E-727 Digital Multi-Channel Piezo Controller

Contents

About this Document ................................................................. 4
Symbols and Typographic Conventions ..................................................... 4
Other Applicable Documents ................................................................. 4
Downloading Manuals ........................................................................ 5

Safety ............................................................................................. 6
Intended Use ..................................................................................... 6
General Safety Instructions ..................................................................... 6
Organizational Measures ....................................................................... 7
Personnel Qualification ......................................................................... 7

Product Description ........................................................................... 8
Model Overview .................................................................................. 8
Product View ....................................................................................... 8
Scope of Delivery ............................................................................... 11
Axes, Channels, Functional Elements .................................................... 11
ID Chip Detection ............................................................................... 15
Overview of PC Software ..................................................................... 15
Installation

General Notes on Installation ................................................................. 17
Installing the PC Software ................................................................. 17
  Performing the Initial Installation ...................................................... 17
  Installing Updates ............................................................................ 19
Ensuring Ventilation ........................................................................... 21
Connecting the E-727 to the Protective Earth Conductor ....................... 21

Quick Start

General Notes on Start-Up ................................................................... 22
Starting the System in PIMikroMove .................................................. 23
Creating Backup Files for Controller Parameters .................................. 25
Executing Test Motions in Open-Loop Operation .................................. 26
Adjusting the Notch Filter(s) in Open-Loop Operation ............................ 27
Checking and Optimizing the Servo-Control Parameters ...................... 33
Saving Optimized Settings in the Single-Axis Window ......................... 36

Operation

Communication ...................................................................................... 37
  Communication via the TCP/IP Interface .......................................... 37
  Communication via the RS-232 Interface ......................................... 39
  Communication via the USB Interface ............................................. 40
AutoZero Procedure ........................................................................... 41
Overtemp Protection of the E-727 ....................................................... 44
Data Recording .................................................................................... 44
External Triggering/Signaling .............................................................. 45
Using the Analog Input ....................................................................... 45
Using the Analog Output ..................................................................... 46
Wave Generator .................................................................................. 47
Dynamic Digital Linearization (DDL) ................................................. 47
Controller Macros ............................................................................... 47
  Commands for Macros .................................................................. 48
  Working with Macros .................................................................... 49
Variables ............................................................................................. 56

GCS Commands

Notation .............................................................................................. 58
GCS Syntax for Syntax Version 2.0 ...................................................... 58
Command Overview ............................................................................ 58
Command Descriptions ....................................................................... 63

Parameters

Parameter Handling ............................................................................ 87
Important New Parameters .................................................................. 87
Technical Note
E727T0005, valid for E-727.3CD, E-727.3CDA, E-727.3SD, E-727.3SDA
BRO, 2015-10-07

Maintenance 90
Updating Firmware ................................................................. 90
Cleaning the E-727 .................................................................... 91
Changing the Fuse ..................................................................... 91

Technical Data 92
Specifications ........................................................................... 92
Maximum Ratings ........................................................................ 94
Ambient Conditions and Classifications ..................................... 94
Dimensions ............................................................................... 95
Pin Assignment .......................................................................... 96
  E-727.3CD, E-727.3CDA: Socket for Piezo Stages ....................... 96
  E-727.3SD, E-727.3SDA: Socket for Piezo Stages ....................... 97
  Digital I/O ............................................................................. 98
  E-727.3CDA, E-727.3SDA: Analog I/O ....................................... 99
  RS-232 ............................................................................... 100
  Power Supply 24 V ............................................................... 101

EC Declaration of Conformity 102

Customer Service 102
About this Document

This Technical Note contains information on the intended use of the E-727.

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

The latest versions of Technical Notes and user manuals are available for download on our website (www.pi.ws).

Symbols and Typographic Conventions

The following symbols and markings are used in this Technical Note:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>☢️</td>
<td>Danger</td>
</tr>
<tr>
<td>☢️</td>
<td>If not avoided, the hazardous situation will result in death or serious injury.</td>
</tr>
<tr>
<td>🔄️</td>
<td>Notice</td>
</tr>
<tr>
<td>🔄️</td>
<td>If not avoided, the dangerous situation will result in damage to the equipment.</td>
</tr>
<tr>
<td>1.</td>
<td>Action consisting of several steps whose sequential order must be observed</td>
</tr>
<tr>
<td>2.</td>
<td>Action consisting of one or several steps whose sequential order is irrelevant</td>
</tr>
<tr>
<td>•</td>
<td>List item</td>
</tr>
<tr>
<td>p. 5</td>
<td>Cross-reference to page 5</td>
</tr>
<tr>
<td>SVO?</td>
<td>Command line or command from PI's General Command Set (GCS) (example: command to get the servo mode)</td>
</tr>
<tr>
<td>RS-232</td>
<td>Operating element labeling on the product (example: socket of the RS-232 interface)</td>
</tr>
<tr>
<td>Device S/N</td>
<td>Parameter name (example: parameter where the serial number is stored)</td>
</tr>
<tr>
<td>Start &gt; Settings</td>
<td>Menu path in the PC software (example: to open the menu, the Start and Settings buttons must be clicked in succession)</td>
</tr>
<tr>
<td>S</td>
<td>Value that must be entered or selected via the PC software</td>
</tr>
</tbody>
</table>

Other Applicable Documents

See the user manual for the E-725.3CD digital piezo controller (document PZ197E; available as PDF file on the E-727 CD) for the following:

- Basic structure (commands, parameters, assignment of channels to axes) and functional principles of a digital piezo controller
- Descriptions of most commands and parameters
Technical Note
E727T0005, valid for E-727.3CD, E-727.3CDA, E-727.3SD, E-727.3SDA
BRO, 2015-10-07

- Communication Details
- Data Recording
- External Triggering/Signaling
- Using the Analog Input
- Wave Generator
- Dynamic Digital Linearization (DDL)

For details regarding the PC software, see the following documents:

<table>
<thead>
<tr>
<th>Software</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI GCS 2 DLL</td>
<td>SM151E Software Manual</td>
</tr>
<tr>
<td>PIMikroMove</td>
<td>SM148E Software Manual</td>
</tr>
<tr>
<td>E-727 GCS LabVIEW Driver Set</td>
<td>PZ260E Software Manual</td>
</tr>
<tr>
<td>GCS Data</td>
<td>SM146E Software Manual</td>
</tr>
<tr>
<td>Updating PI Software</td>
<td>A000T0032 Technical Note</td>
</tr>
<tr>
<td>PI Update Finder</td>
<td>A000T0028 Technical Note</td>
</tr>
</tbody>
</table>

**Downloading Manuals**

**INFORMATION**

If a manual is missing or problems occur with downloading:
- Contact our customer service department (p. 102).

**INFORMATION**

For products that are supplied with software (CD in the scope of delivery), access to the manuals is protected by a password. Protected manuals are only displayed on the website after entering the password.

The password is included on the CD of the product.

**For products with CD: Identify the password**

1. Insert the product CD into the PC drive.
2. Switch to the Manuals directory on the CD.
3. In the Manuals directory, open the Release News (file including releasenews in the file name).
4. Find the user name and the password in the section "User login for software download" in the Release News.

**Downloading Manuals**

1. Open the website http://www.pi.ws.
2. Click Info.
3. If you have a user name and password:
   a) Click Login.
   b) Log in with the user name and password.
4. Click >> Product Downloads.
5. In the Product Files area, click the corresponding product category.
6. On the right-hand side of the page, select the corresponding sub-category.
7. Navigate to the product code on the page.
   The following manuals are displayed:
   − Freely accessible manuals
   − Manuals that are protected by a password
8. Click the desired manual and save it to the hard disk of your PC or to a data storage medium.

Safety

Intended Use

The E-727 is a laboratory device according to DIN EN 61010-1. It is intended to be used in interior spaces and in an environment which is free of dirt, oil and lubricants.

The E-727 is designed and intended for driving capacitive loads (e.g. piezo ceramic actuators).

The E-727 must not be used for purposes other than those named in this Technical Note. In particular, the E-727 must not be used to drive ohmic or inductive loads.

The E-727 can be used for static as well as dynamic applications.

Depending on the model, capacitive sensors or strain gauge sensors must be used for closed-loop operation. PI stages intended for closed-loop operation already have the corresponding sensors. Other sensors can only be used with PI approval.

General Safety Instructions

The E-727 is built according to state-of-the-art technology and recognized safety standards. Improper use can result in personal injury and/or damage to the E-727.

➢ Only use the E-727 for its intended purpose, and only use it if it is in a good working order.
➢ Read the user documentation (Technical Note, user manuals).
➢ Immediately eliminate any faults and malfunctions that are likely to affect safety.

The operator is responsible for the correct installation and operation of the E-727.

➢ Install the E-727 near the power source so that the power plug can be quickly and easily disconnected from the mains.
➢ Use the supplied components (adapter) to connect the E-727 to the power source.
➢ If one of the supplied components for connecting to the power source has to be replaced, use a sufficiently dimensioned component.
➢ Only use cables and connections that meet local safety regulations.
If a protective earth conductor is not or not properly connected, dangerous touch voltages can occur on the E-727 in the case of malfunction or failure of the system. If touch voltages exist, touching the E-727 can result in serious injury or death from electric shock.

- Connect the E-727 to a protective earth conductor before start-up (p. 21).
- Do not remove the protective earth conductor during operation.
- If the protective earth conductor has to be removed temporarily (e.g., in the case of modifications), reconnect the E-727 to the protective earth conductor before starting it up again.

**Organizational Measures**

User documentation (Technical Note, user manuals):

- Always keep this user documentation available by the E-727.
- The latest versions of the user documentation are available from PI.
- Add all information given by the manufacturer to the user documentation, for example supplements or Technical Notes.
- If you pass the E-727 on to other users, also turn over the user documentation as well as other relevant information provided by the manufacturer.
- Only use the device on the basis of the complete user documentation. Missing information due to an incomplete user documentation can result in serious or fatal injury as well as property damage.
- Only install and operate the E-727 after having read and understood this user manual.

**Personnel Qualification**

The E-727 may only be installed, started up, operated, maintained and cleaned by authorized and qualified staff.
Product Description

Model Overview

There are 4 standard versions of the E-727. They differ in regards to the supported sensor type and the available analog I/O lines.

<table>
<thead>
<tr>
<th>Model</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-727.3CD</td>
<td>Digital Multi-Channel Piezo Controller, 3 Channels, -30 to 130 V, Sub-D Connector(s), Capacitive Sensors</td>
</tr>
<tr>
<td>E-727.3CDA</td>
<td>Digital Multi-Channel Piezo Controller, 3 Channels, -30 to 130 V, Sub-D Connector(s), Capacitive Sensors, Analog Inputs</td>
</tr>
<tr>
<td>E-727.3SD</td>
<td>Digital Multi-Channel Piezo Controller, 3 Channels, -30 to 130 V, Sub-D Connector(s), Strain Gauge Sensors</td>
</tr>
<tr>
<td>E-727.3SDA</td>
<td>Digital Multi-Channel Piezo Controller, 3 Channels, -30 to 130 V, Sub-D Connector(s), Strain Gauge Sensors, Analog Inputs</td>
</tr>
</tbody>
</table>

Product View

Figure 1: Front panel of E-727 digital piezo controllers

<table>
<thead>
<tr>
<th>Labeling</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
</table>
| Power    | Toggle switch | Power on/off switch:  
- ○ position: E-727 is switched off  
- | position: E-727 is switched on |
<table>
<thead>
<tr>
<th>Labeling</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fuse</strong></td>
<td>Slotted fuse holder</td>
<td>For cartridge fuse T3.15 AH, 250 V, 5 x 20 mm, changing the fuse see p. 91</td>
</tr>
<tr>
<td><strong>24 VDC / 3.5 A</strong></td>
<td>M8 panel plug, 4-pin (p. 101)</td>
<td>Connection for the supply voltage. To be used with the K050B0003 adapter (included in the scope of delivery, p. 11)</td>
</tr>
<tr>
<td><img src="image" alt="M4 hole with fastening material for protective earth conductor" /></td>
<td>Protective earth connection (p. 21) A protective earth conductor must be connected to the E-727 via the M4 hole and the fastening material, since the E-727 is not grounded via the power supply connector.</td>
<td></td>
</tr>
</tbody>
</table>

**Digital I/O**

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MDR14 (f)</strong> (p. 98)</td>
<td>Digital lines:</td>
</tr>
<tr>
<td></td>
<td>Outputs: Triggering of external devices, output of the servo cycles</td>
</tr>
<tr>
<td></td>
<td>Inputs: Triggering of data recorder or wave generator, use in macros, reboot of E-727 (p. 98)</td>
</tr>
</tbody>
</table>

**Power**

<table>
<thead>
<tr>
<th>LED green/off</th>
<th>Power-on and ready indicator:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Continuously lit: E-727 is ready for normal operation</td>
</tr>
<tr>
<td></td>
<td>Continuously off: E-727 is not connected to the supply voltage</td>
</tr>
<tr>
<td></td>
<td>Alternately lit/off/lit: E-727 performs power-on or reboot sequence</td>
</tr>
</tbody>
</table>

**Error**

<table>
<thead>
<tr>
<th>LED red/off</th>
<th>Error indicator:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Continuously lit: Error (error code ≠ 0)</td>
</tr>
<tr>
<td></td>
<td>Continuously off: No error (error code = 0)</td>
</tr>
<tr>
<td></td>
<td>Alternately lit/off: E-727 performs power-on or reboot sequence</td>
</tr>
</tbody>
</table>

The error code can be queried with the **ERR?** command. The query resets the error code to zero and the LED is switched off.

**Ch1/2 OFL Ch3/4 OFL**

<table>
<thead>
<tr>
<th>LED yellow/off</th>
<th>Overflow indicator for the axes:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Continuously lit: At least one of the axes is in overflow state</td>
</tr>
<tr>
<td></td>
<td>Continuously off: No axis is in overflow state</td>
</tr>
<tr>
<td></td>
<td>Flashing: E-727 searches for a DHCP server during power-on or reboot sequence</td>
</tr>
</tbody>
</table>

The overflow state of the individual axes can be queried with the **OVF?** command. The overflow state can only occur in closed-loop operation. In the overflow state, the axis does not reach the target position because the amplifier(s)
## Labeling

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED green/off</td>
<td>has/have reached the range limit. In this case, readjustment of the sensor zero-point is necessary, using the AutoZero functionality provided by the E-727 firmware (details see p. 41). For an axis in overflow state, the corresponding bit in the response to the #5 command (Request Motion Status) is <strong>not</strong> set (motion state = &quot;not moving&quot;).</td>
</tr>
</tbody>
</table>
| Ch1/2 Servo | Servo mode indicator for the axes:  
  - Continuously lit: Servo mode is on (closed-loop operation) for at least one of the axes  
  - Continuously off: Servo mode is off (open-loop operation) for the axes  
  - Flashing: E-727 initializes parameters and – if the OFL LEDs are flashing in addition - searches for a DHCP server during power-on or reboot sequence. The servo mode of the individual axes can be queried with the SVO? command. |
| Ch3/4 Servo | Servo mode indicator for the axes:  
  - Continuously lit: Servo mode is on (closed-loop operation) for at least one of the axes  
  - Continuously off: Servo mode is off (open-loop operation) for the axes  
  - Flashing: E-727 initializes parameters and – if the OFL LEDs are flashing in addition - searches for a DHCP server during power-on or reboot sequence. The servo mode of the individual axes can be queried with the SVO? command. |
| USB-B socket | Universal serial bus for connection to the PC |
| SPI | Display port | Availability on request from our customer service department (p. 102). |
| -30 V to +130 V | Socket for piezo stages; carries the voltage for the piezo actuators (-30 to 130 V) and the signals of the sensors in the mechanics. |
Labeling

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog I/O</td>
<td>E-727.3CDA, E-727.3SDA: Sub-D 15 (f) (p. 99)</td>
</tr>
<tr>
<td></td>
<td>Only E-727.3CDA and .3SDA - analog lines:</td>
</tr>
<tr>
<td></td>
<td>- Inputs: Used for external sensors or as analog control inputs</td>
</tr>
<tr>
<td></td>
<td>- Outputs: Three sensor monitor lines, 4th line can be used to monitor the position of an axis or for controlling an external motor driver</td>
</tr>
<tr>
<td>RS-232</td>
<td>Sub-D 9 (m) (p. 100)</td>
</tr>
<tr>
<td></td>
<td>Serial connection to PC</td>
</tr>
<tr>
<td>RJ45 socket</td>
<td>Ethernet interface for communication via TCP/IP, see p. 37 for details</td>
</tr>
</tbody>
</table>

Scope of Delivery

<table>
<thead>
<tr>
<th>Item number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-727</td>
<td>Digital piezo controller according to the order</td>
</tr>
<tr>
<td>000023194</td>
<td>Separate 24 V wide-range-input power supply (120 W/5 A) for use with line voltages from 100 to 240 VAC and voltage frequencies of 50 or 60 Hz, with barrel connector</td>
</tr>
<tr>
<td>3763</td>
<td>Power cord</td>
</tr>
<tr>
<td>K050B0003</td>
<td>Adapter for the power supply connection; barrel connector to M8 4-pin connector</td>
</tr>
<tr>
<td>C-815.34</td>
<td>RS-232 Null-Modem Cable, 3 m, 9/9-pin</td>
</tr>
<tr>
<td>C-815.563</td>
<td>Cross-over network cable for direct connection with the PC via TCP/IP</td>
</tr>
<tr>
<td>000011448</td>
<td>USB cable (type A to type B) for connection to the PC</td>
</tr>
<tr>
<td>E-727.CD</td>
<td>Product CD with software and user manuals for the E-727</td>
</tr>
<tr>
<td>E727T0005</td>
<td>Technical Note for the E-727, this document</td>
</tr>
</tbody>
</table>

Axes, Channels, Functional Elements

The following table contains the items that can be accessed with commands of the PI General Command Set (GCS).

<table>
<thead>
<tr>
<th>Item</th>
<th>Number</th>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical axis</td>
<td>3</td>
<td>1 to 3</td>
<td>The logical axes represent the motion of the stage in the firmware of the E-727. A logical axis corresponds to an axis of a linear coordinate system. All commands for the motion of a stage refer to logical axes.</td>
</tr>
</tbody>
</table>
**Item Number Identifier Description**

<table>
<thead>
<tr>
<th>Item</th>
<th>Number Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input signal channels</strong></td>
<td>Number depends on the E-727 model</td>
<td></td>
</tr>
<tr>
<td><strong>E-727.3CD:</strong></td>
<td>3 channels, intended for the capacitive sensors in the stage(s), input on the socket for piezo stages (p. 96).</td>
<td>The value of the <em>Number Of System Axes</em> parameter (ID 0x0E000B02) specifies the number of axes. The input and output signal channels of the E-727 are allocated to the logical axes via matrices (input matrix: parameters 0x07000500 to 0x07000506; output matrix: parameters 0x09000000 to 0x09000003). The E-727 can be configured for a 4th axis with the identifier 4 on request.</td>
</tr>
</tbody>
</table>
| **E-727.3CDA:** | 7 channels. Channels 1 to 3 are intended for the capacitive sensors in the stages(s), input on the socket for piezo stages (p. 96). Channels 4 to 7 are the analog inputs available on the Analog I/O socket (p. 99). The input range of the analog inputs can be configured via the value of the *Sensor Range Factor* parameter (ID 0x02000100) as follows:  
  1: ±5 V  
  2: ±10 V | |
| **E-727.3SD:** | 4 channels. Input on the socket for piezo stages (p. 97). Channels 1 to 3 are intended for the SGS sensors in the stage(s). Via the value of the *Sensor Range Factor* parameter (ID 0x02000100), channel 4 can be configured as follows:  
  1: Use with an SGS sensor (pins 6, 24, 25)  
  2: Use with a PT1000 temperature sensor (pins 1, 20) |  |
| **E-727.3SDA:** | 7 channels. Channels 1 to 3 are intended for the SGS sensors in the stages(s), input on the socket for piezo stages (p. 97). Via the value of the *Sensor Range Factor* parameter (ID 0x02000100), channel 4 can be configured as follows:  
  1: Use with an SGS sensor (input via pins 6, 24, 25 of the socket for the piezo stage(s) (p. 97))  
  2: Use with a PT1000 temperature sensor (input via pins 1, 20 of the socket for the piezo stage(s))  
  3: Use as analog input 1 with a range of ±5 V (input via pins 2 and 9 on the Analog I/O socket (p. 99))  
  4: Use as analog input 1 with a range of ±10 V (input via pins 2 and 9 on the Analog I/O socket)  
  Channels 5 to 7 are the analog inputs 2 to 4 available on the Analog I/O socket. Their input ranges can be configured via the value of the *Sensor Range Factor* parameter (ID 0x02000100) as follows: |  |
### General Notes:

The analog inputs on the Analog I/O socket can be used for an external sensor or as a control source, see “Using the Analog Input” in the E-725 user manual (PZ197E) for details.

The number of sensor channels available on the socket for the piezo stage can be queried via the **Number Of Sensor Channels** parameter (ID 0x0E000B03). Note that this number will change with E-727.3SDA models depending on the usage of channel 4.

The total number of input signal channels can be queried via the **Number Of Input Channels** parameter (ID 0x0E000B00).

### Output signal channels

<table>
<thead>
<tr>
<th>Item</th>
<th>Number</th>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
</table>
|              | 4      | 1 to 4     | E-727.3CD, E-727.3SD:  
The channels are intended for the piezo actuators in the stage(s), output on the socket for piezo stages (p. 96 or p. 97).  

**E-727.3CDA, E-727.3SDA:**  
Channels 1 to 3 are intended for the piezo actuators in the stage(s), output on the socket for piezo stages (p. 96 or p. 97).  
Via the value of the **Select Output Type** parameter (ID 0x0A000003), channel 4 can be configured as follows:  
- 1: Output voltage for a piezo actuator in the stage, output as Piezo Ch 4 on the socket for piezo stages (p. 96 or p. 97).
- 2: Position monitor of an axis, output on pin 8 of the Analog I/O socket (p. 99).
- 5: Control signal for an external motor driver, output on pin 8 of the Analog I/O socket.

Further details see “Using the Analog Output” (p. 46).

**General Notes:**  
The number of channels available on the socket for the piezo stage can be queried via the **Number Of Piezo Channels** parameter (ID 0x0E000B04). Note that this number will change with E-727.3CDA and .3SDA models depending on the usage of channel 4.

The total number of output signal channels can be queried via the **Number Of Output Channels** parameter (ID 0x0E000B01).  
Note that the sensor monitor lines 1 to 3 on the Analog I/O socket are **not** available as output signal channels in the firmware of the E-727 and **not** accessible for commands.

### Digital inputs

<table>
<thead>
<tr>
<th>Item</th>
<th>Number</th>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
</table>
|              | 4      | 1 to 4     | 1 to 4 identify digital input lines 1 to 4 of the Digital I/O socket (p. 98).  
Digital inputs 1 to 3 can be used to trigger the data recorder (DRT command) or wave generator output (WGO command). Furthermore, they
### Technical Note

E727T0005, valid for E-727.3CD, E-727.3CDA, E-727.3SD, E-727.3SDA

BRO, 2015-10-07

---

<table>
<thead>
<tr>
<th>Item</th>
<th>Number</th>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
</table>
| can be used in macros (MAC command). Via the **Reboot On DIO Input** parameter (ID 0x0E001500), the digital input 4 (pin 2 of the Digital I/O socket) can be configured for one of the following options:  
- Same functions as with digital inputs 1 to 3 (default setting)  
- Reset functionality: The input triggers a system reboot (active LOW). Controller behaves just like after power-on. |
| Digital outputs    | 3      | 1 to 3     | 1 to 3 identify digital output lines 1 to 3 of the Digital I/O socket (p. 98). These lines can be used to trigger external devices in conjunction with axis motion (CTO command and corresponding parameters (IDs 0x18000201 to 0x18000209)). Digital output 4 (pin 8 of the Digital I/O socket) is not accessible for commands. It outputs the servo cycles. The number of digital output lines that are accessible by commands can be queried via the **Number Of Trigger Outputs** parameter (ID 0x0E000805). |
| Wave generators    | 3      | 1 to 3     | The number of wave generators corresponds to the number of logical axes. Each wave generator is permanently allocated to a logical axis. If the E-727 is configured for a 4th axis, a 4th wave generator is also present. |
| Wave tables        | 40     | 1 to 40    | The wave tables contain the (temporarily) saved data (a total of 262144 points) for the waveforms that are output by the wave generators. The value of the **Number Of Waves** parameter (ID 0x1300010a) indicates the number of wave tables. |
| DDL tables         | 3      | 1 to 3     | The DDL tables contain the data of the Dynamic Digital Linearization (DDL) feature. The number of DDL tables corresponds to the number of logical axes. Each DDL table is permanently allocated to a logical axis. If the E-727 is configured for a 4th axis, a 4th DDL table is also present. The total number of points provided for the DDL tables is 262144, indicated by the **Max DDL Points** parameter, ID 0x1400000B. |
| Data recorder tables | ≤8    | 1, 2, ...  | The data recorder tables contain the recorded data (a total of 262144 points). The number of data recorder tables can be set with the **Data Recorder Chan Number** parameter (ID 0x16000300). The **Max Number Of Data Recorder Channels** parameter (ID 0x16000100) indicates the maximum number of data recorder tables. |
| Overall system     | 1      | 1          | E-727 as an overall system. |
ID Chip Detection

The piezo stage which is connected to the E-727 may contain an ID-chip (located in the stage connector). The following data is stored in the ID-chip (and cannot be modified there by the customer):

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

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 non-volatile and volatile memory upon controller power-on or reboot. Afterwards, the complete set of ID-chip parameters will be overwritten on power-on or reboot only if the **Power Up Read ID-Chip** option is enabled via the corresponding parameter (ID 0x0f000000) for all input signal channels associated with the stage. By default, this option is disabled to facilitate maintaining optimized parameter settings on the controller.

**INFORMATION**

When you connect a stage when the controller is powered on, the ID-chip of the stage is not read by the controller. To read the ID-chip data, the controller must be power-cycled or rebooted using the RBT command or the corresponding host software functions.

A piezo stage can be easily exchanged due to the functionality of the ID-chip.

Overview of PC Software

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

- Windows: Vista Service Pack 1 (32 bit, 64 bit), Windows 7, 8 and 10 (32 bit, 64 bit)
- Linux: Kernel 2.6, GTK 2.0, glibc 2.4 (configuration used to develop the PC software)

<table>
<thead>
<tr>
<th>PC software</th>
<th>Operating system</th>
<th>Short description</th>
<th>Recommended use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic program library for GCS</td>
<td>Windows, Linux (communication in Linux only via virtual COM port)</td>
<td>Allows software programming for the E-727 with programming languages such as e. g. C++. The functions in the dynamic program library are based on the PI General Command Set (GCS).</td>
<td>For users who would like to use a dynamic program library for their application. Is required for PIMikroMove. Is required for the LabVIEW drivers.</td>
</tr>
<tr>
<td>LabVIEW drivers</td>
<td>Windows, Linux</td>
<td>LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments). The E-727 LabVIEW software is a collection of virtual instrument drivers (VI drivers)</td>
<td>For users who want to use LabVIEW to program their application.</td>
</tr>
</tbody>
</table>
### Technical Note

<table>
<thead>
<tr>
<th>PC software</th>
<th>Operating system</th>
<th>Short description</th>
<th>Recommended use</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LabVIEW Merge Tool</strong></td>
<td>Windows</td>
<td>The LabVIEW Merge Tool allows you to combine product-specific LabVIEW drivers from PI with each other.</td>
<td>For users who want to operate several products from PI at the same time while using LabVIEW.</td>
</tr>
<tr>
<td><strong>PIMikroMove</strong></td>
<td>Windows</td>
<td>Graphic user interface for Windows with which the E-727 and other controllers from PI can be used: The system can be started without programming effort. Graph of motions in open-loop and closed-loop operation. Macro functionality for storing command sequences on the PC (host macros). Support of HID devices. Complete environment for command entry, for trying out different commands. No command knowledge is necessary to operate PIMikroMove. PIMikroMove uses the dynamic program library to supply commands to the controller. To provide the <strong>Device Parameter Configuration</strong> window, PIMikroMove requires the NI LabVIEW Run-Time Engine; see &quot;Performing the Initial Installation&quot; (p. 17).</td>
<td>For users who want to perform simple automation tasks or test their equipment before or instead of programming an application. A log window showing the commands sent makes it possible to learn how to use the commands.</td>
</tr>
<tr>
<td><strong>PI Terminal</strong></td>
<td>Windows</td>
<td>Terminal program that can be used for nearly all PI controllers (see the description of the <strong>Command Entry</strong> window in the PIMikroMove user manual).</td>
<td>For users who want to send GCS commands directly to the controller.</td>
</tr>
<tr>
<td><strong>PI Update Finder</strong></td>
<td>Windows</td>
<td>Checks the PI software installed on the PC. If more current versions of the PC software are available on the PI server, downloading is offered.</td>
<td>For users who want to update the PC software.</td>
</tr>
<tr>
<td><strong>PI Firmware Updater</strong></td>
<td>Windows</td>
<td>Program for user support when updating firmware of the E-727.</td>
<td>For users who want to update the firmware.</td>
</tr>
<tr>
<td><strong>USB driver</strong></td>
<td>Windows</td>
<td>Driver for the USB interface</td>
<td>For all users.</td>
</tr>
</tbody>
</table>
Installation

General Notes on Installation

- Install the E-727 near the power source so that the power plug can be quickly and easily disconnected from the mains.
- Only use cables and connections that meet local safety regulations.

Installing the PC Software

The communication between the E-727 and a PC is necessary to configure the E-727 and send motion commands using the commands of the GCS. Various PC software applications are available for this purpose.

Performing the Initial Installation

Accessories

- PC with a Windows operating system (Vista, 7, 8, 10) or Linux operating system
- Product CD (included in the scope of delivery)

Important information on the procedure for installation on Windows

- Before you start installing the PC software on a PC with a Windows operating system (p. 18), read the following information.

**INFORMATION**

When PIMikroMove is installed (default installation):

To provide the Device Parameter Configuration window, PIMikroMove requires the NI LabVIEW Run-Time Engine. The installation of PIMikroMove therefore includes the installation of the NI LabVIEW Run-Time Engine. A separate window opens for the installation of the NI LabVIEW Run-Time Engine in addition to the InstallShield Wizard window.

The InstallShield Wizard interrupts the installation of the PC software for the E-727 until the installation of the NI LabVIEW Run-Time Engine is started in the separate window.

- Note that the separate window can be covered by the InstallShield Wizard window on the screen. If necessary, display the separate window (e.g. by moving the InstallShield Wizard window).
- Follow the instructions for installing the NI LabVIEW Run-Time Engine that appear in the separate window (see figures below):
  - Note that the files needed for installation have to be unpacked first. This does not complete the installation though; you have to continue according to the instructions in the separate window.
Agree to unpacking with **OK**

**Start unpacking with Unzip**

Finish unpacking with **OK**

Start the installation of the NI LabVIEW Run-Time Engine with **Install NI LabWindows/CVI Run-Time Engine 2012 SP1**

- Note that the installation of the NI LabVIEW Run-Time Engine can take some time.
- If you accidentally close the separate window before the NI LabVIEW Run-Time Engine has been successfully installed: Go to the \SingleSetups directory on the product CD and start the installation by double-clicking the NI_LabWindows-CVI-RTE_2012_SP1_Setup.exe file.

**Installing the PC software on Windows**

1. Read "Important information on the procedure for installation on Windows" (p. 17).
2. Start the installation wizard by double-clicking the PI_E-727.CD_Setup.exe file in the installation directory (main directory of the CD).
The **InstallShield Wizard** window for the installation of programs and manuals for the E-727 opens.

3. Follow the instructions on the screen.

You can choose between default installation (**Complete**) and user-defined installation (**Custom**).

With default installation (recommended), all components are installed. These include among others:
- LabVIEW driver
  
  **Exception:** The Analog LabVIEW drivers component is provided for some PI controllers. This component is only available through user-defined installation.
- Dynamic program library for GCS
- PIMikroMove
- PC software for updating the firmware of the E-727
- For controllers that have a USB interface for communication with the PC: USB drivers

With user-defined installation, you have the option of excluding individual components from the installation.

### Installing the PC software on Linux

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

You can select individual components for installation.

### Installing Updates

PI is constantly improving the PC software.

- Always install the latest version of the PC software.

**Prerequisite**

- Active connection to the Internet.
- If your PC uses a Windows operating system:
  - If the PI Update Finder program is not on your product CD:
    You have downloaded the PI Update Finder from our Update Portal (http://www.update.pi-portal.ws).
  - You have the A000T0028 Technical Note for the PI Update Finder on hand. You can find the document either on the product CD or in the zip file that you have downloaded for the PI Update Finder.
If the PC to be updated is not directly connected to the Internet:
You have the A000T0032 Technical Note for the PI Update Finder at hand. You can find the document either on the product CD or in the zip file that you have downloaded for the PI Update Finder.

- If your PC uses a Linux operating system:
  - You have the user name and password for the E-727 at hand. Both of these can be found in the file “E-727 Releasenews_V_x_x_x.pdf” (x_x_x: version number of the CD) in the \Manuals folder on the product CD.

Updating the PC software on Windows

- Use the PI Update Finder:
  - When the PC to be updated is directly connected to the Internet: Follow the instructions in the A000T0028 Technical Note (TECHNICAL_NOTE PI_UPDATE_FINDER_xx.pdf).
  - When the PC to be updated is not directly connected to the Internet: Follow the instructions in the A000T0032 Technical Note.

Updating the PC software on Linux

INFORMATION
If software is missing or problems occur with downloading:

- Contact our customer service department (p. 102).

1. Open the website http://www.pi.ws.
2. Click Info.
3. Log in with the user name and the password from the "xxx_Releasenews.pdf" file on the product CD.
4. Click >> Product Downloads.
5. In the Product Files area, click the corresponding product category.
6. On the right-hand side of the page, select the corresponding sub-category.
7. Navigate to the product code on the page.
8. Copy the “CD Mirror” archive file to your PC.
9. Unpack the archive file to a separate installation directory.
10. In the directory with the unpacked files, go to the linux subdirectory.
11. Unpack the archive file in the linux directory by entering the command tar -xvpf <name of the archive file> on the console.
12. Read the accompanying information on the software update (readme file and/or "xxx_Releasenews.pdf" file) and decide whether the update makes sense for your application.
   - If no: Stop the update procedure.
   - If yes: Perform the following steps.
13. Log onto the PC as a superuser (root rights).
Ensuring Ventilation

High temperatures can overheat the E-727.

- Set up the E-727 with a distance of at least 10 cm to the top and rear sides and at least 15 cm to the sides. If this is not possible, make sure that the area is cooled sufficiently.
- Set up the E-727 so that the ventilation holes in the housing are freely accessible.
- Ensure sufficient ventilation at the place of installation.
- Keep the ambient temperature to a non-critical level (<40° C).

See also “Overtemp Protection of the E-727” (p. 44).

Connecting the E-727 to the Protective Earth Conductor

INFORMATION

- Observe the applicable standards for mounting the protective earth conductor.

Prerequisite

- You have read and understood the Safety precautions (p. 6).
- The E-727 is switched off via the Power switch.

Tools and accessories

- Suitable protective earth conductor:
  - Cable cross-section ≥0.75 mm²
  - Contact resistance < 0.1 ohm at 25 A at all connection points relevant for mounting the protective earth conductor
- Fastening material for the protective earth conductor (M4 screw, two safety washers, two flat washers), sits on the protective earth connection (M4 hole) upon delivery of the E-727 (p. 8).

Connecting the E-727 to the protective earth conductor

1. If necessary, fasten a suitable cable lug to the protective earth conductor.
2. Remove the M4 screw with the washers from the E-727.
3. Put the washers and the cable lug of the protective earth conductor on the M4 screw in the following order:
   - Safety washer
   - Flat washer
   - Cable lug
   - Flat washer
   - Safety washer
4. Insert the M4 screw into the protective earth connection hole of the E-727.
5. Tighten the M4 screw with at least three rotations and a torque of 1.2 Nm to 1.5 Nm.
Quick Start

The aim of quick start is to start initial test motions of a stage that is connected to a E-727 in the PIMikroMove PC software.

The quick start should comprise the following steps in the given order:
- Starting the system in PIMikroMove (p. 23): Installation, power-on, communication between E-727 and PC in PIMikroMove, configuration of PIMikroMove
- Creating backup file for controller parameters (p. 25)
- Executing test motions in open-loop operation (p. 26): First test of the function
- Measuring the resonant frequency and—if necessary—adjusting the notch filter(s) in open-loop operation (p. 27): System optimization
- Checking and—if necessary—optimizing the servo-control parameters (p. 33): System optimization
- Saving optimized settings in the Single-Axis window (p. 36)

General Notes on Start-Up

CAUTION

Risk of electric shock if the protective earth conductor is not connected!
If a protective earth conductor is not or not properly connected, dangerous touch voltages can occur on the E-727 in the case of malfunction or failure of the system. If touch voltages exist, touching the E-727 can result in serious injury or death from electric shock.
- Connect the E-727 to a protective earth conductor before start-up (p. 21).
- Do not remove the protective earth conductor during operation.
- If the protective earth conductor has to be removed temporarily (e.g. in the case of modifications), reconnect the E-727 to the protective earth conductor before starting it up again.

NOTICE

Damage to the stage and the load from oscillations
Unsuitable settings of the notch filter and the servo-control parameters of the E-727 can cause the stage to oscillate. Oscillations can damage the stage and/or the load affixed to it.
- If the stage is oscillating (unusual operating noise), immediately switch off the servo mode or disconnect the E-727 from the power source.
- Only switch on the servo mode after you have modified the settings of the notch filter and the servo-control parameters of the E-727; see „Adjusting the Notch Filter(s) in Open-Loop Operation“ (p. 27) and "Checking and Optimizing the Servo-Control Parameters" (p. 33).
**INFORMATION**
The E-727 and the stage(s) are supplied as a pre-configured system.
- If a connection assignment is given on the labels of the E-727 and/or stage(s), observe this assignment when connecting the stage(s).

**INFORMATION**
When you connect a stage when the controller is powered on, the ID-chip of the stage is not read by the controller. To read the ID-chip data, the controller must be power-cycled or rebooted using the RBT command or the corresponding host software functions.

**INFORMATION**
Pin 2 of the Digital I/O socket can be configured as Reset input, see p. 98 for details. The Reset input works as follows:
- If nothing is connected to pin 2, the signal level is HIGH (internal pull-up with 10 kohm resistor) which means normal operation of the E-727.
- If the signal level on the Reset input becomes LOW, the E-727 is rebooted.
- Make sure that the configuration of pin 2 of the Digital I/O socket and the connected signal comply with your application.

**INFORMATION**
The write access for the parameters of the E-727 is defined by command levels. After the controller is switched on or rebooted, the active command level is always level 0. For particular parameters, write access is only allowed on command level 1. On command levels > 1, write access is only available to PI service personnel.
- If necessary, send the CCL 1 advanced command or enter the password advanced to change to command level 1.
- Contact the customer service department if there seem to be problems with parameters of command level 2 or higher (p. 102).

---

**Starting the System in PIMikroMove**

Proceed as follows to start the E-727 with the stage(s) in PIMikroMove:

1. Install the following on the PC:
   - The PC software and the USB drivers from the product CD
   - Updates for PC software
   Details see "Installing the PC Software" (p. 17).
2. Make sure that the Power switch of the E-727 is in the OFF position (0).
3. Install the E-727:
   − Observe the general information on installation (p. 17).
   − Ensure the ventilation (p. 21).
   − Connect the E-727 to the protective earth conductor (p. 21).

4. Connect the following to the E-727:
   − The included wide-range-input power supply to the 24 VDC connection via the included adapter.
   − The stage(s) to the socket for the piezo stages.
   − The PC via the RS-232 interface (RS-232 panel plug) or via the USB interface (USB type B) or via the Ethernet interface (RJ45).

5. Switch on the E-727:
   a) Connect the power cord of the wide-range-input power supply to the power socket.
   b) Put the Power switch of the E-727 in the ON position (|).

   During the power-on/reboot sequence, the LEDs of the E-727 behave as follows:
   − Phase 1: All LEDs are lit for up to 4 seconds.
   − Phase 2: If the E-727 is configured to obtain the IP address for TCP/IP communication from a DHCP server (default setting), the E-727 is searching for the DHCP server for a maximum duration of 15 seconds. During this period, the OFL and Servo LEDs are alternately lit, and all other LEDs are off. (If no DHCP server is found, the static IP address set with parameter ID 0x11000600 is used.)
   − Phase 3: E-727 initializes parameters. The Servo LEDs are alternately lit, and all other LEDs are off.

   The power-on/reboot sequence is finished when the Power LED is continuously lit.

6. Start PIMikroMove on the PC.

7. Establish communication between the E-727 and the PC in PIMikroMove via RS-232 or USB or TCP/IP. Further details see "Communication" (p. 37).

8. In the Start up axes step in PIMikroMove, execute the AutoZero procedure for all linear axes of the stage(s) (details see "AutoZero Procedure" (p. 41)).

   Note that starting AutoZero for rotation axes will fail and cause the error code 74 („No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)“).

9. In the Start up controller window, click Close.

   The main window of PIMikroMove opens.

10. Optionally: Configure the PIMikroMove main window.

    It is recommended to see the tab cards for axes, input signal channels and output signal channels (see figure below). You can arrange them by dragging them with the left mouse button pressed so that they become docked e.g. to the bottom border of the window.

    On the Axes tab card, amongst others you can start axis motion. The channel tab cards show the current values of the input signal channels (in the figure below: sensors) and output signal channels (in the figure below: output voltages for piezo actuators).
Note: The input and output signal channels of the E-727 are allocated to the logical axes via matrices (input matrix: parameters 0x07000500 to 0x07000506; output matrix: parameters 0x09000000 to 0x09000003). Depending on the connected stage type, an axis may be driven by more than one piezo actuator and measured by more than one sensor.

Creating Backup Files for Controller Parameters

**INFORMATION**

The properties of the E-727 and the connected stage(s) are stored in the E-727 as parameter values.

- Create a backup copy on the PC before changing the parameter values of the E-727. You can then restore the original settings at any time.
- Create an additional backup copy with a new file name each time after you optimize the parameter values.

To save the parameter values and to load them back to the E-727, use the **Device Parameter Configuration** window provided by PIMikroMove.

Proceed as follows to create a parameter file:

1. In the main window of PIMikroMove, open the **Device Parameter Configuration** window via the **E-727... > Parameter Configuration ...** menu item.
In the figure below, the **Device Parameter Configuration** window shows the Sensor Mechanics 1 parameter group.

2. Save the parameter values from the **Edit Mask** column of the **Device Parameter Configuration** window in a parameter file (file extension .pam) on your PC. Use one of the following options:

   - **File > Save Edit Values** or **File > Save Edit Values As** menu item
   - (Save) or (Save As) button in the icon bar

**Executing Test Motions in Open-Loop Operation**

The first moves should be made in open-loop operation. With the factory default settings of the E-727, open-loop commanding means to give open-loop values which correspond approximately to axis positions.

- In the main window of PI-MikroMove, make some test moves with the individual axes using the controls on the **Axes** tab card. During the test moves, observe the position display for the axes (in the **Current Value / Position** fields) and the current output voltage(s) for the piezo actuator(s) in the stage(s) (in the **Output Value** fields of the **Output Channels** tab card).

Proced as follows for each linear axis (for rotation axes, the given open-loop values correspond approximately to positions in µrad, and the step size can be set to a larger value, for example 100 µrad):

a) Make sure that the **Servo** box is unchecked.

b) Command an open-loop value of 0 (µm) by entering 0 in the **Open-Loop Target Value** field of the axis and pressing Enter on your keyboard.

c) Enter the value 10 (µm) in the **Step size** field of the axis and press Enter.

d) Use the > button next to the **Open-Loop Target Value** field to increment the commanded value by the value given in the **Step size** field (10). Increment the open-loop value this way step by step up to the upper travel range limit of the axis.

e) Use the < button next to the **Open-Loop Target Value** field to decrement the commanded value by the value given in the **Step size** field (10). Decrement the open-loop value this way step by step up back to zero.
The values for position and output voltage should follow the commanded open-loop 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-727, the output voltage range is -30 to +130 V; the output voltage(s) corresponding to a given open-loop value depend(s) on the connected stage(s).

In the example shown in the figure above, the open-loop value for axis 1 was increased to 30 by clicking the > button three times (step size value is 10). The current position approximately corresponds to the commanded open-loop value (29.167 µm). Since axis 1 of the stage is driven by two piezo actuators in the example, the output voltage of the corresponding output signal channels 1 and 2 has changed to appropriate values.

Adjusting the Notch Filter(s) in Open-Loop Operation

The corrections by a notch filter only take place in closed-loop operation by default, but can also be enabled for open-loop operation. The appropriate frequency component is reduced in the control value to compensate for undesired resonances in the mechanical system. Adjusting the notch filter frequency can be useful, particularly in the case of very high loads.

**INFORMATION**

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

- Set the notch filter(s) **before** you optimize the servo-control parameters (p. 33).
INFORMATION

- Before you measure the resonant frequencies as described below, make sure that the notch filters are not enabled in open-loop operation. To do this, check the value of the Enable Notch In Open Loop parameter, ID 0x08000500, for all axes (0 = disable notch filter in open-loop operation; 1 = enable notch filter in open-loop operation). You can do this in the Servo parameter groups of the Device Parameter Configuration window in PIMikroMove.

To measure the resonant frequency and adjust the notch filter(s), a frequency response (axis response to an impulse) is recorded in open-loop operation.

Proceed as follows for each axis:

1. Make sure the stage is mounted in exactly the same way as in the application. The load on the stage is especially important.
2. Open and configure the data recorder in PIMikroMove:
   - In the main window of PIMikroMove, open the Data Recorder window via the E-727… > Show data recorder … menu item.

Configure the data recorder:

a) Open the Configure Data Recorder window by clicking the Configure … button in the bottom right corner of the Data Recorder window.

b) In the Configure Data Recorder window, select the data to be recorded. To measure the resonant frequency, the current position of the axis must be recorded (axis 1 in the figure below).
c) In the **Data Recorder** window, set the number of data points to be read, make sure that the correct axis is selected and that the axis is in open-loop operation (servo mode off), and enter a suitable value for the amplitude of the impulse. Settings in the figure below: 4096 data points, axis 1, **Servo** box unchecked, amplitude of 10 µm.

3. Open and configure the **Single Axis** window in PIMikroMove:
   - In the main window of PIMikroMove, open the **Single Axis** window via the context menu of the axis, see figure below.

   ![Single Axis Window](image)

   Configure the **Single Axis** window:
   a) Expand the view of the **Single Axis** window by clicking on the > button at the right edge of the window.
   b) Display the parameters of the E-727 that are relevant: Select the **Tuning Digital Piezo Controllers** entry from the field in the top right corner of the **Single Axis** window, see figure below.
4. Perform the frequency response measurement in the Data Recorder window and identify the resonant frequency:
   
a) Start the impulse in the positive direction by clicking the corresponding button, see figure below.

   ![Data Recorder window showing positive impulse](image)

   b) Show the button for the FFT calculation using the Show / Hide data toolbar button in the icon bar, see figure below.

   ![Data Recorder window showing FFT calculation](image)

   c) Display the recorded position values as FFT calculation values using the corresponding button in the icon bar, see figure below.

   ![Data Recorder window showing FFT calculation values](image)
d) Show cursors and enable cursor motion using the buttons shown in the figures below.

![FFT display with cursors](image1)

![Cursor settings](image2)

e) Identify the resonance peak(s) in the FFT display. To do so, place a cursor on the peak and read out the cursor value which is displayed on the right hand side of the graph. If there is more than one resonance peak, peak 1 is always the one with the lowest frequency.

In the figure below, the cursor 1 is at the first resonance peak (372.31445 Hz), and cursor 2 is at the second (next higher) resonance peak (622.55859 Hz).
5. If necessary, adjust the notch filter settings in the *Single Axis* window to the measured resonant frequencies (adjustment is necessary if the values significantly differ). Note that 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).

   a) In the *Command Level Too Low* dialog that will open when you try to change a parameter value for the first time, enter the password *advanced* (see figure below).

   ![Command Level Too Low dialog](image)

   ![Password dialog](image)

   b) In the *Single Axis* window, enter the frequency value of the first resonance peak in the *Notch Frequency 1 (0x8000100)* field. If you have measured a second resonance peak, enter the frequency value of the second resonance peak in the *Notch Frequency 2 (0x8000101)* field. If the second notch filter is deactivated (rejection value = 1), change the rejection value to 0.05 in the *Notch Rejection 2 (0x8000201)* field.
Checking and Optimizing the Servo-Control Parameters

Adjusting the servo-control parameters optimizes the dynamic properties of the system (overshoot and settling time). The optimum settings depend on your application and your requirements.

**INFORMATION**

The servo-control parameters which can be set for an axis include:

- **Servo-Loop P-Term** (ID 0x07000300)
- **Servo-Loop I-Term** (ID 0x07000301)
- **Servo-Loop Slew Rate** (ID 0x07000200) (maximum velocity)

**INFORMATION**

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

- Set the notch filter(s) (p. 27) before you optimize the servo-control parameters.

Note that the value of the **Servo-Loop I-Term** parameter can be calculated from the frequency value of the first notch filter with the following formula:

\[
I_{term} = \frac{1}{4 \cdot \pi \cdot \text{Notch Frequency}}
\]

The optimization of the servo-control parameters is typically done empirically: The response of the axes to a step ("step response") is analyzed under various values in closed-loop operation.

Proceed as follows for each axis:

1. Make sure the stage is mounted in exactly the same way as in the application. The load on the stage is especially important.
2. Make sure that **Data Recorder** window and **Single Axis** window of PIMikroMove are still open (see “Measuring the Resonant Frequency in Open-Loop Operation”, p. 27).
3. Switch the servo mode on (closed-loop operation) for the axis. To do this, check the **Servo** box for the axis in one of the following windows: main window of PIMikroMove, **Data Recorder** window, **Single Axis** window.
4. Configure the data recorder:
   a) Open the **Configure Data Recorder** window by clicking the **Configure …** button in the bottom right corner of the **Data Recorder** window.
   b) In the **Configure Data Recorder** window, select the data to be recorded. To analyze the step response, the current position and the target position of the axis must be recorded (axis 1 in the figure below).
c) In the **Data Recorder** window, set the number of data points to be read, make sure that the correct axis is selected and that the axis is in closed-loop operation (servo mode on), enter a suitable value for the amplitude of the step (about 15% of the axis travel range), and check the **Move back after step** box. Settings in the figure below:
4096 data points, axis 1, Servo box checked, amplitude of 15 µm, axis moves back to the start point after the step.

5. Perform and analyze the step response measurement:
   a) In the **Data Recorder** window, start the step in the positive direction by clicking the corresponding button, see cursor position in the figure below.

b) Check the result and compare it with the examples shown in the figures below.
   The rise rate of the step response is very low in the figure below. This means that the P term is too low and has to be increased.
The figure below shows a step response with a small overshoot which means that the P term is too high and has to be decreased.
The result of the step response is satisfactory when there is minimum overshoot, and the settling time is not too long, as in the figure below. No changes are required for the servo-control parameters.

![Step Response Graph]

**c)** If necessary, enter new values for the servo-control parameters in the Single Axis window and repeat the step response measurement (steps a) and b)).

### Saving Optimized Settings in the Single-Axis Window

- If you are satisfied with the results of the frequency response measurement (p. 27) and step response measurement (p. 33), and want to keep the new settings for notch filters and servo-control parameters, save the new settings to the nonvolatile memory of the E-727:
  - In the Single-Axis window, click the left button below the parameter list (see cursor position in the figure below). Only the values from the parameter list will be saved.
Operation

Communication

The E-727 can be controlled with ASCII commands (PI General Command Set) from a PC via the following communication interfaces:

- TCP/IP (p. 37)
- Serial RS-232 connection (p. 39)
- USB connection (p. 40)

**INFORMATION**

The following commands are available for the interface parameters of the E-727:

- Values in the nonvolatile memory:
  - Get with IFS?
  - Set with IFS
- Values in the volatile memory:
  - Get with IFC?
  - Set with IFC (baud rate for RS-232 only)

For querying and setting the interface parameters, it is recommended to use the *Configure Interface* window in PIMikroMove. There you can comfortably read, modify and save the values of the interface parameters. For details, see the PIMikroMove manual.

Communication via the TCP/IP Interface

The TCP/IP settings of E-727 are preset as follows:

- IP address (parameter ID 0x11000600): 192.168.168.10:50000
- IP mask (parameter ID 0x11000700): 255.255.255.0
- IP start (parameter ID 0x11000800): 1 (IP address is obtained from DHCP server; if no DHCP server is present, the IP address defined with parameter 0x11000600 is used as static address)

**Adaptation of the interface parameters**

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

- Network with DHCP server: No adjustment of the factory settings of the interface parameters of the E-727 is necessary
- Network without DHCP server or direct connection (E-727 directly connected to the Ethernet connection socket of the PC):
  - The start-up behavior of the E-727 for configuring the IP address must be changed so that the E-727 uses a static address (set IP start to 0).
  - The IP addresses and subnet masks of the E-727 and PC as well as all other network participants must be adapted to each other (set IP address and IP mask of E-727 to suitable values).
After switching on or rebooting the E-727

The starting procedure of the E-727 must be finished before the communication between the E-727 and PC can be established. The starting procedure takes about 20 seconds and is finished when the Power LED is continuously lit.

Connection of the network cable when the controller is switched on

Establishing communication via TCP/IP can fail if the network cable was connected to the RJ45 Ethernet socket on the front panel of the E-727 while the E-727 was switched on.

- If the establishment of communication fails, switch the E-727 off and back on again while the network cable is plugged in.

Port setting

For communication via TCP/IP, the E-727 only has a single unchangeable port (50000) available, which cannot be used by more than one connection at a time.

Establishing communication via TCP/IP

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

1. Start PIMikroMove.

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

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

2. Select E-727 in the field for controller selection.

3. Select the TCP/IP tab on the right side of the window.

4. Click the Search for controllers button.

   All controllers in the same network are shown in the field underneath the button.
5. Click the **PI E-727 ... SN ...** entry in the list of found controllers (SN stands for serial number).
   - If several **PI E-727... SN ...** entries are shown, identify your E-727 on the basis of its nine-digit serial number.
   - If the E-727 is not displayed in the list of the controllers found, check the network settings. Consult your network administrator if necessary.

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

6. Check the IP address in the **Hostname / TCP/IP Address** field and the port number in the **Port** field.

7. Click the **Connect** button to establish communication.

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

### Communication via the RS-232 Interface

The baud rate of E-727 for RS-232 communication is preset as follows:

- Uart baud rate (parameter ID 0x11000400): 115200

Further possible values are 9600, 19200, 38400, 57600.

To successfully establish communication, the baud rates of the E-727 and PC must match.

### Establishing communication via RS-232

The procedure for PIMikroMove is described in the following. The procedure for other PC software programs (PITerminal, LabVIEW drivers) is similar.
1. Start PIMikroMove.
   The Start up controller window opens with the Connect controller step.
   - If the Start up controller window does not automatically open, select the Connections > New... menu item in the main window.

2. Select E-727 in the field for controller selection.
3. Select the RS-232 tab on the right side of the window.
4. In the COM Port field, select the COM port of the PC to which you have connected the E-727.
5. Set a suitable value in the Baudrate field to adapt the baud rate of the PC to the baud rate of the E-727.
6. Click Connect to establish communication.
   If communication has been successfully established, the Start up controller window switches to the Start up axes step.

Communication via the USB Interface

Establishing communication via USB

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

1. Start PIMikroMove.
   The Start up controller window opens with the Connect controller step.
   - If the Start up controller window does not automatically open, select the Connections > New... menu item in the main window.
2. Select **E-727** in the field for controller selection.
3. Select the **PI USB** tab on the right side of the window.
4. On the **PI USB** tab, select the connected E-727.
5. Click **Connect** to establish communication.

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

### AutoZero Procedure

**INFORMATION**

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

---

**INFORMATION**

AutoZero is to be performed with linear axes only. Starting AutoZero for rotation axes will fail and cause the error code 74 ("No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)").

### Objective and Prerequisites of AutoZero

The AutoZero procedure performs automatic zero point adjustment of the sensors. With the E-727.3SD and .3SDA models for SGS sensors, the AutoZero procedure in addition performs sensor scaling.
Objective of AutoZero:
- Make the entire travel range available:
  Changes in temperature or changes in the mechanical load can cause small deviations of the
  sensor zero point. When the sensor zero-point is set correctly, the complete output voltage
  range of the amplifier can be used in closed-loop operation.
- Prevent the piezo actuators from damage:
  In open-loop operation, the stage displacement with 0 V piezo voltage should already be
  about 10% of the travel range. Then the average applied voltage is reduced which lengthens
  the lifetime of the piezo actuator in the stage without reducing the nominal travel range.
- Ensure the replaceability of stage(s) and controller:
  For optimum sensor scaling, the sensor gain setting of the controller has to be adapted to
  the connected stage.

Prerequisites for AutoZero:
- LowVoltage < HighVoltage
  (LowVoltage is given by the value of the AutoZero Low Voltage parameter (ID 0x07000a00);
  HighVoltage is given by the value of the AutoZero High Voltage parameter (ID 0x07000a01))
- The value of the AutoZero High Voltage parameter (ID 0x07000a01) should be identical with
  the piezo voltage that is required for maximum displacement of the axis.
- If sensor scaling is to be included in the AutoZero procedure: The Sensor Autoscaling Enable
  parameter (ID 0x03003700) must be set to the value 1 (default value is 0).

Settings Changed by AutoZero
The AutoZero procedure changes the values of the parameters Sensor Mech. Correction 1 (ID
0x02000200). With the E-727.3SD and .3SDA models for SGS sensors, the AutoZero procedure also
changes Sensor Offset factor (ID 0x02000102) and—if the Sensor Autoscaling Enable parameter
(ID 0x03003700) has the value 1—Sensor Autoscaling Gain (ID 0x03003701) for all sensor
channels.

Starting AutoZero via Command Entry
Via command entry, you can start the AutoZero procedure of the E-727 as follows:
- Use the ATZ command to perform the AutoZero procedure once (see E-725 user manual
  (PZ197E) for ATZ details). Afterwards save the values of the parameters Sensor Mech.
  Correction 1 (ID 0x02000200), Sensor Offset factor (ID 0x02000102) and Sensor Autoscaling
  Gain (ID 0x03003701) to nonvolatile memory.
  or
- Send the ATZ command after every start or reboot of the E-727.
  or
- Set the value of the Power Up AutoZero Enable parameter (ID 0x07000802) to 1 for all axes
  so that the AutoZero procedure is performed automatically with every start or reboot of the
  E-727.
Starting AutoZero in PIMikroMove

- In the **Start up axes** step of the **Start up controller** window, execute the AutoZero procedure.
  - To re-open the **Start up controller** window with the **Start up axes** step, select the **E-727 > Start up axes...** menu item in the main window.

Proceed as follows for the **linear** axes that are connected:

a) Mark the linear axes in the list.

b) Click **Auto Zero**. The **Auto Zero** dialog opens.

c) In the **Auto Zero** dialog, start the AutoZero procedure by clicking **Start**.

d) After a successful AutoZero procedure, click **OK**.
Overtemp Protection of the E-727

**INFORMATION**

E-727 is equipped with a fan that is automatically switched on when necessary.

A sensor detects the internal temperature of the E-727. Based on the values measured by the sensor, the E-727 supports a “switch-off” temperature threshold for overtemp protection of the amplifier. The value of this threshold is 72 °C.

If the sensor value exceeds the threshold, the amplifier output is switched off automatically.

Use the DIA? command (p. 65) to query if the threshold is exceeded and if the amplifier output is active or not.

Possible measures to avoid overheating:
- Observe the installation instructions (p. 21).
- Keep the ambient temperature below 40 °C.
- Reduce the output power of the E-727 by reducing the frequency and/or the amplitude in dynamic operation.
- If the sensor value exceeds the threshold: Stop the wave generator. Let the system cool down. Ensure sufficient ventilation.

Data Recording

“Data Recording“ in the E-725 user manual (PZ197E) is also valid for E-727. In addition, the information in this section applies to E-727.

**INFORMATION**

With E-727, the settings for record options and data sources (see DRC command) can be saved via parameters in nonvolatile memory:
- **DRC Data Source** (ID 0x16000700)
- **DRC Record Option** (ID 0x16000701)
- For parameter handling see “Controller Parameters” in the E-725 user manual.

Trigger options for data recording can be set with the DRT command (p. 67) and queried with the DRT? command (p. 69).

The number of points comprised by the last recording can be read with the DRL? command(p. 67).
External Triggering/Signaling

„External Triggering/Signaling“ in the E-725 user manual (PZ197E) is also valid for E-727. In addition, the information in this section applies to E-727.

**INFORMATION**

With E-727, the trigger output configuration (see CTO command in the E-725 user manual) can be saved via parameters in nonvolatile memory:

- CTO Trigger Step (ID 0x18000201)
- CTO Axis (ID 0x18000202)
- CTO Trigger Mode (ID 0x18000203)
- CTO Min.Threshold (ID 0x18000205)
- CTO Max.Threshold (ID 0x18000206)
- CTO Polarity (ID 0x18000207)
- CTO Start Threshold (ID 0x18000208)
- CTO Stop Threshold (ID 0x18000209)

➢ For parameter handling see “Controller Parameters” in the E-725 user manual.

In addition to the trigger modes described in the E-725 user manual, E-727 supports mode 9 = Generator Pulse Trigger. With this mode, the trigger output is synchronized with the wave generator output, and the trigger line action must be defined with the TWS command. Difference between mode 9 and mode 4 (Generator Level Trigger):

- Mode 4: The length of a single trigger pulse is the same as the duration of one servo cycle. If the signal level is set to HIGH with TWS for consecutive points of a wave table, the signal level therefore does not change back to LOW between the points.
- Mode 9: A single trigger pulse is shorter than the servo cycle duration. If the signal level is set to HIGH with TWS for consecutive points of a wave table, the signal level therefore changes back to LOW after each point. This way, the trigger output can be used to count the waveform points that are output by the wave generator.

Using the Analog Input

With models E-727.3CDA and E-727.3SDA, four analog inputs are available on the Analog I/O socket (p. 99). These analog inputs each can be used for an external sensor or as a control source, see „Using the Analog Input“ in the E-725 user manual (PZ197E) for details. In addition, see the description of the input signal channels in „Axes, Channels, Functional Elements“ (p. 11) and the information in „E-727.3CDA, E-727.3SDA: Analog I/O“ (p. 99).

When the analog input is used as a control source: In addition to the behaviour and settings described in the E-725 user manual, E-727 supports the **Discon. Target Man. In With Stop** parameter (ID 0x0E001E00). With this parameter, you can select if the analog input channel is to be disconnected from the axis (value 1) or not (value 0) when the axis motion is stopped with STP or #24.
Using the Analog Output

With models E-727.3CDA and E-727.3SDA, one analog output is available on pin 8 of the Analog I/O socket (p. 99). This analog output is accessible in the firmware of the E-727 as output signal channel 4. Note that the analog output shares output signal channel 4 with the Piezo Ch 4 lines on the socket for piezo stages (p. 96 or p. 97).

Output signal channel 4 can therefore be configured as follows via the value of the Select Output Type parameter (ID 0x0A000003):

- 1: Output voltage for a piezo actuator in the stage, output as Piezo Ch 4 on the socket for piezo stages. The internal control voltage for channel 4 is scaled to be in the range of -30 to 130 V.

- 2: Position monitor of an axis, output on pin 8 of the Analog I/O socket. The value of the Select Output Index parameter (ID 0x0A000004) determines the axis whose position is to be output. Note that the output has to be scaled, i.e. the axis position values have to be associated with suitable output levels (= scaled position values). To do this, set the Position Report Scaling parameter, ID 0x07001005, and the Position Report Offset parameter, ID 0x07001006 to suitable values for the appropriate axis.

- 5: Control signal for an external motor driver, output on pin 8 of the Analog I/O socket. The value of the Select Output Index parameter (ID 0x0A000004) determines the output signal channel whose control value is to be output. The internal control voltage for channel 4 is scaled to be in the range of -10 to 10 V.

**INFORMATION**

Irrespective of the selection made with the Select Output Type parameter, signals are always present on both the Piezo Ch 4 lines of the socket for the piezo stages and on pin 8 of the Analog I/O socket. But due to the different scaling settings of the individual configuration options, the correct selection is important for the performance of the stage(s): If a total of four piezo actuators are present in the stage(s), output signal channel 4 must be configured for use as output voltage (Select Output Type must have the value 1).

Note that PI will supply E-727 and the piezo stage(s) as a system with appropriate settings.

- If you are not sure whether your system can be configured for output of position monitor or control signal, contact our customer service department (p. 102).

For further details, see the description of the output signal channels in „Axes, Channels, Functional Elements“ (p. 11) and the information in „E-727.3CDA, E-727.3SDA: Analog I/O“ (p. 99).
Wave Generator

„Wave Generator“ in the E-725 user manual (PZ197E) is also valid for E-727.

In addition to the functionality described in the E-725 user manual, the E-727 provides the Wave Multi Start By Trigger parameter (ID 0x13000202). When the wave generator is to be started by an external trigger signal (WGO bit 1 is set), the value of this parameter determines if the trigger is enabled for only one generator start or for multiple starts:

- 0 = Trigger is enabled for only one generator start. Trigger becomes disabled after the generator has been started. To enable the trigger again, WGO must be sent again with start mode bit 1 set. Default setting.
- 1 = As long as WGO bit 1 is set, the trigger stays enabled for an unlimited number of generator starts. To disable the trigger, the wave generator output must be stopped with WGO, STP or #24.

For the number of wave generators and wave tables provided by the E-727, see „Axes, Channels, Functional Elements“ (p. 11).

Dynamic Digital Linearization (DDL)

„Dynamic Digital Linearization“ in the E-725 user manual (PZ197E) is also valid for E-727.

For the number of DDL tables provided by the E-727, see „Axes, Channels, Functional Elements“ (p. 11).

Controller Macros

The E-727 can save and process command sequences as macros.

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

- Several macros can be saved at the same time.
- Any macro can be defined as the start-up macro. The start-up macro is executed each time that the E-727 is switched on or rebooted.
- Processing a macro and stopping macro execution can be linked to conditions. In this way, loops can be realized as well.
- Macros can call up themselves or other macros in several nesting levels.
- Variables (p. 56) can be set for the macro and in the macro itself and used in different operations.
- Input signals can be evaluated for conditions and variables.
## Commands for Macros

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

<table>
<thead>
<tr>
<th>Command (p.</th>
<th>Syntax</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD (p. 63)</td>
<td>ADD &lt;Variable&gt; &lt;FLOAT1&gt; &lt;FLOAT2&gt;</td>
<td>Adds two values and saves the result to a variable (p. 56). Can only be used for local variables in macros.</td>
</tr>
<tr>
<td>CPY (p.65)</td>
<td>CPY &lt;Variable&gt; &lt;CMD?&gt;</td>
<td>Copies a command response to a variable (p. 56). Can only be used for local variables in macros.</td>
</tr>
<tr>
<td>DEL (p. 65)</td>
<td>DEL &lt;uint&gt;</td>
<td>Can only be used in macros. Delays &lt;uint&gt; milliseconds.</td>
</tr>
<tr>
<td>JRC (p. 74)</td>
<td>JRC &lt;Jump&gt; &lt;CMD?&gt; &lt;OP&gt; &lt;Value&gt;</td>
<td>Can only be used in macros. Triggers a relative jump of the macro execution pointer depending on a condition.</td>
</tr>
<tr>
<td>MAC (p. 76)</td>
<td>MAC BEG &lt;macroname&gt;</td>
<td>Starts the recording of a macro with the name <code>macroname</code> on the controller. <code>macroname</code> can consist of up to 15 characters.</td>
</tr>
<tr>
<td></td>
<td>MAC DEF &lt;macroname&gt;</td>
<td>Defines the given macro as the start-up macro.</td>
</tr>
<tr>
<td></td>
<td>MAC DEF?</td>
<td>Gets the start-up macro.</td>
</tr>
<tr>
<td></td>
<td>MAC DEL &lt;macroname&gt;</td>
<td>Deletes the given macro.</td>
</tr>
<tr>
<td></td>
<td>MAC END</td>
<td>Ends the macro recording.</td>
</tr>
<tr>
<td></td>
<td>MAC ERR?</td>
<td>Reports the last error which occurred during macro execution.</td>
</tr>
<tr>
<td></td>
<td>MAC FREE?</td>
<td>Gets the free memory space for macro recording (unit: number of characters).</td>
</tr>
<tr>
<td></td>
<td>MAC NSTART &lt;macroname&gt; &lt;uint&gt; [&lt;String1&gt; [&lt;String2&gt;]]</td>
<td>Starts the given macro n times in succession (n = number of executions). The values of local variables can be set for the macro with &lt;String1&gt; and &lt;String2&gt;.</td>
</tr>
<tr>
<td></td>
<td>MAC START &lt;macroname&gt; [&lt;String1&gt; [&lt;String2&gt;]]</td>
<td>Starts one execution of the specified macro. The values of local variables can be set for the macro with &lt;String1&gt; and &lt;String2&gt;.</td>
</tr>
<tr>
<td>MAC? (p. 78)</td>
<td>MAC? [macroname]</td>
<td>Lists all macros or the content of a given macro.</td>
</tr>
<tr>
<td>MEX (p. 79)</td>
<td>MEX &lt;CMD?&gt; &lt;OP&gt; &lt;Value&gt;</td>
<td>Can only be used in macros. Stops the macro execution depending on a condition.</td>
</tr>
<tr>
<td>RMC? (p. 81)</td>
<td>RMC?</td>
<td>Lists macros which are currently running.</td>
</tr>
<tr>
<td>VAR (p. 82)</td>
<td>VAR &lt;Variable&gt; &lt;String&gt;</td>
<td>Sets a variable (p. 56) to a certain value or deletes it. Can only be used for local variables in macros.</td>
</tr>
<tr>
<td>VAR? (p. 84)</td>
<td>VAR? {[[&lt;Variable&gt;]]}</td>
<td>Gets variable values.</td>
</tr>
</tbody>
</table>
# Working with Macros

Work with macros comprises the following:

- Recording of macros (p. 49)
- Starting macro execution (p. 51)
- Stopping macro execution (p. 54)
- Setting up a start-up macro (p. 54)
- Deleting of macros (p. 55)

## INFORMATION

For working with controller macros, it is recommended to use the **Controller macros** tab in PIMikroMove. There you can conveniently record, start and manage controller macros. Details are found in the PIMikroMove manual.

## Recording a macro

### INFORMATION

A maximum of 5 nesting levels are possible in macros.

### INFORMATION

Basically all GCS commands (p. 58) can be included in a macro. Exceptions:

- **RBT** for rebooting the E-727
- **MAC BEG** and **MAC END** for macro recording
- **MAC DEL** for deleting a macro

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

To make the use of macros more flexible, you can use local and global variables in macros. See “Variables” (p. 56) for more information.

INFORMATION

The number of write cycles in the nonvolatile memory is restricted by the limited lifetime of the memory chip.

- Only record macros when it is necessary.
- Use variables (p. 56) in macros to make macros more flexible, and give the corresponding variable values when starting macro execution.
- Contact our customer service department (p. 102) if the E-727 shows unexpected behavior.

INFORMATION

A macro must be deleted before a macro with the same name can be re-recorded.

1. Start the macro recording.
   - If you are working with PITerminal or in the Command entry window of PIMikroMove:
     Send the MAC BEG macroname command, where macroname indicates the name of the macro.
   - If you are working in PIMikroMove on the Controller macros tab: Click the Create new empty macro icon to create a tab for entering a new macro. Do not enter the MAC BEG macroname command.

2. Enter the commands to be included in the macroname macro line by line, using the normal command syntax.

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

3. End the macro recording.
   - If you are working with PITerminal or in the Command entry window of PIMikroMove:
     Send the MAC END command.
   - If you are working in PIMikroMove on the Controller macros tab: Do not enter the MAC END command. Click the Send macro to controller icon and enter the name of the macro in a separate dialog window.

The macro has been stored in the nonvolatile memory of the E-727.

4. If you want to check whether the macro has been correctly recorded:
   - If you are working with PITerminal or in the Command entry window of PIMikroMove:
     Get which macros are saved in the E-727 by sending the MAC? command.
     Get the contents of the macroname macro by sending the MAC? macroname command.
If you are working in PIMikroMove on the Controller macros tab:

- Click the Read list of macros from controller icon.
- Mark the macro to be checked in the list on the left side and click the Load selected macro from controller icon.

Example: Moving an axis back and forth

**INFORMATION**

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

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

- Record the macros by sending:

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

**Starting a macro execution**

**INFORMATION**

Any commands can be sent from the command line when a macro is running on the controller. The macro content and move commands received from the command line can overwrite each other.
**INFORMATION**

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

**INFORMATION**

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

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

1. Start the macro execution:
   - If the macro is to be executed once, send the `MAC START macroname string` command, whereby `macroname` indicates the name of the macro.
   - If the macro is to be executed n times, send the `MAC NSTART macroname n string` command, whereby `macroname` indicates the name of the macro and `n` indicates the number of executions.

   *string* stands for the values of local variables. The values only have to be given when the macro contains corresponding local variables. The sequence of the values in the input must correspond to the numbering of the appropriate local variables, starting with the value of the local variable 1. The individual values must be separated from each other with spaces.

2. If you want to check the macro execution:
   - Get whether a macro is being executed on the controller by sending the `#8` command.
   - Get the name of the macro that is currently being executed on the controller by sending the `RMC?` command.

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

**INFORMATION**

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

The axis 1 is to move back and forth. The travel to the left and to the right is to be variably adjustable without having to change the used macros. Local and global variables are therefore used.

1. Create the global variables **LEFT** and **RIGHT** by sending:
   ```
   VAR LEFT 5
   VAR RIGHT 15
   ```
LEFT thus has the value 5, and RIGHT has the value 15. These values can be changed at any
time, e.g., by sending the VAR command again.

– Create the global variables again each time that the E-727 is switched on or rebooted,
since they are only written to the volatile memory of the E-727.

2. Record the MOVLR macro by sending:

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

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

3. Record the MOVWAI macro by sending:

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

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

4. Start the execution of the MOVLR macro by sending:

```
MAC NSTART movlr 5
```

The MOVLR macro is executed five times in succession, i.e., axis 1 alternately moves to the
positions 5 and 15 five times. You can also select any other value for the number of
executions.

Example: Implementing multiple calls of a macro via a loop

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

The TESTDION macro checks the status of the digital input lines on the Digital I/O socket. It uses a
local variable to identify the digital input line (1 to 4). So that the TESTDION macro does not have
to be called separately for each input line, another macro with a loop is recorded.
Record the LOOPDION macro by sending:

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

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

**Stopping a macro execution**

**INFORMATION**

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

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

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

**Setting up a start-up macro**

Any macro can be defined as the start-up macro. The start-up macro is executed each time that the E-727 is switched on or rebooted.

**INFORMATION**

Deleting a macro does not delete its selection as the start-up macro.
In the following, PITerminal or the Command entry window of PIMikroMove is used to enter commands. Details on working with the Controller macros tab in PIMikroMove are found in the PIMikroMove manual.

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

Example: Moving an axis via a start-up macro to a certain position in closed-loop operation

<table>
<thead>
<tr>
<th>INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>When macros are recorded on the Controller macros tab in PIMikroMove, the MAC BEG and MAC END commands must be omitted.</td>
</tr>
</tbody>
</table>

Setting the servo mode via a start-up macro is not necessary because this can be done via the value of the Power Up Servo ON Enable parameter (ID 0x07000800; 0 = Servo mode is not automatically switched on; 1 = Servo mode is automatically switched on). Furthermore, the AutoZero procedure can be executed automatically according to the value of the Power Up AutoZero Enable parameter (ID 0x07000802; 0 = AutoZero procedure is not automatically executed; 1 = AutoZero procedure is automatically executed). For that reason, the STARTCL macro only starts the motion to the desired position which is 10 in this example.

- Send:
  
  ```
  CCL 1 advanced
  SEP 100 1 0x07000800 1
  SEP 100 1 0x07000802 1
  MAC BEG startcl
  MOV 1 10
  MAC END
  MAC DEF startcl
  ```

Deleting a macro

<table>
<thead>
<tr>
<th>INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A running macro may not be deleted.</td>
</tr>
</tbody>
</table>
In the following, PITerminal or the **Command entry** window of PIMikroMove is used to enter
commands. Details on working with the **Controller macros** tab in PIMikroMove are found in the
PIMikroMove manual.

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

**Variables**

For more flexible programming, the E-727 supports variables. While global variables are always
available, local variables are only valid for a given macro. Typically, variables are used when
working with macros.

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

The following conventions apply to variable names:

- Variable names must not contain special characters (especially no “$”).
- The maximum number of characters is 8.
- Names of global variables can consist of characters A to Z and 0 to 9. They must start with an
alphanetic character.
- Names of local variables must not contain alphabetic characters. Possible characters are 0
to 9.
- The variable name can also be given via the value of another variable.

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

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

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

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

**Local variables:**

- Local variables can only be used in macros.
- At present, the controller firmware supports three local variables: 0, 1 and 2.
- The values of the local variables 1 and 2 are given as arguments of the MAC START or MAC
NSTART command when starting the macro.

The command formats are:

```
MAC_START <macroname> [::<STRING1> [::<STRING2>]]
MAC_NSTART <macroname> <uint> [::<STRING1> [::<STRING2>]]
```

- `<STRING1>` and `<STRING2>` indicate the values for the local variables 1 and 2 used in the
macro. `<STRING1>` and `<STRING2>` can be given directly or via the values of variables. `<uint>`
defines the number of times the macro is to be run. See the **MAC** command (p. 76)
description for more information.
- The local variable 0 is read-only. Its value gives the number of arguments (i. e. values of local
variables) set when starting the macro.
• Inside a macro, the values of local variables can be modified using ADD (p. 63), CPY (p. 65) or VAR (p. 82), and can be deleted with VAR (except for the local variable 0).
• As long as the macro is running, the values of the local variables can be queried with

\[
\text{VAR? 0} \\
\text{VAR? 1} \\
\text{VAR? 2}
\]

The queries can be sent inside or outside of the macro.

**Global variables:**
• Global variables can be used inside and outside of macros.
• The maximum number of global variables is 10.
• Global variables are created and modified using ADD, CPY or VAR. They can be deleted with VAR.
• The variable values can be queried with VAR?.


GCS Commands

Notation

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

- **Angle brackets** indicate an argument of a command, can be an item identifier or a command-specific parameter.
- **Square brackets** indicate an optional entry.
- **Curly brackets** indicate a repetition of entries, i.e. that it is possible to access more than one item (e.g. several axes) in one command line.
- **Quotation marks** indicate that the characters enclosed are returned or to be entered.

GCS Syntax for Syntax Version 2.0

See „GCS Syntax“ in the E-725 user manual (PZ197E) for a detailed syntax description.

Command Overview

The table below lists the commands supported by the E-727 in alphabetical order. Commands highlighted in grey are described in this Technical Note. For descriptions of all other commands see the E-725 user manual (PZ197E).

<table>
<thead>
<tr>
<th>Command</th>
<th>Format</th>
<th>Short Description</th>
<th>Details see</th>
</tr>
</thead>
<tbody>
<tr>
<td>#5</td>
<td>#5</td>
<td>Request Motion Status</td>
<td>PZ197E</td>
</tr>
<tr>
<td>#7</td>
<td>#7</td>
<td>Request Controller Ready Status</td>
<td>p. 63</td>
</tr>
<tr>
<td>#8</td>
<td>#8</td>
<td>Query If Macro Is Running</td>
<td>p. 63</td>
</tr>
<tr>
<td>#9</td>
<td>#9</td>
<td>Get Wave Generator Status</td>
<td>PZ197E</td>
</tr>
<tr>
<td>#24</td>
<td>#24</td>
<td>Stop All Axes</td>
<td>PZ197E</td>
</tr>
<tr>
<td>*IDN?</td>
<td>*IDN?</td>
<td>Get Device Identification</td>
<td>PZ197E</td>
</tr>
<tr>
<td>ADD</td>
<td>ADD &lt;Variable&gt; &lt;FLOAT1&gt; &lt;FLOAT2&gt;</td>
<td>Add and Save To Variable</td>
<td>p. 63</td>
</tr>
<tr>
<td>AOS</td>
<td>AOS {&lt;AxisID&gt; &lt;Offset&gt;}</td>
<td>Set Analog Input Offset</td>
<td>PZ197E</td>
</tr>
<tr>
<td>AOS?</td>
<td>AOS? [{&lt;AxisID&gt;}]</td>
<td>Get Analog Input Offset</td>
<td>PZ197E</td>
</tr>
<tr>
<td>ATZ</td>
<td>ATZ [{&lt;AxisID&gt; &lt;LowValue&gt;}]</td>
<td>Set Automatic Zero Point Calibration</td>
<td>p. 41, PZ197E</td>
</tr>
<tr>
<td>Command</td>
<td>Format</td>
<td>Short Description</td>
<td>Details see</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ATZ?</td>
<td>ATZ? [{&lt;AxisID&gt;}]</td>
<td>Get State Of Automatic Zero Point Calibration</td>
<td>PZ197E</td>
</tr>
<tr>
<td>CCL</td>
<td>CCL &lt;Level&gt; [&lt;PSWD&gt;]</td>
<td>Set Command Level</td>
<td>PZ197E</td>
</tr>
<tr>
<td>CCL?</td>
<td>CCL?</td>
<td>Get Command Level</td>
<td>PZ197E</td>
</tr>
<tr>
<td>CPY</td>
<td>CPY &lt;Variable&gt; &lt;CMD?&gt;</td>
<td>Copy Into Variable</td>
<td>p. 65</td>
</tr>
<tr>
<td>CST?</td>
<td>CST? [{&lt;AxisID&gt;}]</td>
<td>Get Assignment Of Stages To Axes</td>
<td>PZ197E</td>
</tr>
<tr>
<td>CSV?</td>
<td>CSV?</td>
<td>Get Current Syntax Version</td>
<td>PZ197E</td>
</tr>
<tr>
<td>CTO</td>
<td>CTO {&lt;TrigOutID&gt; &lt;CTOPam&gt; &lt;Value&gt;}</td>
<td>Set Configuration Of Trigger Output</td>
<td>PZ197E</td>
</tr>
<tr>
<td>CTO?</td>
<td>CTO? [{&lt;TrigOutID&gt; &lt;CTOPam&gt;}]</td>
<td>Get Configuration Of Trigger Output</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DDL</td>
<td>DDL &lt;DDLtableID&gt; &lt;StartPoint&gt; &lt;ValueN&gt;</td>
<td>Set DDL Table Value(s)</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DDL?</td>
<td>DDL? [{&lt;StartPoint&gt; &lt;NumberOfPoints&gt; []&lt;DDLtableID&gt;[]}</td>
<td>Get DDL Table Value(s)</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DEL</td>
<td>DEL &lt;uint&gt;</td>
<td>Delay The Command Interpreter</td>
<td>p. 65</td>
</tr>
<tr>
<td>DIA?</td>
<td>DIA? [{&lt;MeasureID&gt;}]</td>
<td>Get Diagnosis Information</td>
<td>p. 65</td>
</tr>
<tr>
<td>DPO</td>
<td>DPO [{&lt;AxisID&gt;}]</td>
<td>DDL Parameter Optimization</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DRC</td>
<td>DRC {&lt;RecTableID&gt; &lt;Source&gt; &lt;RecOption&gt;}</td>
<td>Set Data Recorder Configuration</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DRC?</td>
<td>DRC? [{&lt;RecTableID}&gt;]</td>
<td>Get Data Recorder Configuration</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DRL?</td>
<td>DRL? [{&lt;RecTableID&gt;}]</td>
<td>Get Number Of Recorded Points</td>
<td>p. 67</td>
</tr>
<tr>
<td>DRR?</td>
<td>DRR? [{&lt;StartPoint&gt; &lt;NumberOfPoints&gt; []&lt;RecTableID&gt;}]</td>
<td>Get Recorded Data Values</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DRT</td>
<td>DRT {&lt;RecTableID&gt; &lt;TriggerSource&gt; &lt;Value&gt;}</td>
<td>Set Data Recorder Trigger Source</td>
<td>p. 67</td>
</tr>
<tr>
<td>DRT?</td>
<td>DRT? [{&lt;RecTableID&gt;}]</td>
<td>Get Data Recorder Trigger Source</td>
<td>p. 69</td>
</tr>
<tr>
<td>DTC</td>
<td>DTC &lt;DDLtableID&gt;}</td>
<td>Clear DDL Table Data</td>
<td>PZ197E</td>
</tr>
<tr>
<td>DTL?</td>
<td>DTL? [&lt;{DDLtableID}&gt;]</td>
<td>Get DDL Table Length</td>
<td>PZ197E</td>
</tr>
<tr>
<td>ERR?</td>
<td>ERR?</td>
<td>Get Error Number</td>
<td>PZ197E</td>
</tr>
<tr>
<td>GWD?</td>
<td>GWD? [&lt;{StartPoint&gt; &lt;NumberOfPoints&gt; [&lt;{WaveTableID}&gt;]}]</td>
<td>Get Wave Table Data</td>
<td>PZ197E</td>
</tr>
<tr>
<td>HDI?</td>
<td>HDI?</td>
<td>Get Help For Interpretation Of DIA? Response</td>
<td>p. 69</td>
</tr>
<tr>
<td>HDR?</td>
<td>HDR?</td>
<td>Get All Data Recorder Options</td>
<td>PZ197E</td>
</tr>
<tr>
<td>HLP?</td>
<td>HLP?</td>
<td>Get List Of Available Commands</td>
<td>PZ197E</td>
</tr>
<tr>
<td>HLT</td>
<td>HLT [&lt;{AxisID}&gt;]</td>
<td>Halt Motion Smoothly</td>
<td>p. 70</td>
</tr>
<tr>
<td>HPA?</td>
<td>HPA?</td>
<td>Get List Of Available Parameters</td>
<td>PZ197E</td>
</tr>
<tr>
<td>Command</td>
<td>Format</td>
<td>Short Description</td>
<td>Details see</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>HPV?</td>
<td>HPV?</td>
<td>Get Parameter Value Description</td>
<td>p. 70</td>
</tr>
<tr>
<td>IDN?</td>
<td>IDN?</td>
<td>Get Device Identification</td>
<td>PZ197E</td>
</tr>
<tr>
<td>IFC</td>
<td>IFC {&lt;InterfacePam&gt; &lt;PamValue&gt;}</td>
<td>Set Interface Parameters Temporarily</td>
<td>PZ197E</td>
</tr>
<tr>
<td>IFC?</td>
<td>IFC? [[&lt;InterfacePam&gt;]]</td>
<td>Get Current Interface Parameters</td>
<td>PZ197E</td>
</tr>
<tr>
<td>IFS</td>
<td>IFS &lt;Ps wd&gt; {&lt;InterfacePam&gt; &lt;PamValue&gt;}</td>
<td>Set Interface Parameters As Default Values</td>
<td>PZ197E</td>
</tr>
<tr>
<td>IFS?</td>
<td>IFS? [[&lt;InterfacePam&gt;]]</td>
<td>Get Interface Parameters As Default Values</td>
<td>PZ197E</td>
</tr>
<tr>
<td>IMP</td>
<td>IMP &lt;AxisID&gt; &lt;Amplitude&gt;</td>
<td>Start Impulse And Response Measurement</td>
<td>PZ197E</td>
</tr>
<tr>
<td>JOG</td>
<td>JOG {&lt;AxisID&gt; &lt;Velocity&gt;}</td>
<td>Start Motion With Given Velocity</td>
<td>p. 72</td>
</tr>
<tr>
<td>JOG?</td>
<td>JOG? [[&lt;AxisID&gt;]]</td>
<td>Get Velocity For Motion Caused By JOG</td>
<td>p. 74</td>
</tr>
<tr>
<td>JRC</td>
<td>JRC &lt;Jump&gt; &lt;CMD?&gt; &lt;OP&gt; &lt;Value&gt;</td>
<td>Jump Relatively Depending On Condition</td>
<td>p. 74</td>
</tr>
<tr>
<td>MAC</td>
<td>MAC &lt;keyword&gt; {&lt;parameter&gt;}</td>
<td>Call Macro Function</td>
<td>p. 76</td>
</tr>
<tr>
<td>MAC?</td>
<td>MAC? [&lt;macroname&gt;]</td>
<td>List Macros</td>
<td>p. 78</td>
</tr>
<tr>
<td>MEX</td>
<td>MEX &lt;CMD?&gt; &lt;OP&gt; &lt;Value&gt;</td>
<td>Stop Macro Execution Due To Condition</td>
<td>p. 79</td>
</tr>
<tr>
<td>MOV</td>
<td>MOV {&lt;AxisID&gt; &lt;Position&gt;}</td>
<td>Set Target Position</td>
<td>PZ197E</td>
</tr>
<tr>
<td>MOV?</td>
<td>MOV? [[&lt;AxisID&gt;]]</td>
<td>Get Target Position</td>
<td>PZ197E</td>
</tr>
<tr>
<td>MVR</td>
<td>MVR {&lt;AxisID&gt; &lt;Distance&gt;}</td>
<td>Set Target Relative To Current Position</td>
<td>PZ197E</td>
</tr>
<tr>
<td>ONT?</td>
<td>ONT? [[&lt;AxisID&gt;]]</td>
<td>Get On-Target State</td>
<td>PZ197E</td>
</tr>
<tr>
<td>OVF?</td>
<td>OVF? [[&lt;AxisID&gt;]]</td>
<td>Get Overflow State</td>
<td>PZ197E</td>
</tr>
<tr>
<td>POS?</td>
<td>POS? [[&lt;AxisID&gt;]]</td>
<td>Get Real Position</td>
<td>PZ197E</td>
</tr>
<tr>
<td>PUN?</td>
<td>PUN? [[&lt;AxisID&gt;]]</td>
<td>Get Axis Unit</td>
<td>p. 81</td>
</tr>
<tr>
<td>RBT</td>
<td>RBT</td>
<td>Reboot System</td>
<td>PZ197E</td>
</tr>
<tr>
<td>Command</td>
<td>Format</td>
<td>Short Description</td>
<td>Details see</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>RMC?</td>
<td>RMC?</td>
<td>List Running Macros</td>
<td>p. 81</td>
</tr>
<tr>
<td>RPA</td>
<td>RPA [{ItemID} &lt;PamID&gt;]</td>
<td>Reset Volatile Memory Parameters</td>
<td>PZ197E</td>
</tr>
<tr>
<td>RTR</td>
<td>RTR &lt;RecordTableRate&gt;</td>
<td>Set Record Table Rate</td>
<td>PZ197E</td>
</tr>
<tr>
<td>RTR?</td>
<td>RTR?</td>
<td>Get Record Table Rate</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SEP</td>
<td>SEP &lt;Pswd&gt; [{ItemID} &lt;PamID&gt; &lt;PamValue&gt;]</td>
<td>Set Nonvolatile Memory Parameters</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SEP?</td>
<td>SEP? [{ItemID} &lt;PamID&gt;]</td>
<td>Get Nonvolatile Memory Parameters</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SPA</td>
<td>SPA [{ItemID} &lt;PamID&gt; &lt;PamValue&gt;]</td>
<td>Set Volatile Memory Parameters</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SPA?</td>
<td>SPA? [{ItemID} &lt;PamID&gt;]</td>
<td>Get Volatile Memory Parameters</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SSN?</td>
<td>SSN?</td>
<td>Get Device Serial Number</td>
<td>p. 81</td>
</tr>
<tr>
<td>STE</td>
<td>STE &lt;AxisID&gt; &lt;Amplitude&gt;</td>
<td>Start Step And Response Measurement</td>
<td>PZ197E</td>
</tr>
<tr>
<td>STP</td>
<td>STP</td>
<td>Stop All Axes</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SVA</td>
<td>SVA {AxisID} &lt;Amplitude&gt;</td>
<td>Set Open-Loop Axis Value</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SVA?</td>
<td>SVA? [{AxisID}]</td>
<td>Get Open-Loop Axis Value</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SVO</td>
<td>SVO {AxisID} &lt;ServoState&gt;</td>
<td>Set Servo Mode</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SVO?</td>
<td>SVO? [{AxisID}]</td>
<td>Get Servo Mode</td>
<td>PZ197E</td>
</tr>
<tr>
<td>SVR</td>
<td>SVR {AxisID} &lt;Difference&gt;</td>
<td>Set Relative Open-Loop Axis Value</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TIO?</td>
<td>TIO?</td>
<td>Tell Digital I/O Lines</td>
<td>p. 82</td>
</tr>
<tr>
<td>TLT?</td>
<td>TLT?</td>
<td>Get Number of DDL Tables</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TMX?</td>
<td>TMX? [{AxisID}]</td>
<td>Get Maximum Commandable Position</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TNR?</td>
<td>TNR?</td>
<td>Get Number Of Record Tables</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TNS?</td>
<td>TNS? [{InputSignalID}]</td>
<td>Get Normalized Input Signal Value</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TPC?</td>
<td>TPC?</td>
<td>Get Number of Output Signal Channels</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TSC?</td>
<td>TSC?</td>
<td>Get Number of Input Signal Channels</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TWC</td>
<td>TWC</td>
<td>Clear All Wave Related Triggers</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TWG?</td>
<td>TWG?</td>
<td>Get Number of Wave Generators</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TWS</td>
<td>TWS &lt;TrigOutID&gt; &lt;PointNumber&gt; &lt;Switch&gt;</td>
<td>Set Trigger Line Action To Waveform Point</td>
<td>PZ197E</td>
</tr>
<tr>
<td>TWS?</td>
<td>TWS? [StartPoint] [NumberOfPoints] [{TrigOutID}]</td>
<td>Get Trigger Line Action At Waveform Point</td>
<td>PZ197E</td>
</tr>
</tbody>
</table>
## Technical Note
E727T0005, valid for E-727.3CD, E-727.3CDA, E-727.3SD, E-727.3SDA
BRO, 2015-10-07

<table>
<thead>
<tr>
<th>Command</th>
<th>Format</th>
<th>Short Description</th>
<th>Details see</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR</td>
<td>VAR &lt;Variable&gt; &lt;String&gt;</td>
<td>Set Variable Value</td>
<td>p. 82</td>
</tr>
<tr>
<td>VAR?</td>
<td>VAR? [[&lt;Variable&gt;]]</td>
<td>Get Variable Value</td>
<td>p. 84</td>
</tr>
<tr>
<td>VCO</td>
<td>VCO {&lt;AxisID&gt; &lt;VelCtrlState&gt;}</td>
<td>Set Velocity Control Mode</td>
<td>p. 84</td>
</tr>
<tr>
<td>VCO?</td>
<td>VCO? [[&lt;AxisID&gt;]]</td>
<td>Get Velocity Control Mode</td>
<td>p. 85</td>
</tr>
<tr>
<td>VEL</td>
<td>VEL {&lt;AxisID&gt; &lt;Velocity&gt;}</td>
<td>Set Closed-Loop Velocity</td>
<td>PZ197E</td>
</tr>
<tr>
<td>VEL?</td>
<td>VEL? [[&lt;AxisID&gt;]]</td>
<td>Get Closed-Loop Velocity</td>
<td>PZ197E</td>
</tr>
<tr>
<td>VOL?</td>
<td>VOL? [[&lt;OutputSignalID&gt;]]</td>
<td>Get Voltage Of Output Signal Channel</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WAC</td>
<td>WAC &lt;CMD?&gt; &lt;OP&gt; &lt;Value&gt;</td>
<td>Wait For Condition</td>
<td>p. 85</td>
</tr>
<tr>
<td>WAV</td>
<td>WAV &lt;WaveTableID&gt; &lt;AppendWave&gt; &lt;WaveType&gt; &lt;WaveTypeParameters&gt;</td>
<td>Set Waveform Definition</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WAV?</td>
<td>WAV? [[&lt;WaveTableID&gt; &lt;WaveParameterID&gt;]]</td>
<td>Get Waveform Definition</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WCL</td>
<td>WCL {&lt;WaveTableID&gt;}</td>
<td>Clear Wave Table Data</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WGC</td>
<td>WGC {&lt;WaveGenID&gt; &lt;Cycles&gt;}</td>
<td>Set Number Of Wave Generator Cycles</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WGC?</td>
<td>WGC? [[&lt;WaveGenID&gt;]]</td>
<td>Get Number Of Wave Generator Cycles</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WGO</td>
<td>WGO {&lt;WaveGenID&gt; &lt;StartMode&gt;}</td>
<td>Set Wave Generator Start/Stop Mode</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WGO?</td>
<td>WGO? [[&lt;WaveGenID&gt;]]</td>
<td>Get Wave Generator Start/Stop Mode</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WGR</td>
<td>WGR</td>
<td>Starts Recording In Sync With Wave Generator</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WOS</td>
<td>WOS {&lt;WaveGenID&gt; &lt;Offset&gt;}</td>
<td>Set Wave Generator Output Offset</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WOS?</td>
<td>WOS? [[&lt;WaveGenID&gt;]]</td>
<td>Get Wave Generator Output Offset</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WPA</td>
<td>WPA &lt;Pswd&gt; [[&lt;ItemID&gt; &lt;PamID&gt;]]</td>
<td>Save Parameters To Nonvolatile Memory</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WSL</td>
<td>WSL {&lt;WaveGenID&gt; &lt;WaveTableID&gt;}</td>
<td>Set Connection Of Wave Table To Wave Generator</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WSL?</td>
<td>WSL? [[&lt;WaveGenID&gt;]]</td>
<td>Get Connection Of Wave Table To Wave Generator</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WTR</td>
<td>WTR {&lt;WaveGenID&gt; &lt;WaveTableRate&gt; &lt;InterpolationType&gt;}</td>
<td>Set Wave Generator Table Rate</td>
<td>PZ197E</td>
</tr>
<tr>
<td>WTR?</td>
<td>WTR? [[&lt;WaveGenID&gt;]]</td>
<td>Get Wave Generator Table Rate</td>
<td>PZ197E</td>
</tr>
</tbody>
</table>
Command Descriptions

This section describes the commands which are not contained in the E-725 user manual (PZ197E). For descriptions of all other commands listed in “Command Overview” (p. 58), see the E-725 user manual.

#7 (Request Controller Ready Status)

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

Arguments: None

Response:

B1h (ASCII character 177 = "±" in Windows) if controller is ready

B0h (ASCII character 176 = "°" in Windows) if controller is not ready
(e. g. performing a reference move)

Troubleshooting: The response characters may appear differently in non-Western character sets or other operating systems.

#8 (Query if Macro Is Running)

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

Format: #8

Arguments: None

Response:

<uint>=0 no macro is running
<uint>=1 a macro is currently running

ADD (Add and Save to Variable)

Description: Adds two values and saves the result to a variable (p. 56).

The variable is present in volatile memory (RAM) only.

Format: ADD <Variable> <FLOAT1> <FLOAT2>
Arguments: <Variable> is the name of the variable to which the result is to be saved.

<FLOAT1> is the first summand.

<FLOAT2> is the second summand.

For the summands, floating point numbers are expected. They can be given directly or via the value of a variable.

Response: None

Notes: Local variables can be set using ADD in macros only.

Example 1: Value $B is added to value $A, and the result is saved to variable C:

```
ADD C $A $B
```

Example 2: The name of the variable to which the result is to be copied is given via the value of another variable:

```
Send: VAR?
Receive:
A=468
B=123
3Z=WORKS

Send: ADD A{3Z} $A $B
Send: VAR?
Receive:
A=468
B=123
AWORKS=591
3Z=WORKS

Send: ADD ${3Z} $A $B
Send: VAR?
Receive:
A=468
B=123
AWORKS=591
WORKS=591
3Z=WORKS
```
**CPY (Copy Into Variable)**

**Description:** Copies a command response to a variable (p. 56).

The variable is present in volatile memory (RAM) only.

**Format:** CPY <Variable> <CMD?>

**Arguments:**
- <Variable> is the name of the variable to which the command response is to be copied.
- <CMD?> is one query command in its usual notation. The response has to be a single value and not more.

**Response:** None

**Notes:** Local variables can be set using CPY in macros only.

**Example:** It is possible to copy the value of one variable (e.g. SOURCE) to another variable (e.g. TARGET):

```plaintext
CPY TARGET VAR? SOURCE
```
Response: \{<MeasureID>"="<MeasuredValue> LF\}

where

<MeasuredValue> gives the current value of the measurand, see below for details.

Notes: Use the response to HDI? (p. 69) to get descriptions and physical units of the supported measurands.

E-727 supports the following measurands:

<table>
<thead>
<tr>
<th>&lt;MeasureID&gt;</th>
<th>&lt;Description&gt; (get with HDI?)</th>
<th>Possible values of &lt;MeasuredValue&gt;</th>
</tr>
</thead>
</table>
| 1          | Temperature Status          | 0 = temperature of amplifier exceeds no threshold  
|            |                             | 2 = temperature of amplifier exceeds switch-off threshold |
| 2          | Amplifier Output Status (On/Off) | 1 = ON: amplifier output is active  
|            |                             | 0 = OFF: amplifier output is not active (temperature of amplifier exceeds switch-off threshold |

Examples:

The temperature threshold is not exceeded:
Send: DIA?
Receive: 1=0  
2=1

Temperature threshold is exceeded, and the amplifier output has been switched off automatically for that reason:
Send: DIA?
Receive: 1=2  
2=0

For possible measures in case of exceeding a threshold see “Overtemp Protection of the E-727” (p. 44).

DIO? (Get Digital Input Lines)

Description: Gets the states of the specified digital input lines.

Use TIO? (p. 82) to get the number of available digital I/O lines.
Format: \( \text{DIO? [\{\text{DIOID}\}] } \)

Arguments: 
\(<\text{DIOID}>\) is the identifier of the digital input line, see below for details.

Response: 
\(\{\text{DIOID}\} = \{\text{InputOn}\} \) LF

where
\(\text{InputOn}\) gives the state of the digital input line, see below for details.

Notes: 
You can use the DIO? command to directly read the digital input lines 1 to 4 that are located on the Digital I/O socket (p. 98).

The \(<\text{DIOID}>\) identifiers to use for the lines are 1 to 4. If the identifier is omitted, all lines are queried.

If \(\text{InputOn} = 0\), the digital input is LOW/OFF; if \(\text{InputOn} = 1\), the digital input is HIGH/ON.

---

**DRL? (Get Number of Recorded Points)**

Description: 
Reads the number of points comprised by the last recording.

Format: \( \text{DRL? [\{\text{RecTableID}\}] } \)

Arguments: 
\(<\text{RecTableID}>\) is one data recorder table of the controller

Response: 
\(\{\text{RecTableID}\} = \{\text{uint}\} \) LF

where
\(\text{uint}\) gives the number of points recorded with the last recording.

Notes: 
The number of points is reset to zero for a data recorder table when changing its configuration with DRC.

---

**DRT (Set Data Recorder Trigger Source)**

Description: 
Defines a trigger source for the given data recorder table.

Format: \( \text{DRT} \ <\text{RecTableID}> \ <\text{TriggerSource}> \ <\text{Value}> \)
Arguments: <RecTableID> is one data recorder table of the controller. See below for details.

<TriggerSource> ID of the trigger source, see below for a list of available options.

<Value> depends on the trigger source, can be a dummy, see below.

Response: none

Notes: The number of available data recorder tables can be read with TNR?. The answer gives the value of the Data Recorder Chan Number parameter, ID 0x16000300. Using SPA or SEP you can change the parameter value in the range of 1 to 8 to increase or decrease the number of data recorder tables.

At present, the specified trigger source is always set for all data recorder tables, irrespective of the <RecTableID> value given in the DRT command.

With HDR? you will obtain a list of available record options and trigger options and additional information about data recording.

For detailed information see "Data Recording" in the E-725 user manual (PZ197E).

Available trigger options:

0 = default setting; data recording is triggered with IMP, STE, WGO, WGR; <Value> must be a dummy

1 = any command changing target position or voltage (MVR, MOV, SVA, SVR; in addition to IMP, STE, WGO, WGR); <Value> must be a dummy

3 = external trigger; <Value> gives the ID of the digital input line to be used for trigger input; if 0, any digital input line is used (see “Digital I/O”, p. 98 for available lines).

4 = immediately (means that the DRT command itself triggers); <Value> must be a dummy
DRT? (Get Data Recorder Trigger Source)

Description: Gets the trigger source for the data recorder tables.

Format:  

DRT? [{<RecTableID>}]

Arguments:  

<RecTableID> is one data recorder table of the controller.

Response:  

{<RecTableID>"="<TriggerSource> <Value> LF}

where

<TriggerSource> is the identifier of the trigger source.

<Value> depends on the trigger source; 0 is a dummy.

Further information is found in the description of the DRT command (p. 67).

HDI? (Get Help For Interpretation Of DIA? Response)

Description: Lists descriptions and physical units for the measurands that can be queried with the DIA? command (p. 65).

Format:  

HDI?

Arguments: None

Response:  

{<MeasureID>"="<Description>TAB<PhysUnit> LF}

where

<MeasureID> is the identifier of the measurand.

<Description> is the name of the measurand.

<PhysUnit> is the physical unit of the measurand.

Notes: With E-727, the response to HDI? is as follows:

HDI?
1=Temperature Status
2=Amplifier Output Status (On/Off)
end of help

The measurands that can be queried with E-727 have no physical unit.

See DIA? for possible values of the measurands.
HLT (Halt Motion Smoothly)

Description: Halts the motion of given axes smoothly. For details see the notes below.

Error code 10 is set.

#24 and STP in contrast abort current motion as fast as possible for the controller without taking care of a given deceleration.

Format: HLT [{<AxisID>}] 

Arguments: <AxisID>: is one axis of the controller, if omitted all axes are halted

Response: none

Troubleshooting: Illegal axis identifier

Notes: HLT stops motion of all axes caused by move commands (MOV, MVR, SVA, SVR).

After the axes have been stopped, if servo is on their target positions are set to their current positions, or if servo is off, their open-loop control values are set to their last valid control values.

HPV? (Get Parameter Value Description)

Description: Responds with a help string which contains possible parameters values. Use HPA? instead to get a help string which contains all available parameters with short descriptions.

Format: HPV?

Arguments: none

Response: <string> has the following format:

“#Possible parameter values are:
{<PamID> <ItemID> “=” <ListType>
 [ {TAB <PossibleValue> “=” <ValueDescription>} ] }
#CCL levels are:
{<PamID> <ItemID> “=”<CmdLevel> }
end of help”
where

<PamID> is the ID of one parameter, hexadecimal format

<ItemID> is one item (axis, channel, whole system) of the controller, if item=0 the description is valid for all items

<ListType> determines how the possible parameter values listed in the string have to be interpreted:
  0 = parameter not applicable for this item
  1 = enumeration
  2 = min/max

<PossibleValue> is one value from the allowed data range

<ValueDescription> is the meaning of the corresponding value

Some parameters are write protected (by a command level > 1) for certain items. These parameters are listed below the “#CCL levels are” line.

.CmdLevel> is the command level which allows write access to the parameter value.
JOG (Start Motion With Given Velocity)

Description: Starts motion with the given (constant) velocity for the given axis. The sign of the velocity value gives the direction of motion.

The servo mode must be switched on when this command is used (closed-loop operation).

When motion started with JOG is executed, the target value is changed continuously according to the given velocity (can be checked with MOV?).

Motion started with JOG is executed in addition to motion started with other move commands (e.g. MOV or MVR).

As long as the axis motion is caused by JOG only, the axis stays on target (i.e. ONT? responds with 1 since the target is continuously adapted to the actual motion).

Motion started by JOG is stopped in the following cases:
- The velocity is set to 0 with JOG.
- #24, STP or HLT is sent: these commands set the velocity for JOG to 0.
- A travel range limit is reached: the velocity for JOG remains unchanged, no error is set, the target value is set equal to the limit value.

JOG can be changed while the axis is moving.

Format: JOG {<AxisID> <Velocity>}

Arguments

<AxisID> is one axis of the controller

<Velocity> gives the velocity and the direction for axis motion started with JOG. With a positive value the target is increased, with a negative value the target is decreased; zero stops the motion caused by JOG. float, signed

Response: none
Example:

Send: MOV? 1
Receive: 1=0.000000000e+00
Send: POS? 1
Receive: 1=-9.843791835e-03
Send: JOG 1 0.001

Note: According to the JOG command, axis 1 now moves with 0.001 µm/s in positive direction. The target is changed with every servo cycle.

Send: ONT?
Receive: 1=1
Send: #5
Receive: 0

Note: Although axis 1 is moving, it is on target since motion is caused by JOG only. Hence the motion status queried with #5 is “no axis is moving”.

Send: MOV? 1
Receive: 1=1.914202720e-01
Send: POS? 1
Receive: 1=1.929343045e-01
Send: MOV? 1
Receive: 1=1.950851232e-01
Send: POS? 1
Receive: 1=1.963570416e-01
Send: MOV? 1
Receive: 1=1.991461724e-01
Send: POS? 1
Receive: 1=2.016617247e-01
Send: MOV? 1
Receive: 1=2.049337626e-01
Send: POS? 1
Receive: 1=2.078935951e-01

Note: MOV? and POS? are sent alternately. The response shows that the target is constantly increased, and that the current position follows the target values.

Send: JOG? 1
Receive: 1=1.000000047e-03
Send: STP
Send: JOG? 1
Receive: 1=0.000000000e+00
Send: TMX? 1
Receive: 1=1.000000000e+02
Send: MOV  1  100
Send: POS?  1
Receive: 1=9.998706818e+01
Send: JOG  1  0.001
Send: ERR?
Receive: 0
Send: JOG?  1
Receive: 1=1.0000000047e-03

Note: The upper travel range limit of axis 1 is 100 µm. (TMX? 1 responds 100). After the corresponding MOV command, axis 1 is at its upper range limit. JOG cannot start motion in positive direction, but the given velocity value remains active. If axis 1 would be moved, for example, to position 10, then the motion commanded with JOG would start.

**JOG? (Get Velocity For Motion Caused By JOG)**

**Description:** Gets the velocity and direction for motion caused by JOG (p. 72).

**Format:** JOG? [[<AxisID>]]

**Arguments:** <AxisID> is one axis of the controller

**Response:** {<AxisID>"="<Velocity>LF}

where

<Velocity> is the velocity and the direction for axis motion caused by JOG. See JOG for details. float, signed

**Troubleshooting:** Illegal axis identifier

**JRC (Jump Relatively Depending On Condition)**

**Description:** Jumps relatively depending on a given condition of the following type: one given value is compared with a queried value according to a given rule.

Can only be used in macros.

**Format:** JRC <Jump> <CMD?> <OP> <Value>

**Arguments:** <Jump> is the size of the relative jump. -1 means the macro execution pointer jumps back to the previous line,
0 means the command is executed again, which is the same behavior as with WAC (p. 85). 1 jumps to the next line, making the command unnecessary, and 2 jumps over the next command. Only jumps within the current macro are allowed.

<CMD?> is one query command in its usual syntax. The response has to be a single value and not more. For an example see below.

<OP> is the operator to be used. The following operators are possible:
= <= < > >= !=
Important: There must be a blank space before and after the operator!

<Value> is the value to be compared with the response to <CMD?>.

Response: None

Troubleshooting: Check proper jump target

Example: Using the following macro, you can stop motion of axis 1 using a stop button connected to a digital input. The stop button is checked until the axis has reached the target position (ONT? query). When the stop button is pressed as long as the target position has not been reached yet: The response to the POS? 1 query is copied to the TARGET variable. Then this variable is used as second argument for the MOV command. Thus the stage stays where it just was. To clean up, TARGET is defined as empty with the VAR command which deletes the variable.

Write the "stop" macro:
MAC BEG stop
MOV 1 20
JRC 2 DIO? 1 = 1
JRC -1 ONT? 1 = 0
CPY TARGET POS? 1
MOV 1 ${TARGET}
VAR TARGET
MAC END
MAC (Call Macro Function)

Description: Calls a macro function. Permits recording, deleting and running macros on the controller.

Format:

MAC <keyword> {<parameter>}

in particular:

MAC BEG <macroname>
MAC DEF <macroname>
MAC DEF?
MAC DEL <macroname>
MAC END
MAC ERR?
MAC FREE?
MAC NSTART <macroname> <uint> [<String1> [<String2>]]
MAC START <macroname> [<String1> [<String2>]]

Arguments <keyword> determines which macro function is called. The following keywords and parameters are used:

MAC BEG <macroname>
Starts recording a macro to be named macroname on the controller; may not be used in a macro; the commands that follow become the macro. End the recording with MAC END. Note that erroneous macro content cannot be detected by sending the ERR? command.

MAC END
Stops macro recording (cannot become part of a macro)

MAC ERR?
Reports the last error which occurred during macro execution.
Response: <macroname> <uint1>"="<uint2> ""<""CMD"">"
where <macroname> is the name of the macro, <uint1> is the line in the macro, <uint2> is the error code and <""CMD""> is the erroneous command which was sent to the parser.

MAC FREE?
Gets the free memory space for macro recording (unit: number of characters)

MAC DEF <macroname>
Sets specified macro as start-up macro. This macro will be
automatically executed with the next switching-on or reboot of the controller. If `<macroname>` is omitted, the current start-up macro selection is canceled.

**MAC DEF?**
Asks for the start-up macro
Response: `<macroname>`
If no start-up macro is defined, the response is an empty string with the terminating character.

**MAC DEL `<macroname>`**
Deletes specified macro.

**MAC NSTART `<macroname>` `<uint>` [ `<String1>` [ `<String2>` ]]**
Repeats the specified macro `<uint>` times. Another execution is started when the last one is finished.
`<String1>` and `<String2>` are optional arguments which give the values for local variables 1 and 2 used in the given macro. `<String1>` and `<String2>` can be given directly or via the values of variables. Macro execution will fail if the macro contains local variables but `<String1>` and `<String2>` are omitted in the MAC NSTART command. See “Variables” (p. 56) for further details.

**MAC START `<macroname>` [ `<String1>` [ `<String2>` ]]**
Starts one execution of the specified macro. `<String1>` and `<String2>` have the same function as with MAC NSTART.

| Response: None |
| Troubleshooting: Macro recording is active (keywords BEG, DEL) or inactive (END) |
| Macro contains a disallowed MAC command |
Notes:
During macro recording no macro execution is allowed.

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

A running macro sends no responses to any interface.

The following commands provided by the E-727 can only be used in macros:
DEL (p. 65), JRC (p. 74), MEX (p. 79) and WAC (p. 85).

You can query with #8 (p. 63) if a macro is currently running on the controller.

For further details, see “Controller Macros” (p. 47).

**MAC? (List Macros)**

**Description:** Lists macros or content of a given macro.

**Format:** MAC? [<macroname>]

**Arguments**

<macroname>: name of the macro whose content shall be listed; if omitted, the names of all stored macros are listed.

**Response:** <string>

If <macroname> was given, <string> is the content of this macro;

If <macroname> was omitted, <string> is a list with the names of all stored macros

**Troubleshooting:** Macro <macroname> not found
MEX (Stop Macro Execution Due To Condition)

Description: Stops macro execution due to a given condition of the following type: a given value is compared with a queried value according to a given rule.

Can only be used in macros.

When the macro interpreter accesses this command, the condition is checked. If it is true, the current macro is stopped; otherwise macro execution is continued with the next line. Should the condition be fulfilled later, the interpreter will ignore it.

See also the WAC command (p. 85).

Format: MEX <CMD?> <OP> <Value>

Arguments <CMD?> is one query command in its usual syntax. The response has to be a single value and not more. For an example see below.

<OP> is the operator to be used. The following operators are possible:

= <= < > >= !=

Important: There must be a blank space before and after the operator!

<Value> is the value that is compared with the response to <CMD?>.

Response: None
Example: Send: **MAC START LOOP**

Note: Macro LOOP has the following contents:
- **MAC START KEY1**
- **MAC START KEY2**
- **MEX DIO? 4 = 1**
- **MAC START LOOP**

Macro KEY1 has the following contents:
- **MEX DIO? 4 = 1**
- **MEX DIO? 1 = 0**
- **MVR 1 1.0**
- **DEL 100**

Macro KEY2 has the following content:
- **MEX DIO? 4 = 1**
- **MEX DIO? 2 = 0**
- **MVR 1 -1.0**
- **DEL 100**

Macro LOOP forms an infinite loop by permanently calling KEY1, KEY2 and itself.

KEY1 checks the state of the digital input channel 1 (located on the Digital I/O socket (p. 98)). If it is not set (0), the macro is aborted, otherwise the macro will move axis 1 by 1.0 in positive direction (relative move).

KEY2 checks the state of the digital input channel 2 and moves axis 1 in negative direction accordingly.

Connecting the digital input channels 1, 2 and 4 with pushbuttons, it is possible to implement interactive control of an axis without any software assistance. The delay (DEL 100) is required to avoid generation of multiple MVR commands while pressing the pushbutton for a short time.

Channel 4 is used as a global exit. Since MEX stops execution of the current macro only, it must also be included in the calling macro, which would otherwise continue.
**PUN? (Get Axis Unit)**

*Description:* Gets the current unit of the axis.

If all arguments are omitted, the current unit for all axes is queried.

*Format:* PUN? [\{<AxisID>\}]

*Arguments:* <AxisID> is one axis of the controller.

*Response:* \{<AxisID>"="<string> LF\}

where

<string> is the current unit of the axis.

*Troubleshooting:* Illegal axis identifier

*Note:* Gets the Axis Unit parameter value in volatile memory (ID 0x07000601).

---

**RMC? (List Running Macros)**

*Description:* Lists macros which are currently running.

*Format:* RMC?

*Arguments:* none

*Response:* \{<macroname> LF\}

where

<macroname> is the name of one macro which is saved on the controller and currently running. The response is an empty line when no macro is running.

---

**SSN? (Get Device Serial Number)**

*Description:* Gets the serial number of the E-727.

*Format:* SSN?

*Arguments:* None

*Response:* <SerialNumber> is the serial number of the device.
TIO? (Tell Digital I/O Lines)

Description: Tells number of installed digital I/O lines

Format: TIO?

Arguments: none

Response: 

\[
I=\text{<uint1>}
\]

\[
O=\text{<uint2>}
\]

where

\[
\text{<uint1>} \text{ is the number of digital input lines.}
\]

\[
\text{<uint2>} \text{ is the number of digital output lines.}
\]

Notes: All digital I/O lines are found on the Digital I/O socket (p. 98) of the E-727.

The digital output lines reported by TIO? are output 1 to output 4. The output lines 1 to 3 can be programmed for trigger output using the CTO command (see „External Triggering/Signaling“, p. 45). The 4th output line (pin 8 of the Digital I/O socket) outputs the servo cycles of the E-727 and is not accessible for commands.

The digital input lines reported by TIO? are input 1 to input 4. The state of the lines can be queried with the DIO? command (p. 66). The input 1 and input 2 lines can be used with WGO to start the wave generator output (see “Wave Generator Started by Trigger Input” in the E-725 user manual (PZ197E)). Input 4 line can be configured as reset input (see p. 98).

VAR (Set Variable Value)

Description: Sets a variable to a certain value.

Local variables can be set using VAR in macros only. See “Variables” (p. 56) for details regarding local and global variables.

The variable is present in RAM only.

Format: VAR <Variable> <String>
Arguments: 

<Variable> is the name of the variable whose value is to be set.

<String> is the value to which the variable is to be set. If omitted, the variable is deleted.

The value can be given directly. In a macro, the value can also be given via the value of a variable.

See “Variables” (p. 56) for conventions regarding variable names and values.

Response: None

Example: In a macro, it is possible to set the value of one variable (e.g. TARGET) to that of another variable (e.g. SOURCE):

```
VAR TARGET ${SOURCE}
```

Use curly brackets if the name of the variable is longer than one character:

```
VAR A ONE
VAR VARB TWO
VAR $A 1
VAR ${VARB} 2
VAR $VARB 2 // this will result in an unwanted behavior
VAR?
A=ONE
VARB=TWO
ONE=1
TWO=2 // ${VARB}: is replaced by its value “TWO”.
ARB=2 // $VARB: $V is replaced by its (empty) value.
```

See ADD (p. 63) for another example.
VAR? (Get Variable Values)

Description: Gets values of variables.

If VAR? is combined with CPY (p. 65), JRC (p. 74), MEX (p. 79) or WAC (p. 85), the response to VAR? has to be a single value and not more.

More information regarding local and global variables can be found in “Variables” (p. 56).

Format: VAR? [{<Variable>}]

Arguments: <Variable> is the name of the variable to be queried. More information on name conventions can be found in "Variables" (p. 56).

If <Variable> is omitted, all global variables present in the RAM are listed.

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

where

<String> gives the value to which the variable is set.

Notes: Local variables can be queried using VAR? only when a macro with local variables is running. See “Variables” (p. 56) for details regarding local and global variables.

Example: See ADD (p. 63) for an example.

VCO (Set Velocity Control Mode)

Description: Sets the Velocity Control Mode of the specified axis to ON or OFF.

Format: VCO {<AxisID> <VelCtrlState>}

Arguments: <AxisID> is one axis of the controller

<VelCtrlState> can have the following values:
0 = Velocity Control Mode OFF
1 = Velocity Control Mode ON (default)

Response: none

Troubleshooting: Illegal axis identifier

Notes: When Velocity Control Mode is ON, the axis is driven with the specified slew rate (closed-loop operation: Servo Loop)
**Slew-Rate** parameter (ID 0x07000200); open-loop operation: **Open Loop Slew-Rate** parameter (ID 0x07000201).

### VCO? (Get Velocity Control Mode)

**Description:** Gets Velocity Control Mode of given axes.

If all arguments are omitted, gets mode of all axes.

**Format:**

VCO? [{<AxisID>}]  

**Arguments:** <AxisID> is one axis of the controller

**Response:** {<AxisID>"="<VelCtrlState> LF}

where

<VelCtrlState> is the current Velocity Control Mode of the axis:

0 = Velocity Control Mode OFF  
1 = Velocity Control Mode ON

**Troubleshooting:** Illegal axis identifier

### WAC (Wait For Condition)

**Description:** Waits until a given condition of the following type occurs: a specified value is compared with a queried value according a specified rule.

Can only be used in macros.

See also the MEX command (p. 79).

**Format:**

WAC <CMD?> <OP> <value>
<table>
<thead>
<tr>
<th>Arguments</th>
<th>&lt;CMD?&gt; is one query command in its usual notation. The response has to be a single value and not more. For an example see below.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;OP&gt; is the operator to be used. The following operators are possible:</td>
</tr>
<tr>
<td></td>
<td>= &lt;= &lt; &gt; &gt;= !=</td>
</tr>
<tr>
<td></td>
<td>Important: There must be a blank space before and after the operator!</td>
</tr>
<tr>
<td></td>
<td>&lt;value&gt; is the value to be compared with the response to &lt;CMD?&gt;.</td>
</tr>
<tr>
<td>Response:</td>
<td>None</td>
</tr>
<tr>
<td>Example:</td>
<td>Send:</td>
</tr>
<tr>
<td></td>
<td>MAC BEG LPMOTION</td>
</tr>
<tr>
<td></td>
<td>MVR 1 1</td>
</tr>
<tr>
<td></td>
<td>WAC ONT? 1 = 1</td>
</tr>
<tr>
<td></td>
<td>MVR 1 -1</td>
</tr>
<tr>
<td></td>
<td>WAC ONT? 1 = 1</td>
</tr>
<tr>
<td></td>
<td>MAC START LPMOTION</td>
</tr>
<tr>
<td></td>
<td>MAC END</td>
</tr>
<tr>
<td></td>
<td>MAC START LPMOTION</td>
</tr>
</tbody>
</table>

Note: Macro LPMOTION is first recorded and then started. WAC ONT? 1 = 1 waits until the response to ONT? 1 is 1=1. To form an infinite loop, the macro calls itself.
Parameters

Parameter Handling

For general information on parameters and their handling see “Controller Parameters” in the E-725 user manual (PZ197E).

**INFORMATION**

To change parameter values, you can work in *Single-Axis* window or in the *Device Parameter Configuration* window of PIMikroMove.

Working in the *Single-Axis* window is recommended if you want to check and adapt a certain selection of parameters, e.g. for tuning the settling behaviour of an axis (see pages 27, 33 and 36).

If you want to work in the *Device Parameter Configuration* window of PIMikroMove (for an example, see p. 25):

- Read "Device Parameter Configuration" in the PIMikroMove manual.
- Determine, modify and save parameter values with the corresponding buttons and menu items in the *Device Parameter Configuration* window of PIMikroMove.

Important New Parameters

The table below gives short descriptions of some parameters that were newly introduced with E-727. For the parameters highlighted in grey, possible values can be queried using the HPV? command.

<table>
<thead>
<tr>
<th>Parameter ID (hexadecimal)</th>
<th>Data Type</th>
<th>Command Level for Write Access</th>
<th>Item Type</th>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x020000100</td>
<td>INT</td>
<td>1</td>
<td>Input signal channel</td>
<td>Sensor Range Factor</td>
<td>Determines the usage of input signal channel 4 and the range of the analog inputs, see the description of the input signal channels in „Axes, Channels, Functional Elements“ (p. 11) and the information in „E-727.3CDA, E-727.3SDA: Analog I/O“ (p. 99).</td>
</tr>
<tr>
<td>0x02000102</td>
<td>INT</td>
<td>1</td>
<td>Input signal channel</td>
<td>Sensor Offset Factor</td>
<td>Only present with E-727.3SD and .3SDA models for SGS sensors; changed by the AutoZero procedure (see p. 41).</td>
</tr>
<tr>
<td>Parameter ID (hexadecimal)</td>
<td>Data Type</td>
<td>Command Level for Write Access</td>
<td>Item Type</td>
<td>Parameter Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------</td>
<td>-------------------------------</td>
<td>-----------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0x03003400</td>
<td>INT</td>
<td>1</td>
<td>Input signal channel</td>
<td>Input Numerical Format</td>
<td>Gives the data type for input values. 6 = SIGN32</td>
</tr>
<tr>
<td>0x08000500</td>
<td>INT</td>
<td>1</td>
<td>Axis</td>
<td>Enable Notch In Open-Loop</td>
<td>Enables usage of notch filter in open-loop operation. In closed-loop operation, the notch filters are always enabled. 0 = disable notch filter in open-loop operation 1 = enable notch filter in open-loop operation</td>
</tr>
<tr>
<td>0x08000600</td>
<td>INT</td>
<td>1</td>
<td>Axis</td>
<td>Notch Filter Calculation Method</td>
<td>See response to HPV? for possible values.</td>
</tr>
<tr>
<td>0x0A000003</td>
<td>INT</td>
<td>1</td>
<td>Output signal channel</td>
<td>Select Output Type</td>
<td>Determines the usage of the output signal channels; for the configuration of output signal channel 4, see “Using the Analog Output” (p. 46).</td>
</tr>
<tr>
<td>0x0A000004</td>
<td>INT</td>
<td>1 or 3</td>
<td>Output signal channel</td>
<td>Select Output Index</td>
<td>For output signal channel 4, this parameter determines the axis whose position is to be output or the output signal channel whose control value is to be output, see “Using the Analog Output” (p. 46).</td>
</tr>
<tr>
<td>0x0E000301</td>
<td>INT</td>
<td>1</td>
<td>System</td>
<td>Disable Error 10</td>
<td>This parameter can be used to avoid that error code 10 is set when axes are stopped with the STP, #24 or HLT commands. 0=OFF (Error code 10 is set.) 1=ON (Error code 10 is not set.)</td>
</tr>
<tr>
<td>0x0E000C01</td>
<td>INT</td>
<td>1</td>
<td>System</td>
<td>OverSampling Filter</td>
<td>Usage of this filter is recommended with SGS sensors and/or with servo sampling rates ≤ 10 kHz. Recommended parameter setting for SGS sensor processing: Cutoff frequency of the filter ≤ Servo sampling rate / 2 See response to HPV? for possible values.</td>
</tr>
</tbody>
</table>
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter ID (hexadecimal)</th>
<th>Data Type</th>
<th>Command Level for Write Access</th>
<th>Item Type</th>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0E001500</td>
<td>INT</td>
<td>1</td>
<td>System</td>
<td>Reboot on DIO Input</td>
<td>The value of the parameter enables/disables the Reset input on pin 2 of the MDR14 socket (details see p. 98).</td>
</tr>
<tr>
<td>0x0E001D00</td>
<td>INT</td>
<td>1</td>
<td>System</td>
<td>Trigger Input Filter Enable</td>
<td>The value of the parameter enables/disables a filter for the digital trigger input as follows: 0 = OFF: Filter is disabled. Processing of the trigger input is done immediately. 1 = ON: Filter is enabled to reduce the influence of noise at the trigger input line. Default setting.</td>
</tr>
<tr>
<td>0x0E001E00</td>
<td>INT</td>
<td>1</td>
<td>System</td>
<td>Discon. Target Man. In With Stop</td>
<td>Determines the behaviour when the analog input is used as a control source. See „Using the Analog Input“ (p. 45) for details.</td>
</tr>
<tr>
<td>0x13000202</td>
<td>INT</td>
<td>1</td>
<td>System</td>
<td>Wave Multi Start By Trigger</td>
<td>When the wave generator is to be started by an external trigger signal (WGO bit 1 is set), the value of this parameter determines if the trigger is enabled for only one generator start or for multiple starts. Details see p. 47.</td>
</tr>
<tr>
<td>0x16000700</td>
<td>INT</td>
<td>1</td>
<td>Data recorder table</td>
<td>DRC Data Source</td>
<td>See „Data Recording“ (p. 44).</td>
</tr>
<tr>
<td>0x16000701</td>
<td>INT</td>
<td>1</td>
<td>Data recorder table</td>
<td>DRC Record Option</td>
<td></td>
</tr>
<tr>
<td>0x18000201</td>
<td>FLOAT</td>
<td>1</td>
<td>Digital output</td>
<td>CTO Trigger Step</td>
<td>See „External Triggering/Signaling“ on p. 45) and in the E-725 user manual (PZ197E).</td>
</tr>
<tr>
<td>0x18000202</td>
<td>INT</td>
<td>1</td>
<td>Digital output</td>
<td>CTO Axis</td>
<td></td>
</tr>
<tr>
<td>0x18000203</td>
<td>INT</td>
<td>1</td>
<td>Digital output</td>
<td>CTO Trigger Mode</td>
<td></td>
</tr>
<tr>
<td>0x18000205</td>
<td>FLOAT</td>
<td>1</td>
<td>Digital output</td>
<td>CTO Min.Threshold</td>
<td></td>
</tr>
<tr>
<td>0x18000206</td>
<td>FLOAT</td>
<td>1</td>
<td>Digital output</td>
<td>CTO Max.Threshold</td>
<td></td>
</tr>
</tbody>
</table>
## Maintenance

### Updating Firmware

**INFORMATION**

The *IDN?* command reads the version number of the firmware among other things.

Example of a response of the E-727:

(c)2015 Physik Instrumente (PI) GmbH & Co. KG, E-727, 115009227, 13.12.00.00e

- 115009227: Serial number of the E-727
- 13.12.00.00e: Firmware version

To update the firmware of E-727, the USB, RS-232 or TCP/IP interfaces can be used with the PI Firmware Updater PC software.

A failed firmware update (e.g. due to power failure) is indicated as follows:

- **Ch1/2 OFL** is permanently blinking after start or reboot of the E-727 while all other LEDs are off: FPGA firmware does not work
- **Ch3/4 OFL** is permanently blinking after start or reboot of the E-727 while all other LEDs are off: DSP firmware does not work

In the case of a failed firmware update, you have to repeat the firmware update via the USB interface (other interfaces will not work).
Cleaning the E-727

**NOTICE**

Short circuits or flashovers!
The E-727 contains electrostatic sensitive devices that can be damaged by short circuits or flashovers when cleaning fluids enter the case.

- Before cleaning, remove the E-727 from the power source by pulling the power plug.
- Prevent cleaning fluid from entering the case.

1. Disconnect the E-727 from the power supply.
2. Wait a minute to be sure that any residual voltage has dissipated.
3. Clean the E-727 case surface with a cloth lightly dampened with a mild cleanser or disinfectant.

Changing the Fuse

**INFORMATION**
The E-727 is equipped with a cartridge fuse with the following characteristics:

T3.15 AH, 250 V, 5 x 20 mm

- Only replace the fuse with a fuse of the same type.

The fuse holder is located on the front panel of the case (p. 8). To access the fuse, proceed as follows:

1. Switch off the E-727 and disconnect the power supply from the E-727.
2. Wait a minute to be sure that any residual voltage has dissipated.
3. Unscrew the fuse holder counterclockwise using a suitable tool and remove it from the case.
4. Be sure to replace the fuse with fuse of the suitable type: T3.15 AH, 250 V, 5 x 20 mm
5. Insert the fuse holder in the case and screw it in clockwise using a suitable tool.
## Technical Data

### Specifications

<table>
<thead>
<tr>
<th>Preliminary Data</th>
<th>E-727.3CD</th>
<th>E-727.3CDA</th>
<th>E-727.3SD</th>
<th>E-727.3SDA</th>
<th>Unit</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function</strong></td>
<td>Digital controller for multi-axis piezo nanopositioning systems with capacitive sensors</td>
<td>Digital controller for multi-axis piezo nanopositioning systems with strain sensors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Axes</strong></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Processor</strong></td>
<td>DSP 32/64-bit, floating point, 375 MHz</td>
<td>DSP 32/64-bit, floating point, 375 MHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sampling rate, servo-control</strong></td>
<td>20 kHz</td>
<td>20 kHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sampling rate, sensor</strong></td>
<td>100 kHz</td>
<td>100 kHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sensor</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Servo characteristics</td>
<td>P-I, two notch filters</td>
<td>P-I, two notch filters</td>
<td>Optional: Advanced piezo control</td>
<td>Optional: Advanced piezo control</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sensor type</strong></td>
<td>Capacitive</td>
<td>Strain gauge sensors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sensor channels</strong></td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sensor bandwidth (-3 dB)</strong></td>
<td>10 kHz</td>
<td>10 kHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sensor resolution</strong></td>
<td>20 at 1 kSPS oversampling</td>
<td>20 at 1 kSPS oversampling</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ext. synchronization</strong></td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Amplifier</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output voltage</td>
<td>-30 to 130 V</td>
<td>-30 to 130 V</td>
<td></td>
<td></td>
<td>±3 V</td>
<td></td>
</tr>
<tr>
<td>Amplifier channels</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peak output power per channel, max. 30 ms</td>
<td>28 W</td>
<td>28 W</td>
<td></td>
<td></td>
<td></td>
<td>max.</td>
</tr>
<tr>
<td>Average output power per channel</td>
<td>14 W</td>
<td>14 W</td>
<td></td>
<td></td>
<td></td>
<td>max. 300 ms</td>
</tr>
<tr>
<td>Peak current per channel, max. 30 ms</td>
<td>200 mA</td>
<td>200 mA</td>
<td></td>
<td></td>
<td></td>
<td>max.</td>
</tr>
<tr>
<td>Average output current per channel</td>
<td>100 mA</td>
<td>100 mA</td>
<td></td>
<td></td>
<td></td>
<td>max.</td>
</tr>
<tr>
<td>Current limitation</td>
<td>Short-circuit-proof</td>
<td>Short-circuit-proof</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resolution DAC</td>
<td>20 bit</td>
<td>20 bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amplifier bandwidth</td>
<td>6.5 kHz</td>
<td>6.5 kHz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Interface and operation

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interface / communication</strong></td>
<td>Ethernet, USB, RS-232, serial SPI, high-speed interface</td>
<td>Ethernet, USB, RS-232, serial SPI, high-speed interface</td>
</tr>
<tr>
<td><strong>Piezo / sensor connection</strong></td>
<td>Sub-D Special</td>
<td>Sub-D 37</td>
</tr>
<tr>
<td><strong>Analog input</strong></td>
<td>Sub-D, 15-pin, 4 inputs, 18-bit</td>
<td>Sub-D, 15-pin, 4 inputs, 18-bit</td>
</tr>
<tr>
<td><strong>Analog outputs</strong></td>
<td>Universal output, 20-bit</td>
<td>Universal output, 20-bit</td>
</tr>
<tr>
<td><strong>Digital input/output</strong></td>
<td>MDR14; 4 inputs, 4 outputs</td>
<td>MDR14; 4 inputs, 4 outputs</td>
</tr>
<tr>
<td><strong>Command set</strong></td>
<td>PI General Command Set (GCS)</td>
<td>PI General Command Set (GCS)</td>
</tr>
<tr>
<td><strong>User software</strong></td>
<td>PIMikroMove</td>
<td>PIMikroMove</td>
</tr>
<tr>
<td><strong>Software drivers</strong></td>
<td>LabVIEW drivers, DLLs</td>
<td>LabVIEW drivers, DLLs</td>
</tr>
<tr>
<td><strong>Supported functionality</strong></td>
<td>Wave generator, data recorder, drift compensation, macros</td>
<td>Wave generator, data recorder, drift compensation, macros</td>
</tr>
<tr>
<td><strong>Display</strong></td>
<td>LEDs for Power, Servo, Error, Overflow</td>
<td>LEDs for Power, Servo, Error, Overflow</td>
</tr>
<tr>
<td><strong>Linearization</strong></td>
<td>4th order polynomials, DDL (Dynamic Digital Linearization)</td>
<td>4th order polynomials, DDL (Dynamic Digital Linearization)</td>
</tr>
<tr>
<td><strong>Separate protective earth connection</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

## Miscellaneous

| **Operating temperature range** | 5 to 40 °C |
| **Overtemp protection** | Max. 72 °C, deactivation of the piezo voltage output |
| **Mass** | 3 kg approx. |
| **Power consumption** | 80 W max. |
| **Power consumption without load** | 24 W max. |
| **Operating voltage** | 24 VDC from external power supply (included) | 24 VDC from external power supply (included) |
Maximum Ratings

The E-727 is designed for the following operating data:

<table>
<thead>
<tr>
<th>Input on:</th>
<th>Maximum Operating Voltage</th>
<th>Operating Frequency</th>
<th>Maximum Power Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>M8 panel plug</td>
<td>24 V</td>
<td>—</td>
<td>80 W</td>
</tr>
</tbody>
</table>

Ambient Conditions and Classifications

The following ambient conditions and classifications must be observed for the E-727:

<table>
<thead>
<tr>
<th>Area of application</th>
<th>For indoor use only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum altitude</td>
<td>2000 m</td>
</tr>
<tr>
<td>Relative humidity</td>
<td>Highest relative humidity 80% for temperatures up to 31°C  Decreasing linearly to 50% relative humidity at 40°C</td>
</tr>
<tr>
<td>Storage temperature</td>
<td>0°C to 70°C</td>
</tr>
<tr>
<td>Transport temperature</td>
<td>–25°C to +85°C</td>
</tr>
<tr>
<td>Overvoltage category</td>
<td>II</td>
</tr>
<tr>
<td>Protection class</td>
<td>I</td>
</tr>
<tr>
<td>Degree of pollution</td>
<td>2</td>
</tr>
<tr>
<td>Degree of protection according to IEC 60529</td>
<td>IP20</td>
</tr>
</tbody>
</table>
Dimensions

Dimensions in millimeters. Decimal places are separated by a comma in the drawings.
### Pin Assignment

**E-727.3CD, E-727.3CDA: Socket for Piezo Stages**

„-30 to +130 V“ - Sub-D special 25W3 (f)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coax inner lines:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A1</td>
<td>input</td>
<td>Sensor Probe Ch 2</td>
</tr>
<tr>
<td>A2</td>
<td>input</td>
<td>Sensor Probe Ch 3</td>
</tr>
<tr>
<td>A3</td>
<td>input</td>
<td>Sensor Probe Ch 1</td>
</tr>
<tr>
<td>Standard pins:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>output</td>
<td>Sensor Target Ch 2</td>
</tr>
<tr>
<td>2</td>
<td>output</td>
<td>Sensor Target Ch 3</td>
</tr>
<tr>
<td>3</td>
<td>GND</td>
<td>ID-Chip GND Ch 1</td>
</tr>
<tr>
<td>4</td>
<td>bidirectional</td>
<td>ID-Chip Ch 1</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
<td>ID-Chip GND Ch 2</td>
</tr>
<tr>
<td>6</td>
<td>bidirectional</td>
<td>ID-Chip Ch 2</td>
</tr>
<tr>
<td>7</td>
<td>output</td>
<td>Piezo Ch 4 +</td>
</tr>
<tr>
<td>8</td>
<td>output</td>
<td>Piezo Ch 3 +</td>
</tr>
<tr>
<td>9</td>
<td>output</td>
<td>Piezo Ch 2 +</td>
</tr>
<tr>
<td>10</td>
<td>output</td>
<td>Piezo Ch 1 +</td>
</tr>
<tr>
<td>11</td>
<td>output</td>
<td>Sensor Target Ch 1</td>
</tr>
<tr>
<td>12</td>
<td>GND</td>
<td>Sensor Target Ch 2 shield</td>
</tr>
<tr>
<td>13</td>
<td>GND</td>
<td>Sensor Target Ch 3 shield</td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td>nc</td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>nc</td>
</tr>
<tr>
<td>16</td>
<td>GND</td>
<td>ID-Chip GND Ch 3</td>
</tr>
<tr>
<td>17</td>
<td>bidirectional</td>
<td>ID-Chip Ch 3</td>
</tr>
<tr>
<td>18</td>
<td>output</td>
<td>Piezo Ch 4 -</td>
</tr>
<tr>
<td>19</td>
<td>output</td>
<td>Piezo Ch 3 -</td>
</tr>
<tr>
<td>20</td>
<td>output</td>
<td>Piezo Ch 2 -</td>
</tr>
<tr>
<td>21</td>
<td>output</td>
<td>Piezo Ch 1 -</td>
</tr>
<tr>
<td>22</td>
<td>GND</td>
<td>Sensor Target Ch 1 shield</td>
</tr>
</tbody>
</table>

With E-727.3CDA models:
The Piezo Ch 4 lines of the socket for piezo stages share output signal channel 4 of the E-727 with analog output 1 (pin 8 on the Analog I/O socket). The use of output signal channel 4 can be configured via the value of the **Select Output Type** parameter (ID 0x0A000003) as follows:

- **1**: Output voltage for a piezo actuator in the stage, output as Piezo Ch 4 (pins 7 and 18)
- **2**: Position monitor of an axis. The value of the **Select Output Index** parameter (ID 0x0A000004) determines the axis whose position is to be output. Output on pin 8 of the Analog I/O socket (p. 99).
- **5**: Control signal for an external motor driver. The value of the **Select Output Index** parameter (ID 0x0A000004) determines the output signal channel whose control value is to be output. Output on pin 8 of the Analog I/O socket.

If a total of four piezo actuators are present in the stage(s), output signal channel 4 must always be configured for use as output voltage (Piezo Ch 4). Note that PI will supply E-727 and the piezo stage(s) as a system with appropriate settings. If you are not sure whether your system can be configured for output of position monitor or control signal, contact our customer service department (p. 102).
E-727.3SD, E-727.3SDA: Socket for Piezo Stages

"-30 to +130 V" - Sub-D 37 (f)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PT1000+</td>
</tr>
<tr>
<td>20</td>
<td>PT1000-</td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
</tr>
<tr>
<td>21</td>
<td>ID-Chip Ch 1</td>
</tr>
<tr>
<td>3</td>
<td>ID-Chip Ch 2</td>
</tr>
<tr>
<td>22</td>
<td>ID-Chip-GND</td>
</tr>
<tr>
<td>4</td>
<td>ID-Chip Ch 3</td>
</tr>
<tr>
<td>23</td>
<td>ID-Chip Ch 4</td>
</tr>
<tr>
<td>5</td>
<td>ID-Chip-GND</td>
</tr>
<tr>
<td>24</td>
<td>SGS Ch 4 -</td>
</tr>
<tr>
<td>6</td>
<td>SGS Ch 4 +</td>
</tr>
<tr>
<td>25</td>
<td>SGS Ch 4 Ref</td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
</tr>
<tr>
<td>26</td>
<td>SGS Ch 3 -</td>
</tr>
<tr>
<td>8</td>
<td>SGS Ch 3 +</td>
</tr>
<tr>
<td>27</td>
<td>SGS Ch 3 Ref</td>
</tr>
<tr>
<td>9</td>
<td>GND</td>
</tr>
<tr>
<td>28</td>
<td>SGS Ch 2 -</td>
</tr>
<tr>
<td>10</td>
<td>SGS Ch 2 +</td>
</tr>
<tr>
<td>29</td>
<td>SGS Ch 2 Ref</td>
</tr>
<tr>
<td>11</td>
<td>GND</td>
</tr>
<tr>
<td>30</td>
<td>SGS Ch 1 -</td>
</tr>
<tr>
<td>12</td>
<td>SGS Ch 1 +</td>
</tr>
<tr>
<td>31</td>
<td>SGS Ch 1 Ref</td>
</tr>
<tr>
<td>13</td>
<td>GND</td>
</tr>
<tr>
<td>32</td>
<td>Reserved</td>
</tr>
<tr>
<td>14</td>
<td>Reserved</td>
</tr>
<tr>
<td>33</td>
<td>Reserved</td>
</tr>
<tr>
<td>15</td>
<td>Reserved</td>
</tr>
<tr>
<td>34</td>
<td>Piezo Ch 1 -</td>
</tr>
<tr>
<td>16</td>
<td>Piezo Ch 1 +</td>
</tr>
<tr>
<td>35</td>
<td>Piezo Ch 2 -</td>
</tr>
<tr>
<td>17</td>
<td>Piezo Ch 2 +</td>
</tr>
<tr>
<td>36</td>
<td>Piezo Ch 3 -</td>
</tr>
<tr>
<td>18</td>
<td>Piezo Ch 3 +</td>
</tr>
<tr>
<td>37</td>
<td>Piezo Ch 4 -</td>
</tr>
<tr>
<td>19</td>
<td>Piezo Ch 4 +</td>
</tr>
</tbody>
</table>

Either the lines for the PT1000 temperature sensor or the lines for the 4th SGS sensor can be used since they share input signal channel 4 of the E-727. The use of input signal channel 4 can be configured via the value of the Sensor Range Factor parameter (ID 0x02000100) as follows:

- 1: Use with an SGS sensor (pins 6, 24, 25)
- 2: Use with a PT1000 temperature sensor (pins 1, 20)

With E-727.3SDA models, the following values of the Sensor Range Factor parameter are supported for input signal channel 4 in addition, deactivating both the PT1000 and 4th SGS sensor:

- 3: Use as analog input 1 with a range of ±5 V (input via pins 2 and 9 on the Analog I/O socket (p. 99))
- 4: Use as analog input 1 with a range of ±10 V (input via pins 2 and 9 on the Analog I/O socket)

With E-727.3SDA models:
The Piezo Ch 4 lines of the socket for piezo stages share output signal channel 4 of the E-727 with analog output 1 (pin 8 on the Analog I/O socket). The use of output signal channel 4 can be configured via the value of the Select Output Type parameter (ID 0x0A000003) as follows:

- 1: Output voltage for a piezo actuator in the stage, output as Piezo Ch 4 (pins 19 and 37)
- 2: Position monitor of an axis. The value of the Select Output Index parameter (ID 0x0A000004) determines the axis whose position is to be output. Output on pin 8 of the Analog I/O socket (p. 99).
- 5: Control signal for an external motor driver. The value of the Select Output Index parameter (ID 0x0A000004) determines the output signal channel whose control value is to be output. Output on pin 8 of the Analog I/O socket.

If a total of four piezo actuators are present in the stage(s), output signal channel 4 must always be configured for use as output voltage (Piezo Ch 4). Note that PI will supply E-727 and the piezo stage(s) as a system with appropriate settings. If you are not sure whether your system can be configured for output of position monitor or control signal, contact our customer service department (p. 102).
Digital I/O

„Digital I/O“ - MDR14

<table>
<thead>
<tr>
<th>Function</th>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3 V out, internal resistance: 100 ohm</td>
<td>14</td>
<td>7 not connected</td>
</tr>
<tr>
<td>Digital In 1</td>
<td>13</td>
<td>6 not connected</td>
</tr>
<tr>
<td>Digital In 2</td>
<td>12</td>
<td>5 Digital Out 1</td>
</tr>
<tr>
<td>Digital In 3</td>
<td>11</td>
<td>4 Digital Out 2</td>
</tr>
<tr>
<td>not connected</td>
<td>10</td>
<td>3 Digital Out 3</td>
</tr>
<tr>
<td>not connected</td>
<td>9</td>
<td>2 Digital In 4 / Reset (active low)</td>
</tr>
<tr>
<td>Output of the servo cycles</td>
<td>8</td>
<td>1 GND</td>
</tr>
</tbody>
</table>

Digital inputs (pins 2, 11, 12, 13):
- TTL (low: 0 to 0.8 V, high: 2 to 5 V, max.: 5 V)
- When nothing is connected to a digital input, the signal level is high due to an internal pull-up with 10 kohm resistor.
- Digital In 4 (pin 2) can be configured as reset input using the Reboot On DIO Input parameter (ID 0x0e001500). Changes of the parameter value become effective immediately. The value of the parameter enables/disables the Reset input as follows:
  - 0 = OFF: Reset input is disabled (default setting)
  - 1 = ON: Reset input is enabled. If the signal level on the Reset input becomes low, the E-727 is rebooted (same behaviour as with the RBT command).

Digital outputs (pins 3, 4, 5, 8):
- High level:
  - at -2 mA output current => min. 2.2 V
  - at -0.1 mA output current => min. 3.0 V
- Low level:
  - at +2 mA output current => max. 0.6 V
  - at +0.1 mA output current => max. 0.21 V
- The servo cycle output on pin 8 is not accessible for commands.
E-727.3CDA, E-727.3SDA: Analog I/O

„Analog I/O“ – Sub-D 15 (f)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
</tr>
<tr>
<td>9</td>
<td>-Analog In 1</td>
</tr>
<tr>
<td>2</td>
<td>+Analog In 1</td>
</tr>
<tr>
<td>10</td>
<td>-Analog In 2</td>
</tr>
<tr>
<td>3</td>
<td>+Analog In 2</td>
</tr>
<tr>
<td>11</td>
<td>-Analog In 3</td>
</tr>
<tr>
<td>4</td>
<td>+Analog In 3</td>
</tr>
<tr>
<td>12</td>
<td>-Analog In 4</td>
</tr>
<tr>
<td>5</td>
<td>+Analog In 4</td>
</tr>
<tr>
<td>13</td>
<td>GND</td>
</tr>
<tr>
<td>6</td>
<td>GND</td>
</tr>
<tr>
<td>14</td>
<td>Sensor Monitor 1</td>
</tr>
<tr>
<td>7</td>
<td>Sensor Monitor 2</td>
</tr>
<tr>
<td>15</td>
<td>Sensor Monitor 3</td>
</tr>
<tr>
<td>8</td>
<td>Analog Out 1</td>
</tr>
</tbody>
</table>

Analog inputs (pins 9, 2, 10, 3, 11, 4, 12, 5):

- Input impedance: 150 kohm
- Max. input voltage (single ended): ±14 V
- Resolution DAC: 18 bit
- E-727.3CDA: Via the value of the Sensor Range Factor parameter (ID 0x02000100), the analog input lines 1 to 4 (accessible as input signal channels 4 to 7) can be configured as follows:
  - 1: Input range ±5 V
  - 2: Input range ±10 V
- E-727.3SDA: Via the value of the Sensor Range Factor parameter (ID 0x02000100), the analog input lines can be configured as follows:
  - Analog input line 1 (accessible as input signal channel 4):
    - 1: No input possible on pins 2 and 9 of Analog I/O; input signal channel 4 is used for an SGS sensor (input via pins 6, 24, 25 of the socket for the piezo stage(s) (p. 97)
    - 2: No input possible on pins 2 and 9 of Analog I/O; input signal channel 4 is used for a PT1000 temperature sensor (input via pins 1, 20 of the socket for the piezo stage(s))
    - 3: Input range ±5 V
    - 4: Input range ±10 V
  - Analog input lines 2 to 4 (accessible as input signal channels 5 to 7):
    - 1: Input range ±5 V
    - 2: Input range ±10 V
Analog outputs (pins 8, 14, 7, 15):

- Min. ±10 V at 5 mA output current
- Resolution ADC: 20 bit
- Sensor Monitor lines 1, 2, 3 (pins 14, 7, 15): The (raw) signals of the sensors 1, 2 and 3 which are fed into the E-727 on the socket for the piezo stage(s) (p. 96 or p. 97) are looped through to these lines. The Sensor Monitor lines are not available as output signal channels in the firmware of the E-727 and not accessible for commands.
- Analog Out 1 (pin 8; accessible as output signal channel 4) can be configured using the value of the Select Output Type parameter (ID 0x0A000003) as follows:
  - 1: Output has no meaning; output signal channel 4 is used as output voltage for a piezo actuator in the stage, output as Piezo Ch 4 on the socket for piezo stages (p. 96 or p. 97).
  - 2: Position monitor of an axis. The value of the Select Output Index parameter (ID 0x0A000004) determines the axis whose position is to be output.
  - 5: Control signal for an external motor driver. The value of the Select Output Index parameter (ID 0x0A000004) determines the output signal channel whose control value is to be output.

If a total of four piezo actuators are present in the stage(s), output signal channel 4 must always be configured for use as output voltage (Piezo Ch 4). Note that PI will supply E-727 and the piezo stage(s) as a system with appropriate settings. If you are not sure whether your system can be configured for output of position monitor or control signal, contact our customer service department (p. 102).

RS-232

„RS-232“ - Sub-D 9 (m)

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>nc</td>
</tr>
<tr>
<td>2</td>
<td>RXD receive data</td>
</tr>
<tr>
<td>3</td>
<td>TXD send data</td>
</tr>
<tr>
<td>4</td>
<td>nc</td>
</tr>
<tr>
<td>5</td>
<td>DGND ground</td>
</tr>
<tr>
<td>6</td>
<td>nc</td>
</tr>
<tr>
<td>7</td>
<td>RTS Hardware handshake, output</td>
</tr>
<tr>
<td>8</td>
<td>CTS Hardware handshake, input</td>
</tr>
<tr>
<td>9</td>
<td>nc</td>
</tr>
</tbody>
</table>
Power Supply 24 V

„24 VDC / 3.5A“ - Phoenix M8 panel plug, 4-pin, male

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND (power)</td>
</tr>
<tr>
<td>2</td>
<td>GND (power)</td>
</tr>
<tr>
<td>3</td>
<td>Input: 24 V DC</td>
</tr>
<tr>
<td>4</td>
<td>Input: 24 V DC</td>
</tr>
</tbody>
</table>

K050B0003 Adapter, Barrel Connector Side

Center: +24 V

Outer conductor: GND
EC Declaration of Conformity

For the E-727, an EC Declaration of Conformity has been issued in accordance with the following European directives:

2006/95/EC, Low Voltage Directive (LVD)
2004/108/EC, EMC Directive
2011/65/EU, RoHS Directive

The applied standards certifying the conformity are listed below.

<table>
<thead>
<tr>
<th>Category</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safety (LVD)</td>
<td>EN 61010-1:2010</td>
</tr>
<tr>
<td>EMC</td>
<td>EN 61326-1:2013</td>
</tr>
<tr>
<td>RoHS</td>
<td>EN 50581:2012</td>
</tr>
</tbody>
</table>

Customer Service

For inquiries and orders, contact your PI sales engineer or send us an e-mail (info@pi.ws).

➢ If you have questions concerning your system, have the following information ready:
  − Product codes and serial numbers of all products in the system
  − Firmware version of the controller (if present)
  − Version of the driver or the software (if present)
  − Operating system on the PC (if present)

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

The latest versions of the user manuals are available for download (p. 5) on our website.