the host PC, remove the line-voltage connection.

DANGER—HAZARDOUS VOLTAGE

Never touch any part that might be connected to the high-voltage output. Especially do not touch any pins and conductors on the E-761 board! Hold the board only by the edges. Hazardous voltage can be present.

WARNING—BURNING HAZARD

Do not touch the aluminum heat sink of the E-761 amplifiers immediately after board operation (see figure below)! The temperature can be up to 60 °C.



CAUTION

Make sure that no electrical contact between the various add-on cards in the PC is possible. Do not remove the protective cover from the E-761 board (labeled with "12" in the figure above).

CAUTION

Screw down the E-761 board in the PC case during installation. Use of additional fastening of the board inside the PC case is recommended. In addition, removing the board temporarily when the PC is to be transported is recommended to avoid damage to the board, the PCI slot or to the PC motherboard.

CAUTION

When connecting the E-761 to the PC power supply as described in step 5 below, make sure that the connection is not shared with another device. Otherwise the supply power for the board may be insufficient, and the PC or the board may malfunction.

NOTES

Due to the board dimensions, 2 adjacent slots are required to install an E-761.

To avoid hardware conflicts, you should not use the slot directly under the graphics card. When problems occur, it might be necessary to install the board in a different slot.

Where possible, you can install an additional fan in the PC housing to handle the heat emission of the E-761. If the heat emission should cause problems with other cards in the PC, it might be also a solution to skip the next slot facing the E-761 heat sink or use it with a card so short that it does not extend into the heat sink area.

NOTES

Another free slot is required if you want to install a trigger output bracket (comes with E-761.3CT and can also be ordered separately as E-761.00T). The bracket should be installed as close as possible to the E-761—the ribbon cables which connect the bracket to the E-761 are 15 cm long.

1. Shut down the host PC and unplug the line cord. Then open the PC housing.
2. Remove the slot cover from the position(s) in the PC which you plan to use.
3. Install the E-761 board in a free PCI slot. Do not cant the board while inserting it, and afterwards check if it fits properly.
4. Screw down the board. Otherwise the board may creep out of its connector and can cause a short circuit.
5. Connect the E-761 to the PC internal power supply.
   If there is no free cable from the power supply or the cable is not long enough, use the Y-cable which comes with the board as an extension. Make sure that the connection is not shared with another device.
6. Only if a trigger output bracket is available (with E-761.3CT or separately as E-761.00T):
   a) If not already done, connect the two ribbon cables (K010B0029) to the connectors J6 and J7 on the PCB of the bracket.
   b) Install the trigger bracket in a free slot.
   c) Screw the bracket down.
   d) Connect the ribbon cables from the bracket to the corresponding sockets on the E-761: J6 to J6 and J7 to J7 (see Fig. 3 on p. 20).
7. Repeat steps 2 to 5 for every other E-761 board and step 6 for every trigger output bracket you want to install in the PC. See Section 2.12.2 on p. 37 for how to synchronize multiple boards.
8. Close the PC case.
9. Connect the stage(s) as described in Section 2.2.
10. Restart the host PC.
11. With Windows 2000 and XP operating systems only:
    When the "Found New Hardware Wizard" appears, do:
    - 1st window: click *Next*
    - 2nd window: choose "Display a list of the known drivers for this device so that I can choose a specific driver" and click *Next*

- 3rd window: in the *Models* list window, "E761 Device Driver" should be highlighted. Otherwise press *Have Disk* and select the CD-ROM drive with the E-761 CD and the path \Driver\Win*(your operating system)*\e761.inf, then click *Next*
- 4th window: click *Next*
- 5th window: click *Finish*

Note that with Windows 2000 and XP operating systems this procedure is required once for every board to register the device, even if the hardware driver has already been installed with the setup procedure in Section 2.1.

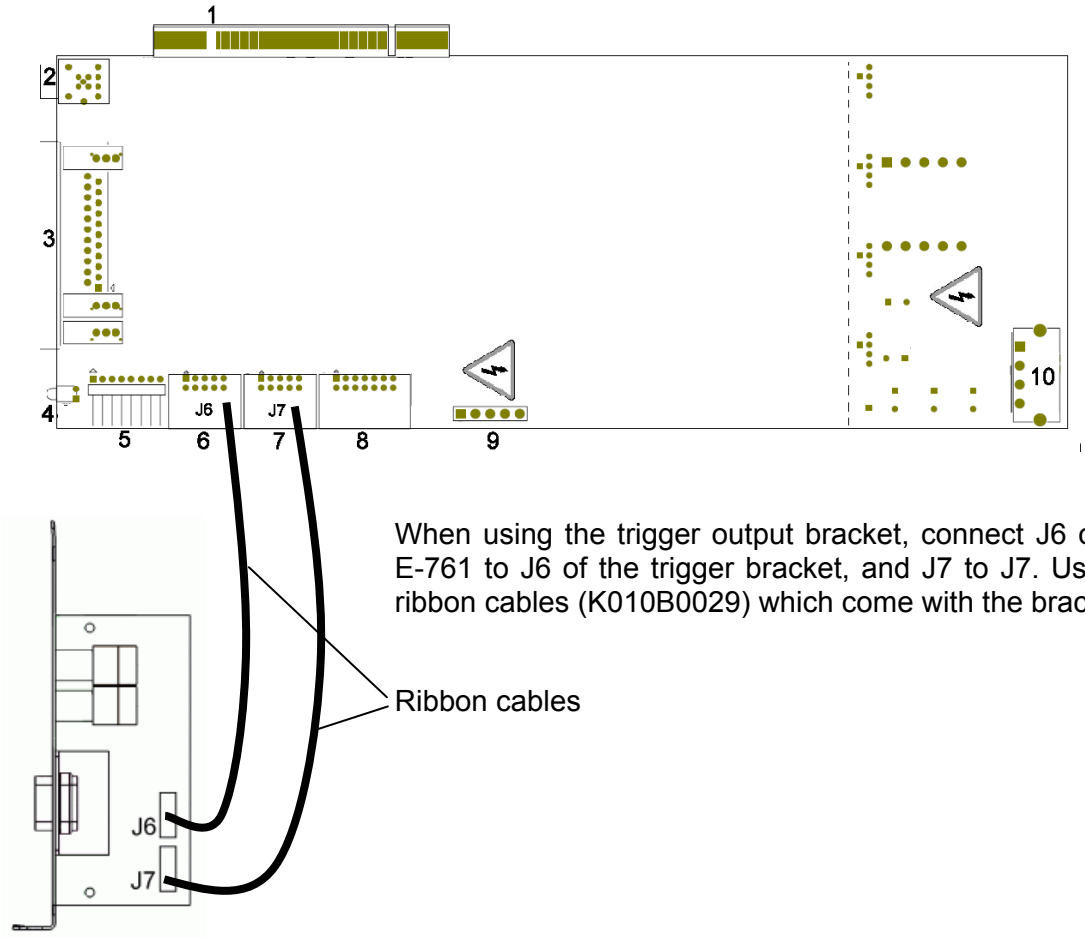


Fig. 3: E-761 and trigger output bracket interconnected with two ribbon cables

2.2 Connect the Stages

DANGER—HAZARDOUS VOLTAGE

The amplifiers used by the E-761 are high-voltage amplifiers capable of generating high output currents. They may cause serious or even lethal injury if used improperly. Working with high-voltage amplifiers requires adequately trained operating personnel. Strictly observe the following:

→ Do not touch the pins of the Sub-D mix connector when the LED is turned on. The high voltage output of the board is active as long as the LED is turned on—normally until the PC is shut down. The output values depend on the last commanded targets (positions or voltages), even if you have quit the terminal or

the program from which the targets were commanded—up to 120 V can be present on the Sub-D mix connector of the board.

→ When providing custom made adapters / connectors, take appropriate safety measures due to the high voltage output which will be present on the hardware!

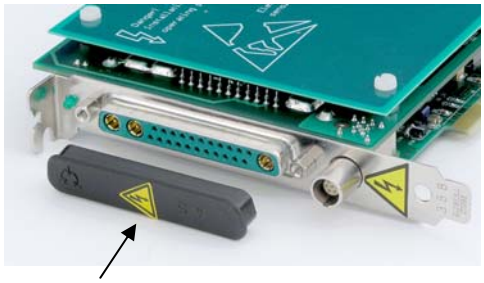
→ When the E-761 is not used or is transported, put the protective cap on the Sub-D Mix connector of the board.

NOTES

Before you connect a stage to your controller for the first time, you should create a parameter backup file as described in Section 2.4 on p. 23.

Connect the stages before you start the host PC to make sure that the content of their ID-chip(s) is written to the E-761—otherwise you will have to reboot the controller using the appropriate commands or functions in the host software before you start working with the system.

Remove the protective cap from the Sub D Mix connector of the board and connect the stage(s).



Protective cap

If the connector on the board does not match those on the PZT stages, an appropriate adapter must be used.

A label on the protective cover of the board indicates which PZT stage was assigned to which controller (axis) during calibration (see figure at right). Be sure to respect this assignment when connecting the stage(s) to the controller.

Calibration label

When you are using a PZT stage with ID-chip together with the E-761, the PZT stage can be easily exchanged due to the functionality of the ID-chip (for details see Section 2.11 on p. 35).

2.3 Start Software on Host Computer

NOTE

With Windows Vista, the E-761 host software must always be started with the "Run as administrator" option. To do this, click on the Start menu entry or the executable file of the appropriate program with the right mouse button and select the "Run as administrator" entry from the context menu.

Starting the host software and establishing the communication is exemplified using the *NanoCapture*™ software package because the E-761 is most conveniently controlled with *NanoCapture*™. To work with *NanoCapture*™, you do not need any knowledge of the E-761 commands.

1. Start *NanoCapture*™ from the Start menu choosing the NanoCapture entry (with Windows Vista: see the Note on p. 22).
2. When *NanoCapture*™ is started for the first time, the *Device Connection* window will be displayed (Fig. 4). (Thereafter, the software can initialize the connection automatically.)
3. Select the *E-761* entry in the *PI Controller* pane.
4. Select the board ID.
5. Press the *OK – Connect* button. When the connection is successfully initialized, the configured axes are automatically got from the controller, and *NanoCapture*™ starts to upload the current parameter settings from the controller.
   When the upload procedure is finished, the *Device Connection* window will be closed automatically, and you can start working in the *NanoCapture*™ main window.

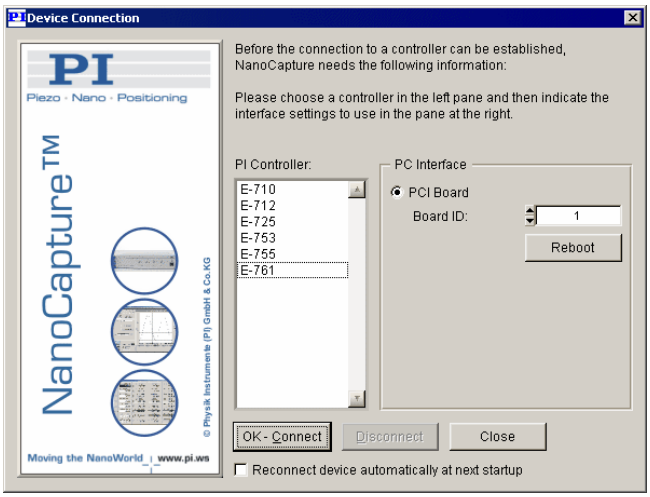


Fig. 4: NanoCapture™ Device Connection *window before connection*

NOTES

If the connection is not successful, press the *Reboot* button in the *Device Connection* window. Additional information can be found in “Troubleshooting” on p. 140.

The configured axes can be changed using the *Config* → *Connected Axes Selection* menu sequence.

When you connect stages to the board after the PC was powered on, the ID-chips of the stages are not read by the controller. To read the ID-chip data, press the *Reboot* button in the *Device Connection* window of *NanoCapture*™ or type the RBT command (p. 90) in the terminal.

2.4 Creating Backup File for Controller Parameters

It is strongly recommended to save the parameter values of the E-761 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-761, use the *Device Parameter Configuration* window of NanoCapture™ (see figure below). See "Install the Software on Host PC" on p. 16 for how to install the program.

Multiple parameter files can be saved, so that you should create the first parameter file before you connect any stage to your controller, and then create further backup files (with different names) whenever you have optimized parameter values or adapted your controller to a certain stage.

Proceed as follows to create a parameter file:

1. Start NanoCapture™ on the host PC and establish a connection to the E-761 as described in Section 2.3.
2. In the NanoCapture™ main window, open the *Device Parameter Configuration* window using the *Config → Device Parameter Configuration* menu sequence.
3. In the *Device Parameter Configuration* window, save the controller parameters to a file. Use the *Save* or *Save As* buttons in the top left-hand corner of the window, or use the *File → Save Edit Values* or *File → Save Edit Values As* menu sequences. In fact, the values from the *Edit Mask* column are saved.

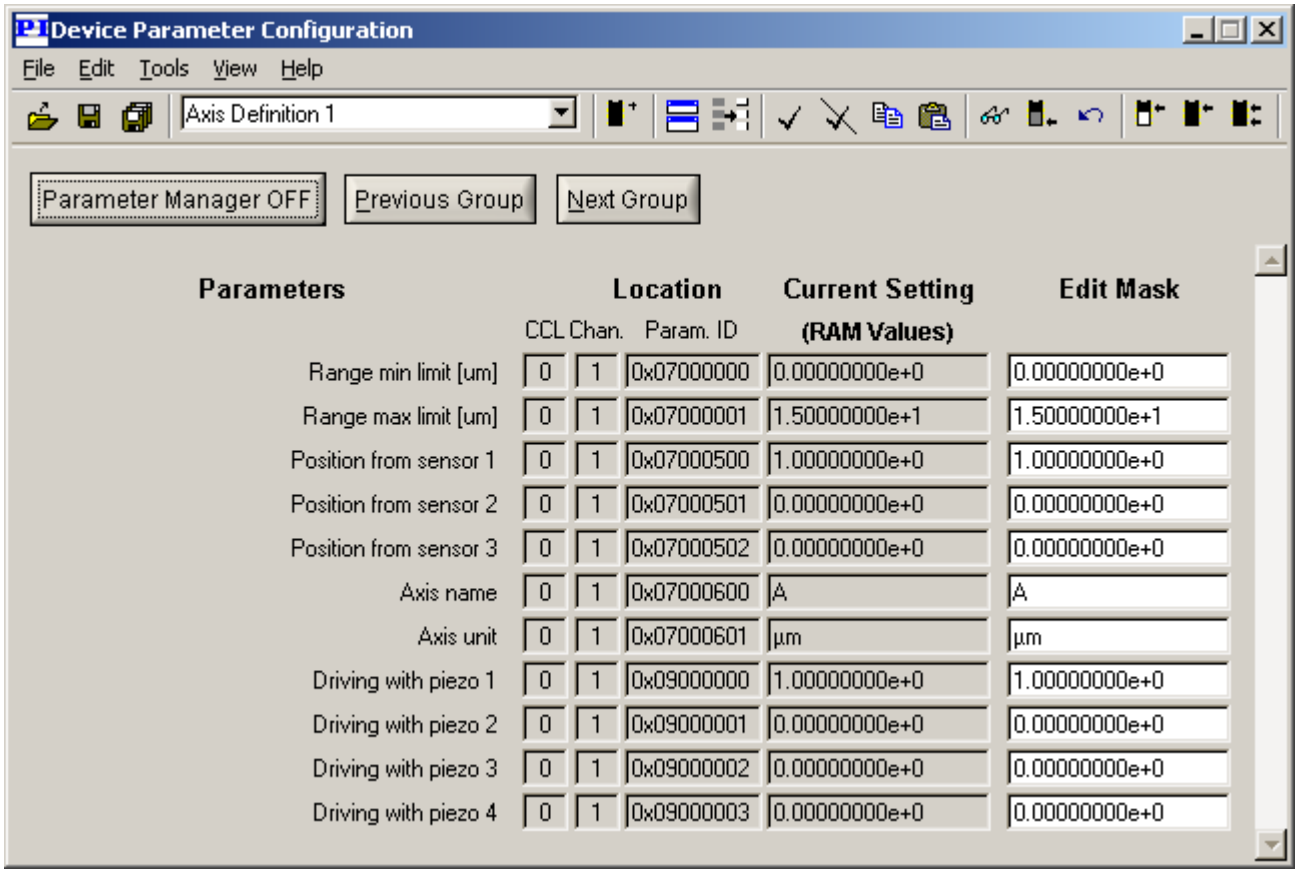


Figure 1: Device Parameter Configuration window for E-761, with the Axis Definition 1 *parameter group displayed*

In the *Device Parameter Configuration* window of NanoCapture™, proceed as follows to load back the content of a parameter file (with the extension .pam):

1. Use the *File → Load* and select menu sequence. The loaded parameter values are written in the corresponding *Edit Mask* fields.
2. You can copy the loaded parameter values from the *Edit Mask* fields to the RAM (volatile memory) or non-volatile memory of your controller using the three *Write selected edit values…* buttons in the top right-hand corner of the window.

See "Parameter Overview" on p. 134 and the NanoCapture™ Manual for further information.

2.5 Check the Servo-Loop State

NOTE

The servo should be switched on only when the servo parameters are set correctly. See "Frequency Response Measurement with NanoCapture™", p. 27 and "Servo-Controller Dynamic Calibration", p. 62 for more information.

Normally the servo loops for all logical axes are OFF when the controller is started for the first time (open-loop operation).

In *NanoCapture*™, you can use the *Current Axis* pane of the *NanoCapture*™ main window (Fig. 5) to set the servo-loop state. The radio buttons in this pane permit selection of the axis that will be used by default for certain actions and measurements that *NanoCapture*™ can perform. All axes supported by the controller (up to 3) will have a radio button shown. Under each axis' radio button is a check box that can be used to turn that axis' servo-loop on (closed-loop operation) or off.

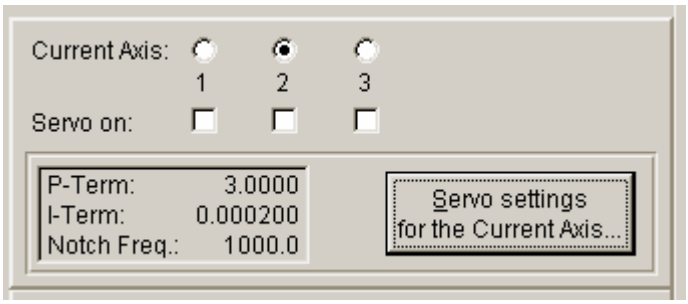


Fig. 5: Axis-select pane of the NanoCapture™ window

The corresponding command for selection of closed-loop or open-loop operation is SVO (p. 98).

See "Operating Modes" on p. 44 for more information.

2.6 Perform AutoZero

Both the range of axis position values (from the sensors, after coordinate transformation) and the range of axis motion control signals are limited. If there is an offset between the input and output ranges caused by a mechanical drift of the piezo stage, then the usable range will be reduced. Such an offset can be compensated by the AutoZero function, as explained with the following example:

Take a one-dimensional piezo stage and its sensor with a position range from 0 µm to 200 µm. The controller has an output voltage range from –10 V to +110 V. The piezo stage has a sensitivity of about 2 µm/V. That means that the piezo stage would be displaced by 200 µm if the output voltage is 100 V. In this case the

controller has a voltage reserve of about 10 V in each direction. Ideally the position of the piezo stage should be 0 µm if the servo loop is switched off and the output voltage of the controller is set to 0 V. In fact there will be a position offset because of some nonideal properties of the piezo. If the position offset is not larger than 10% of the position range (20 µm) this offset can be compensated by the controller when the servo-loop is activated, using the 10% voltage reserve. If the position offset is greater than 20 µm, it will no longer be possible to move the stage to the zero position. This situation can be corrected by using the AutoZero procedure

The position offset of the piezo stage is compensated by adding an offset constant value to the sensor value. The mechanical zero position of the piezo stage will change when AutoZero is executed. But after AutoZero the full travel range can be used.

NOTE

During the AutoZero procedure, the axis will move, and the motion can cover the whole travel range.

2.6.1 Decide Whether Running AutoZero Is Necessary

When the piezo stage is first integrated into the application environment, AutoZero must be run. Afterwards, AutoZero should only be executed in the following cases:

- The stage is subjected to temperature changes.
- The load applied to the stage has changed (note that the effect of load changes depends on the stiffness of the stage).
- The integration of the stage has changed (e.g. the orientation).

Especially if absolute moves are needed, AutoZero should not be executed during normal operation because AutoZero changes the mechanical zero position of the piezo stage.

NOTE

AutoZero is not effective on non-linear axes.

2.6.2 If AutoZero Is Necessary

With *NanoCapture*™, proceed as follows to carry out the AutoZero procedure:

1. In the main window, use the *Tools* → *AutoZero* menu item to open the *AutoZero* window.
2. In the *AutoZero* window, select the axis for which AutoZero is to be run, and select the voltage range which shall be used for this axis.
3. Start the AutoZero procedure by pressing the *Perform AutoZero* button. The process takes about 5 seconds, and the success is shown by a display.

You can also use the ATZ command (see p. 76) to start the AutoZero procedure.

NOTES

Be aware that the result of the AutoZero procedure (new offset value) is automatically written to non-volatile memory (EEPROM).

For stages with ID-chip the option "Read ID-Chip always" must be disabled by default to make the AutoZero result (new offset value) available in the future. See p. 35 for details.

2.7 First Motions and Measurements

For general information on how to command axis motion, see Section 3.4 on p. 44.

CAUTION

Thermally stable systems have the best performance. For a thermally stable system, power on the PC which contains the E-761 at least one hour before you start working with it.

CAUTION

The high voltage output of the E-761 will be deactivated if the internal voltage and / or the board temperature are out of range (LED is turned off). In this case communication with the board is still possible, but move commands will not be executed.

CAUTION

The servo should be switched on only when the servo parameters are set correctly. See "Frequency Response Measurement with NanoCapture™", p. 27 and "Servo-Controller Dynamic Calibration", p. 62 for more information.

CAUTION

Most piezo actuators that can be connected to the E-761 can be destroyed by uncontrolled oscillation near the mechanical resonant frequency. If the piezo stage starts oscillating (humming noise):

→ In closed-loop operation, switch off the servo immediately. Adjust the servo parameters (notch filter frequency, servo-loop P-term (loop gain), servo-loop I-term (time constant), servo-loop slew rate; see "Servo-Controller Dynamic Calibration" on p. 62 and the NanoCapture™ manual for more information).

→ In open-loop operation, stop the axis motion immediately. Do not operate the piezo stage at its resonant frequency because the notch filters are not active in open-loop operation. You can measure the resonant frequency using NanoCapture™; see "Servo-Controller Dynamic Calibration" on p. 62 and the NanoCapture™ manual for more information.

2.7.1 Perform Test Moves with NanoCapture™

Make some test moves with the individual axes, as shown for axis A below (in this example, axis 1 was renamed as A using the SAI command, p. 92):

In the NanoCapture™ main window, make sure that axis A is activated (*Current Axis* radio button must be selected) and in open-loop operation (*Servo on* must be unchecked).

To make first test moves, open the *Visual Panel* window (see figure below) using the *View* ⇒ *Visual Panel* menu entry. In the *Axis Command* field for axis A, enter the control value 0 and press Enter on your keyboard. Then enter the value 10 and press Enter. Increment the *Axis Command* control value this way by steps of 10, up to the upper travel range limit of the axis, and then reduce it in an analogous manner back to zero. In doing so, observe the position display for axis A (in the *Axis Position* field) and the current output voltage(s) for the piezo actuator(s) in the stage (in the *Piezo* fields; depending on the connected stage type, an axis may be driven by more than one piezo actuator). The values should follow the commanded control values: The axis position should always correspond approximately to the commanded value, and the output voltage(s) should become noticeably different from 0 V and then go back to zero again during the procedure (with the E-761, the output voltage range is -20 to +120 V; the output voltage(s) corresponding to a given control value depend(s) on the nominal travel range of the axis).

You can also increment or reduce the control value in the *Axis Command* field by steps of 1 using the arrow controls beside the field.

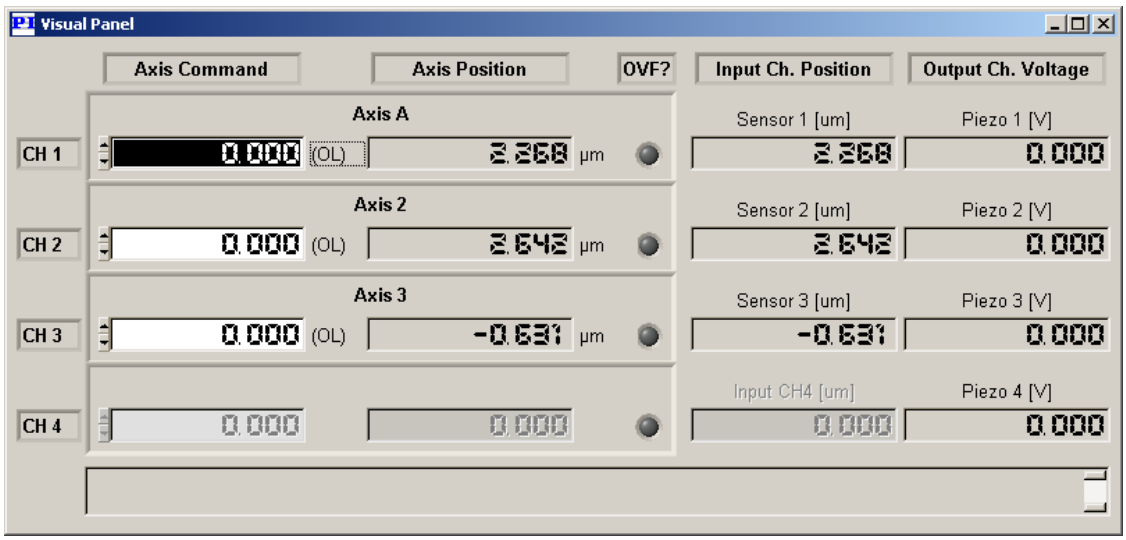


Fig. 6: Visual Panel *window of NanoCapture™, all axes are in open-loop operation*

2.7.2 Perform Frequency Response Measurement with NanoCapture™

Make open-loop frequency response measurements in the NanoCapture™ main window to determine the resonant frequencies of the axes. Proceed as follows for each axis:

Preparation (shown for axis A, see also Fig. 7):

Make sure that the desired axis is activated (corresponding *Current Axis* radio button must be selected) and in open-loop operation (*Servo on* must be unchecked). Select *Frequency Resp. (Impulse)* for *Current Axis Action* and *Current Axis* for *Measurement Axis*. The default entries for *Start Value* and *Amplitude* can be used, but make sure that the values are inside the travel range of the axis. By default, they correspond to axis positions in µm or µrad.

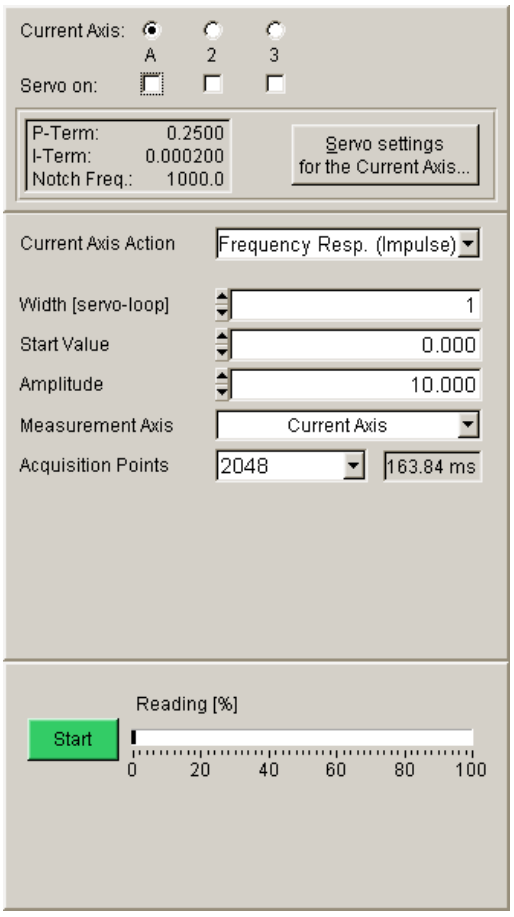


Fig. 7: Settings for open-loop frequeny response measurement with NanoCapture™, shown for axis A

Perform the impulse motion for the axis:

Press the green *Start* button to start performing the impulse, recording the response and analyzing and displaying it in the form of a Bode frequency response diagram (internally, NanoCapture™ sends the required IMP (p. 84) and DRR? (p. 80) commands to the E-761). See Fig. 8 for a typical result (measurement for axis A).

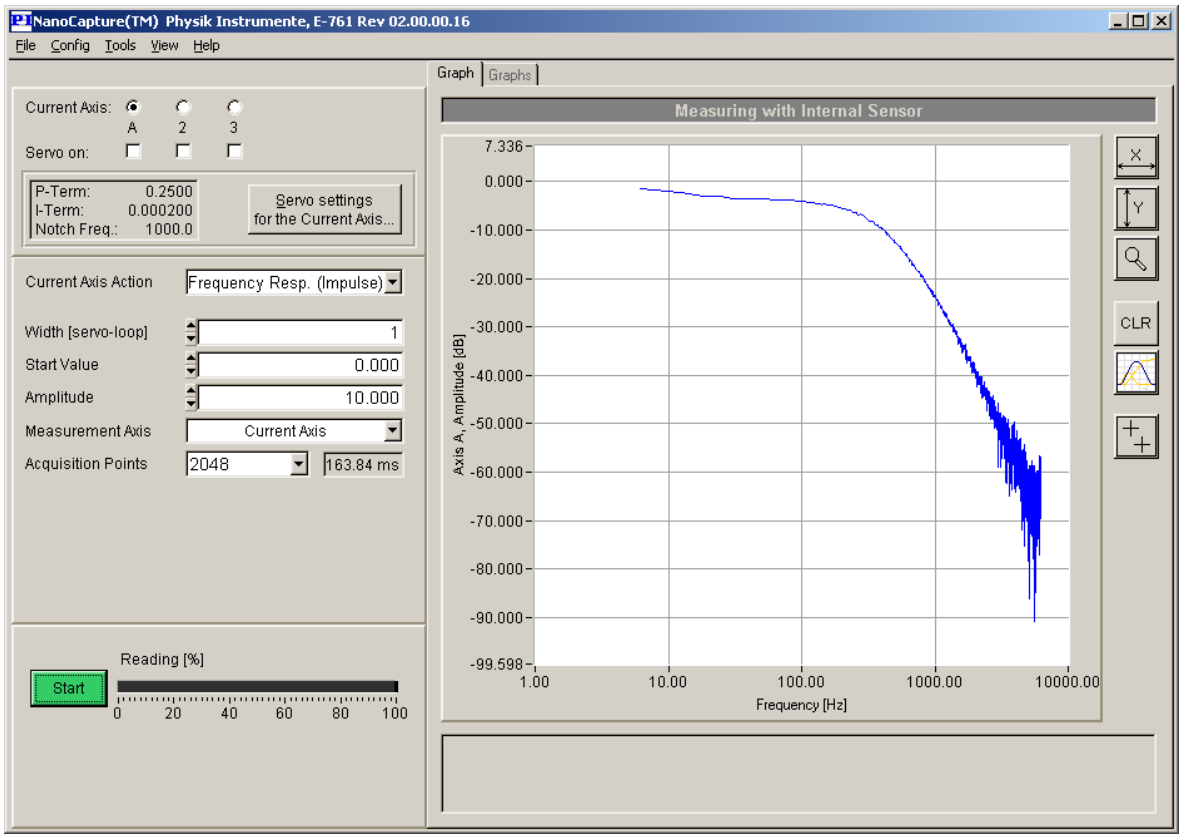


Fig. 8: Bode frequency response diagram for axis A

In the Bode frequency response diagram, you can identify the resonance peak (in the figure above no significant peak is visible).

If there are resonances which are intolerable in your application, adjust the notch filter settings for the axis before you switch to closed-loop operation for the first time (servo on). Furthermore, it might be necessary to readjust the preset servo parameters for the axis. See "Servo-Controller Dynamic Calibration" on p. 62 for more information.

For more information see the *NanoCapture*™ manual.

2.7.3 Working with GCS Commands

If you want to enter GCS commands directly via the command entry facilities of *NanoCapture*™ or *PZTControl*™ or in the *E-761 Terminal*, you can use the following examples for your first steps. See the "GCS Commands" Section beginning on p. 69 for the syntax and detailed command descriptions; see the *NanoCapture*™ manual and the *PZTControl*™ manual for how to enter commands. For axis and channel definitions see p. 40.

1) How to rename an axis:

The default identifiers of the axes are 1 to 3, but they can be changed using the SAI command (p. 92). The configuration made with SAI is automatically saved to non-volatile memory (EEPROM), and the new identifier must now be used to command the axis.

NOTE

Do not mistake the axis identifiers set with SAI with the Axis name parameter (ID 0x07000600) which is only used in the graphical user interface of NanoCapture™.

Action	Content of Program Window	Comment
Send:	SAI? ALL	Check the axis names of all available axes
Response:	1 2 3	
Send:	SAI 1 X	Rename axis "1" to "X"; note that names are limited to one character, case insensitive

2) How to command an axis in open-loop operation:

Perform open-loop moves and check the position values for the individual axes:

Action	Content of Program Window	Comment
Send:	SVO? X	Check servo-control state.
Response:	X=0	Axis X is in open-loop operation, i.e. there is no correction of drift or other effects.
Send:	SVA X 0	Send this command to move axis X to an absolute open-loop value of 0. Note that with the factory default settings of the system, this value should correspond to 0 µm (approximately).
Send:	POS? X	Ask the current position of axis X
Response:	X=+0001.3694	The current position value of axis X 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.
Send:	SVR 1 10	With this command, axis X moves relative by an open-loop value of 10 (corresponds approximately to 10 µm with the factory default system settings).
Send:	POS?	Ask the current position of axis X. The current position value of axis X should now be approximately 10 µm.

3) How to command a position to an axis in closed-loop operation:

If the notchfilter frequency and the servo parameters are adjusted properly (see "Servo-Controller Dynamic Calibration", p. 62) or if the system was calibrated at the factory with a load equal to the current one, you can perform closed-loop moves, in the example shown for axis X:

Action	Content of Program Window	Comment
Send:	SVO X 1	Servo should be ON for axis X
Send:	SPA? X 0x07000000 X 0x07000001	Check the travel range of axis X
Response:	X 0X07000000=0.00000000e+0 X 0X07000001=1.00000000e+2	Axis X can move from 0 to 100 µm.
Send:	NLM? X	Check the lower position soft-limit of axis X
Response:	X=0	
Send:	PLM? X	Check the upper position soft-limit of axis X
Response:	X=100.0	The position soft-limit range is also 0 to 100 µm. (The settings can be changed with NLM and PLM within the moving range limits.)
Send:	VCO X 0	Set velocity control OFF for axis X
Send:	MOV X 10	Move axis X to 10 µm.
Send:	MVR X 1.0	Increase the position of axis X by 1.0 µm
Send:	POS? X	Ask the current position of axis X
Response:	X=11.0	

4) How to set a velocity for an axis in closed-loop operation:

Action	Content of Program Window	Comment
Send:	SPA? X 0x07000200	Check the slew-rate of axis X
Response:	X 0x07000200=10.0	This means that the velocity of axis X is 10 µm/ms (change the parameter setting e.g. with SPA or with VEL)
Send:	VEL X 0.01	Set the velocity of axis X to 0.01 µm/ms = 10 µm/s
Send:	VCO X 1	Set velocity control ON for axis X
Send:	MOV X 0	Move axis X to 0 µm.

2.8 Customize the System

CAUTION

Incorrect parameter values may lead to improper operation or damage to your hardware. Be careful when changing parameters.

It is strongly recommended to save the parameter values of the E-761 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-761, use the *Device Parameter Configuration* window of NanoCapture™. See "Creating Backup File for Controller Parameters" on p. 23 for more information.

NOTES

For stages with ID-chip the option "Read ID-Chip always" must be disabled by default to make new parameter settings available in the future. See p. 35 for details.

Values stored in non-volatile memory are power-on defaults, so that the system can be used in the desired way immediately. Note that PI records the data files of every E-761 controller calibrated at the factory for easy restoration of original settings should that ever be necessary.

To adapt the E-761 to your application, you can modify parameter values—either for the whole system, for the individual axes or for the individual sensor channels and PZT amplifier channels (for the interdependence between axis and channels see Section 3, p. 40). The parameters and parameters types available depend on the controller firmware. Note that many parameters are preset and can not be changed—it is only possible to change the parameters which are listed in Section 8 on p. 134 (can be queried with HPA?, p. 84).

Parameters can be changed temporarily or in non-volatile memory using the *Device Parameter Configuration* window of NanoCapture™ (*Config → Device Parameter Configuration* menu sequence). Alternatively you can enter appropriate GCS commands in the command terminal (see SPA p. 94, SEP p. 93, WPA p. 118), but using the *Device Parameter Configuration* window is much more comfortable because you do not have to deal with any parameter numbers.

NOTES

The parameters which can be changed have the value 0 in the CCL column of the *Device Parameter Configuration* window.

Some parameters are read-only even though their CCL value is 0—see the list in Section 8 on p. 134 for more information.

See "Parameter Overview" on p. 134 for detailed information and the NanoCapture™ manual for how to edit, save or reset parameter values.

2.9 Using the Analog Input

The analog input line is provided for commanding the E-761. To do this, connect an appropriate signal source to the analog input socket (for pinout see p. 145). An appropriate cable with LEMO connector and open end comes with the E-761 (order# K040B0077). The analog input range is -10 V to +10 V.

The analog input can be used in one of the following ways:

- "Direct" axis control
- Control value input as an "external wave generator"
- As trigger input for the internal wave generators

See also "Control Value Generation" on p. 44.

NOTES

For the assignment of axes to wave generators see p. 52. The WGO command (p. 114) which starts the wave generator requires wave generator identifiers and not axis identifiers.

The wave generator output and hence the usage of the analog input as "external wave generator" or wave generator trigger is also stopped by WGO.

When using the analog input for control value generation ("direct" axis control and "external wave generator"), keep in mind that the applicable frequency depends on the available amplifier power. Example: with a capacitive load of 6.6 µF, the frequency should not exceed 15 Hz if three amplifiers are involved in the motion or 50 Hz if only one amplifier is involved (motion covers the whole travel range).

To read the analog input value with the current gain and the offset applied, use the TAV? command (p. 100). To determine the "digital" state of the analog input line, use the DIO? command (p. 80).

The analog input is not included in the value of the "Number of input channels" parameter (ID 0x0E000B00). This parameter refers only to the sensor channels of the E-761.

2.9.1 Preparation: Scaling the Analog Input

When the analog input is used for control value generation ("direct" axis control and "external wave generator") it might be necessary to scale the analog input signal to cover the full travel range of your piezo stage with the output of your signal source. Scaling can be done using SPA (p. 94) with the gain and the offset parameters:

Gain: parameter ID 0x04000001, default value is 1.0
Offset: parameter ID 0x04000101, default value is 0

Since the analog input line is internally handled as the 4th channel of the A/D converter, the ItemID for the appropriate SPA commands must be 4.

Example:

If your signal source has an output of 0 to 5 V DC, gain and offset are to be set as follows:

SPA 4 0x04000001 4
SPA 4 0x04000101 -10

Other gain and offset combinations are:

signal source output: 0 to 10 V DC → gain = 2.0, offset = -10
signal source output: -5 to +5 V DC → gain = 2.0, offset = 0
signal source output: -10 to +10 V DC → gain = 1.0, offset = 0

After parameters were set with SPA, they can be written to the EEPROM using WPA (p. 118) to become the power-up defaults.

2.9.2 Analog Input for "Direct" Axis Control

The scaled analog input signal can be used to control the individual axes "directly" (in open-loop or closed-loop operation). For that purpose, for each axis the signal value is multiplied by a factor ("Aux-Input to target factor", parameter ID 0x06000902). The result is added to the current control value (which can result from a move command, from the wave generator output or from a value given by the `E761_SetDirectTarget` library function, see p. 44 for details). This makes possible, for example, commanding a target by a combination of move command and external signal.

By default, the value of the "Aux-Input to target factor" is 0 for each axis which means that "direct" control via the analog input is disabled. You can set the factor to a value different from 0 with SPA to enable "direct" control.

Example: If you want the analog input signal to participate in the control value generation for axis 2, enter the following command
SPA 2 0x06000902 3
The scaled signal value will then be multiplied by 3, and the result will be added to the current control value of axis 2.
To "amplify" the influence of the analog input signal, increase the factor, e.g. send
SPA 2 0x06000902 10
so that the scaled input will be multiplied by 10.
To disable "direct" control for axis 2, send
SPA 2 0x06000902 0

After parameters were set with SPA, they can be written to the EEPROM using WPA (p. 118) to become the power-up defaults.

When the analog input is being used for "direct" axis control and the servo is switched off (SVO, p. 98), the axis motion will continue in open-loop mode.

2.9.3 Analog Input as an "External Wave Generator"

The scaled analog input signal can be used to generate the control values for the axes as an "external wave generator" (in open-loop or closed-loop operation). To activate the "external wave generator" use the WGO command with the start mode 0x400 (hex format) or 1024 (decimal format).

Example: If you want to command axis 1 and axis 2 by the analog input signal, enter the following command
WGO 1 0x400 2 0x400
and modify the signal according to your application.

2.9.4 Analog Input as Trigger Signal for the Internal Wave Generators

Using the (unscaled) analog input as trigger signal for the internal wave generators means that the signal is interpreted as follows:
If the voltage on the analog input is < 0.8 V, the signal is interpreted as LOW, if the voltage is ≥ 2.4 V, the signal is interpreted as HIGH.
Use the WGO command with the start modes 0x2 or 0x4 (hex format) to activate the analog input for triggering tasks.
Example: If you want to use the analog input signal to start and synchronize wave generator 1, enter the following command
WGO 1 0x4
The wave generator will then output one point of the waveform each time the analog input signal is interpreted as HIGH.

2.10 Using the Wave Generator

The E-761 has four wave tables for creating and (temporarily) storing arbitrary waveforms. These waveforms can then be output by up to four wave generators. The assignment of wave tables, wave generators and axes to each other is fixed.

Waves are created using the WAV command (p. 108) and output using the WGO command (p. 114). The different E-761 software interfaces also support use of the wave generator. Waveforms can be defined, stored and displayed in and by the software in a more user-friendly way than in a terminal using WAV and WGO. If using the wave generator with the GCS library, *PZTControl™, NanoCapture™* or with LabView, read the descriptions in the associated software manual first.

The analog input can be used as an "external wave generator" or as trigger signal for the output of the internal wave generators. See Section 2.9 on p. 33 for more information.

Digital output synchronized with the wave generator output and hence with the axis motion is possible via certain WGO start options. To make the digital output available outside of the PC, a trigger output bracket is required (included with E-761.3CT models; can be ordered separately as E-761.00T). See p. 147 for pinout of the Digital Out socket.

For the interaction of the wave generator with the other control sources the E-761 provides for axis motion, see "How to Command Axis Motion" on p. 44.

Starting the wave generator starts also recording of the actual position of the axes. As long as the wave generator is running, recording can be restarted with WGR (p. 117). You can read the recorded data with DRR? (p. 80).

For a detailed description of the wave generator usage see Section 5 on p. 52.

2.11 ID-Chip Support / Stage Replacment

The PZT stage which is connected to the E-761 may contain an ID-chip (located in the stage connector). The following data is stored in the ID-chip:

- Stage type
- Serial number of the stage
- Calibration data
- Servo-control data (dynamic tuning, load dependent)

The parameters which are stored on the ID-chips are marked in the table in Section 8 on p. 134, but can differ slightly between the different mechanics provided by PI.

When a stage with ID-chip is connected to the controller for the first time, the stage parameters from the ID-chip will be written to the EEPROM on PC power-on (= controller power-on). Afterwards, the stage parameters will be written on power-on only when the "Read ID-Chip always" option is enabled. By default, this option is disabled to maintain optimized parameter settings on the controller.

NOTE

When you connect stages after the PC was powered on, the ID-chips of the stages are not read by the controller. To read the ID-chip data, the controller must be rebooted using the RBT command or the appropriate functions of the host software.

The PZT stage can be easily exchanged due to the functionality of the ID-chip.

Consider the following when replacing stages with ID-chip:

- Normally, when you replace a piezo stage with a new unit and you are using standard factory settings for all parameters, you do not have to adjust anything. The ID-chip holds all information needed. At power-on of the system, the firmware reads the stage type and serial number stored in the ID-chip and compares this data to the data from the last connected stage, stored in the controller:
  - If there is a new stage type connected to the controller, all the data in the ID-chip will be read and the corresponding parameters in the controller overwritten.
  - If there is a stage of the same type but with a different serial number connected to the controller, the calibration data from the ID-chip will be read and only the corresponding parameters overwritten. The servo-control data will not be read, so those parameters will remain unchanged in the controller.

  If you have optimized some parameters for your application, PI recommends that you repeat your optimization routine with any new stage, because there are variations, e.g. in the stiffness and natural frequency, of piezo stages.
- If you send your stage to PI for e.g. upgrade or repair, the calibration data stored in the ID-chip might be changed in the process. When you re-connect this stage to the controller to which it was connected before, the firmware will detect that type and serial number are unchanged and **will not** read any more ID-chip data.

  To force the controller to read the complete data of the ID-chip when the controller is switched on, you have to enable the "Read ID-Chip allways" option (parameter ID 0X0F000000). This has to be done for each axis separately:

  1. In *NanoCapture*™, open the *Device Parameter Configuration* window (*Config → Device Parameter Configuration*) and select the *System Mechanics* groups where you can enable the option. When this is done for all axes, press the "Write current settings to default settings" button in the icon bar of the *Device Parameter Configuration* window.

     Alternatively you can use the following commands in a terminal to enable the option:
     SEP 100 1 0X0F000000 1 for axis 1
     SEP 100 2 0X0F000000 1 for axis 2, etc.
  2. Now reboot the controller by typing the RBT command in the terminal (alternatively you can restart the PC). This time all data is read from the ID-chip and stored on the controller.
  3. To ensure that at next power on the controller will not read all data again and perhaps overwrite your optimized parameters, you now will have to disable the "Read ID-Chip always" option, again for each axis separately.

     In *NanoCapture*™, proceed as described above for enabling but make sure that the parameter has the value "disabled".

     Alternatively you can use the following commands in a terminal to disable the option:
     SEP 100 1 0X0F000000 0 for axis 1,
     SEP 100 2 0X0F000000 0 for axis 2, etc.

If you had optimized parameters before the repair/upgrade, PI recommends you to repeat your optimization routine when the stage is returned.

2.12 Using Multiple E-761s

2.12.1 Installation Details

Up to 4 boards can be installed in one PC.

The system will identify each board on the basis of its factory-set, unique device ID (parameter ID is 0x0D000600).

The registration procedure described on p. 17 in step 11 is required for every E-761 board which is installed in the PC.

For every board a separate instance of the host software must be started.

2.12.2 Synchronisation of Multiple E-761s

If only one E-761 board is used, it will work as a master, i.e. it uses the internal 100 kHz and 200 kHz signals for synchronisation. If multiple boards are synchronized, an auto-detection procedure will dispose the device with the smallest ID to be the master.

Interconnect multiple E-761 boards as follows for synchronisation:

If no trigger output brackets are present in the PC, interconnect the connectors J6 and J7 of the boards using the synchronisation cable(s) which comes with the E-761 (order #K010B0029; for pinouts of J6 and J7 on the E-761 board see p. 146). The connection must be done inside of the PC case as follows before starting the PC:

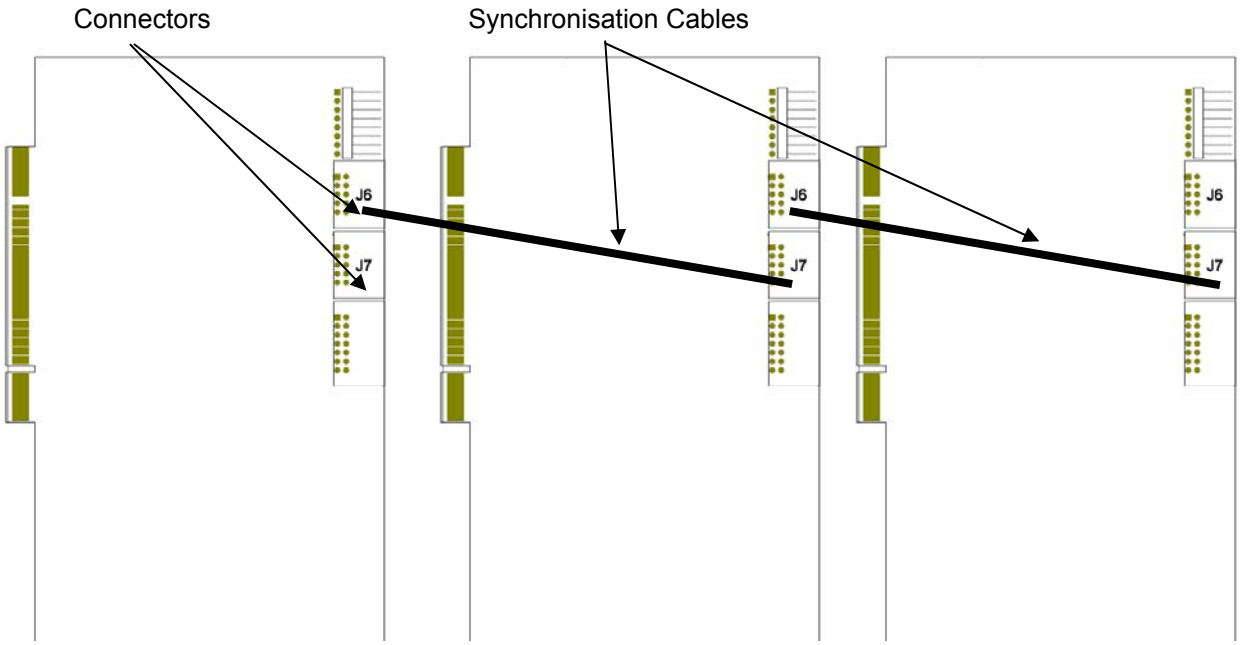


Fig. 9: Interconnecting Multiple E-761 Boards for Synchronisation

If at least one trigger output bracket is present in the PC and connected to an E-761 board, a trigger output bracket is required too for every further E-761 board which is to be synchronised. A trigger output bracket is included with E-761.3CT and can also be ordered separately as accessory, order number is E-761.00T. Install the trigger bracket(s) in the PC as described in Section 2.1.2 on p. 17. Outside of the PC case, connect the Sync Out socket of the bracket belonging to the first E-761 to the Sync In socket of the bracket belonging to the second E-761,

the Sync Out socket of the second bracket to the Sync In socket of the bracket belonging to the third E-761 and so on (see "Sync Out and Sync In Sockets" on p. 147 for pinout). Use the synchronisation cables with LEMO connectors (K030B0296) which come with the trigger output brackets.

2.13 Uninstalling and Updating

2.13.1 Host Software

Windows operating systems:

To uninstall or update the E-761 host software run the Setup Wizard from your E-761 CD or update media.

Note for Windows Vista: The Setup Wizard must be started as administrator. Click on the setup.exe file with the right mouse button and select the "Run as administrator" entry from the context menu.

Linux operating systems:

To uninstall the E-761 host software, proceed as follows:

1. Insert the E-761 CD in the host PC.
2. Open a terminal and go to the /linux directory on the E-761 CD.
3. Log in as superuser (root).
4. Start the uninstall script with ./REMOVE
   Keep in mind the case sensitivity of Linux when typing the command.

To update the E-761 host software run the install script as described on p. 16.

Download of updated releases:

Updated releases of software and manuals are available for download at www.pi.ws. While the manuals are freely accessible, you need a password for the software download. This password is provided on the E-761 CD in the E-761 Releasenews PDF file in the \Manuals directory.

To download the latest software (complete CD mirror) from the PI Website, proceed as follows:

1. On the www.pi.ws front page, click on *Download/Support* in the *Service* section on the left
2. On the *Download/Support* page, click on *Manuals and Software*
3. On the *PI Download Server* page, enter the Username and the Password which are provided in the E-761 Releasenews xxxxx.pdf on the E-761 CD and click on *Login*
4. Click on *Download* in the navigation bar across the top
5. Click on the *E Piezo Drivers & Nanopositioning controllers* category
6. Click on *E-761*
7. Click on *Release* (if you click on *Documents* you will get the latest manuals)
8. Click the download button below the latest CD-Mirror (includes the manual versions that were with the release)

2.13.2 Firmware

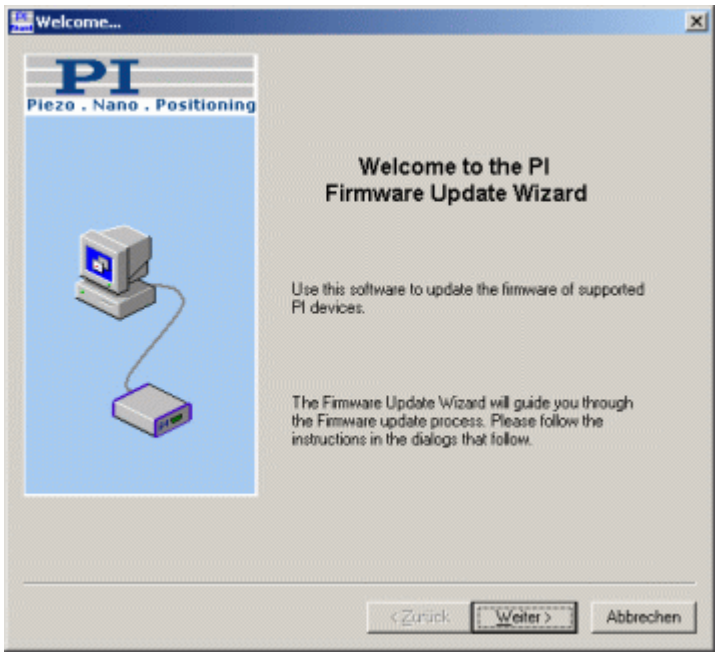
The firmware revision of your E-761 board can be identified in the answer of the IDN? command (p. 75). Contact your PI Sales Engineer or write info@pi.ws to obtain the latest firmware.

Windows operating systems:

Firmware updates can be made by running the Firmware Update Wizard on the host computer. The Firmware Update Wizard is available on the E-761 CD and installed with the procedure described on p. 16 ("typical" installation).

To update the firmware, start the Firmware Update Wizard from the Windows *Start* menu, or start PZTControl™ and select the *DSP Controller Firmware…* menu item from the *Firmware* menu. With Windows Vista, the Firmware Update Wizard must always be started with the "Run as administrator" option. To do this, click on its *Start* menu entry with the right mouse button and select the "Run as administrator" entry from the context menu.

The Firmware Update Wizard guides you through the firmware update.



Linux operating systems:

In the /linux directory of the E-761 CD, enter './pi_e761_fw_update --help' for detailed information. The pi_e761_fw_update script guides you through the update of the firmware of your E-761 board.

3 Principle of Operation

3.1 Axis and Channel Definitions

In dealing with coordinated control of logical axes, the terms “axis” and “channel” are not synonymous. Multiple PZT amplifiers ("piezo channels") and multiple sensors ("sensor channels") can be involved in the motion of one logical axis, as well as one amplifier or one sensor can participate in the motion of more than one logical axis.

Typical system configurations include:

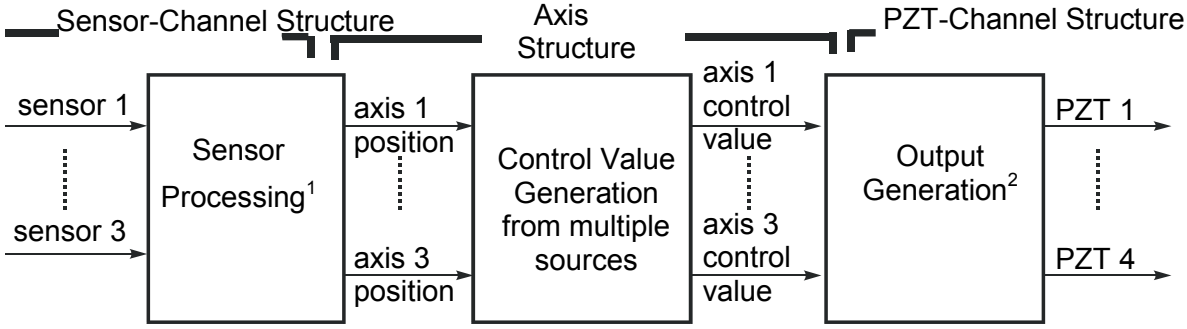
- Independent single-axis piezo actuators or stages. In this case the axes and channels correspond to one another.
- One multi-axis stage; Here the number of axes may be different from the number of channels. The number of piezo channels and sensor channels which are used by the stage are also independent of each other.

The E-761 can control up to 3 logical axes (default axis identifiers are 1, 2, 3; can be changed using the SAI command (p. 92), for example to X, Y, Z) and comprises

- 4 piezo channels, identifiers are 1, 2, 3, 4
- 3 sensor channels, identifiers are 1, 2, 3

3.2 Processing Steps—Overview

The E-761 controls the motion of the logical axes of the connected stage(s). The control values for the axes can be given by multiple sources (see "Control Value Generation", p. 44 for an overview). Depending on the operating mode of the axes, control loops with feedback from multiple sensors can be used to maintain the axis positions (see "Sensor Processing", p. 41 for details). The control values are then transformed to control voltages which drive the PZT amplifiers (piezo channels) for the actuators in the stage(s), see "Output Generation", p. 46 for more information.



[1] Sensor processing details, see Fig. 11;
[2] Details of output generation see Fig. 15

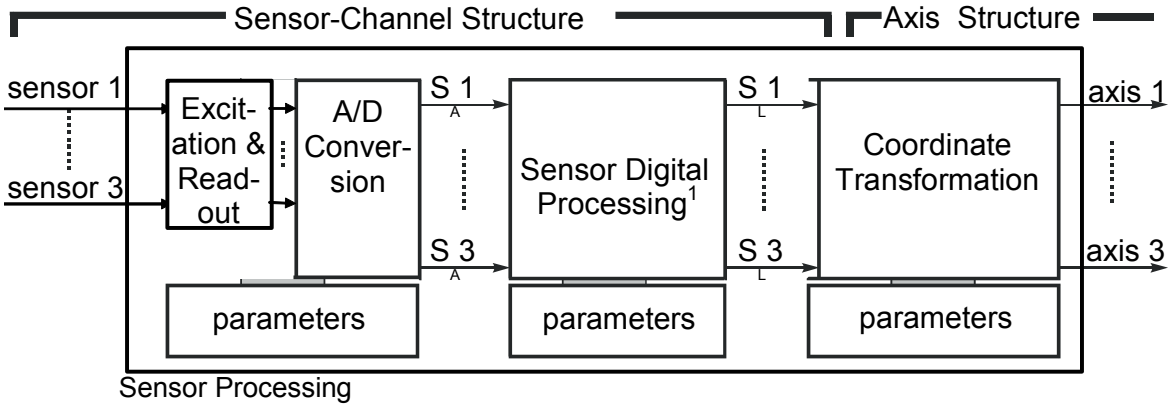
Fig. 10: E-761 Processing Overview

3.3 Sensor Processing

The E-761 is designed to work with highly accurate capacitive position sensors.

The sensor processing section consists of

1. Analog sensor readout followed by analog to digital conversion
2. Digital sensor-value processing
3. Coordinate transformation (channel-to-axis conversion)



[1]Sensor digital processing details, see Fig. 12

Fig. 11: Sensor Processing

3.3.1 Sensor Analog/Digital Conversion Unit

The sensor and associated analog circuitry converts the mechanical distance change to an analog voltage change. In the standard configuration, the distance between the capacitor plates ranges from 50% to 150% of the mid-point distance. The analog voltage is proportional to the distance change. The mid-point distance is also equal to the measurement range.

The important related parameter is the:

- **Sensor Range Factor**, a factor which can be applied to the standard sensor range. A 100 µm sensor can also be used with a measurement range of 200 µm by specifying a factor of 2; at a range of 125 µm with a factor of 1.25 or at 300 µm with a factor of 3.

3.3.2 Sensor Digital Processing Unit

The sensor digital processing section consists of

1. Digital filters, one per sensor channel.
2. Digital compensation for non-linearities in the electronics (per channel)
3. Digital compensation for nonlinearities in the sensor mechanics (per sensor-channel basis)

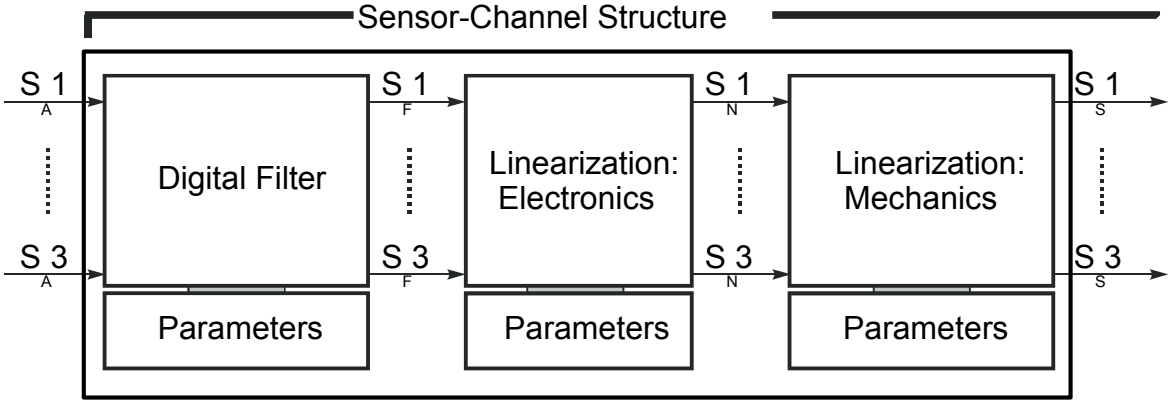


Fig. 12: Sensor Digital Processing

The important related parameters are:

- **Digital Filter Type.** There are three different types of digital filter. IIR filter, FIR filter and user FIR filter. The FIR filter is a simple moving-average filter.
- **Digital Filter Order.** The *order* of the digital filter is number of previous values used in determining the present output. The digital filter order is always 2 for IIR filter and maximum 1000 for FIR filter.
- **Digital Filter Bandwidth**, Only used if the IIR filter has been selected.
- **Polynomial-fit parameters for the electronics**: sensor offset, sensor gain, 2^{nd} and 3^{rd} order sensor correction. These parameters are independent of the connected mechanics. They are set by PI and may not be changed by the user.
- **Polynomial-fit parameters for the mechanics**: sensor offset, sensor gain, 2^{nd}, 3^{rd} and 4^{th} order sensor correction. These parameters depend on the connected stage mechanics. If the connected stage has an ID-chip, the values will be read in from the ID-Chip upon PC power-up or when the controller is rebooted.

The intermediate result values can be reported by special commands as follows:

- The TAD? command (p. 100) reports the value from the A/D converter (–32768 to +32767).
- The TNS? (p. 102) command reports the result after the linearization for the electronics (-100 to 100).
- The TSP? (p. 103) command reports the result after the linearization for the mechanics (the unit is µm).

3.3.3 Sensor Coordinate Transformation

Matrix algebra is used for coordinate transformation. Normally fewer than maximum number of parameters is used for the transformation, but the full range is available. It can be used, for example, to compensate the cross-talk. Fig. 13 shows the principle block diagram for the calculation of one axis position from three sensor values.

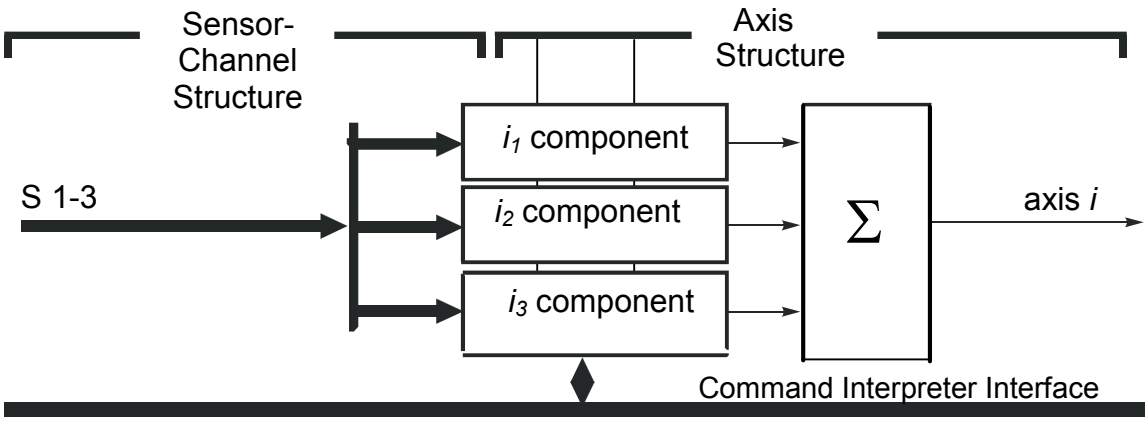


Fig. 13: Sensor-channel-to-axis coordinate transformation (axis i)

The important parameters are the components of the sensor matrix, which are defined as follows:

$$axis_i = a_{i1}S_1 + a_{i2}S_2 + a_{i3}S_3$$

The full transformation algorithm can be expressed in matrix form as follows:

$$\begin{pmatrix} axis1 \\ axis2 \\ axis3 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} S1 \\ S2 \\ S3 \end{pmatrix}$$

Note that the sensor channel signals (*S1* to *S3*) can be read with the TSP command (see previous section), while the *axis1* to *axis3* values can be read with the POS? command (p. 90).

If your controller was ordered as part of a complete system, or if PI had sufficient knowledge of your setup, the coordinate transformation matrices will be filled in with the appropriate values during calibration before shipment.

The matrix elements are accessible with SPA (p. 94) and SEP (p. 93) from a terminal. In *NanoCapture*™ the values are accessible from the *Device Parameter Configuration* window in the *Axis Definition 1* to *Axis Definition 3* parameter groups. There, the element a_{ij} (i=1, ... , 3; *j* = 1, ... ,3) is referred to as the "[axis i] Position [component] from Sensor *j*".

The result of the transformation is single-column matrix, normally representing a position expressed in the logical-axis coordinate system (works for both linear and rotational axes).

If the sensor and logical axes in the system correspond, then the values on the main diagonal of the transformation matrix are the scale factors and the rest will be close to zero. The phenomenon of sensor cross-talk between axes appears as

(and can be compensated by) non-zero values for these coefficients. The cross-talk can be defined as $\delta_{ij} = \frac{\Delta_{ij}}{\Delta_i}$

where i is the commanded axis and *j* is the axis which has a cross-talk error of Δ_{ij}.

The motion of $axis_j$ is then given by:

$$axis_j = a_{j1}S_{s1} + a_{j2}S_{s2} + a_{j3}S_{s3} + \delta_{ij}(a_{i1}S_{s1} + a_{i2}S_{s2} + a_{i3}S_{s3})$$

NOTE

If the controller and actuators are ordered together and/or PI has sufficient knowledge about your application, the system will be delivered preconfigured and precalibrated, so it should not be necessary to change these values.

3.4 Control Value Generation

3.4.1 Operating Modes

The E-761 provides the following operating modes:

- **Open-loop control** (also referred to as "servo-off state" in this document): sensor feedback is not used
- **Closed-loop control** (also referred to as "servo-on state" in this document): sensor feedback participates in the control value generation. For each logical axis, a proportional-integral (P-I) servo-controller is used to generate corrections to the control value. In addition, two notch filters are used for each axis, and the velocity is limited by a special slew rate setting. The settings for the P-I controller, the notch filters and the slew rate are accessible as parameters. See "Servo-Controller Dynamic Calibration", p. 62 and "Parameter Overview", p. 134 for more information.

The operating mode for the axes can be selected with the SVO command (p. 98). In the NanoCapture™ main window, it can be set in the *Current Axis* pane (see Section 2.5 on p. 24).

By default, open-loop control is active after power-on. Using the Servo ON/OFF startup parameter (ID 0x07000800), you can set up the individual axes to start with closed-loop control.

3.4.2 How to Command Axis Motion

The control value for the motion of an axis can result from multiple sources (see below). The interpretation of the control values depends on the settings of the axis-to-PZT matrix (see "Output Generation" on p. 46 for more information). By default, the matrix is set up so that given control values correspond numerically to axis position values (in µm or µrad), irrespective of the current operating mode.

The following control sources are available with E-761:

- Axis-related move commands: SVA (p. 97) and SVR (p. 99) in open-loop operation; MOV (p. 86), MVR (p. 87) and GOH (p. 82) in closed-loop operation; IMP (p. 84) and STE (p. 95) for both operating modes.

- Wave generator output for periodic motion in both operating modes (see "Wave Generator", p. 52 for more information and examples). Note that wave generators output absolute values.
- Analog control input for motion in both operating modes (see "Using the Analog Input", p. 33 for more information and examples).
- Direct access in both operating modes: you can set control values directly in the E-761 RAM using the function `E761_SetDirectTarget` of the GCS library. Using this function, data is written every servo loop, i.e. with up to 25 kHz. See the library manual PZ163E for more information.

NOTES

Make sure that your system (hardware, operating system, configuration) is fast enough to attain the data rate provided by the E-761 board.

To read positions directly from RAM, you can use the `E761_GetDirectPosition` library function.

The GCS library is available as follows:
Windows operating systems: E7XX_GCS_DLL;
Linux operating systems: libpi_e7xx_gcs.so.*x.x.x* and
libpi_e7xx_gcs-*x.x.x*.a where *x.x.x* gives the version of the library

Direct access is only available via the library functions, but not in the other host software provided by PI.

The control sources listed above have different write priorities:

- When the wave generator is running, move commands are not allowed.
- Direct access via the `E761_SetDirectTarget` library function overwrites control values given by move commands or by the wave generator.
- If the analog input is enabled for "direct" control of an axis (parameter ID 0x06000902 set with SPA, see p. 34 for details), its value is added to the current control value resulting from a move command, from the wave generator output or from a value given by the `E761_SetDirectTarget` library function.

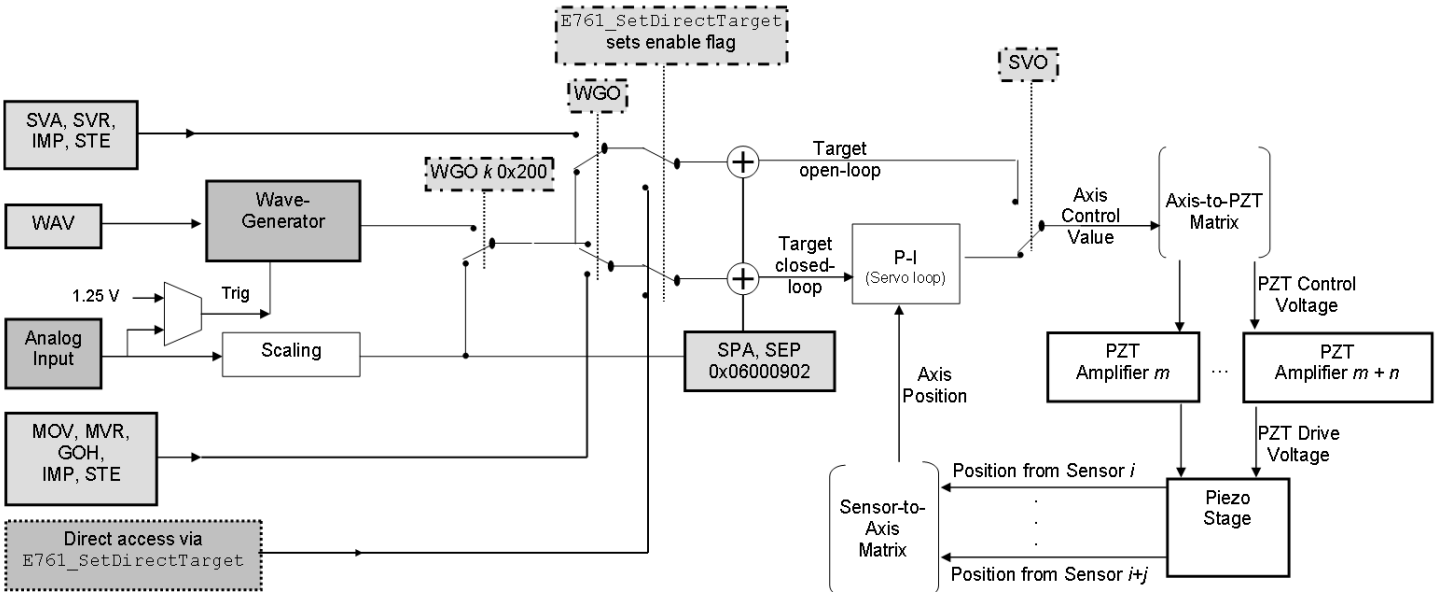
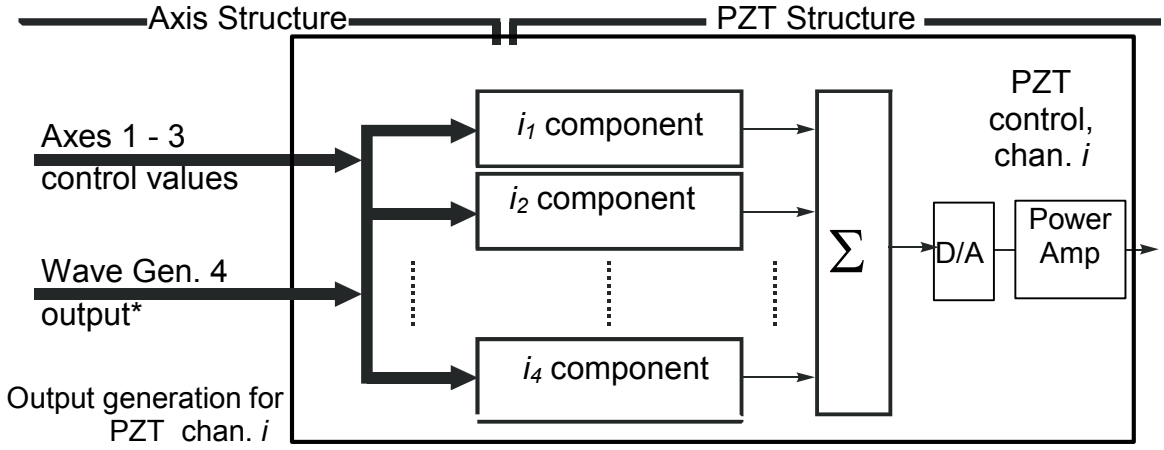


Fig. 14: E-761 control sources, shown for axis k

3.5 Output Generation

To execute the motion of an axis, multiple piezo actuators may be used so that multiple piezo channels (PZT amplifiers) are involved. The control value for an axis is transformed to control voltage values for the piezo channels via the axis-to-PZT matrix. After the digital-to-analog conversion, the resulting control voltage values are sent to the PZT amplifiers whose output drives the actuators in the mechanics.



* Note that a 4 x 4 matrix is used to make the output of the 4th wave generator available (see p. 59 for details).

Fig. 15: Output generation for PZT channel i, i=1 to 4

In the axis-to-PZT-channel matrix, input values of non-supported axes are always 0. The matrix elements are accessible with SPA (p. 94) and SEP (p. 93) from a terminal. In *NanoCapture*™ the values are accessible from the *Device Parameter Configuration* window in the *Axis Definition 1* to *Axis Definition 3* parameter groups, as the "Driving with piezo i" factors (with i = 1 to 4).

NOTES

During calibration at the factory, the coefficients of the axis-to-PZT matrix are set numerically to the number of volts which are required per axis unit by the attached piezo actuators (i.e. the unit of the coefficients is V/µm). Thus both the closed-loop control value and the open-loop control value correspond numerically to axis position values. This means that all control sources always command with axis position values, irrespective of the current operating mode. You should not change the matrix coefficients for the piezo amplifier channels.

If the connected mechanics has an ID-chip, the coefficients will be read in from the ID-chip (see "ID-Chip Support / Stage Replacment" p. 35 for more information).

3.6 Data Recording / Storage

The generation of waveforms and several data recording operations involve data storage on the E-761 (see also Fig. 16):

Command	Data Flow	Comments	Page
WAV	write waveform to wave table		108
GWD?	write wave table content to PC		82
WGO	write actual position data to recorder table	starts the wave generator output	114
WGR	write actual position data to recorder table	only possible when wave generator is running	117
DRR?	write recorder table content to PC	reads data which was recorded with WGO, WGR, IMP and STE	80
STE	write actual position data to recorder table	performs a step and records the step response	95
STE?	write recorder table content to PC	reads the data recorded with STE	96
IMP	write actual position data to recorder table	performs an impulse and records the frequency response	84
IMP?	write recorder table content to PC	reads the data recorded with IMP	85

The content of the wave tables and recorder tables is temporarily stored on the E-761. It is overwritten by every new wave definition or recording operation and lost when the controller is powered down or rebooted.

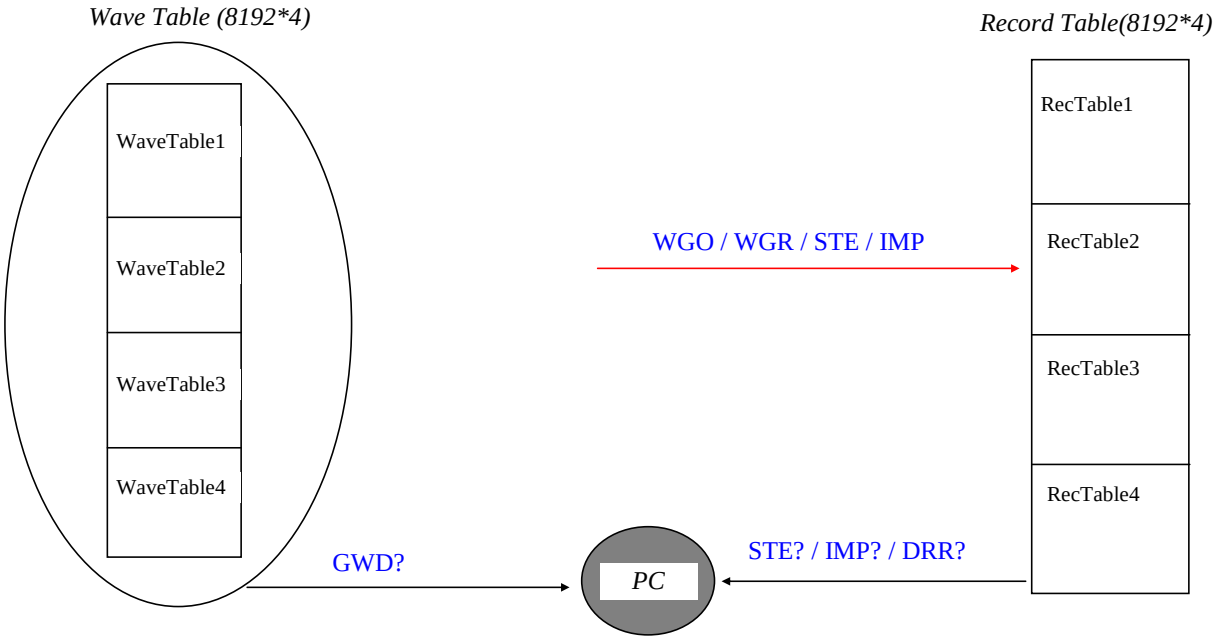


Fig. 16: Data storage on the E-761

4 Hardware Description

4.1 Board Layout

DANGER—HAZARDOUS VOLTAGE

The E-761 digital piezo controller board and the trigger output bracket do not contain any user-serviceable parts. Never re-assemble the E-761 board and the trigger output bracket. Hazardous voltage can be present on the E-761 board components.

Never touch any pins and conductors on the E-761 board and any part that might be connected to the high-voltage output. Hold the E-761 board and the trigger output bracket only by the edges.

Do not remove the protective cover (see Fig. 17) from the E-761 board. Otherwise electrical contact to the various add-on cards in the PC can be possible.

WARNING—BURNING HAZARD

Do not touch the aluminum heat sink immediately after board operation! The temperature can be up to 60 °C.

See Section 11.2 on p. 145 for pinouts.

E-761 board description:

1 PCI bridge
2 LEMO connector for analog input
3 Sub-D Mix connector for Piezo stage(s)
4 Status LED for high voltage output; if on, the high voltage output is active; flashes 5 times in one second when connecting to the board for the first time after PC start
5 CPDL programmer
6 J6, sync and trigger outs
7 J7, sync
8 DSP Emulator
9 High Voltage output
10 PC power supply connection
11 Heat sink of the amplifiers
12 Protective cover with calibration label

The elements 2, 3 and 4 are available outside of the PC on the bracket of the board.

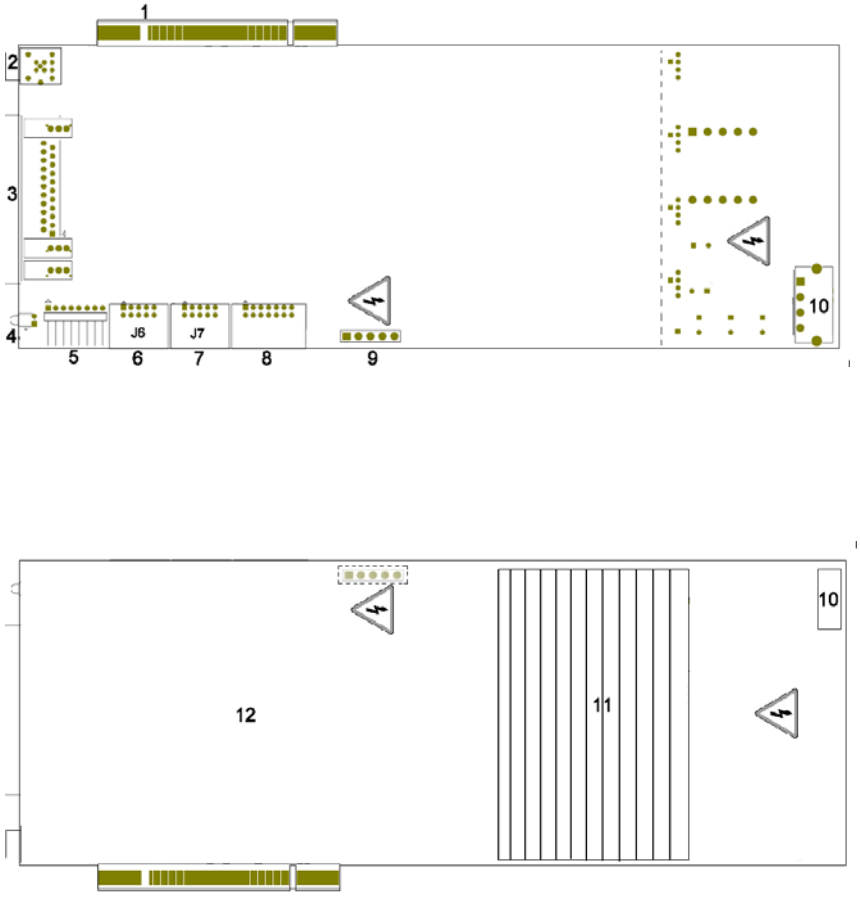


Fig. 17: E-761 board layout, both sides

4.2 Controller Timing Basics

Digital controller operation is based on internal sampling cycles. The ADC sampling rate is 100 kHz, while the sampling rate of the sensor data and the servo sampling rate by default both are 25 kHz (default oversampling factor is 4). The oversampling factor can be changed using the AVG command (p. 77).

4.3 Frequency Response

In order to achieve minimum distortion of the output waveform, it is important to ensure that the control input amplitude is reduced in proportion to the fall-off of the output voltage at higher frequencies. For exact information on maximum operating frequency with a given PZT load, refer to the individual frequency response graphs in Fig. 18.

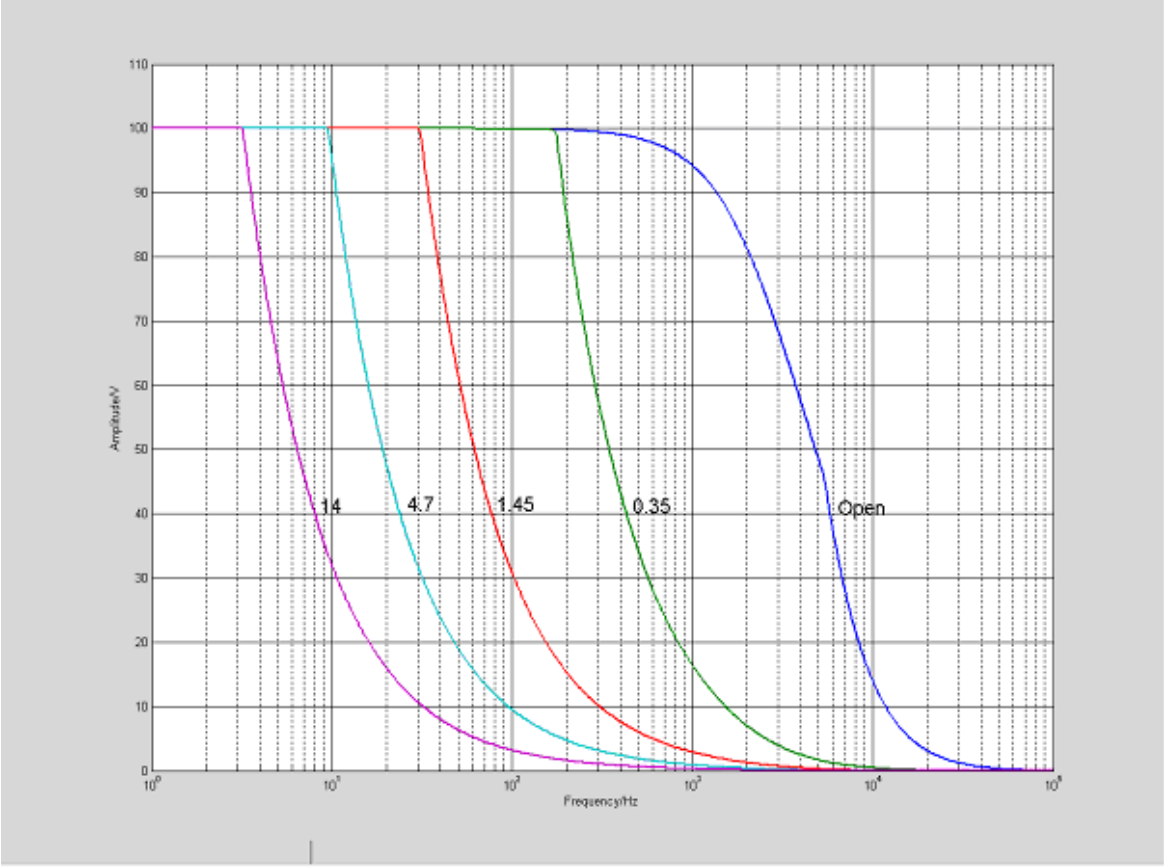


Fig. 18: E-761 frequency response with various PZT loads.
Capacitance values in µF

Note that the frequency response of a given amplifier depends on the amplifier power, the amplifier design, and, of course the PZT capacitance. The capacitance of PZT ceramics significantly changes with amplitude, temperature, and load—up to approximately 200% of the unloaded, small-signal capacitance at room temperature.

The following equations describe the relationship between (reactive) drive power, actuator capacitance, operating frequency and drive voltage.

The average power a piezo driver has to be able to provide for sinusoidal operation is given by:

$P_a \approx C \cdot U_{max} \cdot U_{p\text{-}p} \cdot f$

Peak power for sinusoidal operation is:

$P_{max} \approx \pi \cdot C \cdot U_{max} \cdot U_{p\text{-}p} \cdot f$

Where:

P_a = average power [W]

P_{max} = peak power [W]

C = PZT actuator capacitance [farad (As/v)]

f = operating frequency [Hz]

U_{max} = nominal voltage of the amplifier [V]

$U_{p\text{-}p}$ = peak-peak drive voltage [V]

5 Wave Generator

5.1 General Information

The E-761 has four wave tables for creating and (temporarily) storing arbitrary waveforms. These waveforms can then be output by up to four wave generators. Additionally, the analog input signal can be used as an "external wave generator" or as trigger signal for the output of the four internal wave generators (see section 2.9 on p. 33 for details).

The assignment of wave tables, wave generators and axes to each other is fixed:

- The waveform stored in wave table 1 is output by wave generator 1 for axis 1. If the analog input is used as "external wave generator", the output for axis 1 is enabled with wave generator ID 1 accordingly. For axes 2 and 3 the assignment is implemented in an analogous manner.
- The waveform stored in wave table 4 is output by wave generator 4. Since there is no 4th axis, the output can be assigned to piezo channel 4, if the stage(s) require(s) only piezo channels 1 to 3, see Section 5.4 on p. 59 for details.

Waveforms can be created based on predefined "curve" shapes. This can be polynomial, sine, ramp or single scan line curves. Additionally you can freely define curve shapes and configure several parameters for a waveform. See the WAV command on p. 108 for more information.

When you create waveforms or use the analog input as "external wave generator", keep in mind that the applicable frequency depends on the available amplifier power. Example: with a capacitive load of 6.6 µF, the frequency should not exceed 15 Hz if three amplifiers are involved in the motion or 50 Hz if only one amplifier is involved (motion covers the whole travel range; see also Section "Frequency Response" on p. 50). Otherwise overheating of the amplifiers can occur, and the piezo voltage output will be deactivated automatically. In this case the stage will not move although communication with the board is still possible (see "Troubleshooting" on p. 140 for further information).

A waveform can be output a fixed number of times, or repeated indefinitely. When the wave generator output is stopped and restarted, it will normally continue with the next point of the waveform, even if the axis was moved by a move command in the meantime. Wave generator output will continue even if the terminal or the program from which it was started is quit or if the high voltage output is deactivated. See the WGO command on p. 114 for more information.

The waveform values are absolute values. In closed-loop operation (servo ON), they are interpreted as target positions in either case. In open-loop operation (servo OFF), the interpretation of the wave generator output depends on the settings of the axis-to-PZT matrix (see "Output Generation", p. 46 for more information). By default, the matrix is set up so that commanded open-loop control values numerically correspond to axis position values.

When the wave generator is running, move commands are not allowed. If the analog input is enabled for "direct" control of an axis (parameter ID 0x06000902 set with SPA, see p. 34 for details), its value is added to the current control value resulting from the wave generator output. See "Control Value Generation" on p. 44 for details.

During wave generator output, there is no velocity control, i.e. any settings made with VCO (p. 103) will be ignored.

Digital output synchronized with the wave generator output and hence with the axis motion is possible via the start options bit 3, bit 4 and bit 5 of the WGO command. To make the digital output available outside of the PC, a trigger output bracket is required (included with E-761.3CT models; can be ordered separately as E-761.00T). The assignment of the three trigger output lines to the wave generators (and hence to the axes) is fixed: TrigOut1 belongs to wave generator 1 (axis 1), TrigOut2 belongs to wave generator 2 (axis 2), and TrigOut3 belongs to wave generator 3 (axis 3). With wave generator 4, no digital trigger output is possible. See p. 147 for pinout of the Digital Out socket.

Each time the wave generator is started recording starts automatically as follows (read the data with DRR?, p. 80):
recorder table 1: axis 1 actual position
recorder table 2: axis 2 actual position
recorder table 3: axis 3 actual position
recorder table 4: analog input voltage (same value as read with TAV?, i.e. contains gain and offset for the analog input, see p. 100)
Recording always takes place for all record tables, regardless of which wave generator was started. Recording ends when the record table content has reached the maximum number of points (8192 per table). Recording can be restarted by WGR (p. 117). The record table rate can be set with RTR (p. 91).

The different E-761 software interfaces also support use of the wave generator. Waveforms can be defined, stored and displayed in and by the software in a more user-friendly way than in a terminal using WAV and WGO. If using the wave generator with the GCS library, *PZTControl™, NanoCapture™* or LabView, read the descriptions in the associated software manual first.

5.2 How to Work with the Wave Generator

NOTES

Be sure that you have set correct waveform sequence and configuration before enabling wave output.

Although the WAV command is available when waveform output is enabled, an incorrect command parameter could cause unpredictable stage response, such as overflow and vibration. It is therefore recommended that waveform output be disabled before using the WAV command, and re-enabled only after the waveform sequence has been checked.

Using the wave generators is as follows:

1. Define the waveform using the WAV command. See the detailed command description on p. 108 and the examples in Section 5.3.
2. Optionally: Check the waveform:
   After you sent the waveform definition to the wave table, it is always a good idea to check it by reading back the waveform sequence from the E-761 before actually outputting it. This can be done by the GWD? command (see p. 82).
3. Optionally: Specify additional wave parameters:
   Using the WAV command with the "CFG" wave type, additional parameters can be configured which will be applied to the stored waveform when the output starts. This can be useful e.g. when a very low frequency is required or when the wave is to increase automatically. In principle, the CFG settings are valid until a new WAV command is sent or the wave table content is cleared or the controller is powered down or rebooted. Some CFG settings are also available

as controller parameters and can be modified with SPA and saved to non-volatile memory as power-up defaults—see p. 112 for details.

4. Optionally: Set the number of cycles for the wave generator output using the WGC command (p. 114; can also be done via the "Wave generator cycles" parameter, ID 0x13000003).
5. Optionally: If you want to use the analog input line for triggering tasks, provide the appropriate signal and configure the analog input. See p. 33 for details.
6. Start the wave generator output and hence the motion of the axis using the WGO command (p. 114):
   The start mode is set separately for each wave generator (= axis) via "start bits". With certain start bits, you can specify start options like digital trigger output or start at the endpoint of the last cycle.
   When starting the wave generator, recording is started automatically, and the data can be read with DRR? (p. 80).
7. Optionally: Restart data recording with the WGR command (p. 117).
8. Stop the wave generator output. This can be done with WGO or with #24 (p. 75).

You can check the wave generator activation status with #9 (p. 75). With WGO? (p. 117) you can ask for the last-commanded wave generator start options (WGO settings).

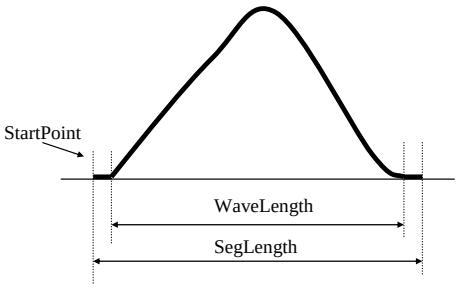
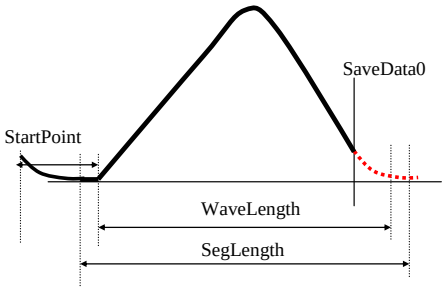
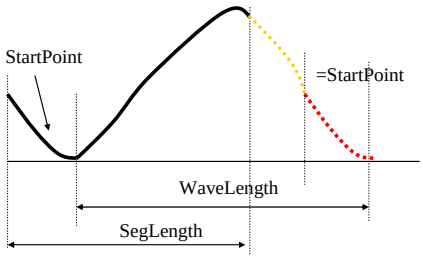
A simple example for your first steps (using the command entry facilities of *NanoCapture*™ or *PZTControl*™ or the *E-761 Terminal*):

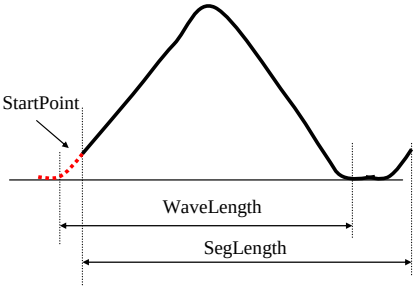
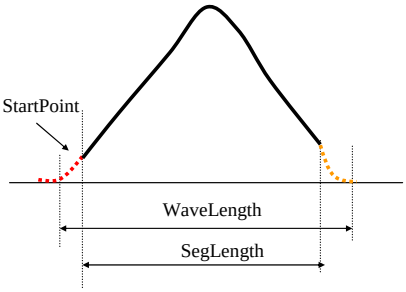
Action	**Content of Program Window**	**Comment**
Send:	SVO X 0	Servo is switched OFF for axis X (when you want to start the wave generator in closed-loop operation, send "SVO X 1")
Send:	WAV 1 ….	Define a waveform for wave table 1 (axis X); see WAV description (p. 108) for details
Send:	WGC 1 10	Set the number of cycles for the wave generator output to 10
Send:	WGO 1 1	Start output of wave generator 1 (axis X) immediately (synchronized by interrupt)
Send:	WGO 1 0	Stop output of wave generator 1 (axis X)

5.3 Examples

5.3.1 Phase Shift for Sine and Ramp Curves

The examples in this section apply to the wave creation with WAV and the SIN_P and RAMP wave types.

Condition 1	Condition 2	Phase Shift	Waveform
StartPoint ≥ 0	SegLength ≥ (WaveLength + StartPoints)	No	
	SegLength ≥ WaveLength	Yes	
	SegLength < WaveLength	Yes	

Condition 1	Condition 2	Phase Shift	Waveform
StartPoint < 0	SegLength ≥ WaveLength	Yes	
	SegLength < WaveLength	Yes	

Phase Shift Conservation

If you want to have a fixed phase shift between the outputs of different wave generators, you have to keep in mind the following behavior:

Example:

Wave generator 1: output SIN
Wave generator 2: output COS

When 1 stops (while 2 is still running), it keeps the last position. When 1 is started again, then the phase shift between 1 and 2 is no longer 90°.

You can keep the fixed phase shift in one of the following ways:

1. Always start and stop 1 and 2 at the same time after you have sent a WAV 1[234] CFG command:

   WGO 1 1 2 1 *to start*
   WGO 1 0 2 0 *to stop*

2. Use the WGC command to limit the number of output periods:

   WGO 1 0 2 0
   WAV 1 2 SIN_P …
   WAV 1 CFG *(as sin)*
   WAV 2 CFG *(as cos*)
   WGC 1 10 2 20
   WGO 1 1 2 1

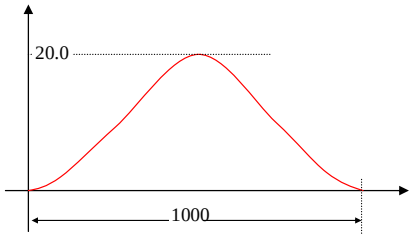
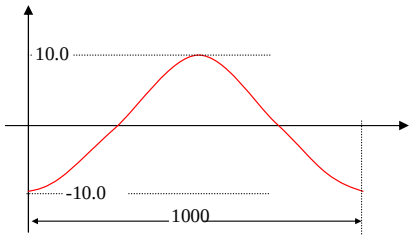
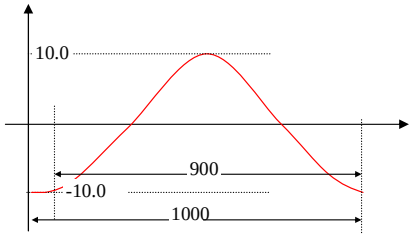
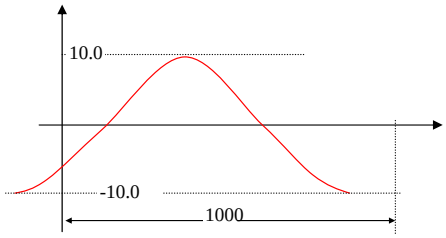
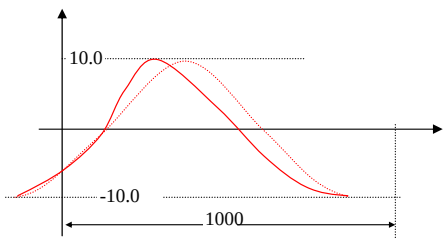
3. Run the CFG again if you stopped 1 and 2 not at the same time (universal method):

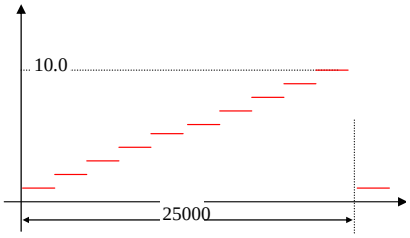
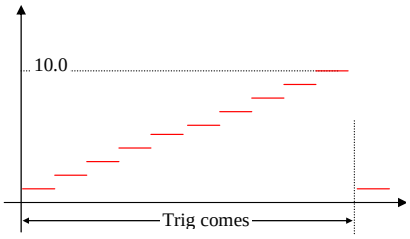
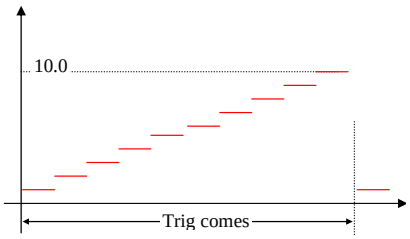
   WGO 1 0
   WGO 2 0 *(Not stopped at the same time)*
   WAV 1 CFG *(as sin)*
   WAV 2 CFG *(as cos)*
   WGO 1 1 2 1

5.3.2 Wave Generation and Output

The interpretation of the control values given by the waveform depends on the settings of the axis-to-PZT matrix (see "Output Generation" on p. 46 for more information). By default, the matrix is set up so that given control values correspond numerically to axis position values (in µm or µrad), irrespective of the current operating mode.

By default, the servo update time is 40 µs, i.e. 25000 points = 1 s. Note that the servo update time can be changed with the AVG command (p. 77).

Description	Commands	Wave Form
Sine wave: 25 Hz, Amp_{p-p}= 20	WAV 1 SIN_P 1000 20.0 WGO 1 1	
Sine wave: 25 Hz, Amp_{p-p} = 20 Offset = -10	WAV 1 SIN_P 1000 20.0 -10.0 WGO 1 1	
Sine wave: 25 Hz, Amp_{p-p} = 20 Offset = -10 , 100 starting points (1-100)	WAV 1 SIN_P 1000 20.0 -10.0 900 WGO 1 1	
Sine wave: 25 Hz, Amp_{p-p} = 20 Offset = -10 , Phase shift 72°	WAV 1 SIN_P 1000 20.0 -10.0 1000 -200 WGO 1 1	
Sine wave: 25 Hz, Amp_{p-p} = 20 Offset = -10 , Phase shift 72° Type: asymmetrical	WAV 1 SIN_P 1000 20.0 -10.0 1000 -200 100 WGO 1 1	

Description	Commands	Wave Form
PNT: Step wave 0.1 s, 1.0 to 10.0	WAV 1 PNT 0 1 1.0 WAV 1 CFG 1 1 0 25000 1 10 WGO 1 1	
PNT: Step wave 0.1 s, 1.0 to 10.0 External trig: analog input signal, for level see Note below	WAV 1 PNT 0 1 1.0 WAV 1 CFG 1 1 0 1 1 10 WGO 1 2	When trig comes, output 
PNT: Step wave 0.1 s, 1.0 to 10.0 External trig: analog input signal, for level and pulse width see Note below	WAV 1 PNT 0 1 1.0 WAV 1 CFG 1 1 0 1 1 10 WGO 1 4	When trig comes, output 

Note: The trig signal on analog input should have the following level and width:

Parameters	Max	Type	Min
V_H, voltage of logical high (V)	10.0	3.3	2.4
V_L, voltage of logical low (V)	0.8	0	-10.0
T_H, time of logical high (µs)		--	50
T_L, time of logical low (µs)		--	50

The following **additional example** shows how to **modify the waveform period** using the CFG wave type (see the WAV description on p. 108 for details):

Assume you want to output a SIN waveform made up of 4096 points with wave generator 1. You type in the following commands:

WGO 1 0

WAV 1 SIN_P 4096 10 0 4096 0 0

WAV 1 CFG 4096 1 0 1

WGO 1 1

Now, wave generator 1 outputs a sine waveform with an amplitude of 10 µm, and a period of about $t = 4096 * t_i$, where t_i is the time interval between interrupts (= 0.01 ms * oversampling factor set with AVG).

If you want to double the waveform period, you can specify 2 interrupt intervals between points with the following command:

WAV 1 CFG 4096 1 0 **2**

If you want to reduce the period of waveform, you can specify ½ interrupt interval per point (2 points per interrupt) with the following command:

WAV 1 CFG 4096 **2** 0 1

5.4 Using the 4th Wave Generator

DANGER—HAZARDOUS VOLTAGE

The amplifiers used by the E-761 are high-voltage amplifiers capable of generating high output currents. They may cause serious or even lethal injury if used improperly. Working with high-voltage amplifiers requires adequately trained operating personnel. Strictly observe the following:

→ Do not touch the pins of the Sub-D mix connector when the LED is turned on. The high voltage output of the board is active as long as the LED is turned on—normally until the PC is shut down. The output values depend on the last commanded targets (positions or voltages), even if you have quit the terminal or the program from which the targets were commanded—up to 120 V can be present on the Sub-D mix connector of the board.

→ When providing custom made adapters / connectors, take appropriate safety measures due to the high voltage output which will be present on the hardware!

The waveform stored in wave table 4 is output by wave generator 4. Since there is no 4th axis, the output can be assigned to piezo channel 4, if the stage(s) require(s) only piezo channels 1 to 3.

The following preparations are required:

- Configure the output matrix (axis-to-PZT-channel transformation, see also p. 46) using the SPA command (p. 94) so that piezo channel 4 is connected to wave generator 4 only[1]:

SPA 1 0x09000003 0	(deactivate piezo channel 4 for axis 1)
SPA 2 0x09000003 0	(deactivate piezo channel 4 for axis 2)
SPA 3 0x09000003 0	(deactivate piezo channel 4 for axis 3)
SPA 4 0x09000003 1	(activate piezo channel 4 for "axis 4", i.e. for the 4th wave generator)

- Prepare an adapter for the Sub-D mix connector which makes the output voltage of piezo channel 4 available separately while one or more stages are connected to the E-761. Take appropriate safety measures due to the high voltage output which will be present on connector and adapter! For pinout see p. 145.

Then you can proceed as described in Section 5.2 on p. 53 for waveform definition and waveform output. Be sure to use wave table ID 4 and wave generator ID 4.

[1] For configuration purposes, the system is handled as if there were a 4th axis connected to wave generator 4.

5.5 Wave Commands and Parameters

The following table summarizes the commands provided for wave generation, wave output and data recording and some useful report commands.

Mne-monic	Description	Type of Function	Page
#9	Get Wave Generator activation status	Wave	75
AVG	Set Number of Values for Averaging, i.e. affects the cycle duration for a waveform	Configuration	77
AVG?	Get Number of Values for Averaging	Report	77
DIO?	Returns the states of the digital input lines (i.e. of the analog input line)	Report	80
DRR?	Data Recorder Read	Report	80
GWD?	Query Waveform Type	Report	82
RTR	Set Record Table Rate	Configuration	91
RTR?	Get Record Table Rate	Report	92
TAV?	Get voltage at analog input.	Report	100
TNR?	Tell Number of Recorder Tables	Report	102
TWG?	Tell Number of Wave Generators	Report	103
WAV	Define Waveform	Wave	108
WAV?	Get Waveform Parameters	Report	113
WCL	Wave clear: Clears the content of the given wave table	Wave	113
WGC	Set Number of Wave Generator Output Cycles	Wave	114
WGC?	Get Number of Wave Generator Output Cycles	Wave	114
WGO	Start and Stop Wave Generator Output	Wave	114
WGO?	Get Wave Generator Output Settings	Wave	117
WGR	Start Recording	Wave	117
WMS	Get Maximum Number of Points per Wave Table	Wave	118

Some settings for wave generation, wave output and data recording are also accessible via controller parameters. See "Parameter Overview" on p. 134 for more information regarding the controller parameters and their handling.

Parameter Number	Parameter Name	Affected by Command	Notes
0x0E000200	System Global: Servo update time	AVG, p. 77	ItemID = 1 only
0x13000001	Wave Generator: Installed wave form		read-only; only in volatile memory
0x13000002	Wave Generator: Connected axis		read-only only in volatile memory
0x13000003	Wave Generator: Wave generator cycles	WGC, p. 114 WGC?, p. 114	

Parameter Number	Parameter Name	Affected by Command	Notes
0x13000004	Wave Generator: Max Wave Points		read-only only in volatile memory
0x13000102	Wave Generator: Total wave form points	WAV with CFG wave type (p. 112)	only in volatile memory
0x13000109	Wave Generator: Wave generator table rate	WAV with CFG wave type (p. 112)	
0x1300010A	Wave Generator: Number of wave tables		read-only; only in volatile memory; ItemID = 1 only
0x1300010B	Wave Generator: Curve offset	WAV with CFG wave type (p. 112)	
0X16000000	Data Record: Table rate	RTR, p. 91	ItemID = 1 only

6 Servo-Controller Dynamic Calibration

If the controller and the attached piezo stages are ordered together and if PI has sufficient knowledge of your application, then the parameters of the closed-loop control algorithm (servo parameters) will be set to suitable values at the factory, and, if present, saved in the stage's ID-chip (Section 2.11 on p. 35). Modification of those parameters will, however, be necessary if the load applied to the piezo stage is changed. In this case, for each servo-controlled axis the following parameters may need to be modified in the controller:

- Notch filter frequencies (first notch filter: parameter ID 0x08000100, second notch filter: parameter ID 0x08000101)
- Servo-loop loop gain (P-term, parameter ID 0x07000300)
- Servo-loop time constant (I-term, parameter ID 0x07000301)
- Servo-loop slew rate (parameter ID 0x07000200)

It is most convenient to use the *NanoCapture*™ software to change these parameters.

NOTES

It is strongly recommended to save the parameter values of the E-761 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-761, use the *Device Parameter Configuration* window of NanoCapture™. See "Creating Backup File for Controller Parameters" on p. 23 for more information.

For stages with ID-chip, to make the optimized settings available in the future, the option "Power Up Read ID-Chip" must have "disabled" as its power-on default (value of parameter 0X0F000000 = 0 in non-volatile memory). See "ID-Chip Support / Stage Replacment" on p. 35 for more information.

When you have finished the adjustment, i.e. when the stage performance is satisfactory, you can press the *Save as Default* buttons in the *Dynamic Tuner* window of NanoCapture™ to save the settings to the non-volatile memory of the E-761 as power-on defaults.

As long as the new settings have not yet been saved, pressing the *Reset all to defaults* buttons in the *Dynamic Tuner* window of NanoCapture™ permits resetting the P, I and Notch Filter terms to the values they had at power-on. This can be very useful to restore stable values if changing the servo parameters produces disturbing noises.

The servo parameters are also listed in the *Servo n* parameter groups in the *Device Parameter Configuration* window of NanoCapture™.

6.1 Setting Notch Filters for Suppressing Mechanical Resonance

NOTES

The *Rejection* value, which scales the damping done by the notch filter, should always be 0.05. A *Rejection* value of 1 deactivates the notch filter.

When the *Notch Freq.* value is set with NanoCapture™, the *Time Constant* servo parameter in the *P-I Controller* pane is adjusted automatically in accordance.

Mechanical resonances of the system exaggerate the response to certain frequencies in the control signal. The E-761 has two notch filters for each axis on the output of the servo-control loop to compensate for resonances in the mechanics by reducing the corresponding frequency components in the control signal.

The frequencies for the notch filters can be optimized by observing the system response to an open-loop impulse.

Using NanoCapture™, proceed as follows for each axis:

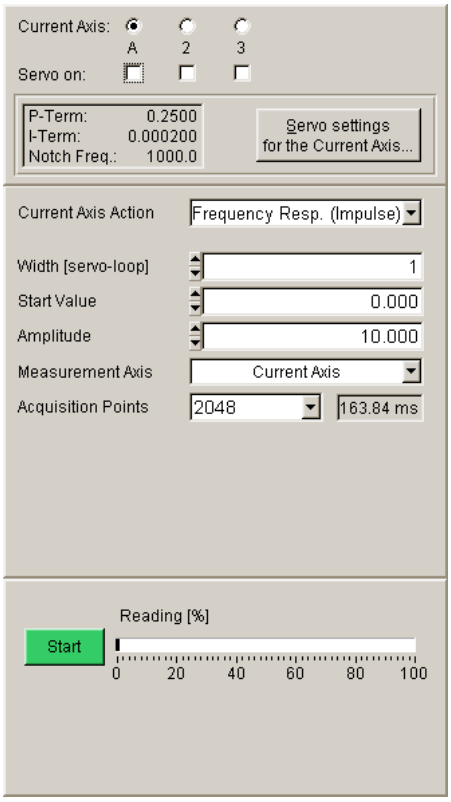
1. Make sure the mechanics is mounted and connected to the controller in exactly the same way as in the application. The load on the mechanics is especially important.

2. Start NanoCapture™ (see the NanoCapture™ software manual on the included CD for details).

3. Use the *Current Axis* radio button(s) to select the axis to set.

4. The measurement must be done in open-loop mode. Uncheck the *Servo ON* box for the selected axis.

5. In the *Current Axis Action* field, select *Frequency Response* (see figure below).

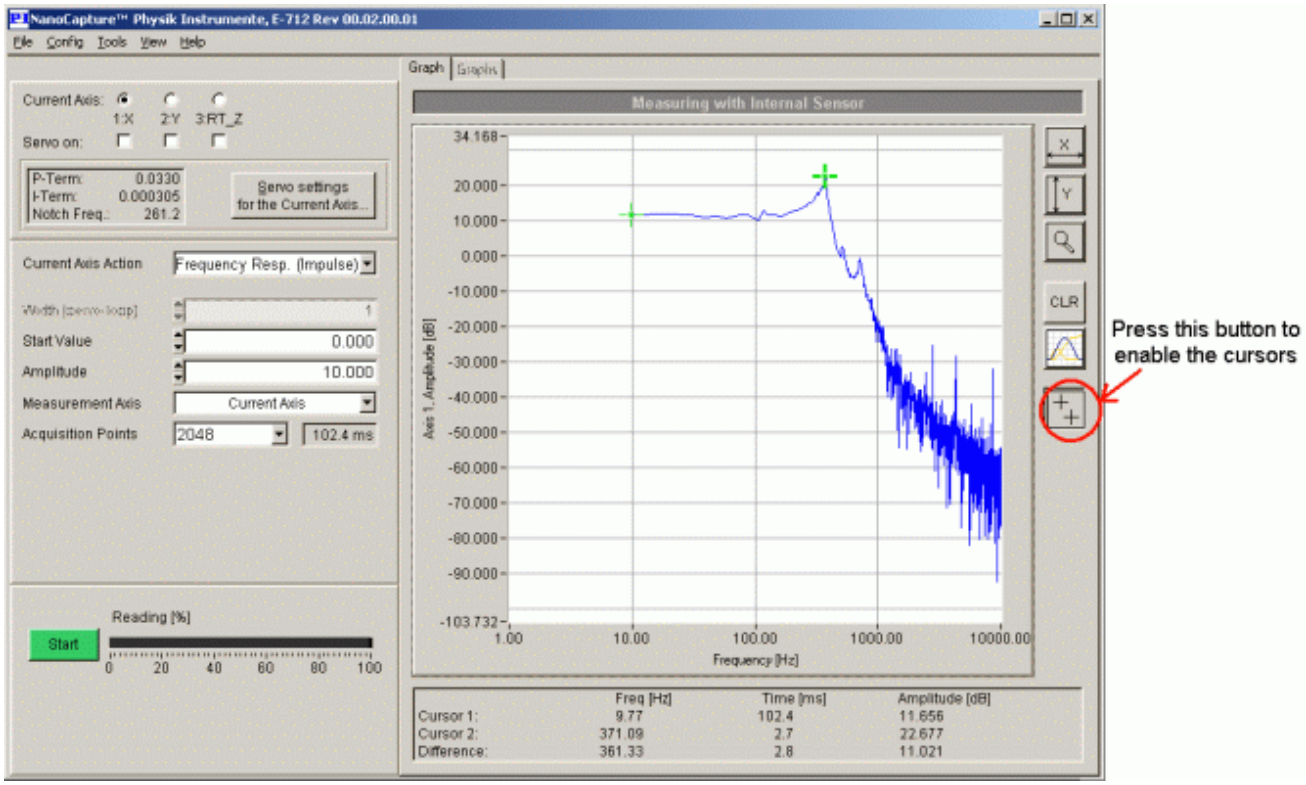


6. Set *Start Value* to 0 and *Amplitude* to about 15% of the axis travel range.

7. Set *Measurement Axis* to *Current Axis* (i.e. measure and display the axis which receives the impulse)

8. Start the measurement by clicking on the *Start* button. An impulse move is commanded (with the IMP command, p. 84), the recorded position data analyzed and displayed as a Bode frequency response diagram (see figure below).

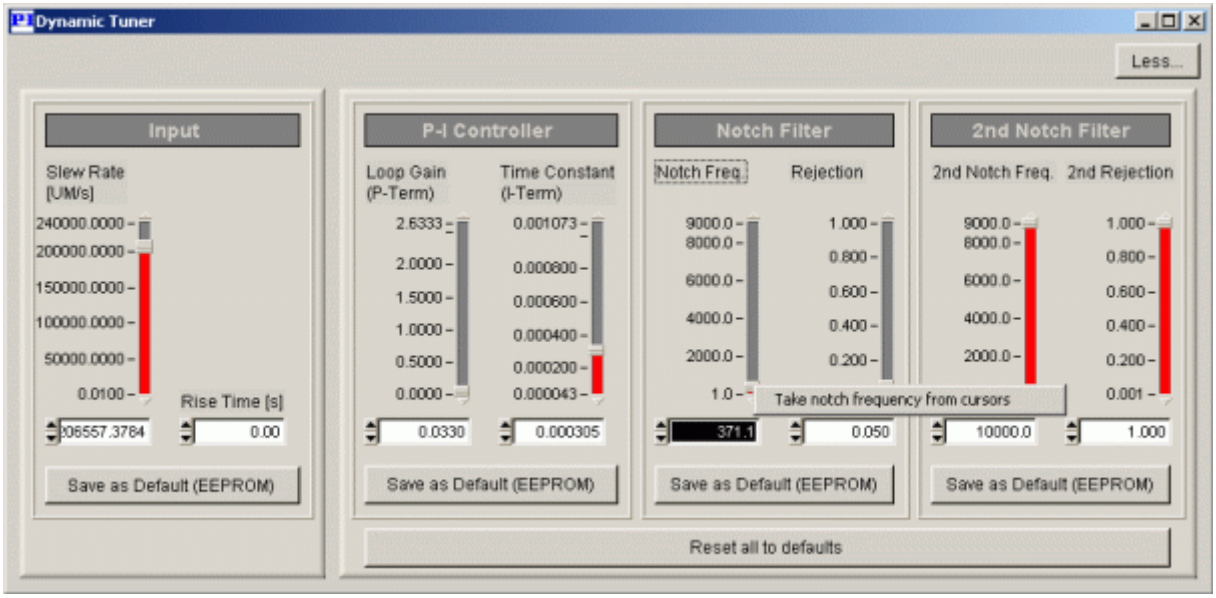
9. On the Bode plot, identify the resonance peak. To do so, you can place a cursor on the peak and read out the cursor value which is displayed below the graph (see figure below for a general example).



It is possible to adapt the notch filter setting automatically to the measured resonance peak (see step 11 below)—if you want to do so, the two cursors must be placed as follows:

With the mouse, move either one of the cursors to the resonance peak. Make sure the other cursor points to a lower frequency, i.e. that it is to the left of the cursor pointing to the resonance peak.

10. Now open the *Dynamic Tuner* window for the axis by pressing F3 or using the *View* ⇒ *Dynamic Tuner* menu sequence. If necessary click *More* to see all the parameters. This window has sliders to adjust the dynamic parameters and buttons to save the settings (see figure below).



11. Move the *Notch Freq.* slider to adjust the notch frequency or right-click on the slider and click on the *Take notch frequency from cursors* item on the context menu that appears (see figure above). The notch filter frequency will then be taken from the cursor which has the higher frequency value.

12. If you have finished the settings, you can press the *Save as Default* buttons to save the settings in the controller as power-on defaults (the other settings in this window will be discussed later).

13. If desired, repeat this procedure for the second notch filter, using the next higher resonance and the *2nd Notch Freq.* slider.

6.2 Setting Servo Parameters

The servo-loop settings for an axis—loop gain (P-term), time constant (I-term) and slew rate (velocity)—affect the settling time and control overshoot. To optimize these parameters, the system response to a closed-loop step move is observed. To modify these parameters, use the *Dynamic Tuner* window of *NanoCapture™* (see adjustment procedure below).

6.2.1 Loop Gain (P-term)

This parameter also affects the velocity of the axis. It is used to optimize the dynamic precision of the stage. Normally the proper loop gain setting is found by observing the response of the axis to an abrupt change of the control value (step response). NanoCapture™ can display the step response of the axis as a graph.

6.2.2 Time Constant

The value of the time constant depends on the frequency of the first notch filter. It is therefore adjusted automatically whenever the frequency of the first notch filter is modified. The value of the time constant is calculated with the following formula:

$$TimeConstont = \frac{1}{4 \cdot \pi \cdot NotchFrequency}$$

If desired, the time constant can also be modified in the *Dynamic Tuner* window of NanoCapture™ explicitly.

6.2.3 Slew Rate

The maximum servo-loop slew rate depends on the size of the piezo actuators and the robustness of the attached mechanics. This parameter limits the maximum velocity at which the axis will be commanded to move from the current position to a new target (only in closed-loop operation).

In order to adjust the servo-loop slew rate setting with NanoCapture™ use the *Dynamic Tuner* window.

NOTES

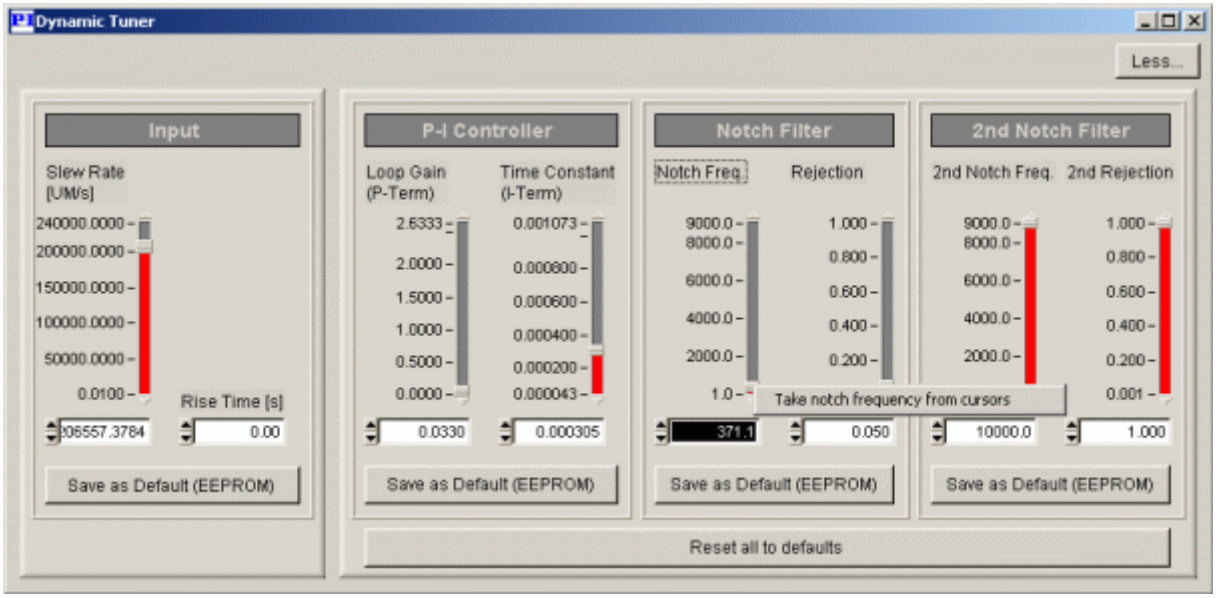
The servo-loop slew rate (parameter ID 0x07000200) is also changed with the VEL command (p. 104).

The slew rate setting will be effective only when velocity control mode is ON for the specified axis (VCO, p. 103).

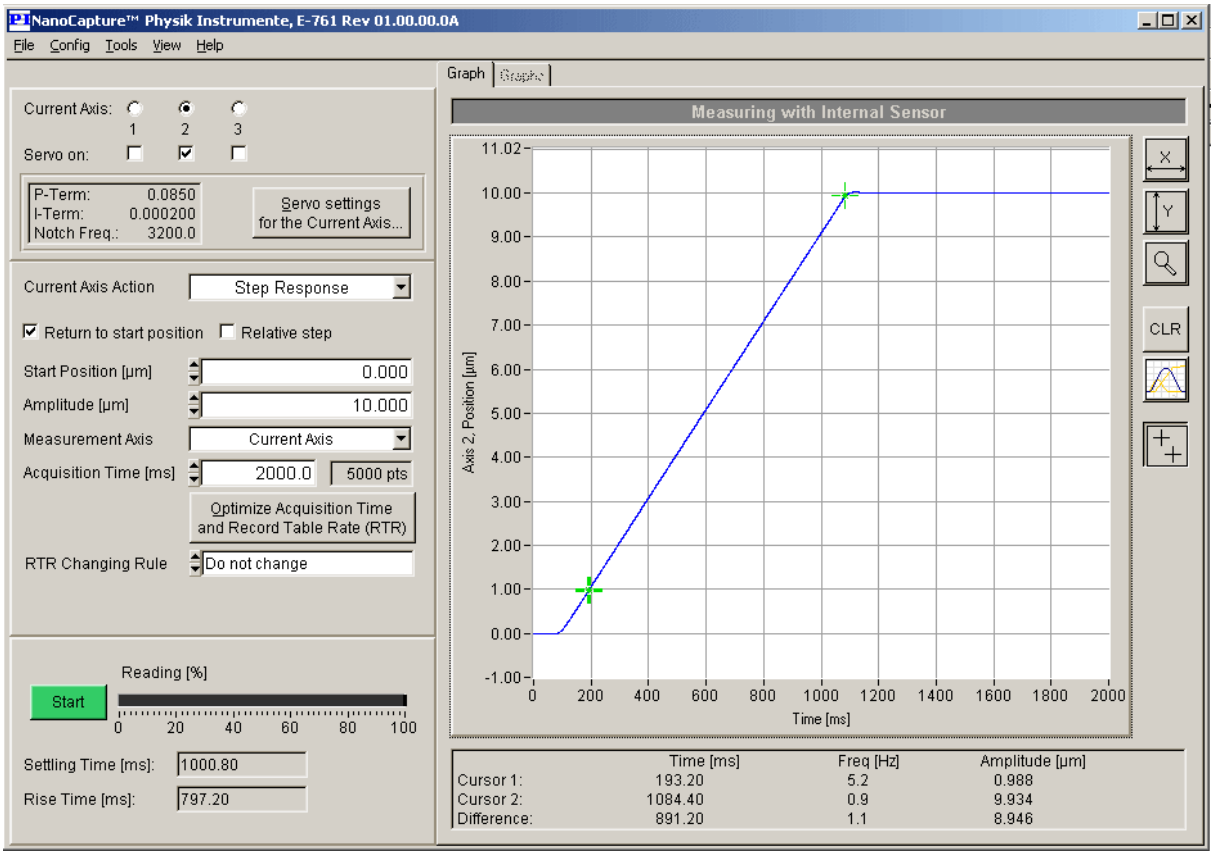
6.2.4 Adjustment Procedure

The servo parameters described above are optimized empirically by observing the effect of different values on a closed-loop Step Response. Proceed as follows with each axis you want to optimize:

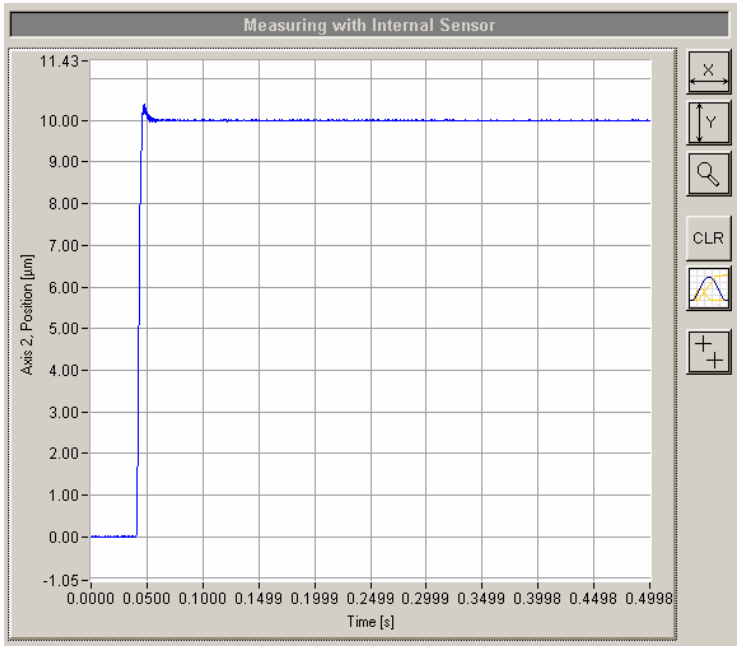
1. Use the *Current Axis* radio button(s) to select the axis to set
2. Set servo-control ON (check the *Servo ON* box) for that axis
3. Select *Step Response* in the *Current Axis Action* field
4. Set *Start Position* to 0 and *Amplitude* to about 15% of the axis travel range (if you activate the *Relative step* checkbox the step will start from the current position).
5. Set *Measurement Axis* to *Current Axis* (so that moved axis and measured axis are the same).
6. Open the *Dynamic Tuner* window for the axis by pressing F3 or using the *View → Dynamic Tuner* menu sequence. If necessary click *More* to see all the parameters (see figure below).



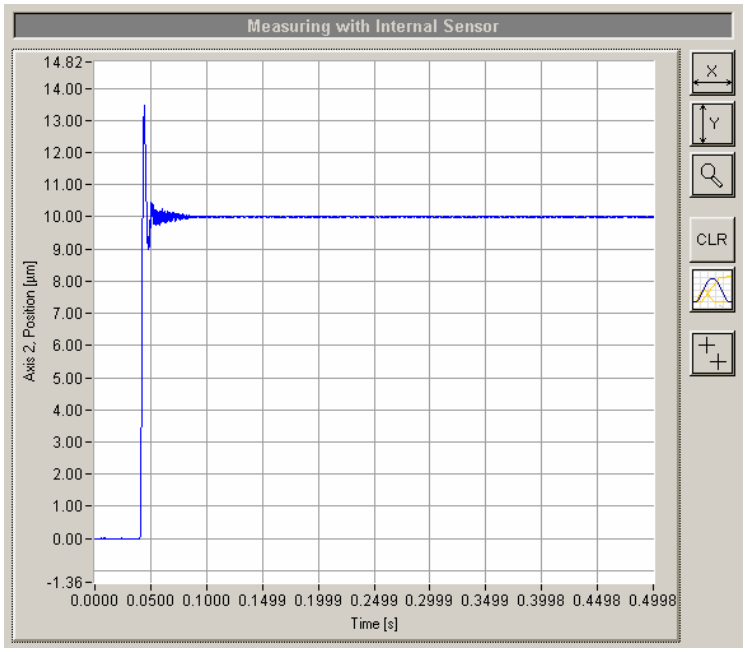
7. Change *Loop Gain (P-Term)*, *Slew Rate* (this is the servo-loop slew rate) and/or *Time Constant (I-Term)* slightly as desired.
8. Click *Start* in the main window to perform a step.
9. Observe the results, comparing with the examples shown in the figures below.
   - If the *loop gain* value is very low then the rise rate of the stage response curve will also very low:



- If the *loop gain* value is increased then the rise rate also rises, but the overshoot will also increase. The figure shows a step response with a small overshoot:



- If *loop gain* is further increased, then the rise rate will not rise significantly, but the overshoot will increase:



The stability of the servo-loop also depends on the *loop gain*. If the value is too high, then the controller will oscillate.

10. Return to the *Dynamic Tuner* window, refine the settings and repeat until satisfied.
11. Press *Save as Default (EEPROM)* for each of the changed parameter groups.

7 GCS Commands

7.1 Format

The firmware of the E-761 understands GCS commands (see Section 7.3 beginning on p. 74 for the alphabetical command reference). If you want to send GCS commands, respect the syntax described below. You can send the commands using the terminals in *NanoCapture*™ or *PZTControl*™ or a simple terminal program. Commands are transmitted as ASCII characters.

7.1.1 Notation

The following notation is used to define the GCS syntax in Section 7.1.2 and to describe the commands in Section 7.3 beginning on p. 74:

parameter		means an argument of the type "parameter" is used here. parameter is "AxisID" for a logical axis of the controller. Note that an axis is always identified with only one character.
[parameter]	means	parameter is optional. Between two arguments an optional space is always allowed.
{parameter}		means repetition of parameter.
|	means	or
LF	means	Linefeed
CR	means	Carriage return
::=	means	“is defined as”.
SP	means	a space character.

Command mnemonic:

<CMD> ::= <character1><character2><character3>[?]

7.1.2 GCS Syntax

A GCS command consists of 3 characters, e.g. ABC. To the corresponding question command a "?" is appended e.g. ABC?.

Exceptions:

- Fast polling commands which consist only of one character. The 24th ASCII character e.g. is called #24.
- *IDN? to support GPIB.

The syntax is not case-sensitive.

General:

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

Example:

Send: `MOV A10.0 B5.0 C20.0`

to move axis A to position 10.0 µm, axis B to 5.0 µm and axis C to 20.0 µm

Definitions for questioning commands:

ABC? { [<parameter>] }

When no <parameter> is given it means all possible <parameters> should be replied. POS? asks for all <axis>, DIO? asks for all <relays>.

The command ABC? <par3><par1><par2> replies in the same order:

<par3>"="<value3> SP LF

<par1>"="<value1> SP LF

<par2>"="<value2> LF

Reply syntax:

[<parameter>"="] <value> LF

Multiple line reply syntax:

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

[<parameter>"="] <value> LF for the last line!

Example:

Send: POS? A B

Report: 10.000000_(LF)
20.000000(LF)

Floating Point Format:

Some commands allow parameters in floating point format.

Format:

sv

sv.v

sv.vEsxx

where:

s	sign (positive values can be sent without sign)
v	integer parameter, will be converted to floating point by firmware
v.v	floating point parameter, be sure to use "." and not ","
E	exponent indicator
xx	2-digit exponent

The format used in reports is:

sv.vvvvvv

where:

s	sign (positive values are without sign)
v.vvvvvv	always with 6 characters after decimal point

If the report contains more than 2 floating point values, each will occupy one line.

7.2 Command Survey

The following table lists the commands according to the function groups "configuration", "status", "move" and "wave". For detailed descriptions see Section "Command Reference (alphabetical)" beginning on p. 74.

Mne-monic	Description	Type of Function	Page
Configuration: settings and queries			
***IDN?**	Get Device Identification String	Report	75
ATZ	AutoZero	Configuration	76
AVG	Set Number of Values for Averaging	Configuration	77
AVG?	Get Number of Values for Averaging	Report	77
CCT	Change Command Type	Configuration	77
CST	Change Assignment of Stages to Axes	Configuration	78
CST?	Get Assignment of Stages to Axes	Report	78
CSV?	Get Command Syntax Version	Report	78
DFH	Define Home	Configuration	79
DFH?	Get Home Positions	Report	79
HLP?	Help	Report	83
HPA?	List Parameter Help String	Report	84
NLM	Set Negative Position Soft Limit	Configuration	88
NLM?	Get Negative Position Soft Limit	Report	88
PLM	Set Positive Position Soft Limit	Configuration	89
PLM?	Get Positive Position Soft Limit	Report	90
RBT	Reboot System	Configuration	90
RPA	Reset Parameter	Configuration	91
RTR	Set Record Table Rate	Configuration	91
RTR?	Get Record Table Rate	Report	92
SAI	Set Axis Identifier	Configuration	92
SAI?	Get Axis Identifier	Report	92
SEP	Set Parameter in Non-Volatile Memory	Configuration	93
SEP?	Get Parameter from Non-Volatile Memory	Report	94
SPA	Set Parameter in Volatile Memory	Configuration	94
SPA?	Get Parameter from Volatile Memory	Report	95
SSN?	Get Serial Number	Report	95
SVO	Set Servo-Control Mode	Configuration	98
SVO?	Get Servo-Control Mode	Report	99
TIO?	Tell Number of Digital I/O Lines	Report	100
TMN?	Tell Minimum Travel Value	Report	101
TMX?	Tell Maximum Travel Value	Report	101

Mne-monic	Description	Type of Function	Page
TPC?	Tell Number of Piezo Channels	Report	102
TSC?	Tell Number of Sensor Channels	Report	102
TVI?	Tell Valid Axis Identifiers	Report	103
VCO	Set Velocity Control Mode	Configuration	103
VCO?	Get Velocity Control Mode	Report	104
VEL	Set Velocity	Configuration	104
VEL?	Set Velocity	Report	105
VER?	Get Version	Report	105
VMA	Set Upper PZT Voltage Soft Limit	Configuration	105
VMA?	Get Upper PZT Voltage Soft Limit	Report	106
VMI	Set Lower PZT Voltage Soft Limit	Configuration	106
VMI?	Get Lower PZT Voltage Soft Limit	Report	107
VST?	Get available Stages	Report	108
WPA	Write Parameter to Non-Volatile Memory	Configuration	118
Status			
#5	Request motion status	Report	74
#6	Ask for position change	Report	74
#7	Asks controller for ready status	Report	74
DIO?	Get State of the Digital Input Lines	Report	80
ERR?	Get Error	Report	81
ONT?	Get ON Target Status	Report	89
OVF?	Get Overflow Status	Report	89
POS?	Read Current Position of specified axis	Report	90
TAD?	Tell Current A/D Converter Value for specified sensor channel	Report	100
TAV?	Get Analog Input Voltage	Report	100
TNS?	Tell Normalized Sensor Value for specified sensor channel	Report	102
TSP?	Tell Current Sensor Position in Physical Unit for specified sensor channel	Report	103
Move: commands and appropriate queries			
#24	Stop all axes	Move	75
GOH	GO Home	Move	82
HLT	Halt	Move	83
IMP	Impulse Response	Move	84

Mne-monic	Description	Type of Function	Page
IMP?	Get Impulse Response Data	Report	85
MOV	MOVe absolute	Move	86
MOV?	Read Target Position	Report	86
MVR	Move Relative	Move	87
STE	Step Response	Move	95
STE?	Get Step Response Data	Report	96
STP	Stop Motion	Move	97
SVA	Set Open-Loop Axis Value for specified axis	Move	97
SVA?	Get Commanded Open-Loop Axis Value for specified axis	Report	98
SVR	Set Relative Open-Loop Axis Value for specified axis	Move	99
VOL	Set Current PZT Voltage for specified piezo channel	Move	107
VOL?	Read Current PZT Voltage for specified piezo channel	Move	107
Wave: wave generation, wave output and data recording			
#9	Get Wave Generator activation status	Wave	75
DRR?	Data Recorder Read	Report	80
GWD?	Query Waveform Type	Report	82
TNR?	Tell Number of Recorder Tables	Report	102
TWG?	Tell Number of Wave Generators	Report	103
WAV	Define Waveform	Wave	108
WAV?	Get Waveform Parameters	Report	113
WCL	Wave clear: Clears the content of the given wave table	Wave	113
WGC	Set Number of Wave Generator Output Cycles	Wave	114
WGC?	Get Number of Wave Generator Output Cycles	Wave	114
WGO	Start and Stop Wave Generator Output	Wave	114
WGO?	Get Wave Generator Output Settings	Wave	117
WGR	Start Recording	Wave	117
WMS	Get Maximum Number of Points per Wave Table	Wave	118

7.3 Command Reference (alphabetical)

For the notation used in the following descriptions and for the GCS syntax see Section 7 on p. 69.

#5 (Request motion status)

Command Type:	Report Command
Description:	Requests motion status of the configured axes. Note that when no stage is connected to an axis (NOSTAGE is returned by CST?), this axis is not included in the bit-coded answer!
Format:	#5 (single ASCII character number 5)
Arguments:	none
Response:	The answer <uint> is bit-coded: hexadecimal sum of the following codes for all axes which are moving: 1=axis 1 (is moving), 2=axis 2 (is moving), 4=axis 3 (is moving), etc.

#6 (Ask for position change)

Command Type:	Report Command
Description:	Ask for position change, answer structure similar to #5. Response is true if since last POS? a motion was commanded. If an axis did not move since the last POS? the answer is false. The answer is bit-coded in the same way like #5 for each axis. Note that when no stage is connected to an axis (NOSTAGE is returned by CST?), this axis is not included in the bit-coded answer!
Format:	#6 (single ASCII character number 6)
Arguments:	none
Response:	The answer <uint> is bit-coded: hexadecimal sum of the following codes for all axes which are moving: 1=axis 1 (position has changed), 2=axis 2 (position has changed), 4=axis 3 (position has changed), etc.

#7 (Asks controller for ready status)

Command Type:	Report Command
Description:	Asks controller for ready status (tests if controller is ready to perform a new command). Note: Use #5 instead of #7 to verify if motion has finished.
Format:	#7 (single ASCII character number 7)

Arguments:	none
Response:	Controller is ready: <uint> = ASCII: '±', Dec:177, Hex: B1 Controller is not ready: <uint> = ASCII: '°', Dec: 176, Hex: B0

#9 (Get Wave Generator activation status)

Command Type:	Report Command
Description:	Polls to determine whether a wave generator is running for any of the configured axes by sending the single ASCII character 9.
Format:	#9 (single ASCII character number 9)
Arguments:	none
Response:	The answer <uint> is bit-coded: hexadecimal sum of the following codes for all axes which are moving: 1=axis 1 (wave generator is running), 2=axis 2 (wave generator is running), c=axis 4 and 3 (wave generators are running), etc.

#24 (Stop all axes)

Command Type:	Move Command
Description:	Stops all motion abruptly. This includes motion of all axes (MOV, MVR, IMP, STE, GOH), wave generators, autozero motion (ATZ), piezo-channel-based motion (VOL). Single-character alias for STP (p. 97). #24 does not take effect to analog input which is used for "direct" axis control (see p. 34). To disable "direct" control for an axis, the value of the corresponding "Aux-Input to target factor" parameter (ID 0x06000902) must be set to 0 with SPA (p. 94). Sets error code to 10. Stops also data recording.
Format:	#24 (single ASCII character number 24)
Arguments:	none
Response:	none

*IDN? (Get Device Identification String) *IDN?

Command Type:	Report Command
Description:	Reports device identity number and firmware version. *IDN is a first choice to test the communication.
Format:	*IDN?
Arguments:	none
Response:	One-line string consisting of <manufacturer>, <model>, <serial_number>, <firmware_rev_level> terminated by LF.

Example: Physik Instrumente, E-761, 200512050003, V1.0.0.3

ATZ (AutoZero) ATZ

Command Type: Configuration Command

Description: Automatic zero-point calibration. All PZTs assigned to the specified axis are moved over a voltage range which is specified by two pre-set controller parameters (Autozero Driving LowVoltage, Autozero Driving HighVoltage). Afterwards the stage is moved back to the defined zero position. For a detailed description of the automatic zero-point calibration see p. 24.

This command can be interrupted by #24.

Notes:

ATZ works independent of servo mode.

AutoZero is not effective on non-linear axes (rotation axes).

Just after execution the current position is 0.

Be aware that the result of the AutoZero procedure (new offset value) is automatically written to non-volatile memory (EEPROM).

For stages with ID-chip the option "Read ID-Chip always" must be disabled by default to make the AutoZero result (new offset value) available in the future. See p. 35 for details.

Format: ATZ {AxisID LowVoltage}

Arguments: AxisID: axis identifier

LowVoltage: A voltage value x.x and NaN are used alternatively:
If x.x (in volts) is given, the zero position is set at this voltage.
If no voltage is given, NaN (not a number) must be entered and the internally saved parameter for the voltage at the zero position is used.

Response: “1” when AutoZero was successful, “0” when not successful.

Problem Solver: ATZ returns "0" (not successful), when

invalid axis identifier was used, e.g.
ATZ 9 NAN

the axis is not configured, e.g.
CST 2 NOSTAGE
ATZ 2 NAN

the wave generator is running for the axis, e.g.
WGO 1 1
ATZ 1 NAN

ADC >= 32765 or ADC <= -32765

AVG (Set Number of Values for Averaging) AVG

Command Type:	Configuration Command
Description:	Sets the number of ADC values to be used for an average (oversampling factor). Larger values mean averaging over more values so fluctuations are damped and the sensor and servo sample rates are reduced. That means that a higher value increases the values for "Sensor sampling time" (ID 0x0e000100) and "Servo update time" (ID 0x0e000200) while the original ADC sampling frequency will remain unchanged. The value set is valid for the whole system. The values influenced by this command are saved in RAM only. To save the currently valid values to non-volatile memory, where they become the power-on defaults, you must use WPA (p. 118). Changes not saved with WPA will be lost when the controller is powered off.
Format:	AVG OverSampling
Arguments:	OverSampling: Can be 4, 8, 16. Default is 4.
Response:	none
Problem Solver:	Invalid oversampling factor.

AVG? (Get Number of Values for Averaging) AVG?

Command Type:	Report Command
Description:	This command reports the current number of ADC values to be used for an average (oversampling factor).
Format:	AVG?
Response:	Oversampling factor. Can be 4, 8, 16.

CCT (Change Command Type) CCT

Command Type:	Configuration Command
Description:	Switch between binary and ASCII protocol. The command type is changed immediately after the command was sent. Notes: After power-up the controller communicates via ASCII protocol. In binary command mode, command mode can only be changed back to ASCII mode by binary command.
Format:	CCT CommandType
Arguments:	CommandType: is the command type to use, 1 = binary, 2 = string
Response:	none
Problem Solver:	Value other than 1 or 2.

CST (Change Assignment of Stages to Axes) CST

Command Type:	Configuration Command
Description:	Axis configuration. When a stage is connected, the appropriate axis must be configured with "ID-STAGE" for stage name. To tell the controller that there is no stage connected to an axis use "NOSTAGE" as stage name. Notes: The configuration made with CST is automatically saved to the non-volatile memory (EEPROM). The axis configuration as "ID-Stage" is required before you can address any move command to this axis (i.e. to the connected stage). When you add or replace stages and configure the axes with CST, the ID-chips of the connected stages are not read by the controller yet. To read the ID-chip data, the E-761 must be rebooted (using RBT (p. 90) or by restarting the PC).
Format:	CST {AxisID StageName}
Arguments:	AxisID: axis identifier StageName: "ID-STAGE for configured axes (a stage should be connected), "NOSTAGE" for non-configured axes (no stage should be connected)
Response:	none
Problem Solver:	Unknown stage name.

CST? (Get Assignment of Stages to Axes) CST?

Command Type:	Report Command
Description:	Get axis configuration, i.e. returns the stage names for queried axes. Note that CST? returns all axes—configured and non-configured.
Format:	CST? {AxisID}
Arguments:	AxisID: axis identifier
Response:	"ID-STAGE for configured axes, "NOSTAGE" for non-configured axes

CSV? (Get Command Syntax Version) CSV?

Command Type:	Report Command
Description:	Ask current GCS syntax version used in the firmware.
Format:	CSV?
Arguments:	none
Response:	1.0 for GCS 1.0, 2.0 for GCS 2.0

DFH (Define Home) DFH

Command Type:	Configuration Command
Description:	Makes the current position the new home position (by setting the current position value to 0.00 and changing the "User Origin" parameter value, ID 0x07010200). Notes: Due to the change of the home position the limits of the travel range are changed accordingly ("Range min limit", ID 0x07000000, and "Range max limit", ID 0x07000001), and also the settings made with NLM (p. 88) and PLM (p. 89) are adapted. The home position is reset to default in the following cases: • by ATZ (p. 76) • on power-on, when the "Read ID-Chip always" parameter is enabled The home position is adapted accordingly when modifying the parameters 0x07000000 and 0x07000001. This command saves the new values in RAM only. To save the currently valid values to flash ROM, where they become the power-on defaults, you must use WPA (p. 118). Changes not saved with WPA will be lost when the controller is powered off.
Format:	DFH [{AxisID}]
Arguments:	AxisID: axis identifier
Response:	none
Troubleshooting:	Illegal axis identifier

DFH? (Get Home Positions) DFH?

Command Type:	Report Command
Description:	Gets home position
Format:	DFH? [{AxisID}]
Arguments:	AxisID: axis identifier
Response:	1=float1 2=float2 3=float3 float*n* is the current physical home position
Troubleshooting:	Illegal axis identifier

DIO? (Get State of the Digital Input Lines) DIO?

Command Type: Report Command

Description: Returns the states of the specified digital input lines. Note that the E-761 has no genuine digital input lines, but the analog input is internally interpreted as digital input for triggering tasks (see Section 2.9 on p. 33), and its signal state can be queried by this command. If the voltage on the analog input is < 0.8 V, the signal is interpreted as LOW, if the voltage is ≥ 2.4 V, the signal is interpreted as HIGH.

Format: DIO?[1]

Arguments: 1: input line specification. The E-761 has only one input line (the analog input)

Response: 1=1: if input line is high
1=0: if input line is low

DRR? (Data Recorder Read) DRR?

Command Type: Report Command

Description: Reading of the last recorded data set.

Notes:

Recording takes place for all recorder tables in the following cases:

- the wave generator is running for an arbitrary axis; recording starts either automatically when a wave generator is started with WGO (p. 114) or can be started "manually" with WGR (p. 117)
- an impulse is started with IMP (p. 84) or a step is started with STE (p. 95)

The E-761 has 4 data recorder tables (ask with TNR?, p. 102). The maximum number of data points is 8192 per recorder table. For the assignment of axis and data sources to the recorder tables see below.

Immediately after recording is started, the record table which is currently filled may still contain some data from the previous run. Within a period of
time [ms] = oversampling factor * wave length [no. of points] * 10^{-2}
you should therefore not query the data to avoid display errors (the oversampling factor can be queried using AVG?, p. 77, the wave length depends on the wave definition with WAV, p. 108).

The recorded data is influenced by the setting for the record table rate (RTR, p. 91).

Format: DRR? [StartPoint[NumberOfPoints [{RecTableID}]]]

Arguments: StartPoint: is the index of first data point

NumberOfPoints: is the number of data points

RecTableID: is the data recorder table; the axis and source assignment is as follows:

table 1: axis 1 actual position

table 2: axis 2 actual position

table 3: axis 3 actual position

table 4: analog input voltage (same value as read with TAV? (p. 100), i.e. contains gain and offset for the analog input, see p. 33)

Response: GCS Array Format, see the separate manual for the GCS array, SM 146E

Example: The wave generator runs for axis 2, axis 1 and 3 are not calibrated and no stage is connected, the voltage on the analog input is 2.8 V. If DRR? is sent for the first 10 points in all recorder tables, the answer is as follows:

```
DRR? 0 10
# REM E-761
#
# TYPE = 1
#
# SEPARATOR = 32
# DIM = 4
# SAMPLE_TIME = 3.99999840e-5
# NDATA = 10
# NAME0 = Actual Position
# NAME1 = Actual Position
# NAME2 = Actual Position
# NAME3 = Aux-input voltage
# END_HEADER
-0008.5990 +0003.0001 -0001.6997 +0002.7957
-0008.5990 +0003.0001 -0001.6997 +0002.7960
-0008.5990 +0003.0001 -0001.6997 +0002.7960
-0008.5990 +0003.0001 -0001.6997 +0002.7957
-0008.5990 +0003.0001 -0001.6997 +0002.7960
-0008.5990 +0002.9993 -0001.6997 +0002.7960
-0008.5990 +0002.9978 -0001.6997 +0002.7960
```

ERR? (Get Error) ERR?

Command Type: Report Command

Description: Get error code of the last occurred error of controller and reset the error to 0.

Only the last error is buffered. Therefore you should call ERR? after each command to check whether it was carried out correctly.

The error codes and their description are fully listed in Section 7.4 on p. 119.

The following table shows a selection of possible controller errors:

0	No error
1	Parameter syntax error
2	Unknown command
5	Unallowable move attempted on unreferenced axis, or move attempted with servo off
7	Position out of limits
8	Velocity out of limits
10	Controller was stopped by command
15	Invalid axis identifier

17	Parameter out of range
22	Axis identifier specified more than once
23	Illegal axis
24	Incorrect number of parameters
26	Parameter missing
27	Soft limit out of range
34	Command not allowed for selected stage(s)
54	Unknown parameter
302	Voltage out of limits
303	Open-loop motion attempted when servo ON

Format: ERR?

Arguments: none

Response: Error number

GOH (GO Home) GOH

Command Type: Move Command

Description: Moves the axis to its home position (is equivalent to moving the axis to position 0 using MOV)

Format: GOH [{AxisID}]

Arguments: AxisID: axis identifier

Response: none

Troubleshooting: Illegal axis identifier

GWD? (Query Waveform Type) GWD?

Command Type: Report Command

Description: Query waveform shape for given wave table.

Note the following fact which affects only the response to the GDW? query and not the waveform output by the wave generator:

The content of a wave table is not completely erased when a new waveform is written to this table. Only the number of points given by the new waveform is written beginning with the first point in the table, but any subsequent data points will keep the old values from the former waveform. You can query the number of points belonging to the current valid waveform using WAV? (p. 113).

Format: GWD? WaveTableID StartPoint NumberOfPoints

Arguments: WaveTableID: is one wave table of the controller

StartPoint: is the index of first data point, starts with 0

NumberOfPoints: is the number of data points

Response: waveform string (values for the queried points)

HLP? (Help) HLP?

Command Type:	Report Command
Description:	Lists all available commands
Format:	HLP?
Arguments:	none
Response:	Command list

HLT (Halt) HLT

Command Type:	Move Command
Description:	Halt the motion of given axes smoothly. Only commands causing non-complex motion (e.g. MOV, GOH, SVR) can be interrupted with HLT. Error code 10 is set. After the stage was stopped, in closed-loop operation the target position is set to the current position. HLT stops motion with given system deceleration with regard to system inertia. STP (p. 97) and #24 (p. 75) in contrast abort current motion as fast as possible for the controller without taking care of systems inertia or oscillations. HLT does not take effect to analog input which is used for "direct" axis control (see p. 34). To disable "direct" control for an axis, the value of the corresponding "Aux-Input to target factor" parameter (ID 0x06000902) must be set to 0 with SPA (p. 94).
Format:	HLT [{AxisID}]
Parameters:	AxisID: axis identifier
Response:	none
Troubleshooting:	Illegal axis identifier

HPA? (List Parameter Help String) HPA?

Command Type: Report Command

Description: Lists a help string which contains the controller parameters with short descriptions.

Note:

The listed parameters can be changed and/or saved using the following commands:

SPA (p. 94) affects the parameter settings in the volatile memory (RAM).

WPA (p. 118) writes parameters settings from the RAM to the non-volatile memory (EEPROM).

SEP (p. 93) writes parameter settings directly into the EEPROM (without changing the RAM settings).

RPA (p. 91) resets the RAM to the EEPROM values

See "Parameter Overview" on p. 134 for more information regarding the controller parameters and their handling.

Format: HPA?

Arguments: none

Response: A string with a list of parameters changeable by the user. The string has following format:

string = {ParameterId “=” TAB FunctionGroupDescription TAB ParameterDescription [{TAB PossibleValue “=” ValueDescription}] }

IMP (Impulse Response) IMP

Command Type: Move Command

Description: Starts performing an impulse and recording the impulse response for the given axis (actual position values).

Notes:

An "impulse" consists of a relative move of the specified amplitude followed by an equal relative move in the opposite direction. Irrespective of the current operating mode (servo on or off), the impulse is performed relative to the current position.

Before you start IMP the wave generator output should be stopped (see WGO, p. 114). For highest speed, VCO (p. 103) should also be OFF.

Command execution is finished when 8192 data points were saved. The required time can be calculated as follows:

time = 8192 * SamplingTime * Oversampling Factor * Record Table Rate

where
SamplingTime is the sampling time of the ADC (0.01 ms for E-761)
Oversampling Factor is the factor set with AVG (p. 77) which influences the actual servo sampling time (without changing the ADC sampling time), default is 4
Record Table Rate is the number of servo-loop cycles to be used in data recording operations; is set with RTR (p. 91).

The recorded data can be read with DRR? (p. 80) or IMP? (p. 85).

Format: IMP AxisID PulseSize [NumberOfServos]

Arguments: AxisID: axis identifier

PulseSize: is the height of the impulse. In closed-loop operation (servo ON), the given amplitude is interpreted as relative position value in either case. In open-loop operation (servo OFF), with the default settings of the axis-to-PZT matrix, the amplitude also corresponds numerically to a relative axis position (see "Output Generation" on p. 46 for more information).

NumberOfServos: is the width of the impulse, in number of servo cycles

Response: none

Troubleshooting: Illegal axis identifier

IMP? (Get Impulse Response Data) IMP

Command Type: Report Command

Description: Get the recorded positions of an impulse response made with IMP (see above).

Format: IMP AxisID StartPoint NumberOfValues

Arguments: AxisID: axis identifier

StartPoint: is the index of first data point

NumberOfValues: is the number of data points

Response: string of values for the queried points

Troubleshooting: Illegal axis identifier

MOV (MOVe absolute) MOV

Command Type:	Move Command
Description:	Set new absolute target position for given axis. Axes will start moving to the new positions if ALL given targets are within the allowed ranges and ALL axes can move. All axes start moving simultaneously. This command can be interrupted by #24 (p. 75), STP (p. 97) and HLT (p. 83). Servo must be enabled for all commanded axes prior to using this command.
Format:	MOV {AxisID Position}
Arguments:	AxisID: axis identifiers Position: target position in physical units (e.g. [µm])
Response:	none
Problem Solver:	Target position out of limits Illegal axis identifier Servo is Off for one given axis
Example 1:	Axis A moves to 10, axis B moves to 20 and axis 3 moves to -30.5, all target positions are in the physical unit valid for the appropriate axis: Send: MOV A 10 B 20 3 -30.5
Example 2:	The axis does not move. The error code "7" replied by the ERR? command indicates that the target position given by the move command is out of limits: Send: MOV A 243 Send: ERR? Receive: 7

MOV? (Read Target Position) MOV?

Command Type:	Report Command
Description:	Returns last valid commanded target position. Notes: The target position is changed by all commands that cause motion (MOV, MVR, STE …), by the wave generator (no matter if the internal wave generators or the analog input is used for target generation), by the analog input ("direct" axis control) and by `E761_SetDirectTarget` function of the GCS library (direct access to E-761 RAM). MOV? gets the commanded positions. Use POS? to get the current actual positions.
Format:	MOV? [{AxisID}]
Arguments:	AxisID: axis identifier
Response:	Last commanded target position(s) in physical units.
Problem Solver:	Illegal axis identifier

MVR (Move Relative) MVR

Command Type:	Move Command
Description:	Move given axes relative to their current position. The new target position is calculated by adding the given Position value to the last commanded target value. Axes will start moving to the new position if ALL given targets are within the allowed range and ALL axes can move. This command can be interrupted by #24, HLT and STP. Servo must be enabled prior to using this command.
Format:	MVR {AxisID Position}
Arguments:	AxisID: axis identifiers Position: This value is added to the last commanded target position and the sum is set as new target position in physical units.
Response:	none
Troubleshooting:	Target position out of limits Illegal axis identifier Servo is Off for one given axis
Example:	First an absolute motion is commanded, then a relative motion. The target position of axis B for the second relative move exceeds the motion range of axis B. Therefore the command is ignored. Send: MOV A -0.5 B 12.3 Send: POS? A B Receive: A=-0.500000 B=12.300000 Send: MVR A 1 B 2 Send: POS? A B Receive: A=0.500000 B=14.300000 Send: MVR A 1 B 2000 Send: POS? A B Receive: A=0.500000 B=14.300000

NLM (Set Negative Position Soft Limit) NLM

Command Type:	Configuration Command
Description:	Set negative position limit ("soft limit") in physical unit for given axis. Before executing this command a range check is done—the LowLimit value must not be smaller than the value of the "Range min limit" parameter (parameter ID is 0x07000000) which can be queried using SPA? (p. 95). Settings made with NLM are only valid for closed-loop operation (servo ON). This command saves the new value in RAM only. To save the currently valid value to flash ROM, where it becomes the power-on default, you must use WPA (p. 118) without any parameter. Changes not saved with WPA will be lost when the controller is powered off.
Format:	NLM {AxisID LowLimit}
Arguments:	AxisID: axis identifier LowLimit: is the value of the negative position limit in physical units
Response:	none
Problem Solver:	Illegal axis identifier LowLimit value is out of the allowed position range of the axis

NLM? (Get Negative Position Soft Limit) NLM?

Command Type:	Report Command
Description:	Get negative position limit ("soft limit") of given axis.
Format:	NLM? [{AxisID}]
Arguments:	AxisID: axis identifier
Response:	The value of the negative soft limit in physical units.
Problem Solver:	Illegal axis identifier

ONT? (Get ON Target Status) ONT?

Command Type:	Report Command
Description:	Get ON target status of given axis. The axis is on target when the current position reaches a certain settle window around the target position in closed-loop operation. The size of the settle window for an axis depends on the "Tolerance" parameter (parameter ID 0x07000900). This parameter can be changed using SPA (p. 94) or SEP (p. 93).
Format:	ONT? [{AxisID}]
Arguments:	AxisID: axis identifier
Response:	“1” when the specified axis is on target, “0” otherwise.
Troubleshooting:	Illegal axis identifier

OVF? (Get Overflow Status) OVF?

Command Type:	Report Command
Description:	Get overflow status of given axes. Overflow means that the control variables are out of range (can only happen if controller is in closed-loop mode).
Format:	OVF? [{AxisID}]
Arguments:	AxisID: axis identifier.
Response:	Is 0 (axis is not in overflow) or 1 (axis is in overflow).
Problem Solver:	Illegal axis identifier

PLM (Set Positive Position Soft Limit) PLM

Command Type:	Configuration Command
Description:	Set positive position limit ("soft limit") in physical unit for given axis. Before executing this command a range check is done—the HighLimit value must not be greater than the value of the "Range max limit" parameter (parameter ID is 0x07000001) which can be queried using SPA? (p. 95). Settings made with PLM are only valid for closed-loop operation (servo ON). This command saves the new value in RAM only. To save the currently valid value to flash ROM, where it becomes the power-on default, you must use WPA (p. 118) without any parameter. Changes not saved with WPA will be lost when the controller is powered off.
Format:	PLM {AxisID HighLimit}
Arguments:	AxisID: axis identifier HighLimit: is the value of the positive position limit in physical units
Response:	none

Problem Solver:	Illegal axis identifier HighLimit value is out of the allowed position range of the axis

PLM? (Get Positive Position Soft Limit) PLM?

Command Type:	Report Command
Description:	Get positive position limit ("soft limit") of given axis.
Format:	PLM? [{AxisID}]
Arguments:	AxisID: axis identifier
Response:	The value of the positive soft limit in physical units.
Problem Solver:	Illegal axis identifier

POS? (Read Current Position) POS?

Command Type:	Report Command
Description:	Reports the current position of the specified axis. Notes: Use TSP? (p. 103) to get the position of an individual sensor channel in physical units.
Format:	POS? [{AxisID}]
Arguments:	AxisID: axis identifier.
Response:	The current axis position in physical units
Problem Solver:	Illegal axis identifier

RBT (Reboot System) RBT

Command Type:	Configuration Command
Description:	Reboot system. Controller behaves like after a cold start. Note: When the connection to the board fails (i.e. no communication is possible and RBT therefore can not be sent), start *NanoCapture*™ and press the *Reboot* button in the *Device Connection* window (see Fig. 4 on p. 22).
Format:	RBT
Arguments:	none
Response:	none
Problem Solver:	

RPA (Reset Parameter) RPA

Command Type:	Configuration Command
Description:	Resets the given parameter of the given item which may be an axis, a sensor or piezo channel or the whole system. The value of the EEPROM is written into the RAM. Note: With HPA? (p. 84) you can obtain a list of the parameters which are changeable by the user. SPA (p. 94) affects the parameter settings in the RAM, WPA (p. 118) writes parameters settings from the RAM to the EPROM, and SEP (p. 93) writes parameter settings directly into the EEPROM (without changing the RAM settings).
Format:	RPA {ItemID ParameterID}
Arguments:	ItemID: is an axis identifier, a sensor or piezo channel or the whole system, the item type depends on the parameter ParameterID: is a parameter ID
Response:	none
Problem Solver:	Invalid parameter ID Invalid item ID

RTR (Set Record Table Rate) RTR

Command Type:	Configuration Command
Description:	Sets the record table rate, i.e. the number of servo-loop cycles to be used in data recording operations ("Table rate", ID 0x16000000). Settings larger than 1 make it possible to cover longer time periods with a limited number of points. Note: The servo update time and hence the RTR setting is influenced by the oversampling factor set with AVG (p. 77). The value set is saved in RAM only. To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA (p. 118). Changes not saved with WPA will be lost when the controller is powered off.
Format:	RTR TableRate
Arguments:	TableRate: is the record table rate to be used (unit: number of servo-loop cycles), must be larger than zero
Response:	none

RTR? (Get Record Table Rate) RTR?

Command Type:	Report Command
Description:	Gets the record table rate, i.e. the number of servo-loop cycles used in data recording operations.
Format:	RTR?
Arguments:	none
Response:	The record table rate (number of servo-loop cycles).

SAI (Set Axis Identifier) SAI

Command Type:	Configuration Command
Description:	Set old axis identifier to new identifier. Notes: The configuration made with SAI is automatically saved to the non-volatile memory (EEPROM). An axis is always identified with only one alphanumeric character. Do not mistake the axis identifiers set with SAI with the Axis name parameter (ID 0x07000600) which is only used in the graphical user interface of NanoCapture™. You can change only the identifiers of axes which were configured with CST (p. 78).
Format:	SAI {AxisID_Old AxisID_New}
Arguments:	AxisID_Old: is one axis identifier currently used by the controller. AxisID_New: is the new identifier for the axis.
Response:	none
Troubleshooting:	Illegal axis identifier or duplicate axis identifier

SAI? (Get Axis Identifier) SAI?

Command Type:	Report Command
Description:	Get axis identifier.
Format:	SAI? [All]
Arguments:	All: when used, all axes are returned—configured and non-configured.
Response:	Axis identifiers. When All is used, all axes are returned—configured and non-configured. When All is not used, only the axes configured with CST (p. 78) are returned (if no axis is configured an empty string is returned).
Example:	Send: SAI? Receive: 321

SEP (Set Parameter in Non-Volatile Memory) SEP

Command Type:	Configuration Command
Description:	Set a parameter of a given item (axis, sensor or piezo channel, whole system) to a different value in non-volatile memory (EEPROM). CAUTION! This command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware! Notes: With HPA? (p. 84) you can obtain a list of the parameters which are changeable by the user. SEP sets the same parameters as SPA (p. 94), but SPA writes them only to volatile memory (RAM), while SEP only writes to non-volatile memory. After parameters were set with SEP, use RPA (p. 91) to activate them (write them to volatile memory), or they become active after next reboot. SEP saves only one parameter each time. See "Parameter Overview" on p. 134 for more information regarding the controller parameters and their handling.
Format:	SEP Password {ItemID ParameterID Value}
Arguments:	Password: the default password to write to EEPROM is "100" ItemID: is an axis identifier, a sensor or piezo channel or the whole system, the item type depends on the parameter ParameterID: is a parameter ID Value: is the value to which the given parameter of the given item is set
Response:	none
Problem Solver:	Invalid parameter ID Invalid item ID Invalid password

SEP? (Get Parameter from Non-Volatile Memory) SEP?

Command Type:	Report Command
Description:	Get the value of a parameter of a given item (axis, channel, whole system) from non-volatile memory (EEPROM). Notes: With HPA? (p. 84) you can obtain a list of the parameters IDs with short descriptions. SEP? only shows the parameters which can be changed by the user. Use the *Device Parameter Configuration* window of *NanoCapture*™ instead to see also the values of "protected" parameters (they have the value "2" in the *CCL* column). See "Parameter Overview" on p. 134 for more information regarding the controller parameters and their handling.
Format:	SEP? [{ItemID ParameterID}]
Arguments:	ItemID: is an axis identifier, a sensor or piezo channel or the whole system, the item type depends on the parameter ParameterID: is a parameter ID
Response:	The value of the given parameter for the given item.
Problem Solver:	Invalid parameter ID Invalid item ID

SPA (Set Parameter in Volatile Memory) SPA

Command Type:	Configuration Command
Description:	Set a parameter of a given item (axis, sensor or piezo channel, whole system) to a different value in volatile memory (RAM). CAUTION! This command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware! Notes: With HPA? (p. 84) you can obtain a list of the parameters which are changeable by the user. SPA sets the same parameters as SEP (p. 93), but SEP writes them directly to non-volatile memory (EEPROM), while SPA only writes to volatile memory. After parameters were set with SPA, they can be written to the EEPROM using WPA (p. 118). See "Parameter Overview" on p. 134 for more information regarding the controller parameters and their handling.
Format:	SPA {ItemID ParameterID Value}
Arguments:	ItemID: is an axis identifier, a sensor or piezo channel or the whole system, the item type depends on the parameter ParameterID: is a parameter ID Value: is the value to which the given parameter of the given item is set

Response:	none
Problem Solver:	Invalid parameter ID
	Invalid item ID

SPA? (Get Parameter from Volatile Memory) SPA?

Command Type:	Report Command
Description:	Get the value of a parameter of a given item (axis, channel, whole system) from volatile memory (RAM).
	Notes:
	With HPA? (p. 84) you can obtain a list of the parameters IDs with short descriptions.
	SPA? only shows the parameters which can be changed by the user. Use the *Device Parameter Configuration* window of *NanoCapture*™ instead to see also the values of "protected" parameters (they have the value "2" in the *CCL* column).
	See "Parameter Overview" on p. 134 for more information regarding the controller parameters and their handling.
Format:	SPA? [{ItemID ParameterID}]
Arguments:	ItemID: is an axis identifier, a sensor or piezo channel or the whole system, the item type depends on the parameter
	ParameterID: is a parameter ID
Response:	The value of the given parameter for the given item.
Problem Solver:	Invalid parameter ID
	Invalid item ID

SSN? (Get Serial Number) SSN?

Command Type:	Report Command
Description:	Get the serial number of the controller.
Format:	SSN?
Arguments:	none
Response:	The serial number of the controller, e.g. 200512050003

STE (Step Response) STE

Command Type:	Move Command
Description:	Starts performing a step and recording the step response for the given axis (actual position values).
	Notes:
	A "step" consists of a relative move of the specified amplitude. Irrespective of the current operating mode (servo on or off), the step is performed relative to the current position.
	Before you start STE the wave generator output should be stopped (see WGO, p. 114). For highest speed, VCO (p. 103) should also be OFF.

Command execution is finished when 8192 data points were saved. The required time can be calculated as follows:

time = 8192 * SamplingTime * Oversampling Factor * Record Table Rate

where
SamplingTime is the sampling time of the ADC (10 us for E-761)
Oversampling Factor is the factor set with AVG (p. 77) which influences the actual servo sampling time (without changing the ADC sampling time), default is 4
Record Table Rate is the number of servo-loop cycles to be used in data recording operations; is set with RTR (p. 91).

The recorded data can be read with DRR? (p. 80) or STE? (p. 96).

Format: STE AxisID StepSize

Arguments: AxisID: axis identifier

StepSize: is the height of the step. In closed-loop operation (servo ON), the given amplitude is interpreted as relative position value in either case. In open-loop operation (servo OFF), with the default settings of the axis-to-PZT matrix, the amplitude also corresponds numerically to a relative axis position (see "Output Generation" on p. 46 for more information).

Response: none

Troubleshooting: Illegal axis identifier

STE? (Get Step Response Data) STE?

Command Type: Report Command

Description: Get the recorded positions of a step response made with STE (see above).

Format: STE AxisID StartPoint NumberOfValues

Arguments: AxisID: axis identifier

StartPoint: is the index of first data point

NumberOfValues: is the number of data points

Response: string of values for the queried points

Troubleshooting: Illegal axis identifier

STP (Stop Motion) STP

Command Type:	Move Command
Description:	Stops the motion of the axes instantaneously. Only commands causing non-complex motion (e.g. MOV, GOH, SVR) can be interrupted with STP. Error code 10 is set. After the stage was stopped, in closed-loop operation the target position is set to the current position. STP always does a hard stop (aborts current motion as fast as possible for the controller without taking care of systems inertia or oscillations). Use HLT (p. 83) instead to stop axis smoothly. Use #24 (p. 75) to stop complex motions like autozero (ATZ), wave generators, piezo-channel-based motion (VOL), etc. STP does not take effect to analog input which is used for "direct" axis control (see p. 34). To disable "direct" control for an axis, the value of the corresponding "Aux-Input to target factor" parameter (ID 0x06000902) must be set to 0 with SPA (p. 94).
Format:	STP [{AxisID}]
Arguments:	AxisID: axis identifier
Response:	none
Troubleshooting:	Illegal axis identifier

SVA (Set Open-Loop Axis Value) SVA

Command Type:	Move Command
Description:	Set absolute open-loop control value to move the axis. This command can be interrupted by #24 and STP. Servo must be switched off when using this command. Notes: Multiple PZT amplifiers ("output channels") can be involved in the motion of one logical axis (see Section "Principle of Operation" on p. 40). Use VOL (p. 107) to set the voltage for a single amplifier. If the PZT control voltage resulting from the commanded open-loop control value exceeds the voltage limit of one of the PZT amplifiers which participate in this axis (see VMA, p. 105, and VMI, p. 106), then the command is not executed (check with ERR? (p. 81)). If the stage velocity needs to be limited, velocity control mode must have been set to ON (VCO, p. 103) and the desired velocity must have been specified (VEL, p. 104). If velocity control is not desired, velocity control mode should be set to OFF.
Format:	SVA {AxisID Amplitude}
Arguments:	AxisID: axis identifier Amplitude: is the new absolute open-loop control value (dimensionless) The interpretation of the amplitude value depends on the settings of the axis-to-PZT matrix (see "Output Generation" on p. 46 for

more information). With the default matrix coefficients, open-loop control values numerically correspond to axis position values.

Response: none

Problem Solver: Illegal axis identifier

Axis is in Servo-On mode

PZT voltage out of limits

The axis has Velocity Control On, but the specified velocity was too small.

SVA? (Get Commanded Open-Loop Axis Value) SVA?

Command Type: Report Command

Description: Reports the commanded open-loop control value of the specified axis.

Format: SVA? [{AxisID}]

Arguments: AxisID: axis identifier

Response: The last commanded open-loop control value of the specified axis.

Problem Solver: Illegal axis identifier

SVO (Set Servo-Control Mode) SVO

Command Type: Configuration Command

Description: Sets servo-control mode on or off for given axes.

When the servo is switched on, the target position is set to the current position. This avoids jumps when servo-control starts.

The current servo state affects the applicable move commands:

- servo-control off: use SVA (p. 97) and SVR (p. 99)
- servo-control on: use MOV (p. 86) and MVR (p. 87)

If servo is switched off while stage is moving the stage stops. Exception: When the analog input is being used for "direct" axis control (p. 34) and servo is switched off, the axis motion will continue in open-loop mode.

Servo status can not be changed while a wave generator is running for the axis.

Format: SVO {AxisID 1|0}

Parameters: AxisID: Axis identifier.

1 or 0;
1 means servo-control mode ON (closed-loop operation);
0 means servo-control mode OFF (open-loop operation).

Response: none

Problem Solver: Illegal axis identifier

SVO? (Get Servo-Control Mode) SVO?

Command Type:	Report Command
Description:	Reports the servo-control mode of the specified axis.
Format:	SVO? [{AxisID}]
Arguments:	AxisID: Axis identifier.
Response:	“1” when the servo-control mode of the specified axis is ON (closed-loop operation), “0” when OFF (open-loop operation).
Problem Solver:	Illegal axis identifier

SVR (Set Relative Open-Loop Axis Value) SVR

Command Type:	Move Command
Description:	Set open-loop control value relative to the current open-loop control value to move the axis. The new open-loop control value is calculated by adding the given value <Difference> to the last commanded open-loop control value. This command can be interrupted by #24 and STP. Servo must be switched off when using this command. Notes: If the PZT control voltage resulting from the commanded open-loop control value exceeds the voltage limit of one of the PZT amplifiers which participate in this axis (see VMA, p. 105, and VMI, p. 106), then the command is not executed (check with ERR? (p. 81)). If the stage velocity needs to be limited, velocity control mode must have been set to ON (VCO, p. 103) and the desired velocity must have been specified (VEL, p. 104). If velocity control is not desired, velocity control mode should be set to OFF.
Format:	SVR {AxisID Difference}
Arguments:	AxisID: axis identifier Difference: is the value which is added to the current open-loop control value (dimensionless) The interpretation of the difference value depends on the settings of the axis-to-PZT matrix (see "Output Generation" on p. 46 for more information). With the default matrix coefficients, open-loop control values numerically correspond to axis position values.
Response:	none
Problem Solver:	Illegal axis identifier Axis is in Servo-On mode PZT voltage out of limits The axis has Velocity Control On, but the specified velocity was too small.

TAD? (Tell Current A/D Converter Value) TAD?

Command Type:	Report Command
Description:	Tell the current value of the A/D converter for the given sensor channel. This value represents the digitized sensor value without filtering, linearization and transformation. Using this command the user can see easily if the sensor is in overflow. Note: Multiple sensors ("input channels") can be involved in the control of one logical axis (see Section "Principle of Operation" on p. 40). TAD? reads the voltage for the individual sensors, not for a logical axis.
Format:	TAD? [{SensorID}]
Arguments:	SensorID: sensor identifier (1, 2 and 3)
Response:	A n-bit value depending on the A/D converter (–32768 to +32767; dimensionless).
Problem Solver:	Illegal sensor identifier

TAV? (Get Analog Input Voltage) TAV?

Command Type:	Report Command
Description:	Get voltage at analog input.
Format:	TAV? [a4]
Arguments:	a4: Input line specification. The E-761 has only one analog input line (which is internally handled as the 4th channel of the A/D converter).
Response:	The current voltage at the analog input, with gain and offset (see Section 2.9 on p. 33). Unit is [V].

TIO? (Tell Number of Digital I/O Lines) TIO?

Command Type:	Report Command
Description:	Tell number of installed digital I/O lines. Note: The E-761 has no genuine digital input and output lines, but the analog input is internally interpreted as digital input for triggering tasks (see Section 2.9 on p. 33), and its signal state can be queried by the DIO? command (p. 80).
Format:	TIO?
Arguments:	none
Response:	I=N (N=Number of installed input lines) O=M (M=Number of installed output lines)

TLT? (Tell Number of DDL Tables) TLT?

Command Type:	Report Command
Description:	Tell the number of DDL tables (dynamic digital linearization – table).
Format:	TLT?
Arguments:	none
Response:	The number of DDL tables (the E-761 has 4 DDL tables).

TMN? (Tell Minimum Travel Value) TMN?

Command Type:	Report Command
Description:	Get the minimum commandable position in physical units. Note: The minimum position which can be commanded is either the value given by the "Range min limit" parameter or—if it is greater than the "Range min limit" parameter value—the value of the negative soft limit set with NLM (p. 88). Both range values are adapted automatically when a new home position is determined with DFH (p. 79). The value of the "Range min limit" parameter can be set using SPA (p. 94) or SEP (p. 93). The parameter ID is 0x07000000. Be careful—wrong values may lead to improper operation or damage of your hardware!
Format:	TMN? [{AxisID}]
Parameters:	AxisID: axis identifier
Response	Value of the minimum commandable position in physical units.

TMX? (Tell Maximum Travel Value) TMX?

Command Type:	Report Command
Description:	Get the maximum commandable position in physical units. Note: The maximum position which can be commanded is either the value given by the "Range max limit" parameter or—if it is smaller than the "Range max limit" parameter value—the value of the positive soft limit set with PLM (p. 89). Both range values are adapted automatically when a new home position is determined with DFH (p. 79). The value of the "Range max limit" parameter can be set using SPA (p. 94) or SEP (p. 93). The parameter ID is 0x07000001. Be careful—wrong values may lead to improper operation or damage of your hardware!
Format:	TMX? [{AxisID}]
Parameters:	AxisID: axis identifier
Response	Value of the maximum commandable position in physical units.

TNR? (Tell Number of Recorder Tables) TNR?

Command Type:	Report Command
Description:	Tell the number of data recorder tables.
Format:	TNR?
Arguments:	none
Response:	The number of data recorder tables (the E-761 has 4 data recorder tables).

TNS? (Tell Normalized Sensor Value) TNS?

Command Type:	Report Command
Description:	Tell the normalized sensor value for the given sensor channel. This value is internally the input for the linearization of the sensor position.
Format:	TNS? [{SensorID}]
Arguments:	SensorID: sensor identifier (1, 2 and 3)
Response:	The normalized sensor value ranging from controller specific minimum to maximum (dimensionless; e.g. -100 to 100).

TPC? (Tell Number of Piezo Channels) TPC?

Command Type:	Report Command
Description:	Tell the number of piezo channels. Note: A piezo channel is the representation of a PZT amplifier in the firmware. Multiple PZT amplifiers can be involved in the motion of one logical axis (see Section "Principle of Operation" on p. 40).
Format:	TPC?
Arguments:	none
Response:	The number of piezo channels in the controller (the E-761 has 4 piezo channels).

TSC? (Tell Number of Sensor Channels) TSC?

Command Type:	Report Command
Description:	Tell the number of sensor channels. Note: A sensor channel is the representation of a physical existing sensor in the firmware. Multiple sensor channels can be involved in the control of one logical axis (see Section "Principle of Operation" on p. 40).
Format:	TSC?
Arguments:	none
Response:	The number of sensor channels supported by the controller (the E-761 supports 3 sensor channels).

TSP? (Tell Current Sensor Position in Physical Unit) TSP?

Command Type:	Report Command
Description:	This command requests the current sensor position of the selected sensor channel in µm. Notes: Multiple sensor channels can be involved in the control of one logical axis (see Section "Principle of Operation" on p. 40). To get the current position of an axis, use POS? (p. 90) instead.
Format:	TSP? [{SensorID}]
Arguments:	SensorID: sensor identifier (1, 2 and 3)
Response:	The current sensor position in µm.

TVI? (Tell Valid Axis Identifiers) TVI?

Command Type:	Report Command
Description:	Tell valid axis identifiers Note: Use SAI (p. 92) to set axis identifiers and SAI? (p. 92) to get the identifiers of all configured axes.
Format:	TVI?
Arguments:	none
Response:	String with allowed axis identifiers, e.g. 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ

TWG? (Tell Number of Wave Generators) TWG?

Command Type:	Report Command
Description:	Tell the number of wave generators.
Format:	TWG?
Arguments:	none
Response:	The number of available wave generators (the E-761 has 4 wave generators).

VCO (Set Velocity Control Mode) VCO

Command Type:	Configuration Command
Description:	Sets the Velocity Control Mode of the specified axis to ON or OFF. When Velocity Control Mode is ON, the stage is driven with the velocity specified with VEL (p. 104). Notes: During wave generator output, there is no velocity control, i.e. any settings made with VCO will be ignored.

This command saves the new value in RAM only. To save the currently valid value to flash ROM, where it becomes the power-on default, you must use WPA (p. 118) without any parameter. Changes not saved with WPA will be lost when the controller is powered off.

Format: VCO {AxisID 0|1}

Arguments: AxisID: Axis identifier.

1 or 0; 1 means to set the Velocity Control Mode to ON, while 0 means to set the Velocity Control Mode to OFF.

Response: none

Problem Solver: Illegal axis identifier

VCO? (Get Velocity Control Mode) VCO?

Command Type: Report Command

Description: Reports the Velocity Control Mode of the specified axis.

Format: VCO? [{AxisID}]

Arguments: AxisID: Axis identifier.

Response: "1" when the velocity control mode of the specified axis is ON, "0" when OFF.

Problem Solver: Illegal axis identifier

VEL (Set Velocity) VEL

Command Type: Configuration Command

Description: Set velocity of given axis ("Servo Loop Slew Rate" parameter, ID 0x07000200). This setting will be effective only when velocity control mode is ON for the specified axis (VCO, p. 103).

Notes:

Before executing this command a range check is done— the velocity value must not be negative.

VEL can be changed while stage is moving.

This command saves the new value in RAM only. To save the currently valid value to flash ROM, where it becomes the power-on default, you must use WPA (p. 118). Changes not saved with WPA will be lost when the controller is powered off.

Format: VEL {AxisID vel}

Arguments: AxisID: axis identifier

vel: is the velocity value in [physical units / ms] and must be positive or zero.

Response: none

Problem Solver: Illegal axis identifier

VEL? (Get Commanded Velocity) VEL?

Command Type:	Report Command
Description:	Get the velocity (the value of the "Servo Loop Slew Rate" parameter, ID 0x07000200).
Format:	VEL? [{AxisID}]
Arguments:	AxisID: axis identifier.
Response:	The velocity value in [physical units / ms].
Problem Solver:	Illegal axis identifier

VER? (Get Version) Ver?

Command Type:	Report Command
Description:	Reports device identity number, DSP firmware version and the versions of the underlying drivers and libraries.
Format:	VER?
Arguments:	none
Response:	Multiple strings, terminated by LF.

Example:	`PI E7XX_GCS_DLL.dll: V4.0.0.61`
	`PI E-761 Device Driver: V1.0.0.4`
	`FW: Physik Instrumente, E-761, 20051028, V02.00.00.16`

VMA (Set Upper PZT Voltage Soft Limit) VMA

Command Type:	Configuration Command
Description:	Set upper PZT voltage soft limit of given piezo channel (the "Output Voltage High Limit" parameter (ID 0x0C000001)). Before executing this command a range check is done—the HighLimit value must not be greater than the value of the "Max Voltage of Amplifier" parameter (ID 0x0B000008) and not be smaller than the "Min Voltage of Amplifier" parameter (ID 0x0B000007). You can query these limits with SPA? (p. 95). Make sure that the VMA HighLimit is not smaller than the lower PZT voltage soft limit set with VMI (p. 106). Notes: A piezo channel is the representation of a PZT amplifier in the firmware. Multiple PZT amplifiers can be involved in the motion of one logical axis (see Section "Principle of Operation" on p. 40). VMA saves the new value in RAM only. To save the currently valid value to non-volatile memory you must use WPA (p. 118). Changes not saved with WPA will be lost when the controller is powered off. On power-up or when rebooting the E-761, the VMA value is replaced by the value of the "Max Voltage of Amplifier" parameter, but can be restored to the saved value using RPA (p. 91).
Format:	VMA {PiezoID HighLimit}

Arguments:	PiezoID: piezo channel identifier (1, 2, 3 and 4)
	HighLimit: is the value of the upper PZT voltage soft limit in [V].
Response:	none
Problem Solver:	Illegal piezo channel identifier
	HighLimit value is out of the allowed voltage range of the piezo channel

VMA? (Get Upper PZT Voltage Soft Limit) VMA?

Command Type:	Report Command
Description:	Get upper PZT voltage soft limit of given piezo channel.
Format:	VMA? [{PiezoID}]
Arguments:	PiezoID: piezo channel identifier (1, 2, 3 and 4)
Response:	The value of the upper PZT voltage soft limit in [V].
Problem Solver:	Illegal piezo channel identifier

VMI (Set Lower PZT Voltage Soft Limit) VMI

Command Type:	Configuration Command
Description:	Set lower PZT voltage soft limit of given piezo channel (the "Output Voltage Low Limit" parameter (ID 0x0C000000)).
	Before executing this command a range check is done—the LowLimit value must not be smaller than the value of the "Min Voltage of Amplifier" parameter (parameter ID 0x0B000007) and not be greater than the "Max Voltage of Amplifier" parameter (parameter ID 0x0B000008). You can query these limits with SPA? (p. 95).
	Make sure that the VMI LowLimit is not greater than the upper PZT voltage soft limit set with VMA (p. 105).
	Notes:
	A piezo channel is the representation of a PZT amplifier in the firmware. Multiple PZT amplifiers can be involved in the motion of one logical axis (see Section "Principle of Operation" on p. 40).
	VMI saves the new value in RAM only. To save the currently valid value to non-volatile memory you must use WPA (p. 118). Changes not saved with WPA will be lost when the controller is powered off.
	On power-up or when rebooting the E-761, the VMI value is replaced by the value of the "Min Voltage of Amplifier" parameter, but can be restored to the saved value using RPA (p. 91).
Format:	VMI {PiezoID LowLimit}
Arguments:	PiezoID: piezo channel identifier (1, 2, 3 and 4)
	LowLimit: is the value of the lower PZT voltage soft limit in [V].
Response:	none
Problem Solver:	Illegal piezo channel identifier

LowLimit value is out of the allowed voltage range of the piezo channel

VMI? (Get Lower PZT Voltage Soft Limit) VMI?

Command Type:	Report Command
Description:	Get lower PZT voltage soft limit of given piezo channel.
Format:	VMI? [{PiezoID}]
Arguments:	PiezoID: piezo channel identifier (1, 2, 3 and 4)
Response:	The value of the lower PZT voltage soft limit in [V].
Problem Solver:	Illegal piezo channel identifier

VOL (Set Current PZT Voltage) VOL

Command Type:	Move Command
Description:	Set absolute PZT voltage output for the given piezo channel. This command can be interrupted by #24. Servo must be switched off when using this command. Notes: A piezo channel is the representation of a PZT amplifier in the firmware. Multiple PZT amplifiers ("output channels") can be involved in the motion of one logical axis (see Section "Principle of Operation" on p. 40). Use SVA (p. 97) or SVR (p. 99) to command an axis in open-loop operation. If the commanded voltage exceeds the voltage limits of the piezo channel (see VMA, p. 105, and VMI, p. 106), then the command is not executed (check with ERR?, p. 81).
Format:	VOL {PiezoID Voltage}
Arguments:	PiezoID: piezo channel identifier. Voltage: is the voltage value to set in V.
Response:	none
Problem Solver:	Illegal piezo channel identifier Axis is in Servo-On mode Voltage out of limits

VOL? (Read Current PZT Voltage) VOL?

Command Type:	Report Command
Description:	Read current PZT voltage value of the given piezo channel.
Format:	VOL? [{PiezoID}]
Arguments:	PiezoID: piezo channel identifier.
Response:	The current voltage value in V.
Problem Solver:	Illegal piezo channel identifier

VST? (Get available Stages) VST?

Command Type: Report Command

Description: List the stage names which can be used for the axis configuration with CST (p. 78).

Format: VST?

Arguments: none

Response: The available stage names. For piezo controllers only ID-STAGE and NOSTAGE are returned.

WAV (Define Waveform) WAV

Command Type: Wave Command

Description: Define waveform of given type for given wave table on the E-761. The necessary parameters depend on the type.

Notes:

The wave table content is temporarily stored on the E-761 until a new WAV command is sent or the controller is powered down or rebooted.

To allow for flexible waveform shapes, the wave table can be divided into "segments" which must each be defined as a separate waveform and concatenated to form the final waveform (see AppendWave argument below). Waveforms can be based on predefined "curve" shapes (see WaveType argument below). See the "Terminology Table" on p. 149 for a detailed description of the "segment" and "curve" terms.

The CFG wave type (see below) is a special type with which you can configure additional parameters for a waveform, e.g. the wave table rate. The settings should be made subsequent to the real waveform definition and will be applied to the waveform when the wave generator output starts. Some CFG settings can be saved to non-volatile memory as power-up defaults—if not, they are valid until a new WAV command is sent or the wave table content is cleared or the controller is powered down or rebooted. See p. 112 for details.

You can check the wave table content using GWD? (p. 82), start the wave generator output with WGO (p. 114) and clear the wave table content using WCL (p. 113). Be sure that you have set correct waveform sequence and configuration before enabling wave output. Although the WAV command is available when waveform output is enabled, an incorrect command parameter could cause unpredictable stage response, such as overflow and vibration. It is therefore recommended that waveform output be disabled before using the WAV command, and re-enabled only after the waveform sequence has been checked.

The content of a wave table is not completely erased when a new waveform is written to this table. Only the number of points given by the new waveform will be written beginning with the first point in the table, but any subsequent data points will keep the old values from the former waveform. This affects only the response to the GWD? command and not the wave generator output which will only comprise the new waveform points.

The waveform values are absolute values. In closed-loop operation (servo ON), they are interpreted as target positions in either case. In open-loop operation (servo OFF), the interpretation of the wave generator output depends on the settings of the axis-to-PZT matrix (see "Output Generation", p. 46 for more information). By default, the matrix is set up so that commanded open-loop control values numerically correspond to axis position values.

The control value which is valid before the wave generator is started has no further influence on the waveform, but large jumps to the start value of the waveform should be avoided.

The periodic time (cycle duration) for a waveform can be calculated as follows:

time = WaveLength * SamplingTime * Oversampling Factor

where WaveLength is the number of points used for the waveform,
SamplingTime is the sampling time of the ADC (10 µs for E-761)
Oversampling Factor is the factor set with AVG (p. 77) which influences the actual servo sampling time (without changing the ADC sampling time), default is 4

When creating the waveform, keep in mind that the applicable frequency depends on the available amplifier power. Example: with a capacitive load of 6.6 µF, the frequency should not exceed 15 Hz if three amplifiers are involved in the motion or 50 Hz if only one amplifier is involved (motion covers the whole travel range; see also Section "Frequency Response" on p. 50). Otherwise overheating of the amplifiers can occur, and the piezo voltage output will be deactivated automatically.

See also the instructions and examples in Section "Wave Generator" beginning on p. 52.

Format: WAV WaveTableID [AppendWave] WaveType WaveTypeParameters

Arguments:

WaveTableID: Wave table identifier.

AppendWave: This can be one of “+” or “&”.
“+” adds the values of the defined wave to the values of an already existing wave.
“&” appends the defined wave to an already existing wave (i.e. concatenates segments to form one waveform).
If the parameter is omitted, the values of the defined wave will overwrite an already existing wave.

WaveType: The wave type. This can be one of
“POL” (polynomial)
“PNT” (user defined)
“SIN_P” (inverted cosine curve)
“RAMP” (ramp curve)
“LIN” (single scan line curve)
"CFG" (special type which configures additionally parameters for a waveform).

WaveTypeParameters: The wave-type-dependent parameters listed below:

For **"POL"** wave type with

$$POL = A_0 + A_1(x - x_0)^1 + \cdots + A_n(x - x_0)^n$$

the wave-type-dependent parameters are:

WaveStartPoint WaveLength x_0 A_0 [{A_n}]

WaveStartPoint:	The index of the starting point.
WaveLength:	The length of the wave as number of points (i.e. the x value of the equation shown above).

n ≤ 5

For **"PNT"** wave type the wave-type-dependent parameters are:

WaveStartPoint WaveLength {WavePoint}

WaveStartPoint:	The index of the starting point. Starts with 0.
WaveLength:	The length of the wave as number of points.
WavePoint:	The value of one single point.

For "**SIN_P**" wave type the wave-type-dependent parameters are:

SegLength Amp Offset WaveLength StartPoint CurveCenterPoint

SegLength:	The length of the wave (segment) as number of points. Only the number of points given by SegLength will be written to the wave table.
Amp:	The amplitude of the wave.
Offset:	The offset of the wave.
WaveLength:	The length of the sine curve as number of points. Determines the shape of the wave.
StartPoint:	The index of the starting point of the wave. Gives the phase shift. Can be negative.
CurveCenterPoint	The index of the center point of the sine curve. Determines the shape of the wave (symmetrically or not).

For "**RAMP**" wave type the wave-type-dependent parameters are:

SegLength Amp Offset WaveLength StartPoint SpeedUpDown CurveCenterPoint

SegLength:	The length of the wave (segment) as number of points. Only the number of points given by SegLength will be written to the wave table.
Amp:	The amplitude of the wave.
Offset:	The offset of the wave.
WaveLength:	The length of the ramp curve as number of points. Determines the shape of the wave.
StartPoint:	The index of the starting point of the wave. Gives the phase shift. Can be negative.
SpeedUpDown:	The number of points for speed up and down.
CurveCenterPoint	The index of the center point of the ramp curve. Determines the shape of the wave (symmetrically or not).

For "**LIN**" wave type the wave-type-dependent parameters are:

SegLength Amp Offset WaveLength StartPoint SpeedUpDown

SegLength:	The length of the wave (segment) as number of points. Only the number of points given by SegLength will be written to the wave table.
Amp:	The amplitude of the wave.
Offset:	The offset of the wave.
WaveLength:	The length of the single scan line curve as number of points. Determines the shape of the wave.
StartPoint:	The index of the starting point of the wave. Can be negative.

SpeedUpDown: The number of points for speed up and down.

For "**CFG**" wave type the wave-type-dependent parameters are

n m p k s L

If not saved to non-volatile memory as power-up defaults (supported for selected parameters, see below), the settings are valid until a new WAV command is sent or the wave table content is cleared or the controller is powered down or rebooted. If less than six parameters are specified, the values are assigned in order left to right and the unspecified parameters retain their previous values.

n integer, the length of the periodic waveform, i.e., how many points are to be included in one period of the waveform. Must be between 1 and 8192 The value is also available in volatile memory as "Total wave form points" parameter, ID 0x13000102.

m integer, amount by which the current-point pointer is incremented each time. Must be between 1 and 8191.

p integer, phase shift of periodic waveform, i.e. which point is the starting point of the waveform. Must be between 1 and 8191.

k integer, the number of interrupts (samples) before incrementing the current-point pointer (note that the sampling time is influenced by AVG, p. 77). Must be equal to or greater than 1. If not specified 1 is used. The value is also available as "Wave generator table rate" parameter, ID 0x13000109 (use WPA to save it to non-volatile memory).

s double, amplitude shift after each complete period. In scanning applications this parameter is the distance between lines. If not specified 0 is used. The value is also available as "Curve Offset" parameter, ID 0x1300010B (use WPA to save it to non-volatile memory). Note that if the wave generator is started with the option "start at the

endpoint of the last cycle" (bit 8), the E-761 at the end of each output cycle equates the offset value with the current generator output.

L double, limit of waveform amplitude. In scanning applications, this parameter defines the field limit. If not specified 1e6 is used.

The period of the resulting output wave (and hence the wave table rate) results from the following equation:

$$T_s = \frac{nk}{m} t_i$$

where t_i is SamplingTime * Oversampling Factor (see above)

The initial phase angle of the output waveform is:

$$\phi_s = \frac{p}{n} \times 360°$$

Response: none

Problem Solver: invalid wave table identifier

total number of points for the waveform (which may consist of several segments) exceeds 8192 points (max. number of points per table)

WAV? (Get Waveform Parameters) WAV?

Command Type: Report Command

Description: Get the value of a wave parameter for a given wave table.

Format: WAV? [{WaveTableID ParameterID}]

Arguments: WaveTableID: Wave table identifier.

ParameterID: is a wave parameter ID; for E-761 only 1 is valid (number of waveform points for currently defined wave).

Response: Number of waveform points for currently defined wave

WCL (Wave Clear) WCL

Command Type: Wave Command

Description: Wave clear: Clears the content of the given wave table.

Format: WCL {WaveTableID}

Arguments: WaveTableID: Wave table identifier.

Response: None

WGC (Set Number of Wave Generator Output Cycles) WGC

Command Type:	Wave Command
Description:	Set the number of output cycles for the given wave generator (the "Wave generator cycles" parameter, ID 0x13000003). Notes: The wave generator output is started with WGO, p. 114. WGC saves the new value in RAM only. To save the currently valid value to non-volatile memory you must use WPA (p. 118). Changes not saved with WPA will be lost when the controller is powered off.
Format:	WGC {WaveGeneratorID Cycles}
Arguments:	WaveGeneratorID: Wave generator identifier. Cycles: Number of cycles. If cycles = 0 then the wave is output without period limitation until it is stopped by WGO or #24.
Response:	None
Example:	WGC 1 10 Set number of wave generator output cycles to 10 for wave generator 1 (without this limitation the output would be continuous until WGO 1 0 was sent to disable the wave generator).

WGC? (Get Number of Wave Generator Output Cycles) WGC?

Command Type:	Report Command
Description:	Reports the number of output cycles set for the given wave generator.
Format:	WGC? [{WaveGeneratorID}]
Arguments:	WaveGeneratorID: Wave generator identifier.
Response:	The number of wave generator cycles that will be performed. If this is zero the wave generator will run continuously.

WGO (Start and Stop Wave Generator Output) WGO

Command Type:	Wave Command
Description:	Start and stop the wave generator with the given mode. Notes: Up to four wave generators can run simultaneously. Digital output synchronized with the wave generator output and hence with the axis motion is possible via the start options bit 3, bit 4 and bit 5. To make the digital output available outside of the PC, a trigger output bracket is required (included with E-761.3CT models, order separately as E-761.00T). See p. 147 for pinout of the Digital Out socket. The number of output cycles can be limited by WGC (p. 114) or by the "Wave generator cycles" parameter, ID 0x13000003.

When the wave generator output is stopped and restarted, it will continue with the next point of the waveform, even if the axis was moved by a move command in the meantime. If you want the wave generator to restart with the first wave point instead, you can

- send WAV *WaveTableID* CFG *n* 1 with the appropriate values for *WaveTableID* and *n* or
- define the complete waveform again with WAV (see p. 108)

See also the example for phase shift conservation which is given in Section "Phase Shift for Sine and Ramp Curves" on p. 55.

Servo status can not be changed (SVO, p. 98) while a wave generator is running for the axis.

Wave generator output will continue even if the terminal or the program from which it was started is quit. The wave generator output will also continue when the high voltage output should be automatically deactivated due to amplifier overheating, i.e. if a certain number of output cycles was set, the output may be already finished when the high voltage output is reactivated.

Each time the wave generator is started recording starts automatically as follows (read the data with DRR?, p. 80):

recorder table 1: axis 1 actual position

recorder table 2: axis 2 actual position

recorder table 3: axis 3 actual position

recorder table 4: analog input voltage (same value as read with TAV?, i.e. contains gain and offset for the analog input, see p. 100)

Recording always takes place for all record tables, regardless of which wave generator was started.

Recording ends when the record table content has reached the maximum number of points (8192 per table).

Recording can be restarted by WGR (p. 117).

You can lengthen the individual output cycles of the waveform: when the wave generator output is synchronized by interrupt (started with bit 0 or 1), the wave table rate (i.e. the output frequency) depends on the servo sampling rate influenced by AVG (p. 77) and on the settings made with the CFG wave type (parameter k) or the "Wave generator table rate" parameter, ID 0x13000109 (see WAV, p. 108).

The #9 single-character command (p. 75) can be used to query the current activation state of the wave generators. The reply shows if a wave generator is running or not, but does not contain any information about the wave generator start mode (e.g. with trigger output). With WGO? you can ask for the last-commanded wave generator start options (WGO settings).

See also the instructions and examples in Section "Wave Generator" beginning on p. 52.

Format: WGO {WaveGeneratorID StartBits}

Arguments: WaveGeneratorID: Wave generator identifier.

StartBits: give the start mode for the specified wave generator.
Note that bit 3 (0x8 or 8), bit 4 (0x10 or 16), bit 5 (0x20 or

32) and bit 8 (0x100 or 256) cannot start the wave generator output by themselves. They simply specify certain start options and must always be combined with one of the start modes specified in bit 0 (0x1 or 1), bit 1 (0x2 or 2), bit 2 (0x4 or 4) and bit 10 (0x400 or 1024). Note that if you should combine bits 0, 1, 2 and 10, the wave generator starts with the mode given by the least significant bit.
The start mode values in detail:

0: wave generator output is stopped. You can also use #24 (p. 75) to stop the wave generator output.

bit 0 = 0x1 (hex format) or 1 (decimal format):
start wave generator output immediately; synchronized by interrupt

bit 1 = 0x2 (hex format) or 2 (decimal format):
start wave generator output triggered by external signal; synchronized by interrupt
The external signal used is the analog input signal (see p. 33). The wave generator runs as long as the signal is HIGH (see DIO?, p. 80, and TAV?, p. 100).

bit 2 = 0x4 (hex format) or 4 (decimal format):
wave generator is started and synchronized by external signal.
The external signal used is the analog input signal (see p. 33). NOTE: the external signal (both high and low level) must have width of more than 50 microseconds.

bit 3 = 0x8 (hex format) or 8 (decimal format):
synchronized trigger pulse is output on digital output line when the wave generator outputs a new data point; start option

bit 4 = 0x10 (hex format) or 16 (decimal format):
synchronized trigger pulse is output on digital output line when the axis finishes each period (end of scan line, see WAV CFG); start option

bit 5 = 0x20 (hex format) or 32 (decimal format):
synchronized trigge pulse is output on digital output line when the axis reaches the amplitude limit (scan field limit, see WAV CFG); start option

bit 8 = 0x100 (hex format) or 256 (decimal format):
wave generator started at the endpoint of the last cycle; start option.
The second and all subsequent output cycles each start at the endpoint of the preceding cycle. The final position is the sum of the endpoint of the last output cycle and any offset defined with WAV for the waveform.

bit 10 = 0x400 (hex format) or 1024 (decimal format):
"external wave generator" is started—i.e. the analog input is enabled for commanding the axis given by the wave generator ID.

The "external wave generator" (e.g. a laboratory power supply) must be connected to the analog input socket according to the description on p. 33.

Response: none

Problem Solver: Illegal wave generator identifier

WGO? (Get Wave Generator Output Settings) WGO?

Command Type: Report Command

Description: Get the start mode of the given wave generator.

Note:

To report the settings for the number of wave generator output periods use the WGC? command, p. 114.

Format: WGO? [{WaveGeneratorID}]

Arguments: WaveGeneratorID: Wave generator identifier.

Response: Start mode of the wave generator, see WGO (p. 114) for a detailed description.

WGR (Start Recording) WGR

Command Type: Wave Command

Description: Starts recording when the wave generator is running.

Notes:

The data can be read out with DRR? (p. 80).

Recording always takes place for all record tables, regardless of which wave generator is running. The assignment of axis and data sources to the data recorder tables is as follows:

table 1: axis 1 actual position

table 2: axis 2 actual position

table 3: axis 3 actual position

table 4: analog input voltage (same value as read with TAV?, p. 100, i.e. contains gain and offset for the analog input, see p. 33)

Recording starts always with the next start point of the waveform, i.e. there might be a short delay between sending WGR and the start of the record. If more than one wave generator is running, recording starts at the waveform start point which occurs first.

Recording ends when the record table content has reached the maximum number of points (8192 per table).

See also WGO, p. 114.

Format: WGR

Arguments: none

Response: None

WMS? (Get Maximum Number of Points per Wave Table) WMS?

Command Type:	Report Command
Description:	Get the maximum number of points for the waveform of the specified wave table.
Format:	WMS? [{WaveTableID}]
Arguments:	WaveTableID: Wave table identifier.
Response:	The number of points available for the wave table.

WPA (Write Parameter to Non-Volatile Memory) WPA

Command Type:	Configuration Command
Description:	Write the currently valid value of a parameter of a given item (axis, sensor or piezo channel, whole system) from volatile memory (RAM) to non-volatile memory (EEPROM). The values saved this way become the power-on defaults. Notes: CAUTION: If current parameter values are incorrect, the system may malfunction. Be sure that you have the correct parameter settings before using the WPA command. Settings not saved with WPA will be lost when the PC is powered off or the E-761 is rebooted. With HPA? (p. 84) you can obtain a list of the parameters IDs. Use SPA? (p. 95) to check the current parameter settings in the volatile memory. Parameters can be changed with SPA (p. 94), AVG (p. 77), DFH (p. 79), RTR (p. 91), VEL (p. 104), VMA (p. 105), VMI (p. 106), WAV (p. 108) and WGC (p. 114). When WPA is used without specifying any parameters, all currently valid parameter values are saved, and additionally the following settings are saved too: velocity control mode (VCO, p. 103), position limits (NLM, p. 88, PLM, p. 89).
Format:	WPA Password [{ItemID ParameterID}]
Arguments:	Password: the default password to write to EEPROM is "100" ItemID: is an axis identifier, a sensor or piezo channel or the whole system, the item type depends on the parameter ID ParameterID: is a parameter ID
Response:	none
Problem Solver:	Incorrect password

7.4 Error Codes

The error codes listed here are those of the PI General Command Set. As such, some may be not relevant to your controller and will simply never occur.

Controller Errors

0	PI_CNTR_NO_ERROR	No error
1	PI_CNTR_PARAM_SYNTAX	Parameter syntax error
2	PI_CNTR_UNKNOWN_COMMAND	Unknown command
3	PI_CNTR_COMMAND_TOO_LONG	Command length out of limits or command buffer overrun
4	PI_CNTR_SCAN_ERROR	Error while scanning
5	PI_CNTR_MOVE_WITHOUT_REF_OR_NO_SERVO	Unallowable move attempted on unreferenced axis, or move attempted with servo off
6	PI_CNTR_INVALID_SGA_PARAM	Parameter for SGA not valid
7	PI_CNTR_POS_OUT_OF_LIMITS	Position out of limits
8	PI_CNTR_VEL_OUT_OF_LIMITS	Velocity out of limits
9	PI_CNTR_SET_PIVOT_NOT_POSSIBLE	Attempt to set pivot point while U,V and W not all 0
10	PI_CNTR_STOP	Controller was stopped by command
11	PI_CNTR_SST_OR_SCAN_RANGE	Parameter for SST or for one of the embedded scan algorithms out of range
12	PI_CNTR_INVALID_SCAN_AXES	Invalid axis combination for fast scan
13	PI_CNTR_INVALID_NAV_PARAM	Parameter for NAV out of range
14	PI_CNTR_INVALID_ANALOG_INPUT	Invalid analog channel
15	PI_CNTR_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
16	PI_CNTR_INVALID_STAGE_NAME	Unknown stage name
17	PI_CNTR_PARAM_OUT_OF_RANGE	Parameter out of range
18	PI_CNTR_INVALID_MACRO_NAME	Invalid macro name

19	PI_CNTR_MACRO_RECORD	Error while recording macro
20	PI_CNTR_MACRO_NOT_FOUND	Macro not found
21	PI_CNTR_AXIS_HAS_NO_BRAKE	Axis has no brake
22	PI_CNTR_DOUBLE_AXIS	Axis identifier specified more than once
23	PI_CNTR_ILLEGAL_AXIS	Illegal axis
24	PI_CNTR_PARAM_NR	Incorrect number of parameters
25	PI_CNTR_INVALID_REAL_NR	Invalid floating point number
26	PI_CNTR_MISSING_PARAM	Parameter missing
27	PI_CNTR_SOFT_LIMIT_OUT_OF_RANGE	Soft limit out of range
28	PI_CNTR_NO_MANUAL_PAD	No manual pad found
29	PI_CNTR_NO_JUMP	No more step-response values
30	PI_CNTR_INVALID_JUMP	No step-response values recorded
31	PI_CNTR_AXIS_HAS_NO_REFERENCE	Axis has no reference sensor
32	PI_CNTR_STAGE_HAS_NO_LIM_SWITCH	Axis has no limit switch
33	PI_CNTR_NO_RELAY_CARD	No relay card installed
34	PI_CNTR_CMD_NOT_ALLOWED_FOR_STAGE	Command not allowed for selected stage(s)
35	PI_CNTR_NO_DIGITAL_INPUT	No digital input installed
36	PI_CNTR_NO_DIGITAL_OUTPUT	No digital output configured
37	PI_CNTR_NO_MCM	No more MCM responses
38	PI_CNTR_INVALID_MCM	No MCM values recorded
39	PI_CNTR_INVALID_CNTR_NUMBER	Controller number invalid
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave

42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter) missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error
49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	MOV! motion still in progress
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data Record Table does not exist
58	PI_CNTR_INVALID_RECORDER_SRC_OPT	Source does not exist; number too low or too high
59	PI_CNTR_INVALID_RECORDER_SRC_CHAN	Source Record Table number too low or too high
60	PI_CNTR_PARAM_PROTECTION	Protected Param: current Command Level (CCL) too low
61	PI_CNTR_AUTOZERO_RUNNING	Command execution not possible while Autozero is running

62	PI_CNTR_NO_LINEAR_AXIS	Autozero requires at least one linear axis
63	PI_CNTR_INIT_RUNNING	Initialization still in progress
64	PI_CNTR_READ_ONLY_PARAMETER	Parameter is read-only
65	PI_CNTR_PAM_NOT_FOUND	Parameter not found in non-volatile memory
66	PI_CNTR_VOL_OUT_OF_LIMITS	Voltage out of limits
67	PI_CNTR_WAVE_TOO_LARGE	Not enough memory available for requested wave curve
68	PI_CNTR_NOT_ENOUGH_DDL_MEMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DELAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_ARRAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN_SINGLE_MODE_RESTART	Attempt to restart the generator while it is running in single step mode
72	PI_CNTR_ANALOG_TARGET_ACTIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERATOR_ACTIVE	Motion commands are not allowed when wave generator is active
74	PI_CNTR_AUTOZERO_DISABLED	No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)
75	PI_CNTR_NO_WAVE_SELECTED	Generator started (WGO) without having selected a wave table (WSL).
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly
77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_SERVO_ON	Open-loop commands (SVA, SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM

81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command
82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
100	PI_LABVIEW_ERROR	PI LabVIEW driver reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on
206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header
207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error
210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller

212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken
216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, call CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop
218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATION	Position was calculated during MOV motion
230	PI_CNTR_INVALID_HANDLE	Invalid handle
231	PI_CNTR_NO_BIOS_FOUND	No bios found
232	PI_CNTR_SAVE_SYS_CFG_FAILED	Save system configuration failed
233	PI_CNTR_LOAD_SYS_CFG_FAILED	Load system configuration failed
301	PI_CNTR_SEND_BUFFER_OVERFLOW	Send buffer overflow
302	PI_CNTR_VOLTAGE_OUT_OF_LIMITS	Voltage out of limits
303	PI_CNTR_OPEN_LOOP_MOTION_SET_WHEN_SERVO_ON	Open-loop motion attempted when servo ON
304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long
305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus
307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro

310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined
402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
601	PI_CNTR_NOT_ENOUGH_MEMORY	not enough memory
602	PI_CNTR_HW_VOLTAGE_ERROR	hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	hardware temperature out of range
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros
1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
2000	PI_CNTR_ALREADY_HAS_SERIAL_NUMBER	Controller already has a serial number
4000	PI_CNTR_SECTOR_ERASE_FAILED	Sector erase failed

4001	PI_CNTR_FLASH_PROGRAM_FAILED	Flash program failed
4002	PI_CNTR_FLASH_READ_FAILED	Flash read failed
4003	PI_CNTR_HW_MATCHCODE_ERROR	HW match code missing/invalid
4004	PI_CNTR_FW_MATCHCODE_ERROR	FW match code missing/invalid
4005	PI_CNTR_HW_VERSION_ERROR	HW version missing/invalid
4006	PI_CNTR_FW_VERSION_ERROR	FW version missing/invalid
4007	PI_CNTR_FW_UPDATE_ERROR	FW update failed

Interface Errors

0	COM_NO_ERROR	No error occurred during function call
-1	COM_ERROR	Error during com operation (could not be specified)
-2	SEND_ERROR	Error while sending data
-3	REC_ERROR	Error while receiving data
-4	NOT_CONNECTED_ERROR	Not connected (no port with given ID open)
-5	COM_BUFFER_OVERFLOW	Buffer overflow
-6	CONNECTION_FAILED	Error while opening port
-7	COM_TIMEOUT	Timeout error
-8	COM_MULTILINE_RESPONSE	There are more lines waiting in buffer
-9	COM_INVALID_ID	There is no interface or DLL handle with the given ID
-10	COM_NOTIFY_EVENT_ERROR	Event/message for notification could not be opened
-11	COM_NOT_IMPLEMENTED	Function not supported by this interface type
-12	COM_ECHO_ERROR	Error while sending "echoed" data

-13	COM_GPIB_EDVR	IEEE488: System error
-14	COM_GPIB_ECIC	IEEE488: Function requires GPIB board to be CIC
-15	COM_GPIB_ENOL	IEEE488: Write function detected no listeners
-16	COM_GPIB_EADR	IEEE488: Interface board not addressed correctly
-17	COM_GPIB_EARG	IEEE488: Invalid argument to function call
-18	COM_GPIB_ESAC	IEEE488: Function requires GPIB board to be SAC
-19	COM_GPIB_EABO	IEEE488: I/O operation aborted
-20	COM_GPIB_ENEB	IEEE488: Interface board not found
-21	COM_GPIB_EDMA	IEEE488: Error performing DMA
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended operation
-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost
-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full
-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits
-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)

-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources
-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device
-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present
-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error
-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port: was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again

-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)

DLL Errors

-1001	PI_UNKNOWN_AXIS_IDENTIFIER	Unknown axis identifier
-1002	PI_NR_NAV_OUT_OF_RANGE	Number for NAV out of range--must be in [1,10000]
-1003	PI_INVALID_SGA	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	PI_UNEXPECTED_RESPONSE	Controller sent unexpected response
-1005	PI_NO_MANUAL_PAD	No manual control pad installed, calls to SMA and related commands are not allowed
-1006	PI_INVALID_MANUAL_PAD_KNOB	Invalid number for manual control pad knob
-1007	PI_INVALID_MANUAL_PAD_AXIS	Axis not currently controlled by a manual control pad
-1008	PI_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
-1009	PI_THREAD_ERROR	Internal error--could not start thread
-1010	PI_IN_MACRO_MODE	Controller is (already) in macro mode--command not valid in macro mode
-1011	PI_NOT_IN_MACRO_MODE	Controller not in macro mode--command not valid unless macro mode active

-1012	PI_MACRO_FILE_ERROR	Could not open file to write or read macro
-1013	PI_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
-1014	PI_MACRO_EDITOR_ERROR	Internal error in macro editor
-1015	PI_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	PI_AXIS_ALREADY_EXISTS	Axis identifier is already in use by a connected stage
-1017	PI_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
-1018	PI_COM_ARRAY_ERROR	Could not access array data in COM server
-1019	PI_COM_ARRAY_RANGE_ERROR	Range of array does not fit the number of parameters
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?
-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error while axis in motion, call CLR to resume operation
-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed
-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup

-1032	PI_USER_CANCELLED	The user cancelled a dialog
-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL
-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter file open. Close it before opening a new file
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid
-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis
-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started
-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC

-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication with OPM (Optical Power Meter), maybe no OPM connected
-1057	PI_WAVE_EDITOR_WRONG_PARAMNUM	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	PI_WAVE_EDITOR_FREQUENCY_OUT_OF_RANGE	WaveEditor: Frequency out of range
-1059	PI_WAVE_EDITOR_WRONG_IP_VALUE	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	PI_WAVE_EDITOR_WRONG_DP_VALUE	WaveEditor: Error during wave creation, incorrect index for floating point parameter
-1061	PI_WAVE_EDITOR_WRONG_ITEM_VALUE	WaveEditor: Error during wave creation, could not calculate value
-1062	PI_WAVE_EDITOR_MISSING_GRAPH_COMPONENT	WaveEditor: Graph display component not installed
-1063	PI_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
-1064	PI_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
-1065	PI_EXT_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
-1066	PI_EXT_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
-1067	PI_PROFILE_GENERATOR_NO_PROFILE	ProfileGenerator: No profile has been created yet
-1068	PI_PROFILE_GENERATOR_OUT_OF_LIMITS	ProfileGenerator: Generated profile exceeds limits of one or both axes

-1069	PI_PROFILE_GENERATOR_UNKNOWN_PARAMETER	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	PI_PROFILE_GENERATOR_PAR_OUT_OF_RANGE	ProfileGenerator: Parameter out of allowed range
-1071	PI_EXT_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
-1072	PI_EXT_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
-1073	PI_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.
-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by another function. Please try again later.

8 Parameter Overview

CAUTION

Incorrect parameter values may lead to improper operation or damage to your hardware. Be careful when changing parameters.

It is strongly recommended to save the parameter values of the E-761 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-761, use the *Device Parameter Configuration* window of NanoCapture™. See "Creating Backup File for Controller Parameters" on p. 23 for more information.

To adapt the E-761 to your application, you can modify parameter values—either for the whole system, for the individual axes or for the individual sensor channels and PZT amplifier channels (for the interdependence between axis and channels see Section 3, p. 40). The parameters and parameter types available depend on the controller firmware. Note that many parameters are "protected" and can not be changed—it is only possible to change the parameters which are listed in the table below (can be queried with HPA?, p. 84).

NOTE

The parameters which are available in the controller—protected and unprotected—are listed in the *Device Parameter Configuration* window of *NanoCapture*™. The unprotected parameters have the value 0 in the *CCL* column.

Parameters can be changed temporarily or in non-volatile memory using the *Device Parameter Configuration* window of *NanoCapture*™ (*Config* → *Device Parameter Configuration* menu sequence). Alternatively you can enter appropriate GCS commands in the command terminal (see SPA p. 94, SEP p. 93, WPA p. 118), but using the *Device Parameter Configuration* window is much more comfortable because you do not have to deal with any parameter numbers. The parameters which can be changed have the value 0 in the *CCL* column of the *Device Parameter Configuration* window.

NOTE

See the Notes column in the list below for special parameter characteristics:

→ Parameters may be read-only even though their CCL value is 0

→ Parameters may only be present in volatile memory

→ Parameters may be modifiable only for a certain axis or channel

→ Parameters may refer to the whole system. For those parameters the ItemID in the appropriate commands must always have the value 1

See also the *NanoCapture*™ manual for how to edit, save or reset parameter values.

In addition to the "general" modification commands SPA and SEP, there are commands which change certain specific parameters. All the commands listed

below change the parameter value only in volatile memory, and WPA must be used to save the value to non-volatile memory:

AVG (p. 77; "Sensor sampling time" (ID 0x0e000100) and "Servo update time" (ID 0x0e000200))

DFH (p. 79; "User Origin" (ID 0x07010200))

RTR (p. 91; "Table rate" (ID 0x16000000))

VEL (p. 104; “Servo loop slew rate” (ID 0x07000200))

VMA (p. 105; "Output Voltage High Limit" (ID 0x0C000001))

VMI (p. 106; "Output Voltage Low Limit" (ID 0x0C000000))

WAV (p. 108; "Wave generator table rate" (0x13000109), "Curve Offset" (ID 0x1300010B), "Total wave form points" (ID 0x13000102); note that the "Total wave form points" parameter is only present in volatile memory (can not be saved))

WGC (p. 114; "Wave generator cycles" (ID 0x13000003))

Values stored in non-volatile memory are power-up defaults, so that the system can be used in the desired way immediately. Note that PI records data files of every E-761 controller calibrated at the factory for easy restoration of original settings after shipping.

Note that when a stage with ID-chip is connected to the controller for the first time, the stage parameters from the ID-chip will be written to the EEPROM on PC power-on (= controller power-on). Afterwards, the stage parameters will be written on power-on only when the "Read ID-Chip always" parameter is enabled, in this case the home-position is reset. By default, this option is disabled to maintain optimized parameter settings on the controller. The parameters which are stored on the ID-chip are marked in the Notes column of the table below, but can differ slightly between the different mechanics provided by PI. See Section 2.11 on p. 35 for more information about the handling of stages with ID-chip.

Parameter Number	Parameter Name	Range	Notes
0X02000000	Sensor Mechanic: Sensor/Analog enable	0 = Disabled 1 = Enabled	ID-Chip
0X02000001	Sensor Mechanic: Sensor type		
0X02000100	Sensor Mechanic: Sensor range factor	0 = Board Range 3.00X 1 = Option 3.00X 21 2 = Option 3.00X 31 3 = Option 3.00X 41 4 = Option 3.00X 51 5 = Option 3.00X 61 6 = Option 3.00X 71 7 = Board Range 2.13X 8 = Option 2.13X 32 9 = Option 2.13X 42 10 = Option 2.13X 52 11 = Option 2.13X 62 12 = Option 2.13X 72 13 = Board Range 1.25X 14 = Option 1.25X 43 15 = Option 1.25X 53 16 = Option 1.25X 63 17 = Option 1.25X 73 18 = Board Range 1.00X 19 = Option 1.00X 54 20 = Option 1.00X 64 21 = Option 1.00X 74 22 = Board Range 0.75X 23 = Option 0.75X 65 24 = Option 0.75X 75 25 = Board Range 0.68X 26 = Option 0.68X 76 27 = Board Range 0.56X	ID-Chip
0X02000101	Sensor Mechanic: Board Gain	0 = Gain 0.5 64 = Gain 1.0 128 = Gain 2.0 192 = Gain 3.0	ID-Chip
0X02000102	Sensor Mechanic: Electrical poti selected		
0X04000001	ADC: PGA correction of gain 1.0		Can only be changed for the analog input line which is handled as the 4th sensor (channel of the A/D converter), i.e. ItemID must be 4 for write operations
0X04000101	ADC: PGA correction offset x (x = index)		Can only be changed for the analog input which is handled as the 4th sensor (channel of the A/D converter), i.e. ItemID must be 4 for write operations

Parameter Number	Parameter Name	Range	Notes
0X05000000	Sensor Filter: Digital filter type	0 = No Filter 1 = IIR Filter 2 = FIR filter	ID-Chip
0X05000001	Sensor Filter: Digital filter Bandwidth/Hz		ID-Chip
0X05000002	Sensor Filter: Digital filter order		
0X05000101	Sensor Filter: User filter parameter A0		
0X05000102	Sensor Filter: User filter parameter A1		
0X05000103	Sensor Filter: User filter parameter B0		
0X05000104	Sensor Filter: User filter parameter B1		
0X05000105	Sensor Filter: User filter parameter B2		
0x06000902	Target Manipulation: Aux-Input to target factor		
0X07000000	Servo: Range min limit (µ)		ID-Chip
0X07000001	Servo: Range max limit (µ)		ID-Chip
0X07000200	Servo: Servo loop slew rate (axis unit/ms)		ID-Chip
0X07000300	Servo: Servo loop P-Term		ID-Chip
0X07000301	Servo: Servo loop I-Term		ID-Chip
0X07000500	Servo: Position from sensor 1		ID-Chip
0X07000501	Servo: Position from sensor 2		ID-Chip
0X07000502	Servo: Position from sensor 3		ID-Chip
0X07000600	Servo: Axis name		ID-Chip
0X07000601	Servo: Axis unit		ID-Chip
0X07000800	Servo: servo ON/OFF start up	0 = Disabled 1 = Enabled	ID-Chip
0X07000900	Servo: Tolerance		ID-Chip
0X07000A00	Servo: Auto-Zero driving low voltage (V)		ID-Chip
0X07000A01	Servo: Auto-Zero driving high voltage (V)		ID-Chip
0X07000A02	Servo: AutoZero voltage [V]		
0X07000C00	Servo: Default position		
0X07000C01	Servo: Default voltage		
0X07010200	Servo: User origin		
0X08000100	Servo output filter: Notch frequency of filter nr. 1		ID-Chip

Parameter Number	Parameter Name	Range	Notes
0X08000101	Servo output filter: Notch frequency of filter nr. 2		ID-Chip
0X08000200	Servo output filter: Notch rejection of filter nr. 1		ID-Chip
0X08000201	Servo output filter: Notch rejection of filter nr. 2		ID-Chip
0X08000300	Servo output filter: Notch bandwidth of filter nr. 1		ID-Chip
0X08000301	Servo output filter: Notch bandwidth of filter nr. 2		ID-Chip
0X09000000	Output Matrix: Driving with piezo 1		ID-Chip
0X09000001	Output Matrix: Driving with piezo 2		ID-Chip
0X09000002	Output Matrix: Driving with piezo 3		ID-Chip
0X09000003	Output Matrix: Driving with piezo 4		ID-Chip
0X0C000000	Piezo: Output voltage low limit (V)		ID-Chip
0X0C000001	Piezo: Output voltage high limit (V)		ID-Chip
0X0D000600	System Local: Device ID		ItemID = 1 only
0X0E000100	System Global: Sensor sampling time		ItemID = 1 only
0X0E000200	System Global: Servo update time		ItemID = 1 only
0X0E000A00	System Global: Min temperature		ItemID = 1 only
0X0E000A01	System Global: Max temperature		ItemID = 1 only
0X0F000000	System Mechanic: Read ID-Chip always	0 = Disabled 1 = Enabled	
0x13000001	Wave Generator: Installed wave form		read-only; only in volatile memory
0x13000002	Wave Generator: Connected axis		read-only only in volatile memory
0x13000003	Wave Generator: Wave generator cycles		
0x13000004	Wave Generator: Max Wave Points		read-only only in volatile memory
0x13000102	Wave Generator: Total wave form points		only in volatile memory
0x13000109	Wave Generator: Wave generator table rate		

Parameter Number	Parameter Name	Range	Notes
0x1300010A	Wave Generator: Number of wave tables		read-only; only in volatile memory; ItemID = 1 only
0x1300010B	Wave Generator: Curve offset		
0X16000000	Data Record: Table rate		ItemID = 1 only

9 Troubleshooting

Problem	Possible Causes	Solutions
Hardware conflicts	Board creeps out of its connector	Screw down the board.
	Board is installed in an improper slot	Install the board in a different slot. You should not use the slot directly under the graphics card because it may be shared with the slot of the graphics card.
	Heat emission of the board affects other hardware components in the PC	Skip the next slot facing the E-761 heat sink or use it for a card which is so short that it does not extend into the heat sink area. You can also install an additional fan in the PC case. To protect your equipment, you can reduce the maximum allowed temperature for the board by changing value of the "Max temperature" parameter (ID 0x0E000A01). Note that the high voltage output of the board will be automatically deactivated if the current board temperature exceeds the allowed range. See also " Stage does not move → The high voltage output of the board is deactivated, while the communication with the board is still possible (LED is turned off) → If error code 603 is returned …" in this table.
	PC switches itself off / malfunctions unexpectedly while the connection to the board is opened or the board is rebooted.	Make sure that the board connection to the PC power supply is not shared with another device. Otherwise the supply power for the board may be insufficient, and the PC or the board may malfunction. If the problems continue to be, you should replace your PC power supply by a device with higher power output. Contact your Physik Instrumente sales engineer or write info@pi.ws.
Communi-cation with controller does not work	Another program still uses the PCI interface	Close the other program.
	Undefined state where connection with device fails	Press the *Reboot* button in the *NanoCapture™ Device Connection* window (see Fig. 4 on p. 22).
	Wrong board number selected when trying to connect.	Select the correct board number.
	Specific software has problems with operating system.	Compare if another software is running, e.g. a terminal or development environment. You can, for example, test the communication by simply starting a terminal program, e.g. *WinTerm32*, and entering commands like *IDN? (p. 75) or HLP? (p. 83). Note that the commands are transferred as terminated by a line feed LF character. The command is executed only after the LF is received.

Stage does not move	Cable not connected or connected to wrong connector (if adapter cable is used)	Check the connecting cable
	Adapter cable is defective	Connect the stage to a different connector of the adapter cable to test its function
	Stage or stage cable is defective	Exchange stage with a working stage to test a new combination of board and stage (only with stages which are equipped with ID-chips)
	Wrong axis commanded	Check if commanded axis is that of the desired stage
	Axes are not available for commanding	Check the axis configuration with CST?. If there is returned "NOSTAGE" for the axis you want to use, configure the axis with CST (see p. 78). The axis can then be commanded.
	The high voltage output of the board is deactivated, while the communication with the board is still possible (LED is turned off).	Check the error state with ERR?: If error code 602 is returned (PI_CNTR_HW_VOLTAGE_ERROR), some internal voltage value on the board is out of range. Shut down the PC and disconnect it from the line voltage. Open the case and check if the E-761 is connected to the PC power supply (see also p. 19, step 5). If the error persists even though the board is properly connected to the PC power supply, it might be necessary to replace the PC power supply by a device with higher power output. Contact your Physik Instrumente sales engineer or write info@pi.ws. If error code 603 is returned (PI_CNTR_HW_TEMPERATURE_ERROR), the board temperature is out of the allowed range which is set by the "Min temperature" (ID 0x0E000A00) and "Max temperature" (ID 0x0E000A01) parameters. Wait a few minutes to let the board cool down. If the LED turnes on, the high voltages output is activated again, otherwise the temperature is still out of range. Note that the wave generator output will continue even if the high voltage output is deactivated, i.e. if a certain number of output cycles was set, the output may be already finished when the high voltage output is reactivated. If you have two E-761 boards installed side by side, the temperature may be permantently out of range. In this case install the boards so that at least one slot is skipped between them. When using the wave generator, it is recommended to reduce the frequency and/or the amplitude and/or the output duration to avoid overheating. See Section "Wave Generator" on p. 52 for more information. See also "Hardware conflicts → Heat emission of the board affects other hardware components in the PC" in this table.
Self-written program with PI drivers does not run.	Wrong combination of driver routines/Vis.	Check if system runs with PI Control or Terminal program. If yes read software manual or compare sample code to check the necessary driver routines.

Still problems? Please call your local distributor or write to info@pi.ws and know the following about your system:

Product codes and serial numbers of all used products

Current firmware version of the controller

Software version of driver or host software

Operating system

10 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.

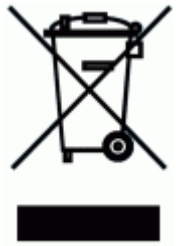
To meet the manufacturer's product responsibility with regard to this product, Physik Instrumente (PI) GmbH & Co. KG will ensure environmentally correct disposal of old PI equipment that was first put into circulation after 13 August 2005, free of charge.

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



11 Technical Data

11.1 Specifications

	E-761.3CD / E-761.3CT	Tolerance
Function	Digital NanoAutomation® PZT controller and power amplifier PCI board	
Axes	3	
Processor	DSP 32-bit floating point, 60 MHz	
Sampling rate, servo-control	25 kHz, with oversampling factor 4 (default)	
Sampling rate, sensor	25 kHz, with oversampling factor 4 (default)	
Sensor		
Servo characteristics	PI, two notch filters	
Sensor type	Capacitive	
Sensor channels	3	
Sensor resolution	16 bit	
Ext. synchronization	Yes	
Amplifier		
Min. output voltage	-20 V	
Max. output voltage	120 V	
Amplifier channels	4	
Peak output power per channel	5.3 W	max.
Average output power per channel	1.7 W	max.
Peak output current 20 ms, per channel	50 mA	
Average output current per channel	10 mA	
Current limitation	Short-circuit proof	
Resolution DAC	24 bit	
Interfaces and operation		
Communication interfaces	PCI slot	
Piezo connector	Sub-D special connector	
Sensor connector	Sub-D special connector	
Analog input	-10 to +10 V, 16-bit ADC resolution LEMO (EPG.0B.307.HLN)	

	E-761.3CD / E-761.3CT	Tolerance
Digital output	3 x LVTTL available with firmware revision 2.0.1.0 and newer; requires trigger output bracket with Sub-D connector, 9-pin (included in E-761.3CT; order separately as E-761.00T)	
Command set	PI General Command Set (GCS) 1.0	
User software	NanoCapture™, PZTControl™	
Software drivers	LabVIEW Driver, Libraries for Windows (DLL) and Linux	
Supported functionality	Wave-Gen, Trigger Output	
Display	Status LED for high voltage output	
Linearization	4th order polynomials	
Miscellaneous		
Operating temperature range	5 to 50°C (over 40°C, max. av. power derated 10%); deactivation of the piezo voltage output if the internal temperature exceeds 60°C	
Mass	E-761 PCI board: 564 g	
Dimensions	E-761 PCI board: 287 x 108 x 25 mm; 2 free slots required Trigger output bracket (w/o cables): 122 x 45x 26 mm; 1 free slot required	
Power consumption max.	20 W, 4 A	
Power consumption min.	10 W, 2.1 A	
Operating voltage	5 V (from PC power supply)	

11.2 Pinouts of the E-761 PCI Board

11.2.1 Sub-D Special Connector

The sub-D special connector which is used to connect piezo stages to the controller is illustrated below:

Sub-D Mix Connector with 3 coax lines and 22 single pins:

Pin	Signal
Coax inner lines	
A1	Sensor Probe Ch 2
A2	Sensor Probe Ch 3
A3	Sensor Probe Ch 1
Standard pins	
1	Sensor Target Ch 2
12	Sensor Target Ch 2 shield
2	Sensor Target Ch 3
13	Sensor Target Ch 3 shield
3	GND
14	nc
4	ID-Chip Axis 1
15	nc
5	GND
16	GND
6	ID-Chip Axis 2
17	ID-Chip Axis 3
7	Piezo Output Ch 4
18	GND Piezo
8	Piezo Output Ch 3
19	GND Piezo
9	Piezo Output Ch 2
20	GND Piezo
10	Piezo Output Ch 1
21	GND Piezo
11	Sensor Target Ch 1
22	Sensor Target Ch 1 shield

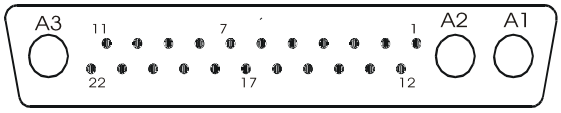


Fig. 19: Sub-D Mix Connector with 3 coax lines and 22 single pins, front view

DANGER

Up to 120 V can be present on the Piezo Output lines.

11.2.2 Analog Input Socket

Using the ANALOG IN socket (EPG.0B.307.HLN LEMO) analog input can be transferred to the controller. The analog input can be used for control value generation ("direct" axis control and "external wave generator") or for triggering the internal wave generator. See p. 33 for details. An appropriate cable with LEMO connector and open end comes with the E-761 (order# K040B0077).

Pin	Signal
1	internal use
2	AGND
3	internal use
4	internal use
5	internal use
6	internal use
7	Analog Input, -10 V to +10 V DC

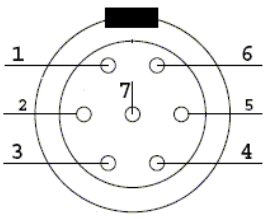


Fig. 20: EPG.0B.307.HLN LEMO *socket, front view*

11.2.3 Connectors J6 and J7—Sync and Digital Output

The connectors J6 and J7 carry synchronization lines. With firmware rev. 2.0.1.0 and newer, J6 in addition carries three digital output lines.

NOTE

To use the digital output lines, a bracket with Digital OUT socket is required (see Section 11.3.1 on p. 147). This trigger output bracket is included in E-761.3CT and can also be ordered separately as E-761.00T.

J6 and J7 can be connected as follows:

- If no trigger output brackets are available:
  When it is necessary to synchronise multiple E-761 boards, interconnect their connectors J6 and J7 inside the PC housing using the synchronisation cable (order #K010B0029). See p. 37 for connection details and connector locations on the board.

- If trigger output brackets are available:
  Connect J6 of the E-761 to J6 on the PCB of the corresponding trigger output bracket, and J7 to J7. See Section 2.1.2 starting on p. 17 for connection details and connector locations.
  To synchronize multiple E-761 boards, use the Sync sockets on the trigger output brackets, see p. 37 for details and p. 147 for pinout.

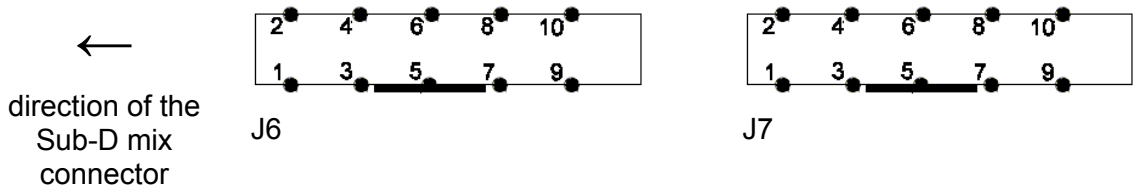


Fig. 21: Connector front view when the E-761 is installed in PCI slot

J6 Pinout		J7 Pinout	
Pin	**Signals**	**Pin**	**Signals**
1	TrigOut3	1	reserved
2	GND	2	GND
3	TrigOut2	3	reserved
4	GND	4	GND
5	TrigOut1	5	reserved
6	GND	6	GND
7	SYNC 100 kHz	7	SYNC 100 kHz
8	GND	8	GND
9	DC_SYNC 200 kHz	9	DC_SYNC 200 kHz
10	GND	10	GND

11.3 Pinouts of the Trigger Output Bracket

The trigger output bracket is required to make the three digital output lines of the E-761 available outside of the PC. Furthermore, it provides access to the synchronization lines of the E-761.

NOTE

This bracket is included in E-761.3CT and can also be ordered separately as E-761.00T.

The digital output lines are available with firmware rev. 2.0.1.0 and newer.

Inside of the PC, the connectors J6 and J7 of the E-761 must be connected to the connectors J6 and J7 of the PCB on the trigger output bracket to carry the signals from the E-761 board to the Digital OUT and Sync sockets. See Section 2.1.2 starting on p. 17 for connection details and connector locations.

11.3.1 Digital OUT Socket

The Digital OUT socket (Sub-D, 9-pin, female) on the trigger output bracket provides three lines for trigger signals. Those signals can be output by the E-761 in conjunction with wave generator usage (see Section 5 starting on p. 52 and WGO, p. 114, for how to use digital output).

Pin	Signal
1	TrigOut1 (belongs to wave generator 1 (axis 1))
6	AGND
2	TrigOut2 (belongs to wave generator 2 (axis 2))
7	AGND
3	TrigOut3 (belongs to wave generator 3 (axis 3))
8	AGND
4	Reserved
9	Reserved
5	Reserved

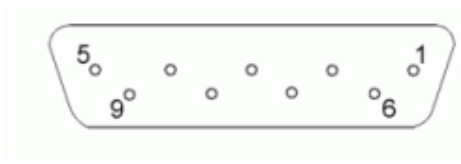


Fig. 22: Sub-D 9-pin (f) Connector, front view

11.3.2 Sync Out and Sync In Sockets

The Sync sockets (EPL.0S.303.HLN LEMO) can be used to synchronize multiple boards, see p. 37 for details.

Pin	Signal
1	AGND
2	DC_SYNC 200 kHz
3	SYNC 100 kHz

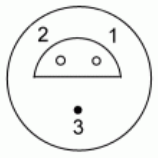


Fig. 23: EPL.0S.303.HLN LEMO *socket, front view*

11.4 Dimensions

Dimensions in mm, decimal places separated by commas in drawings.

11.4.1 E-761 PCI Board

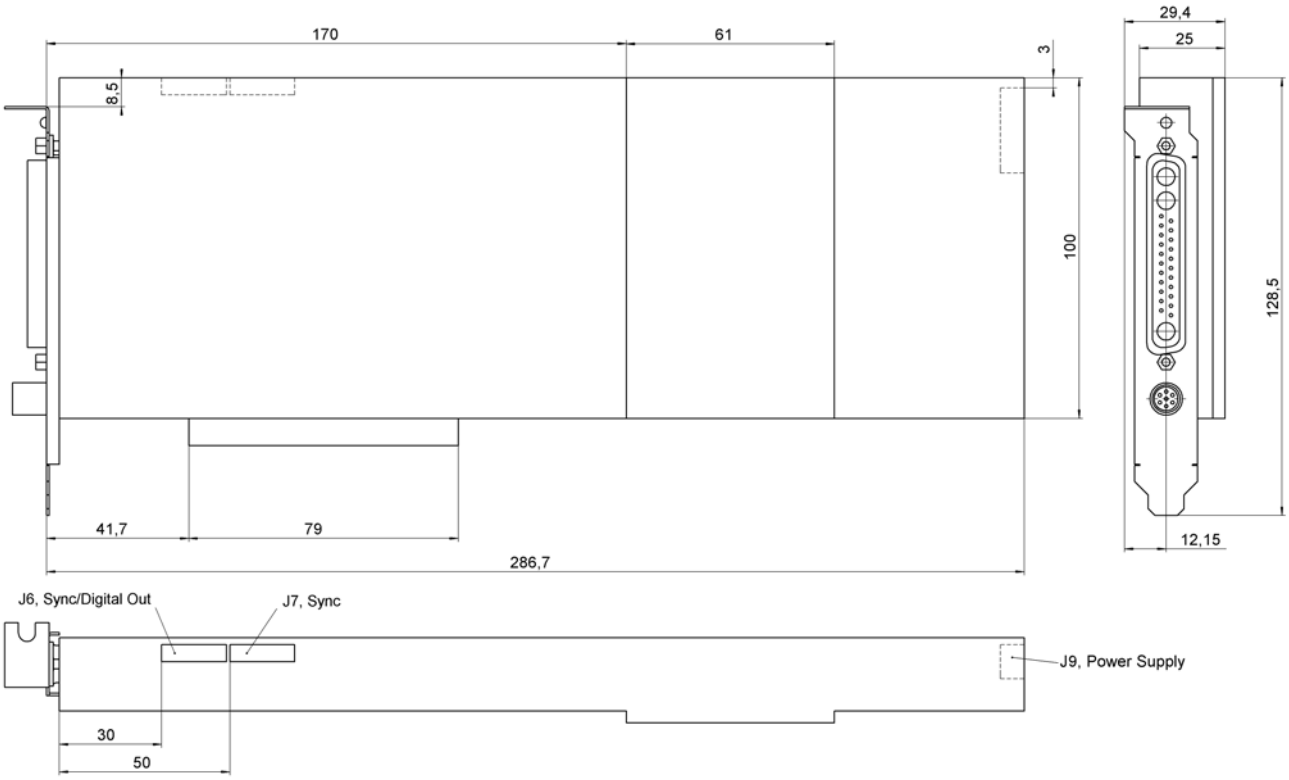


Fig. 24: E-761 dimensions

11.4.2 Additional Bracket for Digital Output

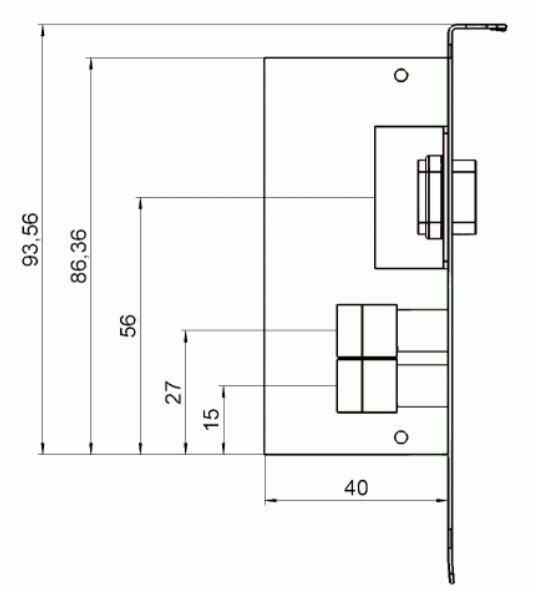


Fig. 25: Dimensions of the additional bracket for digital output

12 Terminology Table

Term	Definition
axis	one of the set of orthogonal logical axes for which servo-control can be enabled
actual position	position values for an axis as measured by the sensors, digitized by the ADC and calculated according to the sensor-to-axis transformation matrix (see Section 3.3.3 on p. 43)
current position	position at this very moment
curve	a series of data points within a waveform or segment forming one ramp, single scan line or sine wave
data point	one of an ordered collection of amplitude values in data point memory
data point memory	ordered collection of data points; can be a wave table (contains waveform points) or a record table (contains actual position values recorded by WGO, WGR, IMP or STE)
piezo channel	the representation of a PZT amplifier in the firmware. Multiple PZT amplifiers can be involved in the motion of one logical axis.
segment	a group of contiguous data points in a wave table as defined with the WAV command, one or more segments form a waveform
sensor channel	the representation of a physical existing sensor in the firmware. Multiple sensor channels can be involved in the control of one logical axis.
wave generator	one of four internal function blocks which can, on command, send a series of control values (the wave table content) for the axes. In closed-loop operation (servo ON), the control values are interpreted as target positions in either case. In open-loop operation (servo OFF), the interpretation of the wave generator output depends on the settings of the axis-to-PZT matrix (see "Output Generation", p. 46 for more information). By default, the matrix is set up so that commanded open-loop control values numerically correspond to axis position values.
waveform	the content of a wave table (series of up to 8192 data points) which can be sent sequentially as control values by a wave generator