User Manual
E727T0005, valid for E-727
BRO, 2019-06-28

E-727 Digital Multi-Channel Piezo Controller

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

Contents

Original instructions
First printing: 2019-06-28
Document number: E727T0005, BRo

Subject to change without notice. This manual is superseded by any new release. The latest respective release is available for download on our website (p. 10).

About this Document

This user manual 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 user manuals and Technical Notes are available for download on our website (p. 10).

Symbols and Typographic Conventions

The following symbols and markings are used in this user manual:

CAUTION

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

➢ Actions to take to avoid the situation.

NOTICE

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

➢ Actions to take to avoid the situation.

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

Other Applicable Documents

Additional documentation for the E-727:

Description	Document
PI General Command Set (GCS)	PZ281E GCS Commands Manual for E-727

Documentation for the available PC software:

Description	Document
GCS Data	SM146E Software Manual
E-727 GCS Driver Set for use with NI LabVIEW software	PZ260E Software Manual
Analog Controller GCS Driver Set for use with NI LabVIEW software	PZ181E Software Manual
Merge Tool for use with driver sets for NI LabVIEW software	SM154E Software Manual
PI GCS 2 DLL	SM151E Software Manual
PI MATLAB Driver GCS 2.0	SM155E Software Manual
PIMikroMove	SM148E Software Manual
Updating PI Software	A000T0032 Technical Note
PI Update Finder	A000T0028 Technical Note

Basic information on EtherCAT networks and the CiA402 drive profile:

Description	Document
Adjustable speed electrical power drive systems - Part 7-201: Generic interface and use of profiles for power drive systems - Profile type 1 specification	IEC 61800-7-201:2015
Adjustable speed electrical power drive systems - Part 7-301: Generic interface and use of profiles for power drive systems - Mapping of profile type 1 to network technologies	IEC 61800-7-301:2015
EtherCAT Implementation Directive for CiA402 Drive Profile: Directive for using IEC 61800-7-201 within EtherCAT-based servo drives	ETG.6010 D (R) V1.1.0

Documentation of EtherCAT samples on the E-727 CD:

Description	Document
Implementing a PI Controller in TwinCAT 3.1 for a Single-Axis Motion	A000T0071 Technical Note

Downloading Manuals

INFORMATION

If a manual is missing or problems occur with downloading:

➢ Contact our customer service department (p. 234).

INFORMATION

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

You need the CD of the product to obtain the access data.

For products with CD: Obtain access data

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 access data for the download of protected contents in the section "User login for software download" in the Release News. Possibilities for the provision:
   - Link to a registration page for request of access data
   - Direct display of user name and password
5. If the access data have to be requested via a registration page:
   a) Follow the link in the Release News.
   b) Enter the information required for registration.
   c) Click ***Show login data***.
   d) Find the user name and the password in the browser window.

Downloading Manuals

If you have requested via registration page the access data for protected contents (see above):

➢ Click the links on the registration page to move to the contents for your product, and log in there with the access data.

General procedure:

1. Open the website **www.pi.ws**.
2. If access to the manuals is protected by a password:
   a) Click ***Login***.
   b) Log in with the user name and password.
3. Click ***Search***.
4. Enter the product code up to the period (“E-727”) into the search field.
5. Click ***Start search*** or press the Enter key.
6. Open the corresponding product detail page in the list of search results:
   a) If necessary: Scroll down the list.

   b) If necessary: Click ***Load more results*** at the end of the list.
   c) Click the corresponding product in the list.
7. Click the ***Downloads*** tab.

   The manuals are displayed under ***Documentation***.
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 user manual. 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 or piezoresistive sensors (semiconductor 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 (user manuals, Technical Notes).
- 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. 48).
- 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 (user manual, Technical Notes):

- 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

Model	Name
E-727.3CD	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, capacitive sensors, Sub-D 25W3 socket
E-727.3CDA	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, capacitive sensors, Sub-D 25W3 socket, analog inputs
E-727.3CDP	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, capacitive sensors, Sub-D 25W3 socket, 1,5 A peak output current
E-727.3CDAP	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, capacitive sensors, Sub-D 25W3 socket, 1,5 A peak output current, analog inputs
E-727.3CDF	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, capacitive sensors, Sub-D 25W3 socket, EtherCAT interface
E-727.3CDAF	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, capacitive sensors, Sub-D 25W3 socket, EtherCAT interface, analog inputs
E-727.3RD	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, piezoresistive sensors, Sub-D 37 socket
E-727.3RDA	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, piezoresistive sensors, Sub-D 37 socket, analog inputs
E-727.3RDP	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, piezoresistive sensors, Sub-D 37 socket, 1,5 A peak output current
E-727.3RDAP	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, piezoresistive sensors, Sub-D 37 socket, 1,5 A peak output current, analog inputs
E-727.3RDF	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, piezoresistive sensors, Sub-D 37 socket, EtherCAT interface
E-727.3RDAF	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, piezoresistive sensors, Sub-D 37 socket, EtherCAT interface, analog inputs
E-727.4RD	Digital multi-channel piezo controller, 4 axes, -30 to 130 V, piezoresistive sensors, Sub-D 37 socket
E-727.3SD	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, strain gauge sensors, Sub-D 37 socket
E-727.3SDA	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, strain gauge sensors, Sub-D 37 socket, analog inputs
E-727.3SDP	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, strain gauge sensors, Sub-D 37 socket, 1,5 A peak output current
E-727.3SDAP	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, strain gauge sensors, Sub-D 37 socket, 1,5 A peak output current, analog inputs
E-727.3SDF	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, strain gauge sensors, Sub-D 37 socket, EtherCAT interface
E-727.3SDAF	Digital multi-channel piezo controller, 3 axes, -30 to 130 V, strain gauge sensors, Sub-D 37 socket, EtherCAT interface, analog inputs
E-727.4SD	Digital multi-channel piezo controller, 4 axes, -30 to 130 V, strain gauge sensors, Sub-D 37 socket

Product View

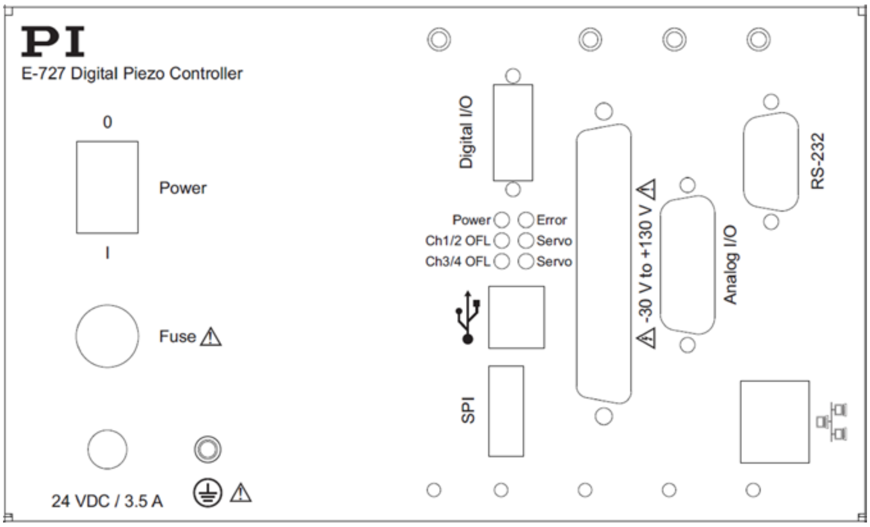


Figure 1: Front panel of E-727 digital piezo controllers, models E-727.xxx, .xxxA

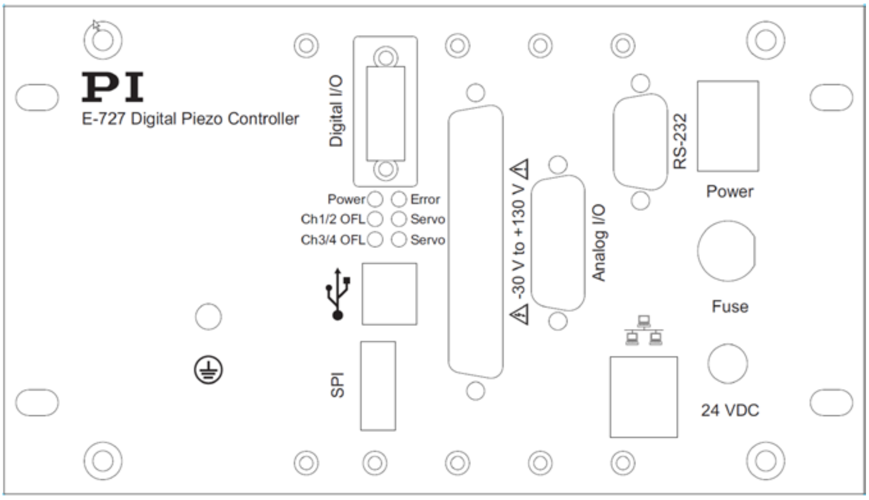


Figure 2: Front panel of E-727 digital piezo controllers, models E-727.xxxP, E-727.xxxAP

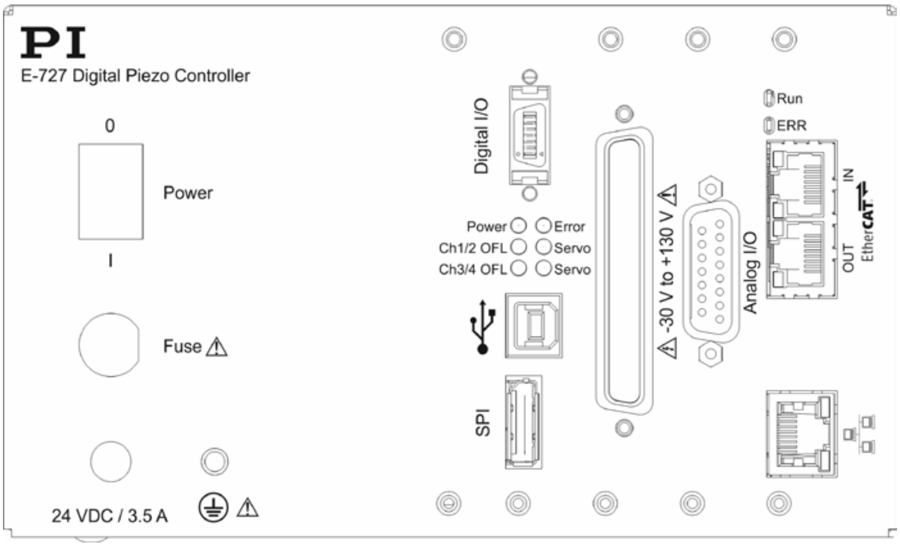


Figure 3: Front panel of E-727 digital piezo controllers, models E-727.xxxF, E-727.xxxAF

Labeling	Type	Function
Power	Toggle switch 	Power on/off switch: ▪ O position: E-727 is switched off ▪ | position: E-727 is switched on
Fuse ⚠	Slotted fuse holder 	For cartridge fuse 5 x 20 mm, changing the fuse see p. 212
24 VDC	M8 panel plug, 4-pin (p. 232)	Connection for the supply voltage. To be used with the K050B0003 adapter (included in the scope of delivery, p. 19)
⏚ ⚠	M4 hole with fastening material for protective earth conductor	Protective earth connection (p. 48) 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.
Digital I/O	MDR14 (f) (p. 227)	Digital lines: ▪ Outputs: Triggering of external devices, output of the servo cycles ▪ Inputs: Triggering of data recorder or wave generator, use in macros, reboot of E-727 (p. 227)

Labeling	Type	Function
Power	LED green	Power-on and ready indicator: ▪ Continuously lit: E-727 is ready for normal operation ▪ Continuously off: E-727 is not connected to the supply voltage ▪ Alternately lit/off/lit: E-727 performs power-on or reboot sequence
Error	LED red	Error indicator: ▪ Continuously lit: Error (error code ≠ 0) ▪ Continuously off: No error (error code = 0) ▪ Alternately lit/off: E-727 performs power-on or reboot sequence 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**	LED yellow	Overflow indicator for the axes: ▪ Continuously lit: At least one of the axes is in overflow state ▪ Continuously off: No axis is in overflow state ▪ Both **OFL** LEDs are flashing together with the **Servo** LEDs: E-727 searches for a DHCP server during power-on or reboot sequence ▪ At least one **OFL** LED is permanently flashing while all other LEDs are off: Firmware update failed (details see “Updating Firmware”, p. 204) 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) 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. 62). For an axis in overflow state, the corresponding bit in the response to the #5 command (Request Motion Status) is **not** set (motion state = “not moving”).

Labeling	Type		Function
Ch1/2 Servo **Ch3/4 Servo**	LED green		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		Connection to SPI (serial peripheral interface) master. Can be used for transferring position data from and to the E-727 with minimum latency and update rates as high as the servo update rate of the E-727. It is also possible to send and receive ASCII data so that the connected master has full access to the PI General Command Set (GCS). See "SPI-Interface" (p. 144) for details.
-30 V to +130 V	E-727.xRxxx*, .xSxxx*: D-Sub 37 (f) (p. 226) 	E-727.xCxxx*: D-Sub special 25W3 (f) (p. 225) 	Socket for piezo stages; carries the voltage for the piezo actuators (-30 to 130 V) and the signals of the sensors in the mechanics.
Analog I/O	E-727.xxxA*, E-727.xxxAx*: D-Sub 15 (f) (p. 228) 	E-727.xxx*, E-727.xxxF*: - 	Only E-727.xxxA and .xxxAx - analog lines: ▪ Inputs: Used for external sensors or as analog control inputs ▪ Outputs: Three sensor monitor lines, and one line that can be used to monitor the position of an axis or for controlling an external amplifier

Labeling	Type	Function
RS-232	D-Sub 9 (m) (p. 230)	Serial connection to PC via UART, voltage level RS-232 If the E-727 is equipped with an EtherCAT interface, the RS-232 connection is not present.
IN **OUT**	RJ45 socket with green and yellow LED 	E-727.xxxF* and .xxxAF* only: IN (on top): Connection for EtherCAT master OUT (below): Connection for the next EtherCAT slave Green LED: ▪ On: The E-727 is linked via EtherCAT, but does not send/receive frames. ▪ Flickering (load-dependent): The E-727 is linked via EtherCAT and sends/receives frames. ▪ Off: The E-727 is not linked via EtherCAT. Yellow LED: Not used.
RUN	LED green	E-727.xxxF* and .xxxAF* only: EtherCAT communication state of the E-727: ▪ Off: E-727 is in INIT state. ▪ Blinking (2.5 Hz): E-727 is in PRE-OPERATIONAL state ▪ Single flash: E-727 is in SAFE-OPERATIONAL state ▪ On: E-727 is in OPERATIONAL state
ERR	LED red	E-727.xxxF* and .xxxAF* only: EtherCAT communication state of the E-727: ▪ Off: No error, EtherCAT communication is running ▪ Blinking (2.5 Hz): Invalid configuration. General configuration error. Possible reason: State change commanded by master is impossible due to register or object settings. ▪ Single flash: Local error. E-727 has changed the EtherCAT state autonomously. Possible reason 1: A host watchdog timeout has occurred. Possible reason 2: Synchronization error, E-727 changes to SAFE-OPERATIONAL automatically. ▪ Double flash: An application watchdog timeout has occurred. Possible reason: Sync Manager Watchdog timeout.
	RJ45 socket	Ethernet interface for communication via TCP/IP, see p. 56 for details

* x is a placeholder; for the individual models see "Model Overview" (p. 13).

Scope of Delivery

Item number	Description
E-727	Digital piezo controller according to the order
000023194	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
3763	Power cord
K050B0003	Adapter for the power supply connection; barrel connector to M8 4-pin connector
C-815.34	RS-232 null-modem cable, 3 m, 9/9-pin Not for E-727.xxxF and .xxxAF models.
C-815.563	Cross-over network cable for direct connection with the PC via TCP/IP
000011448	USB cable (type A to type B) for connection to the PC
E-727.CD	Product CD with software and user manuals for the E-727
E727T0005	User Manual for the E-727, this document
Models with additional analog interfaces only (E-727.xxxA, E-727.xxxAx):	
E-727.IO3x	Analog input cable, D-Sub 15 (m) to open end, 1 m. For further details, see p. 230.

Accessories

Order number	Description
P-895.4LDS	Adapter cable for four channels, Sub-D 37 (m) to 8 x Lemo (4 x piezo, 4 x sensor); 0.3 m. For piezo stages with piezoresistive or strain gauge sensors. PI has to adapt the cable to the piezo stage with which the cable is to be used. Therefore the cable and the piezo stage have to be ordered together.
E-727.IO8	Adapter cable for the **Analog I/O** socket of E-727.xxxA and .xxxAx models, D-Sub 15 (m) to 8 x BNC, 0.4 m. For further details, see p. 231.
E-710.SCN	Dynamic Digital Linearization (DDL) option. The DDL makes it possible to achieve significantly better position accuracy for dynamic applications with periodic motion. It is used in conjunction with the wave generator output and in addition to the "normal" servo algorithm in closed-loop operation. You can activate the DDL after purchase and without opening the device. See "Dynamic Digital Linearization (DDL)" (p. 114) for more information.
E-712.U1	Advanced Piezo Control option. Advanced Piezo Control (APC) is an alternative control algorithm for closed-loop operation of piezo actuator systems. You can activate the license after purchase and without opening the device. See the separate E712T0007 Technical Note for more information.

To order, contact our customer service department (p. 234).

Functional Principles

Block Diagrams

The following block diagrams show the basic structure of the E-727.

For clear readability, the block diagrams do not contain the additional analog inputs and outputs which are present with E-727.xxxA and E-727.xxxAx models.

Block diagram for models without EtherCAT interface:

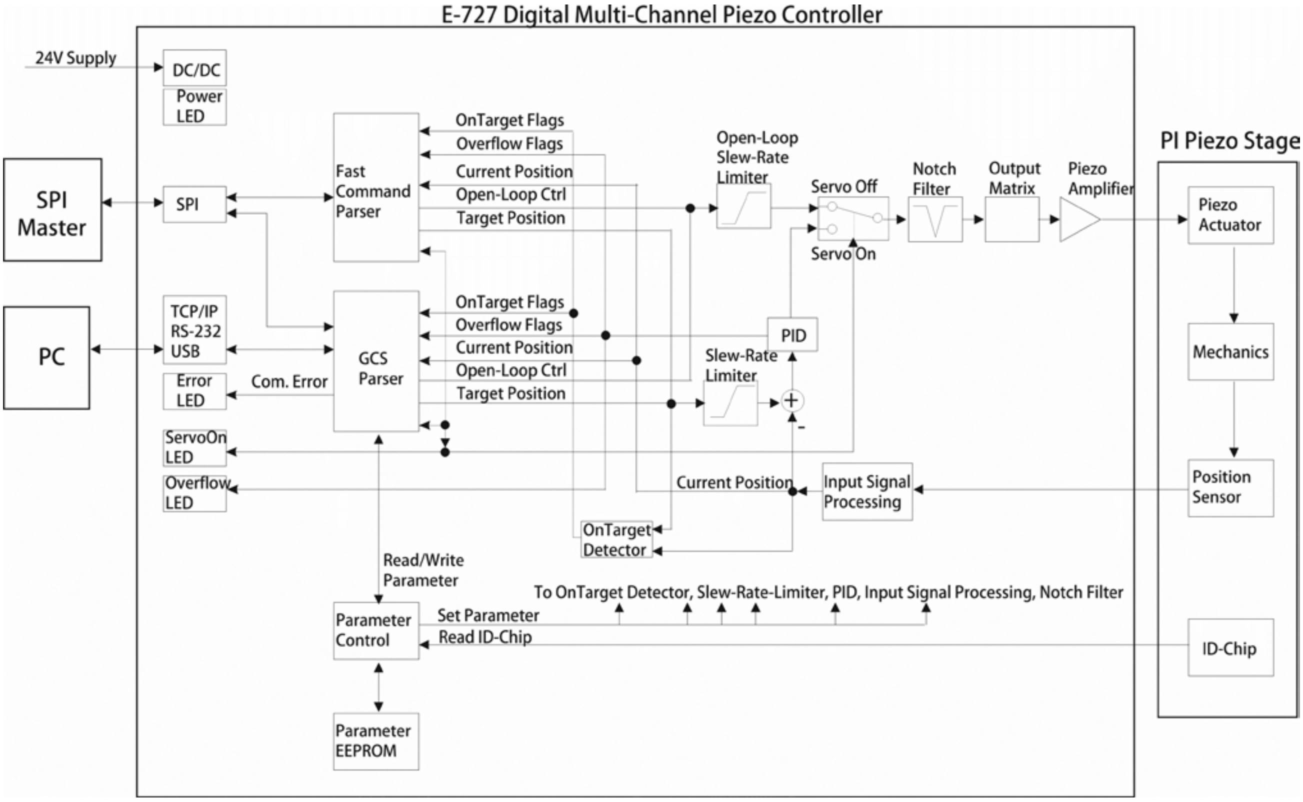


Figure 4: Block diagram of E-727 models without EtherCAT interface

Block diagram for models with EtherCAT interface:

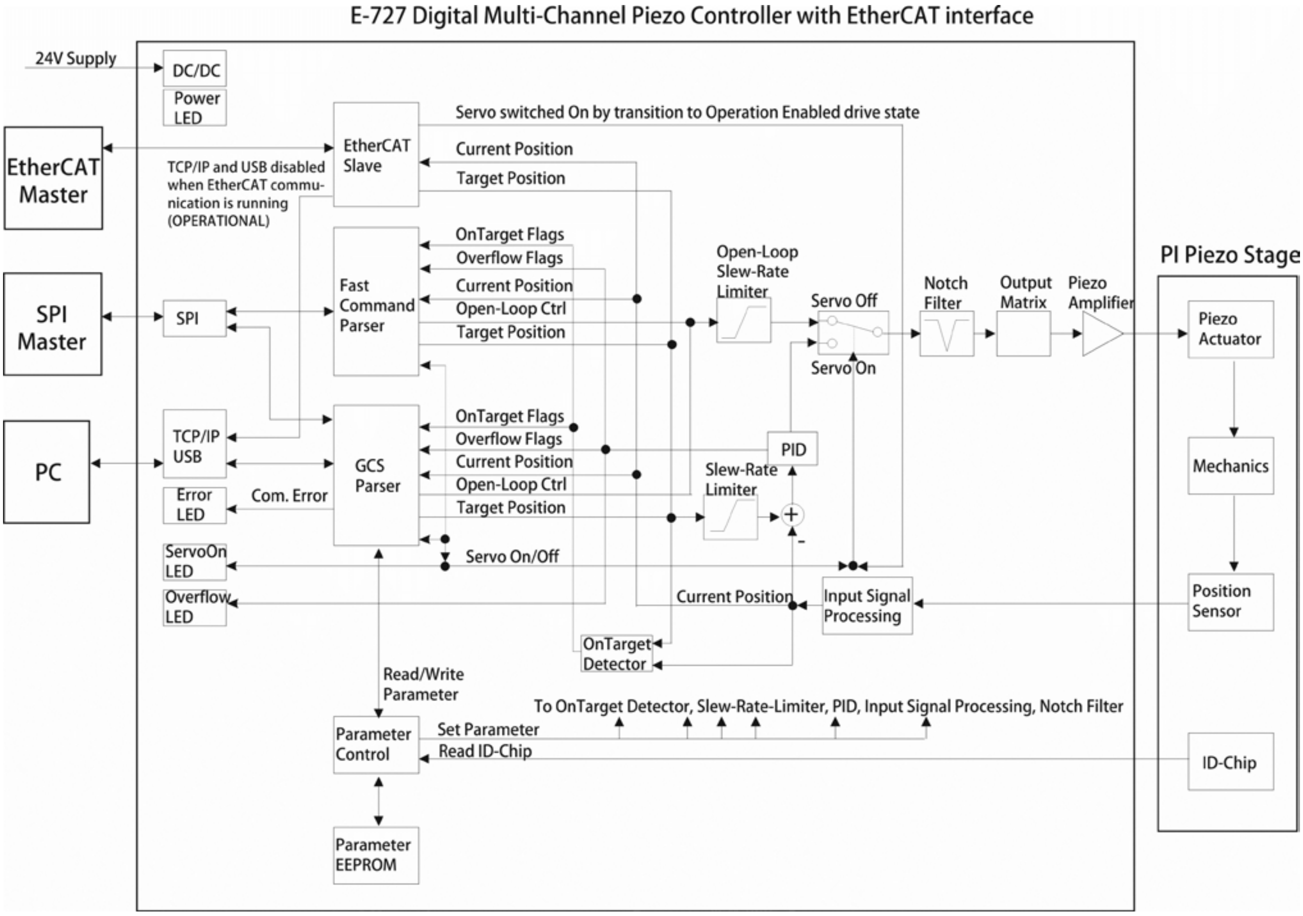


Figure 5: Block diagram of E-727 models with EtherCAT interface

The E-727 controls the motion of the logical axes of the connected stage(s) in open-loop or closed-loop operation. The block diagram below shows the signal path for an axis in closed-loop operation.

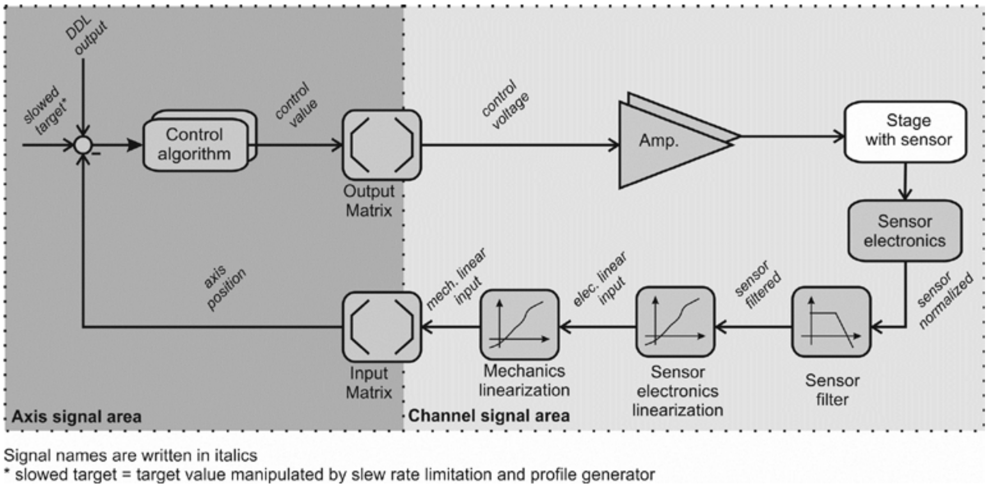


Figure 6: Block diagram of an axis in closed-loop operation

Axes, Channels, Functional Elements

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

When controlled by an EtherCAT master (E-727.xxxF and .xxxAF only), the EtherCAT master specifies target positions for the logical axes of the E-727. For further information, see "EtherCAT Interface" (S. 164).

Item	Num-ber	Identi-fier	Description
Logical axis	3 or 4	1 to 3 or 1 to 4	The supported number of logical axes depends on the E-727 model. 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. The value of the ***Number Of System Axes*** 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).

<table>
<tr><th>Item</th><th>Number</th><th>Identifier</th><th>Description</th></tr>
<tr><td>Input signal channels</td><td>Number depends on the E-727 model</td><td>1 to max. number of channels</td><td>E-727.3CD, .3CDF, .3CDP:
3 channels, intended for the capacitive sensors in the stage(s), input on the socket for piezo stages (p. 225).

E-727.3CDA, .3CDAF, .3CDAP:
7 channels. Channels 1 to 3 are intended for the capacitive sensors in the stages(s), input on the socket for piezo stages (p. 225). Channels 4 to 7 are the analog inputs available on the Analog I/O socket (p. 228). 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, .3SDF, .3SDP. .4SD, .3RD, .3RDF, .3RDP, .4RD:
4 channels. Input on the socket for piezo stages (p. 226). Channels 1 to 3 are intended for the piezoresistive or strain gauge 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 a piezoresistive or strain gauge sensor (pins 6, 24, 25)
▪ 2: Use with a PT1000 temperature sensor (pins 1, 20)

E-727.3SDA, .3SDAF, .3SDAP, .3RDA, .3RDAF, .3RDAP:
7 channels. Channels 1 to 3 are intended for the piezoresistive or strain gauge sensors in the stages(s), input on the socket for piezo stages (p. 226). Via the value of the Sensor Range Factor parameter (ID 0x02000100), channel 4 can be configured as follows:
▪ 1: Use with a piezoresistive or strain gauge sensor (input via pins 6, 24, 25 of the socket for the piezo stage(s) (p. 226))
▪ 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. 228))
▪ 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:
▪ 1: ±5 V
▪ 2: ±10 V

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” (p. 85) for</td></tr>
</table>

Item	Num-ber	Identi-fier	Description
			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-727.3RDA, .3RDAF, .3RDAP, .3SDA, .3SDAF and .3SDAP 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	4	1 to 4	**E-727.xxx, .xxxF, .xxxP:** The channels are intended for the piezo actuators in the stage(s), output on the socket for piezo stages (p. 225 or p. 226). **E-727.xxxA, .xxxAx:** Channels 1 to 3 are intended for the piezo actuators in the stage(s), output on the socket for piezo stages (p. 225 or p. 226). 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. 225 or p. 226). ▪ 2: Position monitor of an axis, output on pin 8 of the **Analog I/O** socket (p. 228). ▪ 5: Control signal for an external amplifier, output on pin 8 of the **Analog I/O** socket. Further details see "Using the Analog Output" (p. 94). **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.xxxA and .xxxAx 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	4	1 to 4	1 to 4 identify digital input lines 1 to 4 of the **Digital I/O** socket (p. 227). Digital inputs 1 to 3 can be used to trigger the data recorder (DRT command) or wave generator output (WGO command). Furthermore, they can be used in macros (MAC command). Via the ***Reboot On DIO Input*** parameter (ID 0x0E001500), the digital input 4 (pin 2 of **Digital I/O**) 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.

Item	Num-ber	Identi-fier	Description
Digital outputs	3	1 to 3	1 to 3 identify digital output lines 1 to 3 of the **Digital I/O** socket (p. 227). 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 0x0E000B05).
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 4^{th} axis, a 4^{th} 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 4^{th} axis, a 4^{th} 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.
Firmware units	2	1, 2	The number of different firmware units present in the E-727.

Important Components of the Firmware

The firmware of the E-727 provides the following functional units:

Firmware Component	Description
Commands	Communication with the E-727 can be managed using the commands of the PI General Command Set (GCS; version 2.0). The GCS is independent of the hardware (controller, stages connected). See the GCS commands manual PZ281E for more information. You can find a list of available GCS commands in the "Command Overview" section (p. 181). The E-727 can also be controlled by an SPI master, see “SPI Interface” (p. 144). E-727 models with EtherCAT interface can also be controlled by an EtherCAT master, see “EtherCAT Interface” (p. 164).
Parameters and command levels	Parameters reflect the properties of the E-727 and the connected stage and define the behavior of the system (e.g. settings for the control algorithm and notch filters (p. 35)). The parameters can be divided into the following categories: ▪ Protected parameters whose default settings cannot be changed ▪ Parameters that can be set by the user to adapt to the application The write permission for the parameters is determined by command levels. The current command level can be changed with the `CCL` command. This may require entering a password. For more information, see "Parameters" (p. 185). The values of some parameters are stored on the ID chip (p. 39) of the stage. They are loaded to the volatile and nonvolatile memory when the E-727 is switched on or rebooted.
Control algorithm for closed-loop operation	For better position accuracy and performance, the E-727 can be operated in closed-loop mode. A control algorithm (with sensor feedback) will then apply corrections to the control value. For more information, see “Control Details” (p. 35).
Wave generator	Each axis can be controlled by a wave generator that outputs waveforms. The wave generator is especially suited for dynamic applications in which periodic motions of the axis are executed (p. 99).
Data recorder	The E-727 contains a real-time data recorder (p. 65). This can record different input and output signals (e. g. current position, sensor input, output current) from different data sources (e. g.logical axes, input and output signal channels).
Macros	The E-727 can save macros—command sequences can be defined and permanently stored in the nonvolatile memory of the device via the macro function. A start-up macro can be defined that is executed each time when the E-727 is switched on or rebooted. This simplifies stand-alone operation (operation without a connection to the PC). Further information can be found in the "Controller Macros" section (p. 121).

Input Signal Processing

The following processing is applied to all input signal channels of the E-727 (channels for internal sensors of the mechanics and for analog input lines):

- Analog to digital conversion
- Digital processing (filtering and linearization / scaling)
- Allocation of input signal channels to axes via the input matrix to calculate the axis positions from the input signals

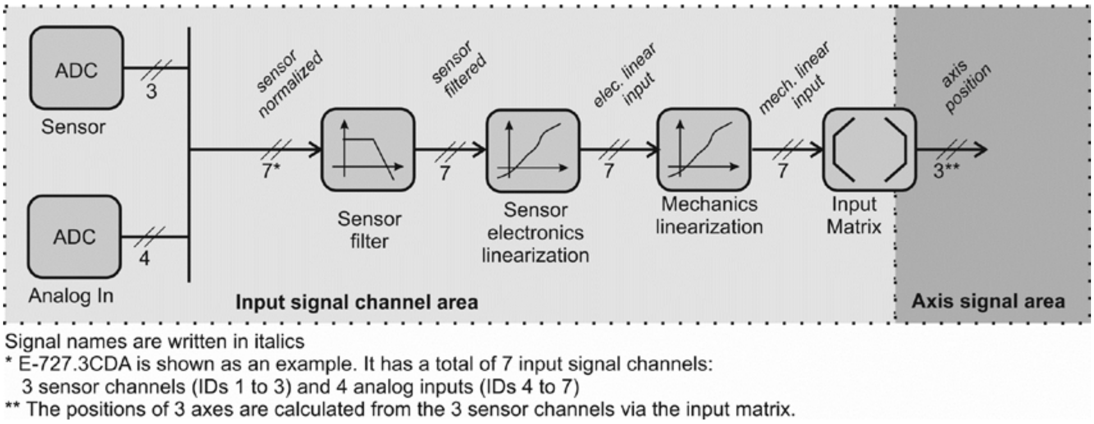


Figure 7: Input signal processing for E-727.3CDA

Analog to digital conversion

The results of the analog to digital conversion can be queried with the TAD? command for all channels.

Digital processing

The digital processing of the input signals comprises the following steps:

- Digital filtering
- Electronics linearization
- Mechanics linearization

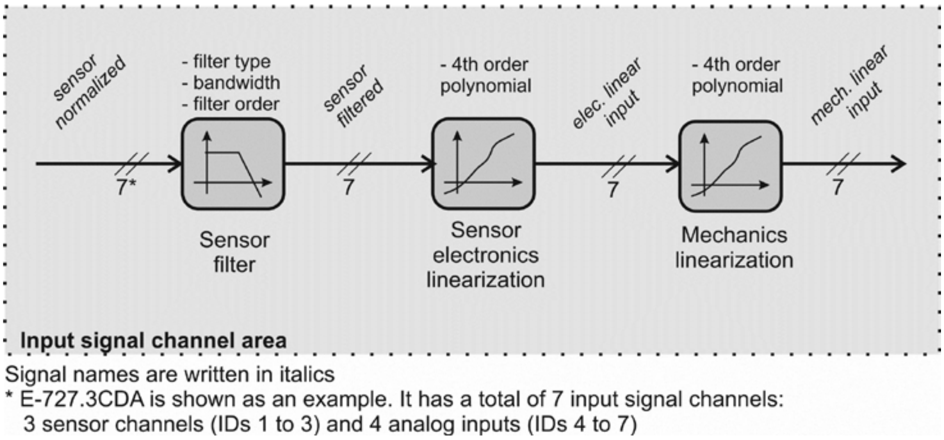


Figure 8: Digital processing of the input signals, shown for E-727.3CDA

The following parameters determine the digital filter settings:

- ***Digital Filter Type***
  parameter ID 0x05000000
  0 = no filter
  1 = IIR low-pass filter, 2nd order
  99 = USER filter (coefficients of the filter are given by ***User Filter Param. 1*** to ***5*** (IDs 0x05000101 to 0x05000105)

- ***Digital Filter Bandwidth***
  parameter ID 0x05000001
  Gives the cut-off-frequency f_g of the IIR low-pass filter. Only used if "Digital Filter Type" is set to "IIR low-pass filter, 2nd order".
  Note that the duration of the signal processing for a sensor results from two portions:
  1) Duration of the analog sensor processing, which takes about 100 µs
  2) Duration of the digital filtering which depends on the f_g setting: for signal frequencies
  $f < f_g/2$, the duration of the filtering can be estimated as follows: $t \approx 0.216 / f_g$

In PIMikroMove, the digital filter parameters are available in the *Sensor Mechanics* parameter groups in the ***Device Parameter Configuration*** window.

Polynomial linearization is used to correct system performance. The basic form of the polynomials is as follows:

$$y = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + a_4 \cdot x^4 + a_5 \cdot x^5$$

x	–	*filtered sensor ADC value*
y	–	*linearized sensor value*

To make the system components easily replaceable, sensor (i.e. mechanics) and electronics use separate polynomials. The coefficients of the polynomials are determined at the factory. Some terms of the polynomials are provided for future application and presently set to zero. The following terms are currently in use:

- Electronics linearization: offset, gain, 2nd, 3rd and 4th order correction. The corresponding coefficients of the polynomial are given by the parameters 0x03000100 to 0x03000500. They are independent of the connected mechanics and may not be changed by the user. In PIMikroMove, these parameters are available in the *Sensor Electronics* parameter groups in the ***Device Parameter Configuration*** window.

- Mechanics linearization: offset, gain, 2nd, 3rd and 4th order correction. The corresponding coefficients of the polynomial are given by the parameters 0x02000200 to 0x02000600. They depend on the connected mechanics. In PIMikroMove, these parameters are available in the *Sensor Mechanics* parameter groups in the ***Device Parameter Configuration*** window.
  For the internal sensors in the mechanics, the parameters should not be changed by the user. For analog input lines (E-727.xxxA, .xxxAx only; see p. 22 for details), changing the offset and gain values is required to scale the analog input to suitable

position values (see "Using the Analog Input" (p. 85) for more information and examples).
If the connected mechanics has an ID-chip, the coefficients will be read in from the ID-chip (see "ID-Chip Support / Stage Replacement" (p. 131) for more information).

The TNS? command reports the result after the linearization for the electronics (normalized value, dimensionless), while the TSP? command reports the result after the linearization for the mechanics (scaled value, the unit is µm).

Allocation of input signal channels to axes

Multiple sensors can be used to monitor the position of an axis, especially with rotation axes. The internal sensors in the mechanics are active by default, while additional, external sensors can optionally be connected to the analog input lines of E-727.xxxA or .xxxAx models. The axis positions are calculated from the position values of the input signal channels using the input matrix. The number of rows and columns of the input matrix depends on the E-727 model. The example below shows the matrix for an E-727.3CDA model which controls 3 axes and has 3 channels for internal sensors and 4 additional analog input lines (the last four input signal channels):

$$\begin{pmatrix} Axis1 \\ Axis2 \\ Axis3 \end{pmatrix} = \begin{pmatrix} a11 & a12 & a13 & a14 & a15 & a16 & a17 \\ a21 & a22 & a23 & a24 & a25 & a26 & a27 \\ a31 & a32 & a33 & a34 & a35 & a36 & a37 \end{pmatrix} * \begin{pmatrix} InputCh1 \\ InputCh2 \\ InputCh3 \\ InputCh4 \\ InputCh5 \\ InputCh6 \\ InputCh7 \end{pmatrix}$$

In equation form:

$Axis_1 = a_{11}InputCh_1 + a_{12}InputCh_2 + a_{13}InputCh_3 + a_{14}InputCh_4 + a_{15}InputCh_5 + a_{16}InputCh_6 + a_{17}InputCh_7$

$Axis_2 = a_{21}InputCh_1 + a_{22}InputCh_2 + a_{23}InputCh_3 + a_{24}InputCh_4 + a_{25}InputCh_5 + a_{26}InputCh_6 + a_{27}InputCh_7$

$Axis_3 = a_{31}InputCh_1 + a_{32}InputCh_2 + a_{33}InputCh_3 + a_{34}InputCh_4 + a_{35}InputCh_5 + a_{36}InputCh_6 + a_{37}InputCh_7$

The matrix coefficients are given by the ***Position From Sensor n*** parameters (*n* depends on the number of input signal channels present in the E-727). With an E-727.3CDA, for example, the following parameters are used (with i = 1 to 3 for the three axes of the system, i.e. each parameter has a different value for each of the logical axes):

- a_{i1} = ***Position From Sensor 1***, parameter ID 0x07000500
  a_{i2} = ***Position From Sensor 2***, parameter ID 0x07000501
  a_{i3} = ***Position From Sensor 3***, parameter ID 0x07000502
  These coefficients are for the internal sensors in the mechanics (socket for piezo stages).
- a_{i4} = ***Position From Sensor 4***, parameter ID 0x07000503
  a_{i5} = ***Position From Sensor 5***, parameter ID 0x07000504
  a_{i6} = ***Position From Sensor 6***, parameter ID 0x07000505
  a_{i7} = ***Position From Sensor 7***, parameter ID 0x07000506
  These coefficients are for the analog input lines (**Analog I/O** socket).

In PIMikroMove, these parameters are available in the *Axis Definition* parameter groups in the ***Device Parameter Configuration*** window. In addition, you can check the matrix coefficients in the ***Axis Matrices*** window (open via ***View -> Axis Matrices*** menu item of the ***Device Parameter Configuration*** window).

INFORMATION

The coefficients of the input matrix are determined during calibration at the factory. The preset values of the coefficients for the internal sensors should not be changed.

If the connected mechanics has an ID-chip, the coefficients will be read in from the ID-chip (see "ID-Chip Support / Stage Replacement" (p. 131) for more information).

E-727.xxxA and .xxxAx models only:

The coefficients for the analog input lines should be set to zero as long as no external sensors are connected to the analog input lines or when the analog input is used for control value generation (see "Using the Analog Input" (p. 85) for more information).

While TSP? reports the position values of the input signal channels, the POS? command reports the axis positions calculated via the input matrix. Note that the physical units to be used for the axes can be queried with the PUN? command.

Control Value Generation

The control value for the motion of an axis can result from multiple sources (see below). Furthermore, the feedback from multiple sensors can be used to maintain the axis position, depending on the current operating mode. The interpretation of the control values depends on the settings of the output matrix (see "Output Generation" (p. 34) for more information). By default, the output matrix is set up so that control values correspond numerically to axis position values.

The E-727 provides the following operating modes:

- **Open-loop control** (also referred to as "servo-off state" in this document): No control algorithm is used, and the sensor feedback does not participate in the control value generation.

- **Closed-loop control** (also referred to as "servo-on state" in this document): Sensor feedback participates in the control value generation.
  For each logical axis, a control algorithm is used to generate corrections to the control value (default: PID algorithm). In addition, two notch filters are used for each axis (default: only active in closed-loop operation). The settings for control algorithm and notch filters are accessible as parameters. See "Control Details" (p. 35), "Servo Controller Dynamic Tuning" (p. 134) and " Parameters" (p. 185) for more information.

The operating mode can be selected with the SVO command for each axis. By default, open-loop control is active after power-on. Using the ***Power Up Servo On Enable*** parameter (ID 0x07000800), you can set up the individual axes to start with closed-loop control. When switching from open-loop to closed-loop control, the behaviour depends on the setting made with parameter

0x0e002000. Default: The current axis position is set as the target position. For further details, see "Parameter Overview" (p. 189). Switching from closed-loop to open-loop control sets the current closed-loop control value as the open-loop control value.
E-727 models with EtherCAT interface: A transition to the Operation Enabled state of the drive state machine switches the axis to closed-loop control. The axis remains in closed-loop operation until it is switched to open-loop control by an SVO command via TCP/IP or USB interface.

The E-727 supports the following control sources:

- Move commands, sent from the command line or from a macro:
  MOV and MVR in closed-loop operation
  SVA and SVR in open-loop operation
  IMP and STE commands generate an impulse or step response

- Wave Generator: The wave generator is enabled with WGO.
  An offset value can be added to the wave generator output using the WOS command.

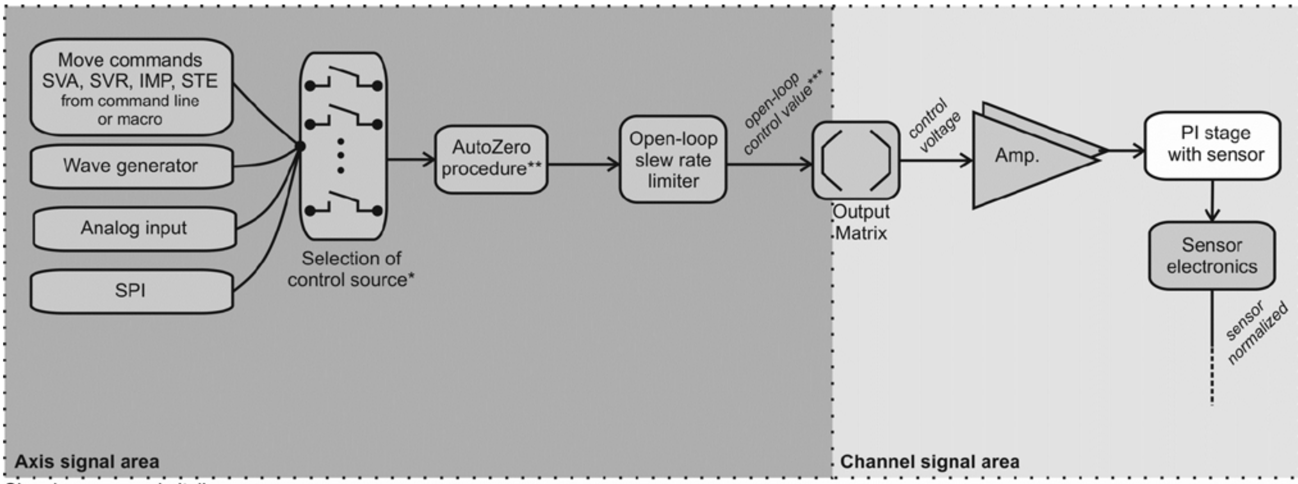
- Analog Input (only possible with E-727.3CDA, .3RDA, .3SDA models): The analog control input is enabled via parameter settings, see "How to work with the Analog Input" (p. 85) for more information.
  An offset value can be added to the analog input scaled value using the AOS command.

- AutoZero procedure: this procedure is started by the ATZ command and performed in open-loop operation only (if servo is on, it will be switched off automatically during the AutoZero procedure and on again afterwards). The AutoZero procedure has the highest priority, i.e. it will overwrite the control values given by all other sources. When the analog control input is enabled, it will be disabled automatically at the start of the AutoZero procedure and reenabled again when AutoZero is finished. See "AutoZero Procedure" (p. 62) for details.

The E-727 can also be commanded via an SPI master. Depending on the SPI data segment used, target values (with same write priority as analog control input) or GCS commands can be sent from the SPI master. See "SPI Interface" (p. 144) for details.

E-727 models with EtherCAT interface can also be commanded by an EtherCAT master. The EtherCAT master sends target values for closed-loop operation. The PC interfaces are disabled when EtherCAT communication is running (OPERATIONAL state). For further details on the interaction of the EtherCAT interface with other control sources, see "EtherCAT Interface" (p. 164).

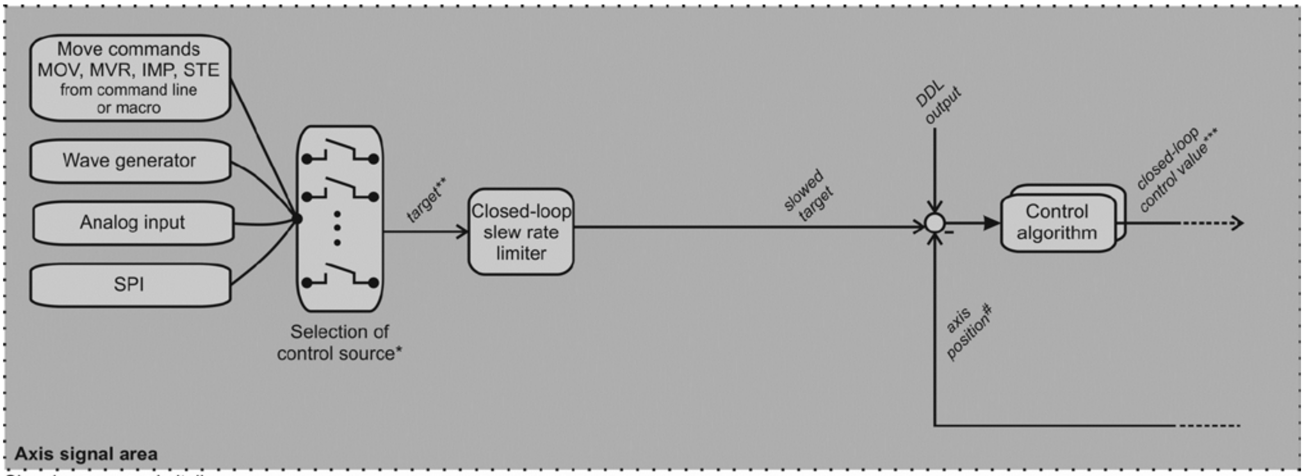
Open-Loop Control (all E-727 models):



Signal names are in italics
* Only one control source can be used at a time.
** The AutoZero procedure is triggered with the ATZ command and has the highest priority for signal manipulation.
If the AutoZero procedure is not running, it is bypassed. For further details, see „AutoZero Procedure“.
*** The open-loop control value is reported by the SVA? command.

Figure 9: Control sources for an axis in open-loop operation

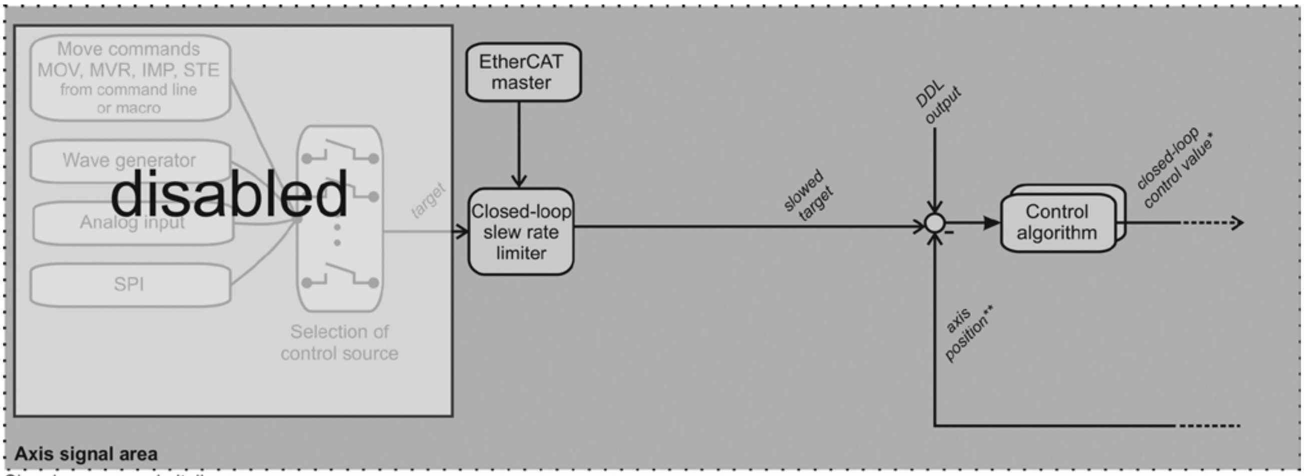
Closed-Loop Control (E-727 models without EtherCAT interface):



Signal names are in italics
* Only one control source can be used at a time.
** The target position is reported by the MOV? command.
*** The closed-loop control value is the input signal for the output matrix,
see block diagram of closed-loop operation for details.
\# The axis position is the output signal of the input matrix,
see block diagram of closed-loop operation for details.

Figure 10: Control sources for an axis in closed-loop operation

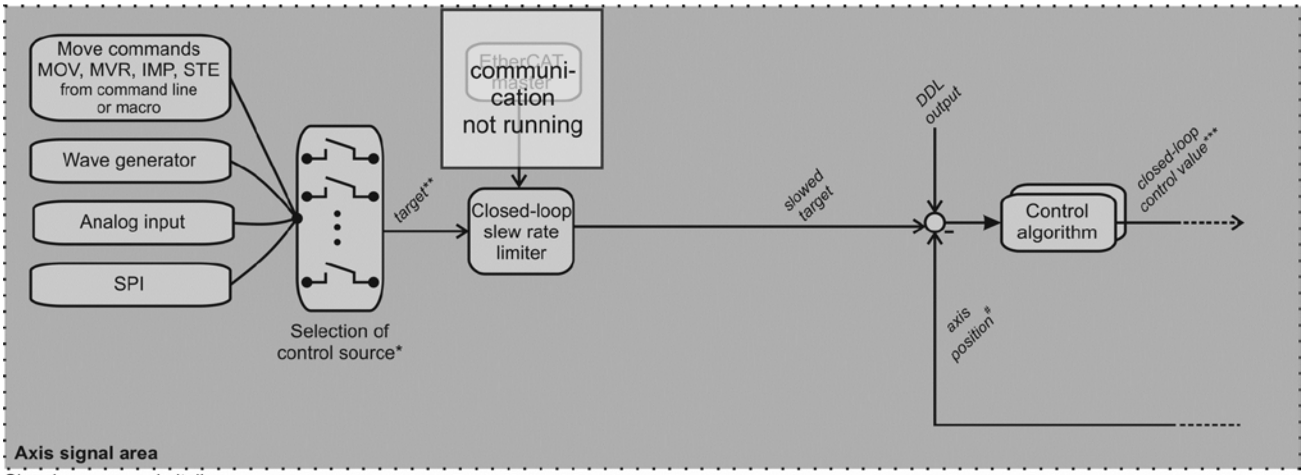
Closed-Loop Control when EtherCAT communication is running (E-727.xxxF, .xxxAF only):



Signal names are in italics
* The closed-loop control value is the input signal for the output matrix, see block diagram of closed-loop operation for details.
** The axis position is the output signal of the input matrix, see block diagram of closed-loop operation for details.

Figure 11: EtherCAT master controls an axis in closed-loop operation

Closed-Loop Control when EtherCAT communication does not run (E-727.xxxF, .xxxAF only):



Signal names are in italics
* Only one control source can be used at a time.
** The target position is reported by the MOV? command.
*** The closed-loop control value is the input signal for the output matrix, see block diagram of closed-loop operation for details.
# The axis position is the output signal of the input matrix, see block diagram of closed-loop operation for details.

Figure 12: EtherCAT communication does not run; control sources for an axis in closed-loop operation

Output Generation

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

With E-727.xxxA and .xxxAx models, the control voltage values can also be output by the analog output line to control an external amplifier (see "Using the Analog Output" (p. 94) for more information).

Output matrix of the E-727:

$$\begin{pmatrix} Output\ Ch\ 1 \\ Output\ Ch\ 2 \\ Output\ Ch\ 3 \\ Output\ Ch\ 4 \end{pmatrix} = \begin{pmatrix} p11 & p12 & p13 \\ p21 & p22 & p23 \\ p31 & p32 & p33 \\ p41 & p42 & p43 \end{pmatrix} * \begin{pmatrix} Axis\ 1 \\ Axis\ 2 \\ Axis\ 3 \end{pmatrix}$$

In equation form:

$OutputCh_1 = p_{11}Axis_1 + p_{12}Axis_2 + p_{13}Axis_3$
$OutputCh_2 = p_{21}Axis_1 + p_{22}Axis_2 + p_{23}Axis_3$
$OutputCh_3 = p_{31}Axis_1 + p_{32}Axis_2 + p_{33}Axis_3$
$OutputCh_4 = p_{41}Axis_1 + p_{42}Axis_2 + p_{43}Axis_3$

The matrix coefficients are given by the following parameters (with i = 1 to 3, or 1 to 4, depending on the number of axes provided by the E-727, i.e. each parameter has a different value for each of the logical axes):

p_{1i} = ***Driving Factor of Piezo 1***, parameter ID 0x09000000,
p_{2i} = ***Driving Factor of Piezo 2***, parameter ID 0x09000001,
p_{3i} = ***Driving Factor of Piezo 3***, parameter ID 0x09000002
P_{4i} = ***Driving Factor of Piezo 4***, parameter ID 0x09000003

In PIMikroMove, these parameters are available in the *Axis Definition* parameter groups in the ***Device Parameter Configuration*** window. In addition, you can check the matrix coefficients in the ***Axis Matrices*** window (open via ***View -> Axis Matrices*** menu item of the ***Device Parameter Configuration*** window).

INFORMATION

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

If the connected mechanics has an ID-chip, the coefficients will be read in from the ID-chip (see "ID-Chip Support / Stage Replacement" (p. 131) for more information).

The VOL? command reports the current voltage output of the output signal channel (in volts).

Control Details

Slew Rate Limitation

In closed-loop operation, the target signal can be manipulated by a slew rate limitation. For fastest possible settling, the slew rate limitation can be switched off with the VCO command.

The slew rate limit for an axis is given by the ***Servo Loop Slew-Rate*** parameter (ID 0x07000200).

Control Algorithms

The control algorithm to be used in closed-loop operation can be selected for each axis via the ***Closed-Loop Control Mode*** parameter (ID 0x07030100). Possible values of the parameter:

0 = None

1 = PID algorithm (position control)

Optional: 2 = Advanced Piezo Control (APC) algorithm, licence must be ordered separately, see “Accessories” (p. 19)

With the PID algorithm, up to two notch filters can be used. The PID algorithm in principle has the following structure:

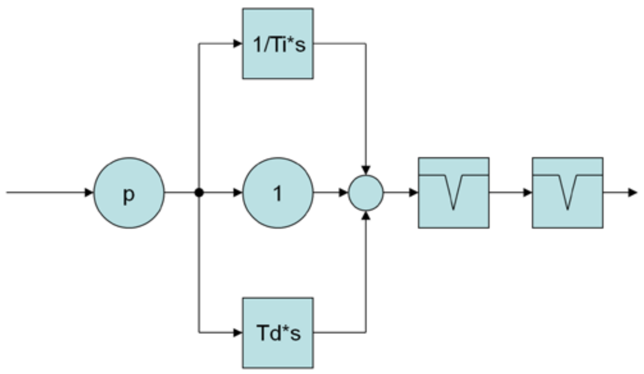


Figure 13: Structure of the PID algorithm; two notch filters are available

Position control (parameter value = 1):

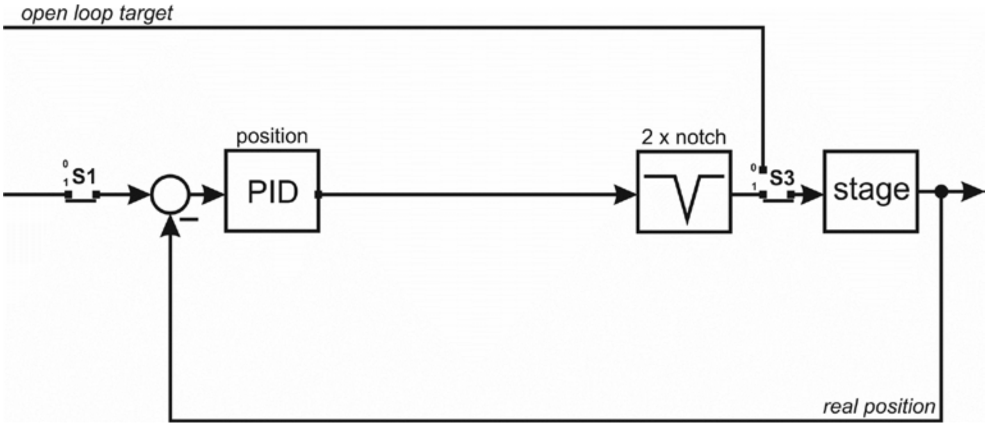


Figure 14: Control structure for position control (mode 1). Here it is assumed that the notch filters are disabled in open-loop operation (default setting of the Enable Notch In Open-Loop parameter).

Control mode	Servo-control state	S1	S3
Position control (mode 1)	Open-loop operation	0	0
	Closed-loop operation	1	1

The PID algorithm can be configured with the following parameters:

Parameter	Notes
Servo-Loop P-Term, ID 0x07000300	P constant for position control Must be > 0. For further details, see "Servo-Controller Dynamic Tuning" (p. 131).
Servo-Loop I-Term, ID 0x07000301	Integrator time constant Ti for position control output = Ts / Ti · ∑ input where Ts is the servo update time (parameter 0x0E000200). When the time constant Ti is zero, then the integrator is turned off. For further details, see "Servo-Controller Dynamic Tuning" (p. 131).
Servo-Loop D-Term, ID 0x07000302	Differentiator time constant Td for position control output = Td / Ts · Δinput where Ts is the servo update time (parameter 0x0E000200). Must be > 0. For further details, see "Servo-Controller Dynamic Tuning" (p. 131).

Feedforward

In closed-loop operation, for positioning systems with piezo actuators the main component regarding the tracking error (tracking error = commanded position – real position) is the phase shift between the commanded position and the real position. By adding a feedforward signal to the control algorithm this phase shift can be reduced.

For every axis, an input signal channel of the controller can be selected as feedforward signal using the ***Feed Forward Input Channel Index*** parameter (ID 0x07030900; S1 in the figure below). This way, an external preshaping sequence can be used to compensate for a known error, for example. If no input signal channel is selected, the target signal is used as feedforward signal (value of parameter 0x07030900 is zero).

The feedforward signal is fed in before the notch filters to avoid exciting the resonances of the system.

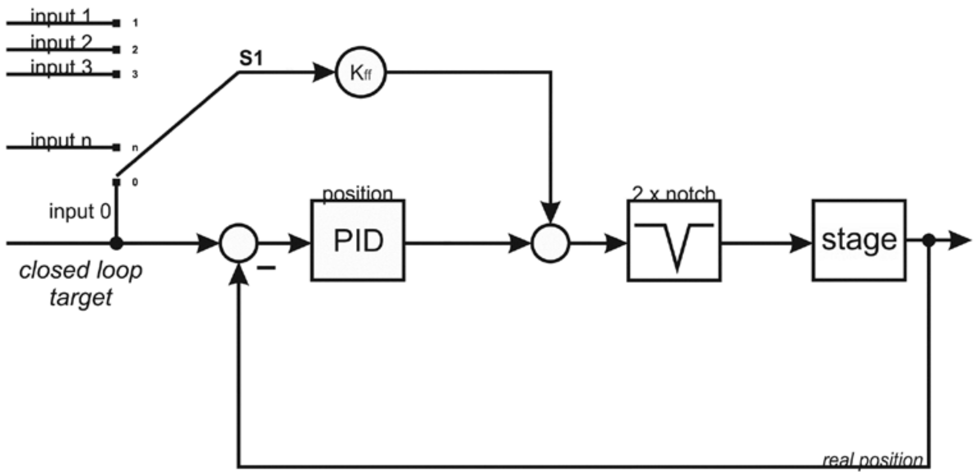


Figure 15: Control structure for position control with feedforward signal

For every axis, the usage of the feedforward signal can be configured using the ***Feedforward Gain*** parameter (ID 0x07030600).

Notch Filters

The E-727 provides two notch filters per axis. 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 mechanics.

The transfer function of a notch filter is as follows:

$$G(s) = k^2 \times \frac{s^2 + 2 \times \omega \times r \times s + \omega^2}{s^2 + 2 \times \omega \times k \times s + \omega^2 \times k^2}$$

Where
G(s) is the transfer function of the notch filter
k is the bandwidth of the notch filter
s is the input signal
ω is the angular frequency, with ω = 2*π*f0, where f0 is the notch filter frequency in Hz
r is the notch rejection

The notch filters can be configured using the following parameters:

Parameter	Notes
Notch frequency 1 ID 0x08000100 ***Notch frequency 2*** ID 0x08000101	Frequency f_0 of notch filter 1 and notch filter 2, in Hz. The maximum value is: $f_{0max} = 0.45*f_{sample}$ where f_{sample} is the servo rate in Hz (1/***Servo Update Time*** (ID 0x0e000200)) Adjusting the notch filter frequency can be useful, particularly in the case of very high loads. For further details, see “Adjusting the Notch Filter(s) in Open-Loop Operation” (p. 136).
Notch Rejection 1 ID 0x08000200 ***Notch Rejection 2*** ID 0x08000201	Notch rejection value r for notch filter 1 and notch filter 2. 0 to 0.98 Recommended value is 0.05. A notch rejection value of 1 deactivates the notch filter. The notch rejection value determines the filter width of the notch filter, i.e. it scales the damping done by the notch filter: The greater the rejection value, the wider the frequency spectrum of the damping, but the smaller the damping effect.
Notch Bandwidth 1 ID 0x08000300 ***Notch Bandwidth 2*** ID 0x08000301	Bandwidth k of notch filter 1 and notch filter 2 ≥ 0.1 The notch filter bandwidth determines the effect of the low-pass filtering: The smaller the bandwidth, the smaller the low-pass filter frequency.
Creep factor T1/sec ID 0x08000400 ***Creep factor T2/sec*** ID 0x08000401	Currently not used; provided for future applications.
Enable Notch in Open Loop ID 0x08000500	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 (default) 1 = enable notch filter in open-loop operation
Notch Filter Calculation Method ID 0x08000600	0 = bilinear 1 = zero-order hold 2 = frb (rejection rate is independent of bandwith)

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 nonvolatile 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. For further details, see p. 131.

Overtemp Protection

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 the following temperature thresholds for overtemp protection of the amplifier:

- E-727.xxxP and .xxxAP models for higher output current only: Temperature threshold 1 (“alert threshold”): 66 °C
- All E-727 models: Temperature threshold 2 (“switch-off threshold”): 72 °C

If the sensor value exceeds threshold 2, the amplifier output is switched off automatically, and all axes are switched to open-loop control.

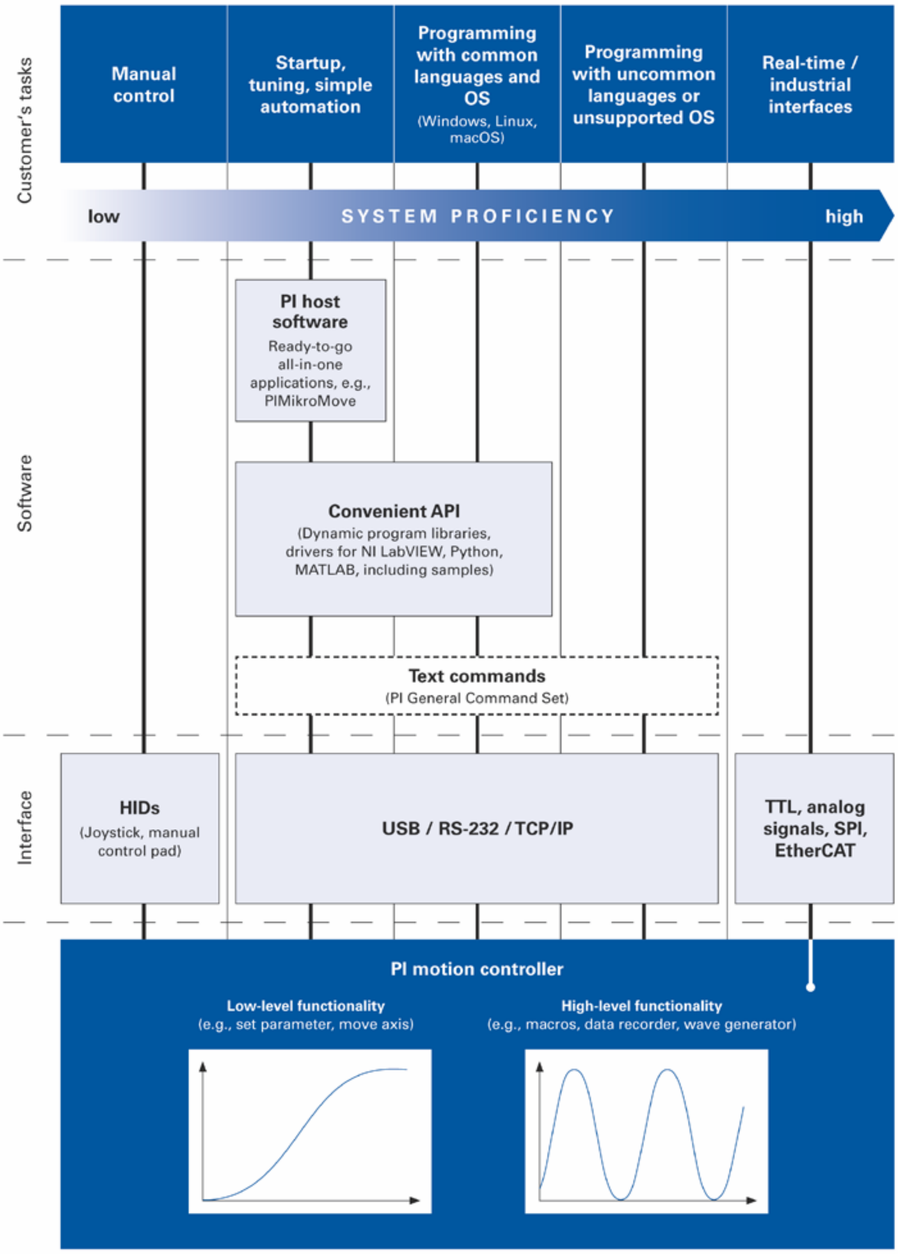
Use the DIA? command to query if a threshold is exceeded and if the amplifier output is active.

Possible measures to avoid overheating:

- Observe the installation instructions (p. 47).
- With high output power, keep the ambient temperature below 35 °C.
- If the sensor value exceeds threshold 1: Reduce the output power of the E-727 by reducing the frequency and/or the amplitude in dynamic operation. Ensure adequate ventilation.
- If the sensor value exceeds threshold 2: Stop the wave generator. Let the system cool down. Ensure adequate ventilation.

Overview of PC Software

PI's systems can generally be controlled as follows:



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

- Windows: Windows 7, 8 and 10 (32 bit, 64 bit)
- Linux: Kernel 2.6, GTK 2.0, from glibc 2.15

PC software	Operating system	Short description	Recommended use
Dynamic program library for GCS	Windows, Linux	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).	For users who would like to use a dynamic program library for their application. Is required for PIMikroMove. Is required for the drivers for NI LabVIEW software.
Drivers for use with NI LabVIEW software	Windows, Linux	NI LabVIEW is a software for data acquisition and process control (must be ordered separately from National Instruments). The E-727 software is a collection of virtual instrument drivers (VI drivers) for the E-727 controller. In addition to the product-specific drivers, the product CD also contains the *Analog drivers*, a collection of drivers for generating an analog control signal; see "Performing the Initial Installation" (p. 43). The drivers support the PI General Command Set.	For users who want to use NI LabVIEW to program their application.
Merge Tool for use with drivers for NI LabVIEW software	Windows	The Merge Tool allows you to combine product-specific drivers from PI with each other.	For users who want to operate several products from PI at the same time while using NI LabVIEW.
MATLAB drivers	Windows	MATLAB is a development environment and programming language for numerical calculations (must be ordered separately from MathWorks). The PI MATLAB driver consists of a MATLAB class that can be included in any MATLAB script. This class supports the PI General Command Set. The PI MATLAB driver does not require any additional MATLAB toolboxes.	For users who want to use MATLAB to program their application.

PC software	Operating system	Short description	Recommended use
PIMikroMove	Windows	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 ***Device Parameter Configuration*** window, PIMikroMove requires the NI LabVIEW Run-Time Engine; see "Performing the Initial Installation" (p. 43).	For users who want to perform simple automation tasks or test their equipment before or instead of programming an application. A log window showing the commands sent makes it possible to learn how to use the commands.
PITerminal	Windows	Terminal program that can be used for nearly all PI controllers (see the description of the ***Command Entry*** window in the PIMikroMove user manual).	For users who want to send GCS commands directly to the controller.
PI Update Finder	Windows	Checks the PI software installed on the PC. If more current versions of the PC software are available on the PI server, downloading is offered.	For users who want to update the PC software.
PI Firmware Update Wizard	Windows	Program for user support when updating firmware of the E-727.	For users who want to update the firmware.
USB driver	Windows	Driver for the USB interface	For all users.

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.

E-727 models for capacitive sensors:

Electromagnetic signals in the range of the sensor frequency (100 kHz) can interfere with the sensor signal.

- Avoid electromagnetic signals in the range of 100 kHz.
- Keep in mind that low-frequency signals may have harmonics in the range of 100 kHz.
- If interfering signals in the range of 100 kHz cannot be avoided, take particular care to ensure suitable shielding and grounding.

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 (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. 45), read the following information.

INFORMATION

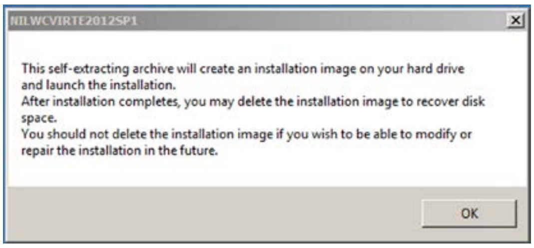
When PIMikroMove **is installed (default installation):**

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.

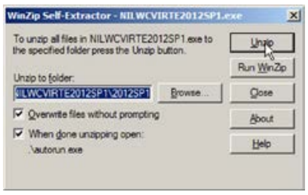
The InstallShield Wizard interrupts the installation of the PC software for the E-727 until the installation of the NI LabVIEW Run-Time Engine is started in the separate window.

➢ 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 accidently 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. 43).
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:

   – Driver for use with NI LabVIEW software
     Exception: The Analog 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
   – PI Update Finder for updating the PC software
   – 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:

- You have installed the PI Update Finder from th product CD (p. 43).
- You have the A000T0028 Technical Note for the PI Update Finder at hand. You can find the document on the product CD.
- 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 on the product CD.

➢ If your PC uses a Linux operating system:

- You have the user name and password for the E-727 at hand. See the instructions on p. 10 for how to obtain the access data.

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. 234).

1. Open the website **www.pi.ws**.
2. Log in with the access data for E-727 (user name, password).
3. Click ***Search***.
4. Enter the product code up to the period (“E-727”) into the search field.
5. Click ***Start search*** or press the Enter key.
6. Open the corresponding product detail page in the list of search results:
   a) If necessary: Scroll down the list.
   b) If necessary: Click ***Load more results*** at the end of the list.
   c) Click the corresponding product in the list.
7. Click the ***Downloads*** tab.

   The “CD Mirror” archive file is displayed under ***Software Files***.
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).
14. Install the update.

Ensuring Ventilation

Only a correct installation as described below will ensure correct operation of the E-727. To avoid overheating of the E-727:

- Set up the E-727 so that all ventilation holes in the housing are freely accessible.
- Ensure adequate ventilation at the place of installation.
- Keep the ambient temperature to a non-critical level (< 40 °C).
- E-727.xxx, .xxxA, .xxxF and .xxxAF models:
  – Allow at least 10 cm clearance from the top and rear, and a clear space of at least 15 cm from the sides of the E-727. If this is not possible, make sure that the place of installation is cooled sufficiently.
- E-727.xxxP and .xxxAP models for higher output current:
  – Allow at least 2 cm clearance from the sides and top, and a clear space of at

least 15 cm x 15 cm x 15 cm from the rear of the E-727.
- Do not remove the rubber feet from the bottom of the E-727 if you use it as a benchtop device. The rubber feet ensure that the ventilation holes in the bottom of the E-727 act as the air intake for convective cooling of internal components (except for the amplifiers).
- If you place the E-727 in a rack: Remove the rubber feet from the bottom of the E-727 only if the rack ensures that the ventilation holes in the bottom of the E-727 act as the air intake for convective cooling.

See also "Overtemp Protection" (p. 39).

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. 11).
✓ The E-727 is switched off via the **Power** switch.

Tools and accessories

- Suitable protective earth conductor:
  - Cable cross-section ≥0.75 mm^2
  - 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. 14).

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.

Start-Up

This chapter is intended to enable you to start initial test motions of a stage that is connected to a E-727 in the PIMikroMove PC software.

The start-up should comprise the following steps in the given order:

- Starting the system in PIMikroMove (p. 51): Installation, power-on, communication between E-727 and PC in PIMikroMove, configuration of PIMikroMove
- Creating backup file for controller parameters (p. 53)
- Executing test motions in open-loop operation (p. 54): First test of the function

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. 48).
- 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. 136) and "Checking and Optimizing the Servo-Control Parameters" (p. 140).

NOTICE

Damage to piezo tip/tilt systems with differential drive!

PI piezo tip/tilt systems with differential drive, such as S-334 models, provide motion in two axes (θ_X, θ_Y). The motion is controlled by axes 1 and 2 of the E-727. If a third axis is provided by the E-727, it is only used to give access to the fixed voltage (100 V) that is required by the differential drive. A piezo tip/tilt system with differential drive can be damaged by oscillations when the servo mode is switched on for axis 3 of the E-727.

- Do **not** switch on the servo mode for axis 3 of the E-727.
- Make sure that the ***Power Up Servo On Enable*** parameter (ID 0x07000800) has the value 0 (= servo mode is **not** automatically switched on) for axis 3 of the E-727.

Changing the piezo output voltage too fast can cause damage to a piezo tip/tilt system with differential drive.

- Make sure that for axes 1, 2 and 3 of the E-727, the velocity for rising and falling of the piezo output voltage in open-loop operation is limited to a suitable value. The velocity is given by the ***Open Loop Slew-Rate*** parameter (ID 0x07000201). For axes 1 and 2, the parameter value should be set to the axis travel range in µrad (e.g., 50000 (µrad/s) with S-334). For axis 3, the parameter value should be set to 100 (= 100 V/s).

When the E-727 is not used but is to remain switched on to ensure the temperature stability, the piezo output voltages should be set to 0 V to increase the lifetime of the piezo ceramics. When a third axis is present in the E-727, zeroing and restoring the piezo output voltages in wrong order can cause damage to a piezo tip/tilt system with differential drive.

1. Because zeroing/restoring the piezo output voltages will cause motion of axes 1 and 2, make sure that the axes can move safely.
2. Proceed as follows for zeroing the piezo output voltages when the E-727 is not used but is to remain switched on:
   a. Switch off the servo mode for axes 1 and 2; corresponding command:
   `SVO 1 0 2 0`
   b. Set the piezo output voltage to 0 V for axes 1 and 2; corresponding command:
   `SVA 1 0 2 0`
   c. Wait 1 second, or check the piezo output voltages for channels 1 and 2 by sending:
   `VOL? 1 2`
   d. When the piezo output voltages for channels 1 and 2 are 0 V, set the fixed voltage to 0 V by sending the corresponding command for axis 3:
   `SVA 3 0`
3. Proceed as follows for restoring the piezo output voltages when the E-727 is still switched on and the piezo tip/tilt system is to be used again:
   a. Set to piezo output voltage to 100 V for axis 3; corresponding command:
   `SVA 3 100`
   b. Wait 1 second, or check the piezo output voltage for channel 3 by sending:
   `VOL? 3`
   c. Switch on the servo mode for axes 1 and 2; corresponding command:
   `SVO 1 1 2 1`.

INFORMATION

The E-727, the stage(s) and any adapter or adapter cable for the stage connection are supplied as a pre-configured system.

➢ If a connection assignment is given on the labels of the E-727 and/or stage(s) and/or adapter/adapter cable, 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. 227 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 E-727 performance can be reduced directly after power on due to thermal instability.

➢ Switch the E-727 on at least one hour before starting work.

➢ If the E-727 is not used, but should remain switched on to ensure the temperature stability: Make sure that the servo mode is switched off (open-loop operation) and the piezo output voltage is set to 0 V. To set the piezo output voltage to 0 V, set the axis position to a corresponding value with the SVA command.

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. 43).
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. 43).
   - Ensure the ventilation (p. 47).
   - Connect the E-727 to the protective earth conductor (p. 48).
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. If necessary, use the adapter or adapter cable which was delivered with the system.
   - 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.

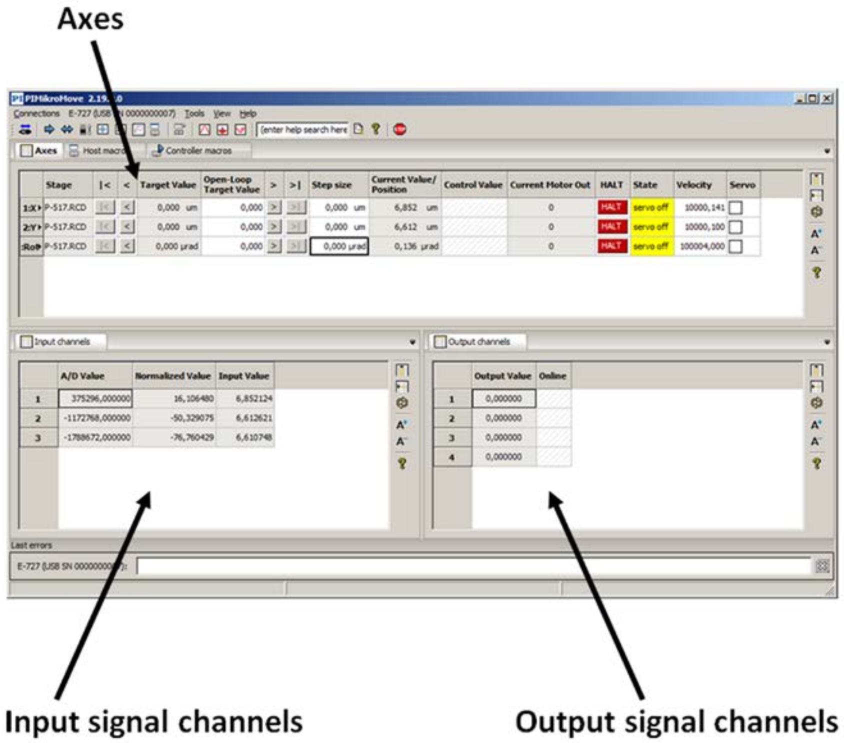
   The power-on/reboot sequence is finished when the **Power** LED is continuously lit.
6. Start PIMikroMove on the PC.
7. When the power-on/reboot sequence of the E-727 is finished, establish communication between the E-727 and the PC in PIMikroMove via RS-232 or USB or TCP/IP. Further details see "Communication" (p. 56).
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. 62)).

   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 (sensors, analog input) and output signal channels (output voltages for piezo actuators, analog output).



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.

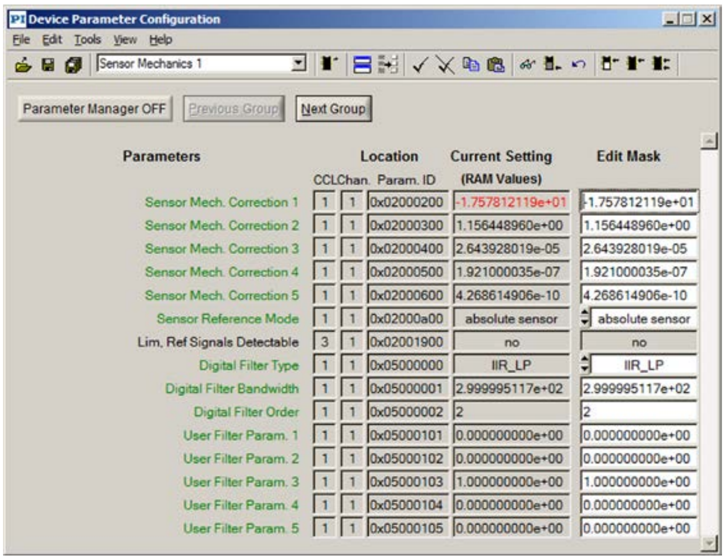
- ➢ Create a backup copy on the PC before changing the parameter values of the E-727. You can then restore the original settings at any time.
- ➢ Create an additional backup copy with a new file name each time after you optimize the parameter values.

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

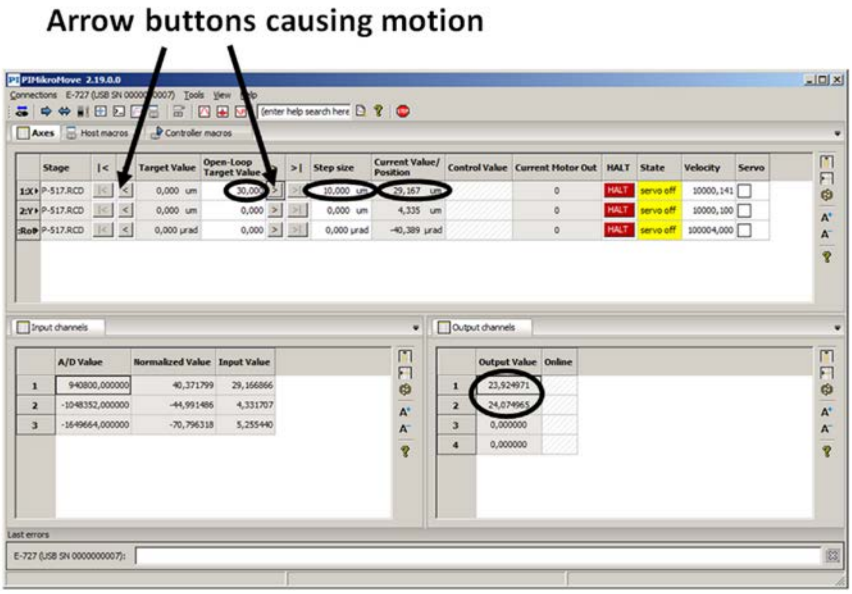
The first moves should be made in open-loop operation. With the factory default settings of the E-727, open-loop commanding means to give open-loop values which correspond approximately to axis positions.

1. In the main window of PIMikroMove, 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).

   Proceed as follows for each linear axis (for rotation axes, the given open-loop values correspond approximately to positions in µrad or mrad, 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 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). Because 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.

2. Make open-loop frequency response measurements in the ***Piezo Dynamic Tuner*** window of PIMikroMove to determine the resonant frequencies of the axes. See "Servo Controller Dynamic Tuning" (p. 134) for more information.

   If there are resonances which are intolerable in your application:

   – Adjust the notch filter settings for the axis before you switch to closed-loop operation for the first time (servo on). Furthermore, it might be necessary to readjust the preset servo parameters for the axis.

Operation

Communication

PC Interfaces

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

- TCP/IP (p. 56)
- Serial RS-232 connection (p. 61; not present if an EtherCAT interface is available)
- USB connection (p. 62)

With TCP/IP and USB connections, communication cannot be maintained after the E-727 is power-cycled or rebooted, or when EtherCAT communication is no longer running and TCP/IP and USB are enabled again. The TCP/IP or USB connection must then be closed and reopened.

INFORMATION

The following commands are available for the PC 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.

Additional Interfaces

In addition, the E-727 can be controlled by an SPI master, see “SPI Interface” (p. 144) for details. The PC interfaces and the SPI interface can be active simultaneously. The commands from the interfaces are queued in the order the completed command lines are received.

E-727 models with EtherCAT interface can also be controlled by an EtherCAT master, see “EtherCAT Interface” (p. 164) for details. The PC interfaces are disabled when EtherCAT communication is running (OPERATIONAL communication state).

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)

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).

  See “Adaptation of the interface parameters” below for details.

When the IP address is obtained via DHCP server

You should be aware of the following: When a TCP/IP connection is established at which the IP address is obtained from DHCP server, this address will automatically be written to the ***IP Address*** parameter (ID 0x11000600) in the volatile memory of the E-727. When you save the current settings to nonvolatile memory (e.g., with WPA 100) while the current command level is 1, the obtained IP address will become the new default address of the E-727.

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.

With E-727 firmware version 14.02.01.00 and newer (see response to *IDN?): The network cable can be connected when the E-727 is switched on. The E-727 will start searching for the DHCP server in the following cases:

- You try to establish communication via TCP/IP in the PC software, e. g., in PIMikroMove or PITerminal. Note that the E-727 may not be found at the first attempt. Try again after the first attempt has failed.
- E-727 models with EtherCAT interface: The PC interfaces are enabled again after stop of EtherCAT communication (= the E-727 has left the OPERATIONAL communication state).

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.

Adaptation of the interface parameters

The IP address and IP mask settings of PC and controller must be compatible with each other in the following cases:

- The E-727 is directly connected to an Ethernet connection of the PC.
- E-727 and the host PC both are connected to the same network where no DHCP server is available (in this case, the settings must also be compatible with those of any other devices in the same network).

Otherwise no connection can be established. You can configure either the PC or the controller settings to be compatible. If you have a network with multiple E-727, the settings of the individual controllers must be changed to have unique IP addresses for all devices in the network. See below for how to proceed.

If you want to change the PC settings:

Configure the connection on the PC according to the IP address and IP mask settings of the controller (see above for the default controller settings). Note that the following steps may vary in some details depending on the version of your Windows operating system:

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

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

2. In the ***Internet Protocol (TCP/IP) Properties*** window, activate ***Use the following IP address***. Make a note of the current ***IP address*** and ***Subnet mask*** settings, if any, in case they need to be restored later. Then adapt the ***IP address*** and ***Subnet mask*** settings to make them compatible with the settings of your E-727:

   Set the first three portions of IP address identical to those of the IP address of the E-727, while the last portion must be different. One possible IP address setting would be, for example, 192.168.168.2 (the default IP address of the E-727 is 192.168.168.10). Do not use "255" for the last portion.

   Set Subnet mask to 255.255.255.0 (if the IP mask of the E-727 is 255.255.255.0).

   Confirm with OK. An example is shown in the figure below.

3. Connect the E-727 to the Ethernet socket of the PC using the included, special, cross-over cable ("point-to-point" connection). If you connect the E-727 to a free access point (e.g. to a hub) on a network to which the PC is connected, it might be necessary to use a straight-through network cable.
4. Power on the E-727 system.
5. Establish the connection between PC and the E-727 as described below in "Establishing communication via TCP/IP".
6. If the connection fails: Change the last portion of the IP address setting on the PC and try again to connect.

Adapt *IP address* and *Subnet mask*; do not change the *Default gateway* setting

Press *OK*

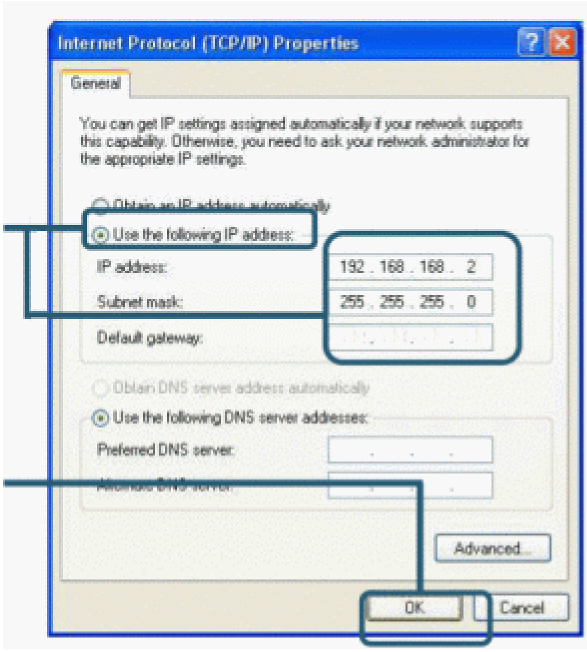


Figure 16: Internet Protocol (TCP/IP) Properties window, the settings shown are only examples, maybe they do not match that of your controller

If you want to change the controller settings:

1. Establish a serial connection between PC and E-727 as described in "Communication via the RS-232 Interface" (p. 61).
2. Use the IFS command in the command entry facility of the program to adapt the IP address and IP mask settings of the E-727 to those of the PC (to check the PC settings, you can open the ***Internet Protocol (TCP/IP) Properties*** window as described above):

   To change the IP mask, send
   `IFS 100 IPMASK mask`
   mask must be identical to the Subnet mask setting of the PC.

   To change the IP address, send
   `IFS 100 IPADR address`
   address: At least the last portion of the IP address must be different from that of the PC and any other device in the same network (the applicable address settings depend on the IP mask setting). If, for example, the PC has the IP address 172.21.0.1, send
   `IFS 100 IPADR 172.21.0.2:50000`
   Do not use "255", and do not change the port setting (must always be 50000).
3. Close the connection.
4. Connect the E-727 to the Ethernet socket of the PC using the included, special, cross-over cable ("point-to-point" connection). If you connect the E-727 to a free access point (e.g. to a hub) on a network to which the PC is connected, it might be necessary to use a straight-through network cable.

5. Power-cycle the E-727.
6. Establish the connection between PC and the E-727 as described in " Establishing communication via TCP/IP" below.

Establishing communication via TCP/IP

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

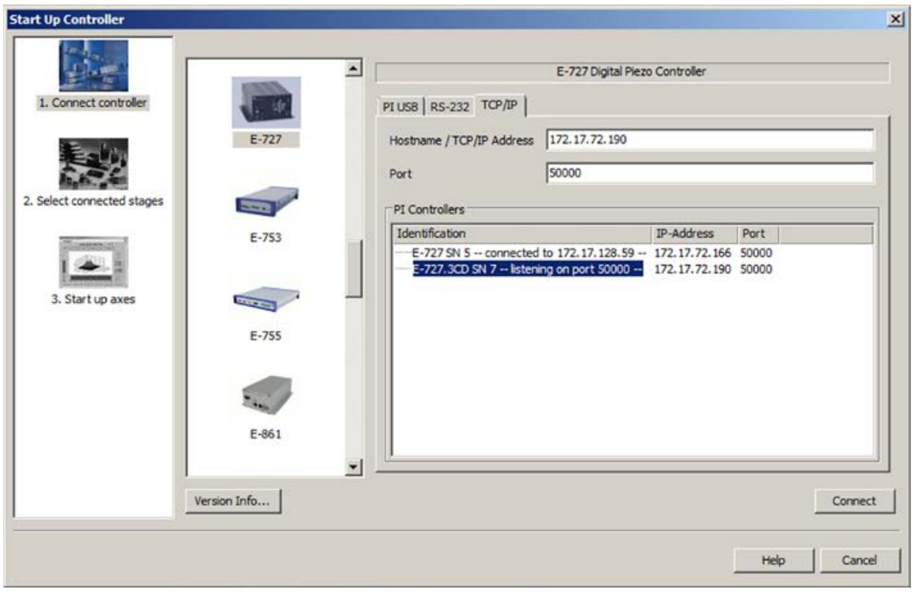
1. Start PIMikroMove.

   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.

   All E-727 controllers in the same network are listed in the ***PI Controllers*** field.
4. Click the ***PI E-727 … SN*** ... entry in the controller list (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 controller list, 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.
5. Check the IP address in the ***Hostname / TCP/IP Address*** field and the port number in the ***Port*** field.
6. 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.

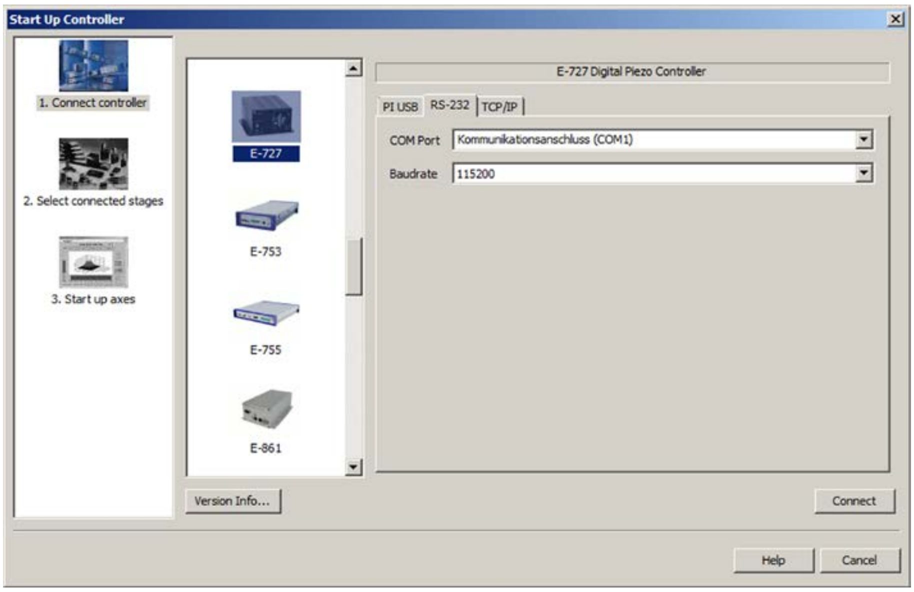
Establishing communication via RS-232

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

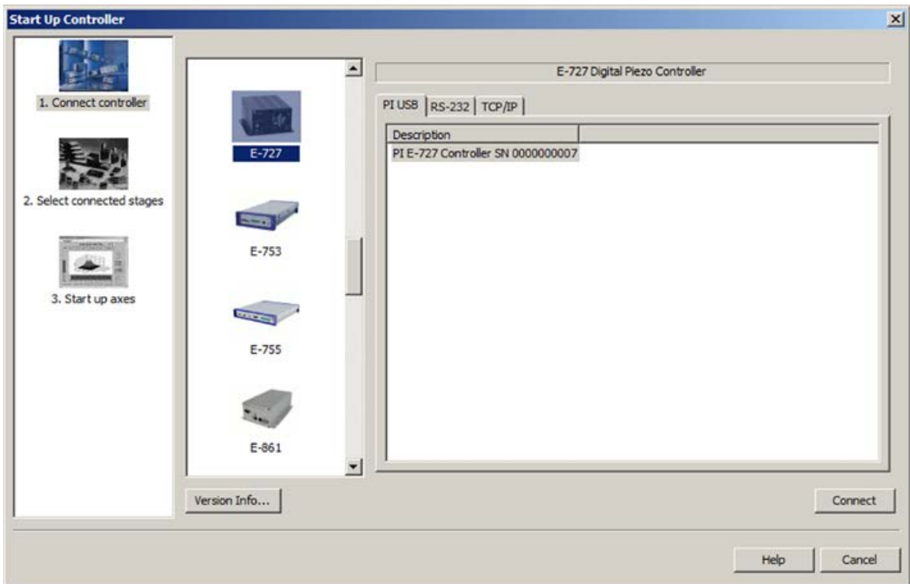
Establishing communication via USB

The procedure for PIMikroMove is described in the following. The procedure for other PC software programs (PITerminal, drivers for use with NI LabVIEW software) 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 ***USB*** tab on the right side of the window.
4. On the ***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.

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.

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.

Settings Changed by AutoZero

The AutoZero procedure changes the values of the parameters ***Sensor Mech. Correction 1*** (ID 0x02000200). With the E-727 models for piezoresistive sensors and for strain gauge sensors (details see “Model Overview”, p. 13), the AutoZero procedure also changes ***Sensor Offset factor*** (ID 0x02000102).

Starting AutoZero via Command Entry

Via command entry, you have the following options to start the AutoZero procedure of the E-727:

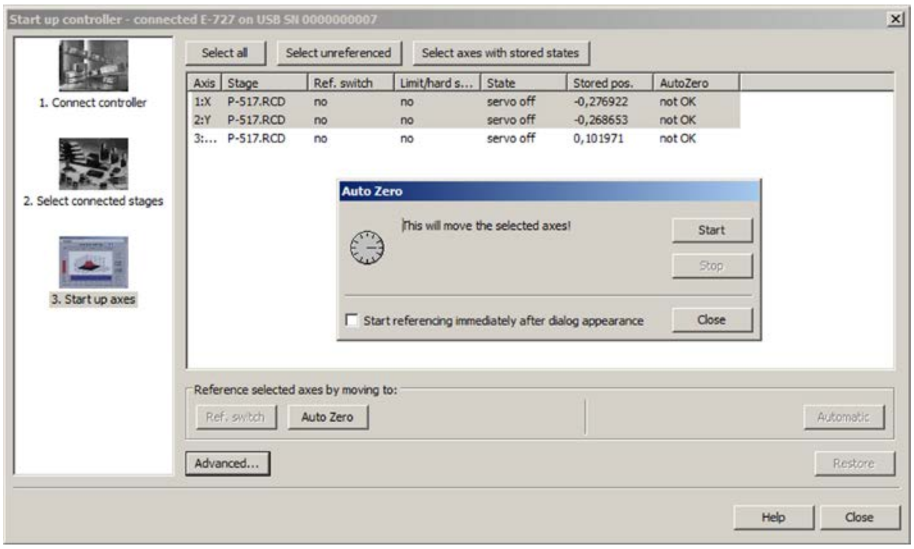
- Use the ATZ command to perform the AutoZero procedure once (see E-727 commands manual (PZ281E) for ATZ details). Afterwards save the values of the parameters ***Sensor Mech. Correction 1*** (ID 0x02000200) and ***Sensor Offset factor*** (ID 0x02000102) to nonvolatile memory.
- Send the ATZ command after every start or reboot of the E-727.
- 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

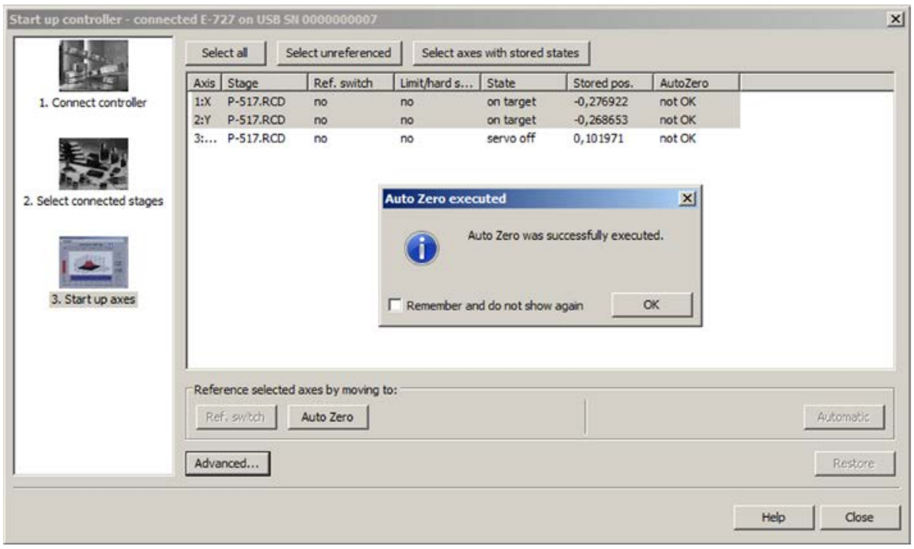
- In the ***Start up axes*** step of the ***Start up controller*** window, execute the AutoZero procedure.
  - To re-open the ***Start up controller*** window with the ***Start up axes*** step, select the ***E-727 > Start up axes...*** menu item in the main window.

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***.



Starting AutoZero via EtherCAT Master

E-727 models with EtherCAT interface (p. 164): The AutoZero procedure is the only homing method supported by the E-727. For details, see “Homing Mode” (p. 173).

In order for the EtherCAT master to be able to initiate a transition to the **Ready to switch on** state, the AutoZero procedure must be finished.

Special Function: Sensor Autoscaling

Sensor autoscaling can be included in the AutoZero procedure, if necessary.

INFORMATION

With the default settings of the relevant parameters, sensor autoscaling is excluded from the AutoZero procedure:
Sensor Autoscaling Enable (ID 0x03003700): default value is 0

Sensor Autoscaling Gain (ID 0x03003701): default value is 1.0

➢ Change the values of the parameters **only if sensor autoscaling is necessary**. Sensor autoscaling is necessary only if all of the following conditions are met:
  - The stage has no ID chip.
  - The stage has piezoresistive or strain gauge sensors.
  - PI has informed you that the stage design is suitable for sensor autoscaling (currently, this can apply to customized stages only).

Objective of sensor autoscaling:

- 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:

- If autoscaling of a sensor is to be included in the AutoZero procedure: The ***Sensor Autoscaling Enable*** parameter (ID 0x03003700) must be set to the value 1 for that sensor.

Sensor autoscaling setting changed by AutoZero:

If the ***Sensor Autoscaling Enable*** parameter (ID 0x03003700) has the value 1, the AutoZero procedure changes the ***Sensor Autoscaling Gain*** (ID 0x03003701) for the corresponding sensor channel (in addition to the settings mentioned on p. 63).

Data Recording

How to Use the Data Recorder

The E-727 includes a real-time data recorder. It is able to record several input and output signals (e.g. current position, sensor input, output voltage) from different data sources (e.g. controller axes, input and output channels, digital inputs and outputs). The gathered data is stored (temporarily) in "data recorder tables"—each table contains the signal from one data source. You can configure the data recorder flexibly, e.g. select the type of data and the data source. Furthermore, you can choose the number of record tables and hence influence their size.

For general information regarding the data recording you can send HDR?, which lists available options, and gives information about additional parameters and commands concerned with data recording.

How to Define What to Record—Set Record Options

The data recorder configuration, i.e. the assignment of data sources and record options to the recorder tables, can be read with the DRC? command. The answer gives the values of the parameters ***DRC Data Source*** (ID 0x16000700) and ***DRC Record Option*** (ID 0x16000701). Use the DRC command or change the parameter values directly in volatile memory. The default setting is that the current positions of the axes are recorded.

How to Start Recording—Set Trigger Options

Recording can be triggered in several ways. Ask with DRT? for the current trigger option and use DRT to change it.

INFORMATION

Trigger settings cannot be saved in the nonvolatile memory of the E-727. After the E-727 has been switched on or rebooted, factory default settings will therefore be active unless a configuration takes place with a start-up macro.

Irrespective of the DRT settings, data recording is always triggered by the following four commands:

- STE (step response measurement)
- IMP (impulse response measurement)
- WGO (wave generator start)
- WGR (restarts recording when the wave generator is running)

Recording always takes place for all data recorder tables and ends when the data recorder tables are completely filled.

If digital input lines are used to trigger data recording (configuration with DRT): For reliable triggering, the pulse width of the input signal has to be at least 2 x the servo update time of the E-727 system. The servo update time is given in seconds by parameter 0x0E000200.

How to Read Recorded Data

The last recorded data can be read with the DRR? command. The data is reported in GCS array format. For details regarding GCS array see the separate manual (SM146E), which is provided on the E-727 CD. Reading out recorded data can take some time, depending on the number of points to be read! It is possible to read the data while recording is still in progress.

The number of points comprised by the last recording can be read with the DRL? command. This can be useful, for example, if you restart recording with WGR and want to read data while recording is still in progress.

How to Configure Number of Tables and Sampling Period

The number of available data recorder tables can be read with the TNR? command. The answer gives the value of the ***Data Recorder Chan Number*** parameter, ID 0x16000300. You can change the parameter value to increase or decrease the number of data recorder tables. For the E-727, the number of tables must be in the range of 1 to 8.

The total number of points available for data recording is given by the ***Data Recorder Max Points*** parameter, ID 0x16000200. The controller allocates these points in equal shares to the available tables (i.e. to the number of tables given in the TNR? answer). For the E-727, the total number of points is 262144. If, for example, TNR? replies 8, each table is comprised of 32768 points.

The data recorder sampling period can be read with the RTR? command. The answer gives the value of the ***Data Recorder Table Rate*** parameter (ID 0x16000000) whose default value is one servo cycle. You can cover longer periods by increasing this value. Use the RTR command or change the parameter value directly in volatile memory.

Data-Recorder Related Commands and Parameters

Command	Description	Notes
DRC	Set Data Recorder Configuration	Assigns data sources and record options to data recorder tables in volatile memory (***DRC Data Source*** parameter, ID 0x16000700, and ***DRC Record Option*** parameter, ID 0x16000701)
DRC?	Get Data Recorder Configuration	Reads current data recorder settings
DRL?	Get Number Of Recorded Points	Reads the number of points comprised by the last recording.
DRR?	Get Recorded Data Values	Note: Reading can take some time, depending on the number of points.
DRT	Set Data Recorder Trigger Source	Determines how recording is to be triggered. Settings will be lost on controller power down or reboot.
DRT?	Get Data Recorder Trigger Source	Reads current trigger option
HDR?	Get All Data Recorder Options	Lists available record options, gives information about additional parameters and commands concerned with data recording
IMP	Start Impulse and Response Measurement	Triggers recording
RTR	Set Record Table Rate	Changes the data recorder table rate in volatile memory (***Data Recorder Table Rate*** parameter, ID 0x16000000)
RTR?	Get Record Table Rate	Reads the current setting of the data recorder table rate (***Data Recorder Table Rate*** parameter, ID 0x16000000)
STE	Start Step and Response Measurement	Triggers recording
TNR?	Get Number of Record Tables	Reads the number of available data recorder tables (***Data Recorder Chan Number*** parameter, ID 0x16000300)

Command	Description	Notes
WGO	Set Wave Generator Start/Stop Mode	Triggers recording
WGR	Start Recording Synchronous to Wave Generator	Triggers recording

For detailed command descriptions see the GCS commands manual PZ281E. For the identifiers of the items which can be addressed with the commands see "Axes, Channels, Functional Elements" (p. 22).

Parameter ID	CCL for Write Access	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x16000000	1	System	1	INT	Data Recorder Table Rate
0x16000100	3	System	1	INT	Max Number of Data Recorder Channels
0x16000200	3	System	1	INT	Data Recorder Max Points
0x16000300	1	System	1	INT	Data Recorder Chan Number; the available data recorder points are allocated in equal shares to the number of tables given by this parameter
0x16000700	1	Data recorder table	8	INT	DRC Data Source
0x16000701	1	Data recorder table	8	INT	DRC Record Option

See "Parameters" (p. 185) for more information regarding the controller parameters and their handling.

Using Digital Input

The digital input lines of theE-727 are available on the **Digital I/O** socket, see p. 227 for the lines and pinout.

The values of all digital input lines can be recorded, see the DRC command for details.

The IN1 to IN4 input lines can be used to start data recording, see the DRT command for details.

The IN 4 line can also be configured as reset input, see p. 227 for details. In this case, do not use IN4 to start data recording.

The IN1 and IN2 input lines can be used in conjunction with the WGO command to trigger the wave generator output (IN1 and IN2) and to stop it (IN2). See "Wave Generator Started by Trigger Input" (p. 111) for an example.

Configuring Trigger Output

Overview of Trigger Options, Commands and Parameters

The digital output lines of theE-727 are available on the **Digital I/O** socket, see p. 227 for the lines and pinout.

The values of the digital output lines OUT1 to OUT3 can be recorded, see the DRC command for details.

You can program the digital output lines OUT1 to OUT3 to trigger other devices using the CTO command.

The general format of the CTO command is as follows, i.e. all trigger-related settings for a digital output line can normally be made with one command line (the number of arguments following a command mnemonic is limited to 32):

CTO {<TrigOutID> <CTOPam> <Value>}

The following trigger modes are supported by the E-727:

- 0 = Position Distance; a trigger pulse is written whenever the axis has covered a given distance. Optionally, values for StartThreshold and StopThreshold can be defined to enable the trigger output for a limited position range and a certain direction of motion only (negative or positive). When StartThreshold and StopThreshold are set to the same value, they will not be used. Further options which cannot be configured with CTO but only via parameters: The length of the trigger pulses can be set to a constant value, and filters can be applied to the axis position to reduce false triggers caused by position noise. See "Example—"Position Distance" Trigger Mode" (p. 71).
- 2 = OnTarget; the on-target status of the axis is written to the trigger output line (this status can also be read with the ONT? command). See "Example—"On Target" Trigger Mode" (p. 80).
- 3 = MinMaxThreshold; values for MinThreshold and MaxThreshold must be defined. When the axis position is inside the band specified by the MinThreshold and MaxThreshold values, the trigger output line is set high, otherwise it is set low. See "Example—"MinMax Threshold" Trigger Mode" (p. 81).
- 4 = Generator Level Trigger; the trigger line action must be defined with TWS. The trigger output will be synchronized with the wave generator output. 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. See "Example—"Generator Level Trigger" Mode" (p. 82).
- 9 = Generator Pulse Trigger; the trigger line action must be defined with TWS. The trigger output will be synchronized with the wave generator output. The length of 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. See "Example—"Generator Pulse Trigger" Mode" (p. 83).
- 14 = TriggerOutAND; the digital output line <TrigOutID> outputs the signal states of the output lines selected with TriggerOutMask (<CTOPam> ID 16). The states of the

selected lines are combined via AND bit operation. See "Examples—"TriggerOutAND" and "TriggerOutOR" Modes" (p. 84).

- 15= TriggerOutOR; the digital output line <TrigOutID> outputs the signal states of the output lines selected with TriggerOutMask (<CTOPam> ID 16). The states of the selected lines are combined via OR bit operation. See "Examples—"TriggerOutAND" and "TriggerOutOR" Modes" (p. 84).

To select the mode, set <CTOPam> = 3 and <Value> to the code of the mode; default selection is On Target (2).

Furthermore, it is possible to select the signal polarity for the digital output line (active high / active low). See "Example—Polarity Setting" (p. 84).

CTO changes the values of the parameters listed below in volatile memory. The current values in volatile memory can be read with the CTO? command. You can also change these parameters using SPA (volatile memory) or SEP (non-volatile memory). Furthermore, you can use WPA to copy the current values from volatile memory to non-volatile memory, where they become the power-on defaults. When using SPA, SEP or WPA, it is necessary to switch to command level 1 with CCL to have write access to the parameters. To read the parameter values, you can also query with the SPA? or SEP? commands.

Parameter ID	Command Level	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x18000201	1	Digital Output Line	3	FLOAT	CTO Trigger Step (<CTOPam> = 1)
0x18000202	1	Digital Output Line	3	INT	CTO Axis (<CTOPam> = 2)
0x18000203	1	Digital Output Line	3	INT	CTO Trigger Mode (<CTOPam> = 3)
0x18000205	1	Digital Output Line	3	FLOAT	CTO Min.Threshold (<CTOPam> = 5)
0x18000206	1	Digital Output Line	3	FLOAT	CTO Max.Threshold (<CTOPam> = 6)
0x18000207	1	Digital Output Line	3	INT	CTO Polarity (<CTOPam> = 7)
0x18000208	1	Digital Output Line	3	FLOAT	CTO Start Threshold (<CTOPam> = 8)
0x18000209	1	Digital Output Line	3	FLOAT	CTO Stop Threshold (<CTOPam> = 9)
0x18000210	1	Digital Output Line	3	INT	CTO Trigger Out Mask (<CTOPam> = 16)

The following examples can be reproduced using the command entry facilities of PIMikroMove or PI Terminal.

Example—"Position Distance" Trigger Mode

The "Position Distance" trigger mode is designed for scanning applications. A trigger pulse is written whenever the axis has covered the distance set with CTO (<TriggerStep>). The unit of <TriggerStep> is the physical unit of the axes (query with the PUN? command).

In addition to the basic configuration, the “Position Distance” trigger mode provides several options:

- Definition of pulse length, p. 72: variable (depending on velocity) or constant
- Definition of thresholds for trigger limitation to a certain position range and direction of motion, p. 74
- Filter definition for reduction of false triggers, p. 76

Basic trigger configuration

The following parameters must be set for the digital output line which is to be used for trigger output (<TrigOutID>):

- Axis (<CTOPam> = 2)
- TriggerMode (<CTOPam> = 3)
- TriggerStep (<CTOPam> = 1)

General notation of the CTO command for this mode (in fact, the command arguments can be divided in three "portions", each starting with the <TrigOutID> declaration):

Command mnemonic	Axis selection	Trigger mode selection	Step size setting
CTO	<TrigOutID> 2 *Axis*	<TrigOutID> 3 0	<TrigOutID> 1 *Stepsize*

Instead of using the CTO command, you can also set the values of the corresponding parameters with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

INFORMATION

Possible values for TriggerStep depend on the axis velocity.

- With high velocities, the minimum TriggerStep value is limited as follows:
  TriggerStep > 4 * velocity * Servo Update Time
  Servo Update Time is given in seconds by parameter 0x0E000200
  Examples (the Servo Update Time of the E-727 is 50 µs):
  - With a velocity of 1000 µm/s, the minimum TriggerStep value is 200 nm.
  - With a TriggerStep value of 100 nm, the maximum velocity is 500 µm/s.
- With very small velocities, the minimum TriggerStep value is limited by the noise of the position sensor. For reliable triggering, the TriggerStep setting has to be at least 5 times larger than the peak-to-peak level of the sensor noise. The sensor noise level can be reduced by digital filtering of the signal (p. 27). To detect the noise, you can record, for example, the position error of the axis using the data recorder.

Example: A pulse on the digital output line 1 is to be generated whenever axis 1 of the stage has covered a distance of 0.1 µm. Send:

CTO 1 2 1 1 3 0 1 1 0.1

Pulse length definition

The value of the ***Pos. Distance Trig. High Time Definition*** parameter (ID 0x18000400) determines how the trigger pulse length is set (i.e., the time the trigger output line is set to high):

- Parameter value = 0 (default setting): The trigger pulse length varies depending on the position distance set with <TriggerStep> and on the current velocity of the axis. The trigger output line changes its level each time the axis has covered half the TriggerStep distance.
- Parameter value = 1: The trigger pulse length is constant. The trigger high time is adjustable using the ***Position Distance Trigger High Time*** parameter (ID 0x18000401)

Switch to command level 1 with CCL to have write access to the parameters. To set the parameters, use SPA or SEP. Furthermore, you can use WPA to copy the current values from volatile memory to non-volatile memory, where they become the power-on defaults. To read the parameter values, you can query with the SPA? or SEP? commands.

Variable pulse length depending on <TriggerStep> setting and current axis velocity:

The ***Pos. Distance Trig. High Time Definition*** parameter has the value 0.

Example: If TriggerStep is 100 nm, a rising edge of the trigger signal (low -> high) will be followed by a falling edge (high -> low) when the axis has covered a distance of 50 nm. The next rising edge follows when the axis has covered another distance of 50 nm, and so on.

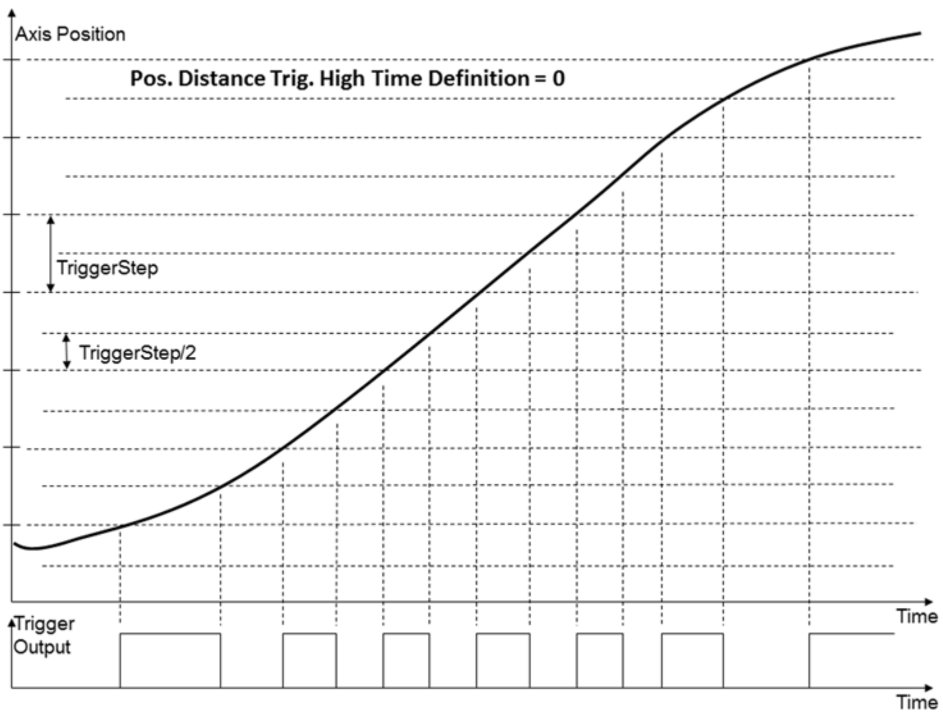


Figure 17: Position Distance" Trigger Mode, trigger pulse length depends on the axis velocity

Constant pulse length:

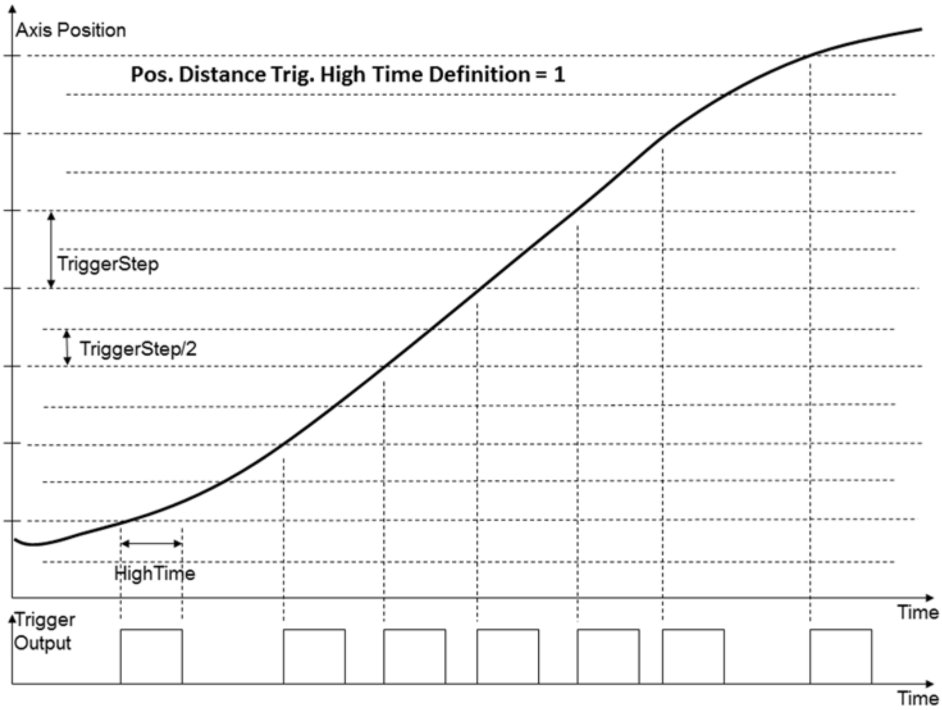


Figure 18: Position Distance" Trigger Mode, constant trigger pulse length

The ***Pos. Distance Trig. High Time Definition*** parameter has the value 1.

The trigger pulse length is constant. The trigger high time is adjustable using the ***Position Distance Trigger High Time*** parameter (ID 0x18000401). The parameter value is given in seconds. If the parameter value is set to 0 (default setting), the length of a trigger pulse is one servo cycle (50 µs).

Possible values for the ***Position Distance Trigger High Time*** parameter (ID 0x18000401) depend on the axis velocity and the TriggerStep value. Pulses can get lost if the pulse length and/or TriggerStep values are not suitable for the current axis velocity, see figure below for an example.

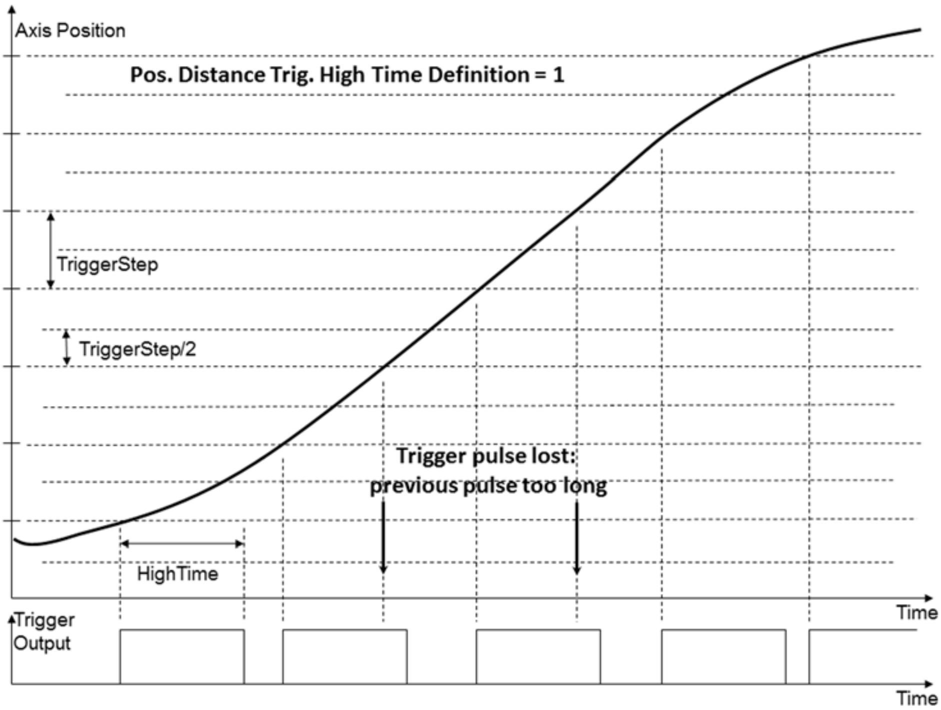


Figure 19: Position Distance" Trigger Mode, constant trigger pulse length; pulse length and/or step size not suitable for the axis velocity

Configuration of threshold values

Optionally, start and stop values can be set with CTO (<StartThreshold> and <StopThreshold>) to enable the trigger output for a limited position range and a certain direction of motion only (positive or negative). Should the direction of motion be reversed when the axis position is still between the start and stop values, the trigger output depends on the value of the ***Pos. Distance Trig. Single Direction*** parameter (ID 0x18000300). Default: Trigger pulses will continue to be generated. For further details, see “Filtering for reduction of false triggers”, p. 76.

When <StartThreshold> and <StopThreshold> are set to the same value, they will not be used.

The following parameters must then be set for the digital output line which is to be used for trigger output (<TrigOutID>):

- Axis (<CTOPam> = 2)
- TriggerMode (<CTOPam> = 3)
- TriggerStep (<CTOPam> = 1)
- StartThreshold (<CTOPam> = 8)
- StopThreshold (<CTOPam> = 9)

General notation of the CTO command for this option (in fact, the command arguments can be divided in five "portions", each starting with the <TrigOutID> declaration):

Command mnemonic	Axis selection	Trigger mode selection	Step size setting	Start threshold setting	Stop threshold setting
CTO	<TrigOutID> 2 Axis	<TrigOutID> 3 0	<TrigOutID> 1 *Stepsize*	<TrigOutID> 8 *Startpos*.	<TrigOutID> 9 *Stoppos*.

Instead of using the CTO command, you can also set the values of the corresponding parameters with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

INFORMATION

For reliable activation of the trigger output, the axis has to move through the “Trigger ready, not active” position range, or the motion has to start in this range (see Figure 20 and Figure 21).

For reliable deactivation of the trigger output, the axis has to move through the “Trigger stopped” position range, or the motion has to end in this range (see Figure 20 and Figure 21). Of course, the trigger output is also deactivated when the axis leaves the trigger range via the StartThreshold.

The distance between the start position of the axis and the StartThreshold setting has to be at least 3 times larger than the noise of the axis position (peak-to-peak value), as well as the distance between the end position of the axis and the StopThreshold setting. In addition, the distance between the start position of the axis and the StartThreshold has to be at least TriggerStep/4.

See the examples 2 and 3 below for details.

Example 2: A pulse on the digital output line 1 is to be generated whenever axis 1 of the stage has covered a distance of 0.1 µm, as long as axis 1 moves in positive direction in the range of 0.2 µm to 0.55 µm (start threshold < stop threshold). For reliable activation and deactivation of the trigger output, the axis motion should start at 0.15 µm and end at 0.6 µm. Send:

CTO 1 2 1 1 3 0 1 1 0.1 1 8 0.2 1 9 0.55

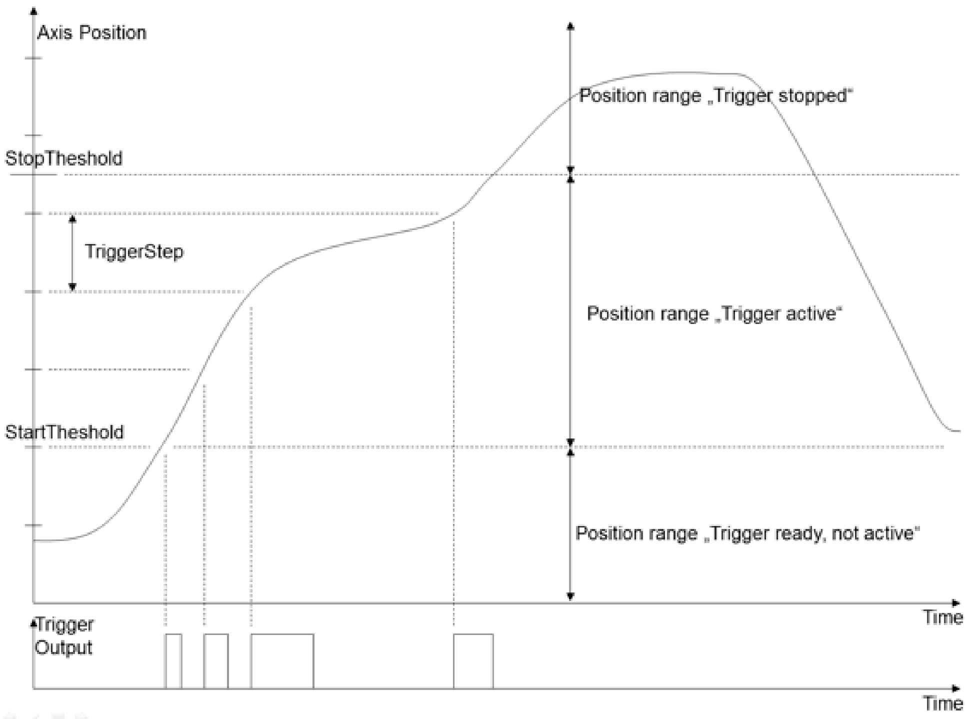


Figure 20: "Position Distance" Trigger Mode with threshold settings for positive direction of motion; variable pulse length (Pos. Distance Trig. High Time Definition parameter has the value 0)

Example 3: A pulse on the digital output line 1 is to be generated whenever axis 1 of the stage has covered a distance of 0.1 µm, as long as axis 1 moves in negative direction in the range of 0.55 µm to 0.2 µm (start threshold > stop threshold). For reliable activation and deactivation of the trigger output, the axis motion should start at 0.6 µm and end at 0.15 µm. Send:

```
CTO 1 2 1 1 3 0 1 1 0.1 1 8 0.55 1 9 0.2
```

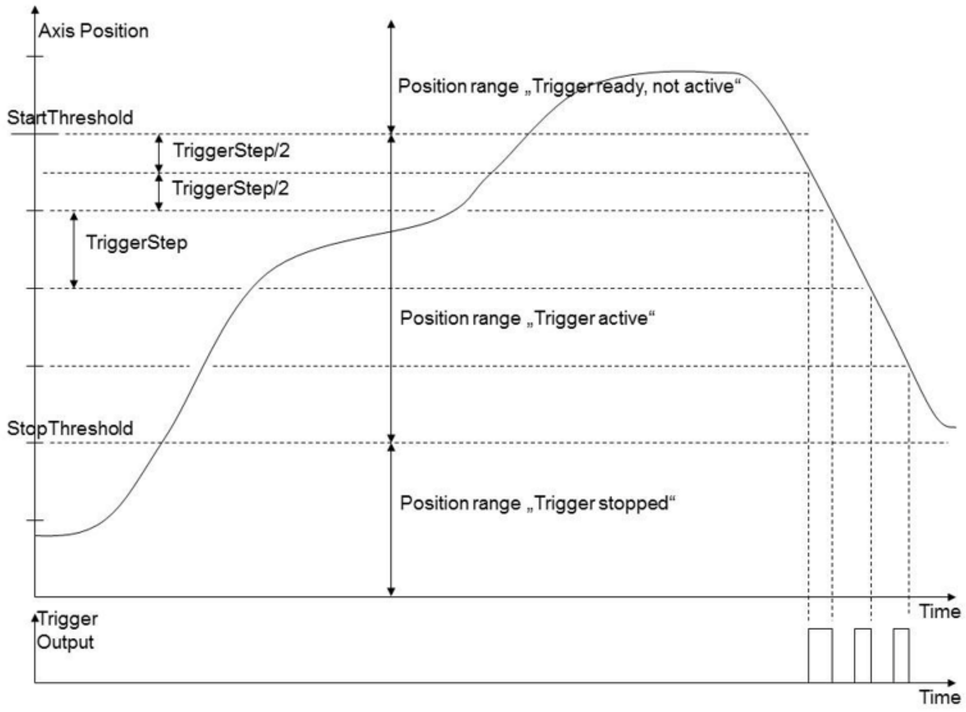


Figure 21: "Position Distance" Trigger Mode with threshold settings for negative direction of motion; here, the trigger output is not yet disabled reliably at the end of the curve

Filtering for reduction of false triggers

If the peak-to-peak level of the sensor position noise is greater than one fifth of the TriggerStep value set, this can lead to unwanted generation of multiple trigger pulses at the same position in "Position Distance" trigger mode. The occurrence of these false triggers can be reduced with the following parameters:

- Only allow the trigger output in one direction within the start/stop threshold values: ***Pos. Distance Trig. Single Direction*** parameter (ID 0x18000300)
- To prevent false triggering within the start threshold range also: ***Pos. Distance Trig. Filter Time*** parameter (ID 0x18000301) or ***Pos. Distance Trig. Filter Level*** parameter (ID 0x18000302)

However, the position precision cannot be retrieved by using the parameters to reduce the trigger impact of the sensor position noise.

Switch to command level 1 with CCL to have write access to the parameters. To set the parameters, use SPA or SEP. Furthermore, you can use WPA to copy the current values from volatile memory to non-volatile memory, where they become the power-on defaults. To read the parameter values, you can query with the SPA? or SEP? commands.

Direction of motion for trigger output:

If the start and stop thresholds are active in "Position Distance" trigger mode (start threshold ≠ stop threshold), the value of the ***Pos. Distance Trig. Single Direction*** parameter (ID 0x18000300) determines the direction of motion for trigger output:

- 0 (default setting) = Trigger is generated whenever the position is within the range defined by the start/stop thresholds. Trigger output is independent of the current direction of motion, i.e. continues when the motion direction is reversed before the axis position has reached the stop threshold (Figure 22).
- 1 = Trigger is generated only when the position is within the range defined by the start/stop thresholds **and** when the axis moves in the direction determined by the start/stop thresholds (Figure 23)

If start and stop thresholds are not active (start threshold = stop threshold), the ***Pos. Distance Trig. Single Direction*** parameter has no effect.

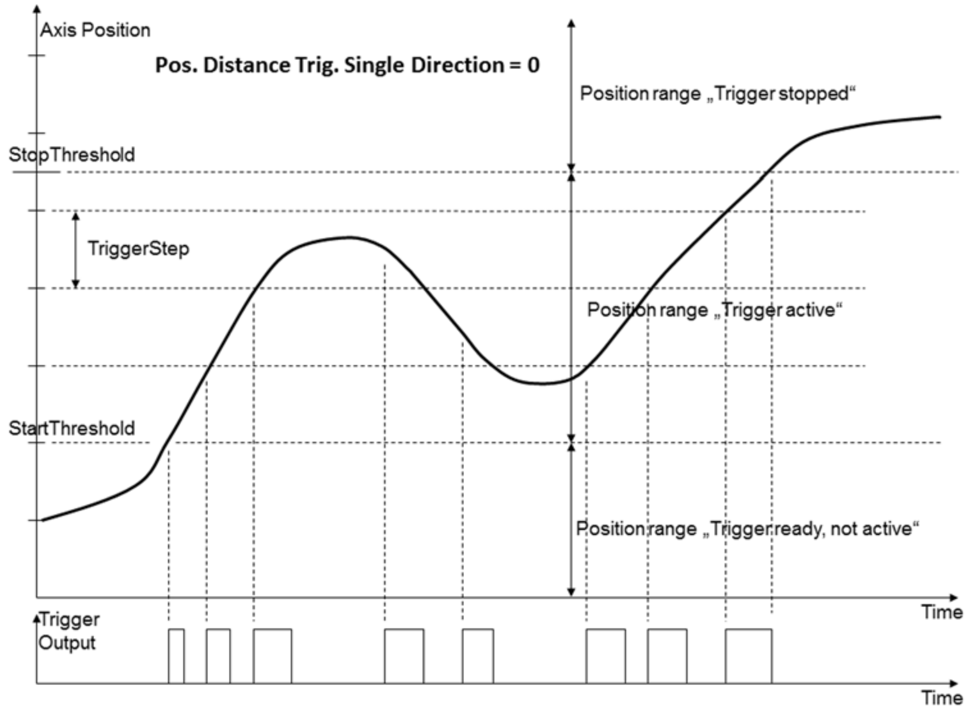


Figure 22: "Position Distance" Trigger Mode with threshold settings, trigger output continues when the motion direction is reversed

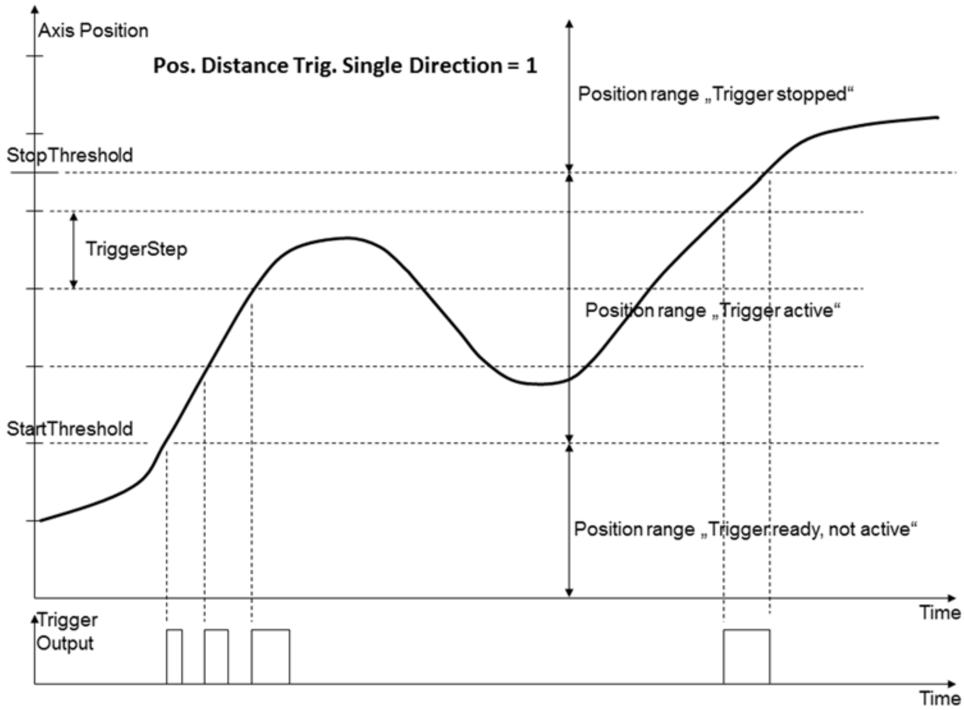


Figure 23: "Position Distance" Trigger Mode with threshold settings, trigger output only for positive direction of motion

Filter settings for start threshold range:

If the "Position Distance" trigger mode is selected and the start and stop thresholds are active (start threshold ≠ stop threshold), a filter can reduce the influence of noise around the start threshold. In combination with the ***Pos. Distance Trig. Single Direction*** parameter (ID 0x18000300) the filter setting can be used to avoid multiple triggers at the same position in cases where the noise is too high. Filter options:

- The ***Pos. Distance Trig. Filter Time*** parameter (0x18000301) specifies a time filter. The time filter is used to delay the trigger deactivation when the axis position falls below the start threshold due to noise (Figure 25). The time value is given in seconds. Recommended minimum value: period of the noise signal. Default value: 0 (= filter deactivated)
- The ***Pos. Distance Trig. Filter Level*** parameter (ID 0x18000302) specifies a position level filter. The position level filter is used to delay the trigger deactivation when the axis position falls below the start threshold due to noise (Figure 26). The filter level value is given in axis units. Recommended minimum value: amplitude of the noise signal. Default value: 0 (= filter deactivated)

Note: Although the activation of multiple triggers at the same position can be avoided by the filter setting, the precision of the position trigger is lost if the trigger step value is not at least 5 times higher than the peak-to-peak level of the sensor position noise.

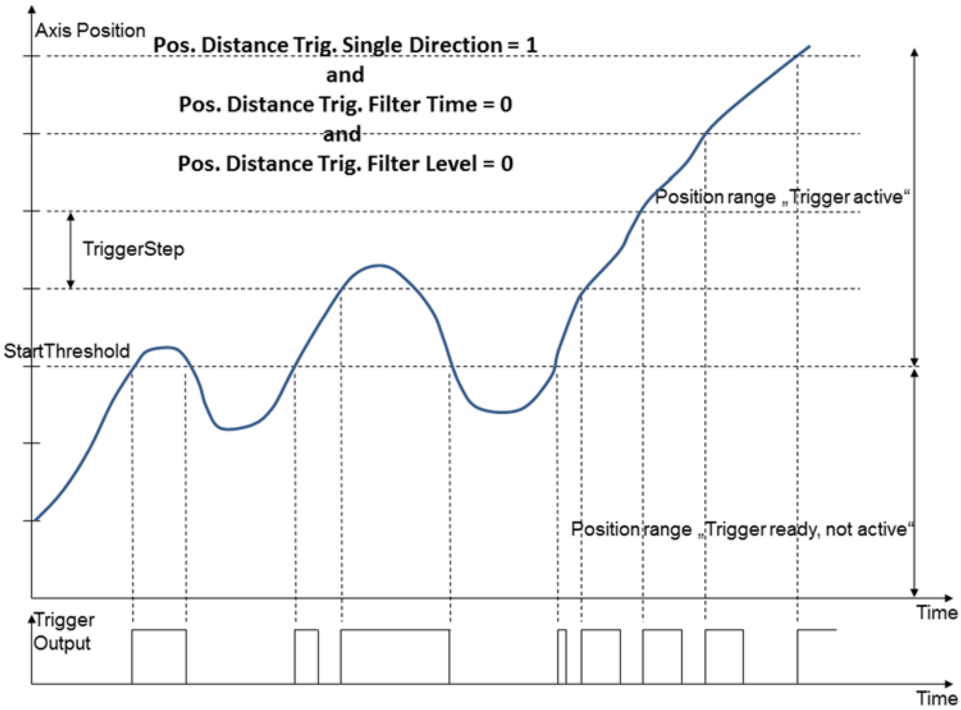


Figure 24: "Position Distance" Trigger Mode with threshold settings, without any filter settings

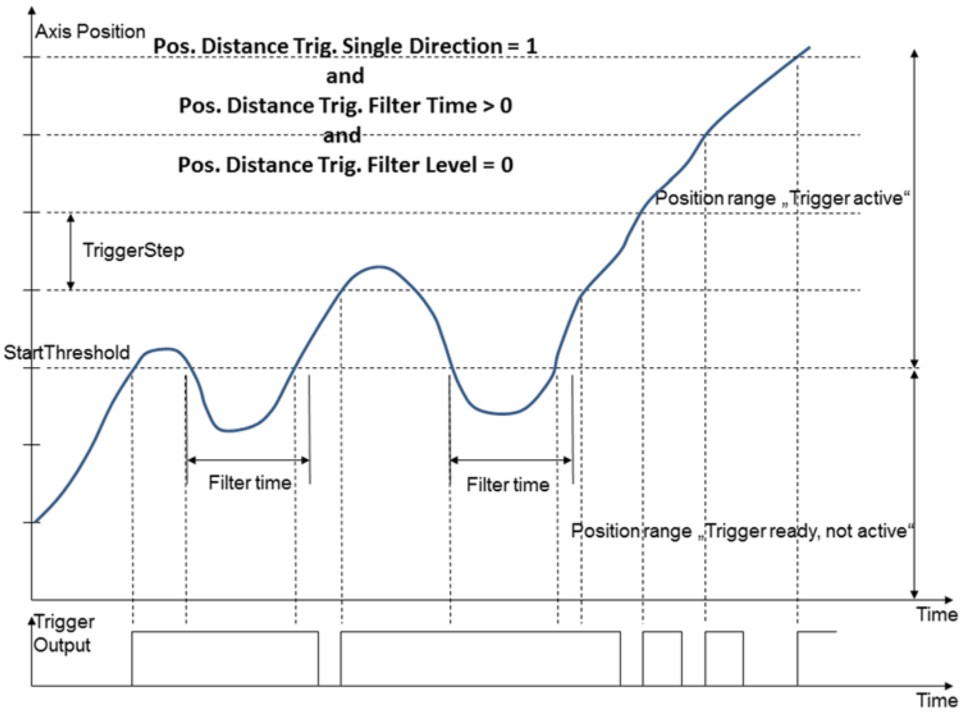


Figure 25: "Position Distance" Trigger Mode with threshold settings, with time filter

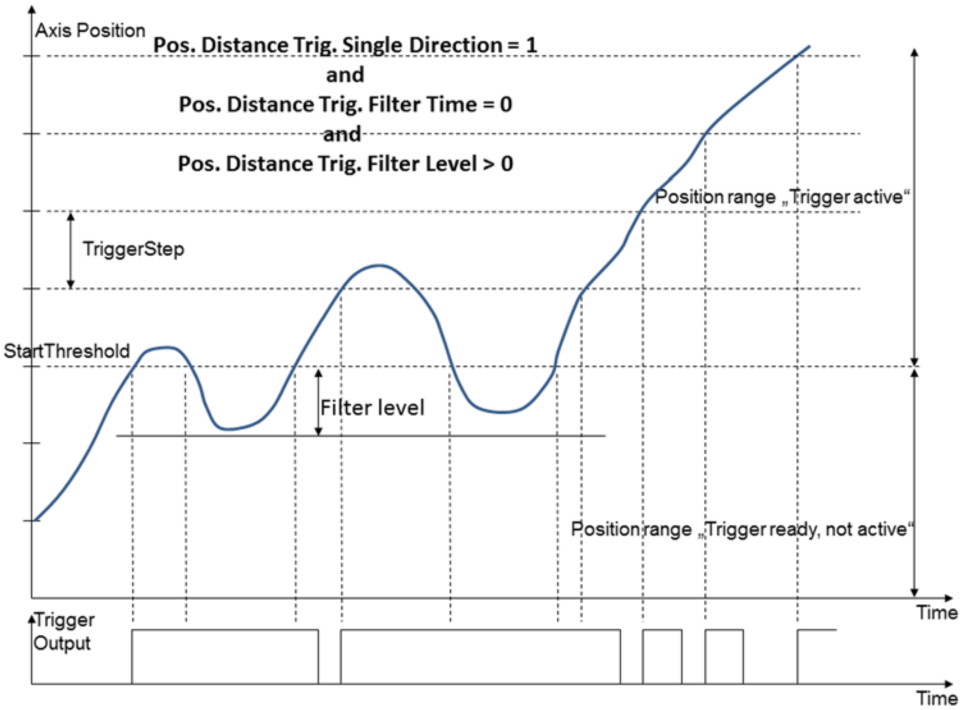


Figure 26: "Position Distance" Trigger Mode with threshold settings, with position level filter

Example—"On Target" Trigger Mode

With the "On Target" trigger mode, the on-target status of the selected axis is written to the selected trigger line. It is the same on-target status flag which can also be read by the ONT? command. The on-target status is influenced by two parameters: settling window (On Target Tolerance, ID 0x07000900) and settling time (Settling Time, ID 0x07000901). The on-target status is true when the current position is inside the settling window and stays there for at least the settling time. The settling window is centered around the target position.

The following parameters must be set for the digital output line which is to be used for trigger output (<TrigOutID>):

- Axis (<CTOPam> = 2)
- TriggerMode (<CTOPam> = 3)

General notation of the CTO command for this mode (in fact, the command arguments can be divided in two "portions", each starting with the <TrigOutID> declaration):

Command mnemonic	Axis selection	Trigger mode selection
CTO	<TrigOutID> 2 *Axis*	<TrigOutID> 3 2

Instead of using the CTO command, you can also set the values of the corresponding parameters with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

Example: The On-Target status flag of axis 1 is to be written to the digital output line 1. Send:

```
CTO 1 2 1 1 3 2
```

Example—"MinMax Threshold" Trigger Mode

With the "MinMax Threshold" trigger mode, a band is specified with MinThreshold and MaxThreshold (<CTOPam> IDs 5 and 6). When the axis position is inside the specified band then the trigger output line is set high, otherwise it is set low.

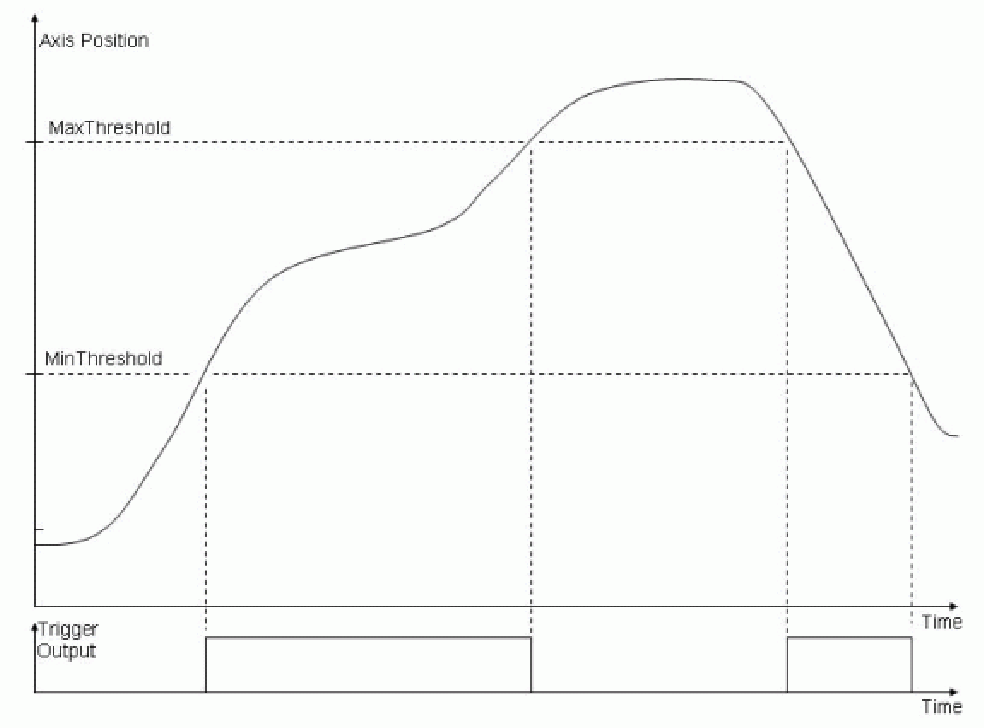


Figure 27: "MinMax Threshold" Trigger Mode

The following parameters must be set for the digital output line which is to be used for trigger output (<TrigOutID>):

- Axis (<CTOPam> = 2)
- TriggerMode (<CTOPam> = 3)
- MinThreshold (<CTOPam> = 5)
- MaxThreshold (<CTOPam> = 6)

General notation of the CTO command for this mode (in fact, the command arguments can be divided in four "portions", each starting with the <TrigOutID> declaration):

Command mnemonic	Axis selection	Trigger mode selection	Min threshold setting	Max threshold setting
CTO	<TrigOutID> 2 *Axis*	<TrigOutID> 3 3	<TrigOutID> 5 *min.pos.*	<TrigOutID> 6 *max.pos.*

Instead of using the CTO command, you can also set the values of the corresponding parameters with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

Example: The digital output line 1 is to be set high whenever the axis position of axis 1 is higher than 0.3 µm and lower than 0.6 µm. Send:

```
CTO 1 2 1 1 3 3 1 5 0.3 1 6 0.6
```

Example—"Generator Level Trigger" Mode

With the "Generator Level Trigger" mode, the trigger output will be synchronized with the wave generator output, and CTO must be used in combination with TWS.

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.

The following parameter must be set for the digital output line which is to be used for trigger output (<TrigOutID>):

- TriggerMode (<CTOPam> = 3)

General notation of the CTO command for this mode:

Command mnemonic	Trigger mode selection
CTO	<TrigOutID> 3 4

Instead of using the CTO command, you can also set the value of the corresponding parameter with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

Example 1: Generate single trigger pulses synchronized with the wave generator in Generator Level Trigger mode.

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for Wave Table 2, the segment length and hence the number of points in the wave table is 2000
TWC	Clears all output trigger settings related to the wave generator by switching the signal state for all points to "low" (the power-on default state is also "low"). It is recommended to use TWC before new trigger actions are defined.
TWS 1 500 1 1 1500 1 1 1900 1 1 2000 1	Set trigger actions for the digital output line OUT1 (identifier is 1): at the waveform points 500, 1500, 1900 and 2000 it is set high; at all other points the state of the line is low (due to the TWC usage).
CTO 1 3 4	The digital output line OUT1 is set to "Generator Level Trigger" mode.
WSL 1 2	Connect Wave Generator 1 (Axis 1) to Wave Table 2
WGO 1 1	Start output of Wave Generator 1 immediately (synchronized by servo cycle). Now the trigger output action will take place as specified.
WGO 1 0	Stop output of Wave Generator 1 and hence also the trigger output.

Example 2: Use Generator Level Trigger mode to switch the digital output line to a certain level for a certain range of the waveform.

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for Wave Table 2, the segment length and hence the number of points in the wave table is 2000
CTO 1 3 4	The digital output line OUT1 is set to "Generator Level Trigger" mode.
TWS 1 1 3 1 750 2 1 1150 3	For all waveform points from point 1 to point 749, the output line is set low. At point 750, there is a rising edge on the output line. Therefore, the output line is set high from point 751 to point 1149. At point 1150, there is a falling edge on the output line, and for all subsequent points the line will therefore be set low.
WSL 1 2	Connect Wave Generator 1 (Axis 1) to Wave Table 2
WGO 1 1	Start output of Wave Generator 1 immediately (synchronized by servo cycle). Now the trigger output action will take place as specified.
WGO 1 0	Stop output of Wave Generator 1 and hence also the trigger output.

Example—"Generator Pulse Trigger" Mode

With the "Generator Pulse Trigger" mode, the trigger output will be synchronized with the wave generator output, and CTO must be used in combination with TWS.

The length of 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, for example, to count the waveform points that are output by the wave generator.

The following parameter must be set for the digital output line which is to be used for trigger output (<TrigOutID>):

- TriggerMode (<CTOPam> = 3)

General notation of the CTO command for this mode:

Command mnemonic	Trigger mode selection
CTO	<TrigOutID> 3 9

Instead of using the CTO command, you can also set the value of the corresponding parameter with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

The examples given for the “Generator Level Trigger” mode (p. 82) can also be used in “Generator Pulse Trigger” mode. Note: If example 2 is used in “Generator Pulse Trigger” mode, the digital output line will toggle between high and low with each waveform point from point 751 to point 1149.

Example—Polarity Setting

It is possible to select the signal polarity (active high = 1, default / active low = 0) for the digital output line which is to be used for trigger output.

The following parameter must be set for the digital output line (<TrigOutID>):

- Polarity (<CTOPam> = 7)

General notation of the CTO command for polarity selection:

Command mnemonic	Trigger mode selection
CTO	<TrigOutID> 7 *pol.code*

Instead of using the CTO command, you can also set the value of the corresponding parameter with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

Example: The signal polarity for the digital output line 1 is to be set to "active low". Send:

```
CTO 1 7 0
```

Examples—"TriggerOutAND" and "TriggerOutOR" Modes

With the "TriggerOutAND" and "TriggerOutOR" trigger modes, the signal states of multiple digital output lines are combined by a logical operation, and the result is written to another trigger line. A bit-mapped mask selects the digital output lines whose signal states are to be logically combined. The mask can be specified in hex or decimal format.

Bit-mapped mask values of the digital output lines:

Digital output line	Bit	Value in hex format	Value in decimal format
Digital Out 1	0	0x1	1
Digital Out 2	1	0x2	2
Digital Out 3	2	0x4	4

The following parameters must be set for the digital output line which is to be used for trigger output (<TrigOutID>):

- TriggerMode (<CTOPam> = 3)
- TriggerOutMask (<CTOPam> = 16)

General notation of the CTO command for the TriggerOutAND mode:

Command mnemonic	Trigger mode selection	Axis selection
CTO	<TrigOutID> 3 14	<TrigOutID> 16 *TriggerOutMask*

General notation of the CTO command for the TriggerOutOR mode:

Command mnemonic	Trigger mode selection	Axis selection
CTO	<TrigOutID> 3 15	<TrigOutID> 16 *TriggerOutMask*

Instead of using the CTO command, you can also set the values of the corresponding parameters with SPA or SEP, see “Configuring Trigger Output” (p. 69) for a parameter list.

Example 1: The signal states of digital output lines 1 and 3 are to be combined by a logical AND operation. The result is to be written to the digital output line 2. The mask for selection of the digital output lines to be combined is specified in hex format. Send:

```
CTO 2 3 14 2 16 0x5
```

Example 2: The signal states of digital output lines 2 and 3 are to be combined by a logical OR operation. The result is to be written to the digital output line 1. The mask for selection of the digital output lines to be combined is specified in decimal format. Send:

```
CTO 1 3 15 1 16 6
```

Using the Analog Input

How to Work with the Analog Input - Overview

With models E-727.xxxA and .xxxAx, four analog inputs are available on the **Analog I/O** socket (p. 228). In the firmware of the E-727, the analog input lines are represented by the input signal channels 4 to 7.

The voltage range of an analog input (±5 V or ±10 V) can be configured via the value of the corresponding ***Sensor Range Factor*** parameter (ID 0x02000100).

You can use an analog input line as follows:

- Connect an external sensor
- Connect a source for control value generation

INFORMATION

E-727.3SDA, .3SDAx, .3RDA and .3RDAx models only:

Input signal channel 4 can also be used for sensor input on the socket for piezo stages (p. 226). The corresponding configuration is done via the value of the ***Sensor Range Factor*** parameter (ID 0x02000100) as follows:

- 1: Use with a piezoresistive or strain gauge sensor (input via pins 6, 24, 25)
- 2: Use with a PT1000 temperature sensor (input via pins 1, 20)

For further details, see the description of the input signal channels in „Axes, Channels, Functional Elements“ (p. 22).

Irrespective of the intended usage, the analog input values must first be scaled to suitable position values (see "Scaling the Analog Input" (p. 87)). Then, to set the usage of the analog input, it is furthermore necessary to change certain controller parameters. See "Use as Control Value Generation Source" (p. 90) or "Use as External Sensor Input" (p. 91) for details. Analog input lines which are not used should be deactivated to avoid interferences, see "Deactivation of Unused Analog Input Lines" (p. 92) for details.

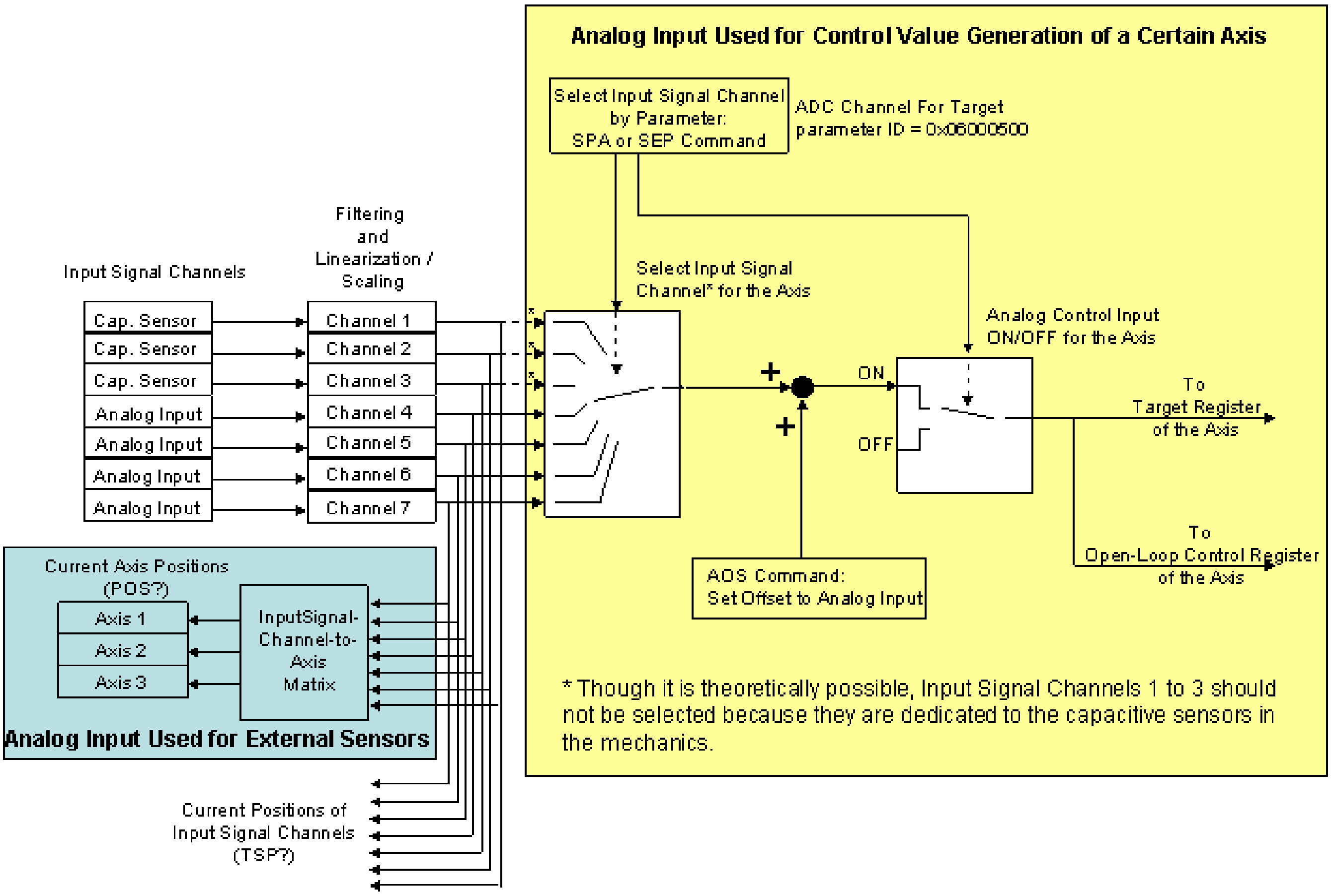


Figure 28: Overview over the usage of the analog input lines, exemplified by E-727.3CDA

INFORMATION

Changing the parameter values for configuration of the analog input requires command level 1. Switch to command level 1 as follows:

- In a terminal program: Send `CCL 1 advanced`
- When prompted to enter a password in any PC software, e.g. in PIMikroMove: Enter `advanced`.

If you want to work in the ***Device Parameter Configuration*** window of PIMikroMove (for an example, see p. 53):

- 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.

Scaling the Analog Input

Before the analog input line can be used with an external sensor or with a control-signal source, the input levels must be associated with suitable position values. To do this, adjust the OFFSET (parameter ID 0x02000200) and the GAIN (parameter ID 0x02000300) of the Mechanics linearization polynomial according to the travel range of the axis and the input signal range. See below for details. The TSP? command reports the analog input values after the scaling as position values in µm.

In addition, the digital filter parameters can be adjusted. See "Digital processing" (p. 27) for details.

How to adjust OFFSET and GAIN to map the analog input voltage to a suitably scaled position value for a certain axis:

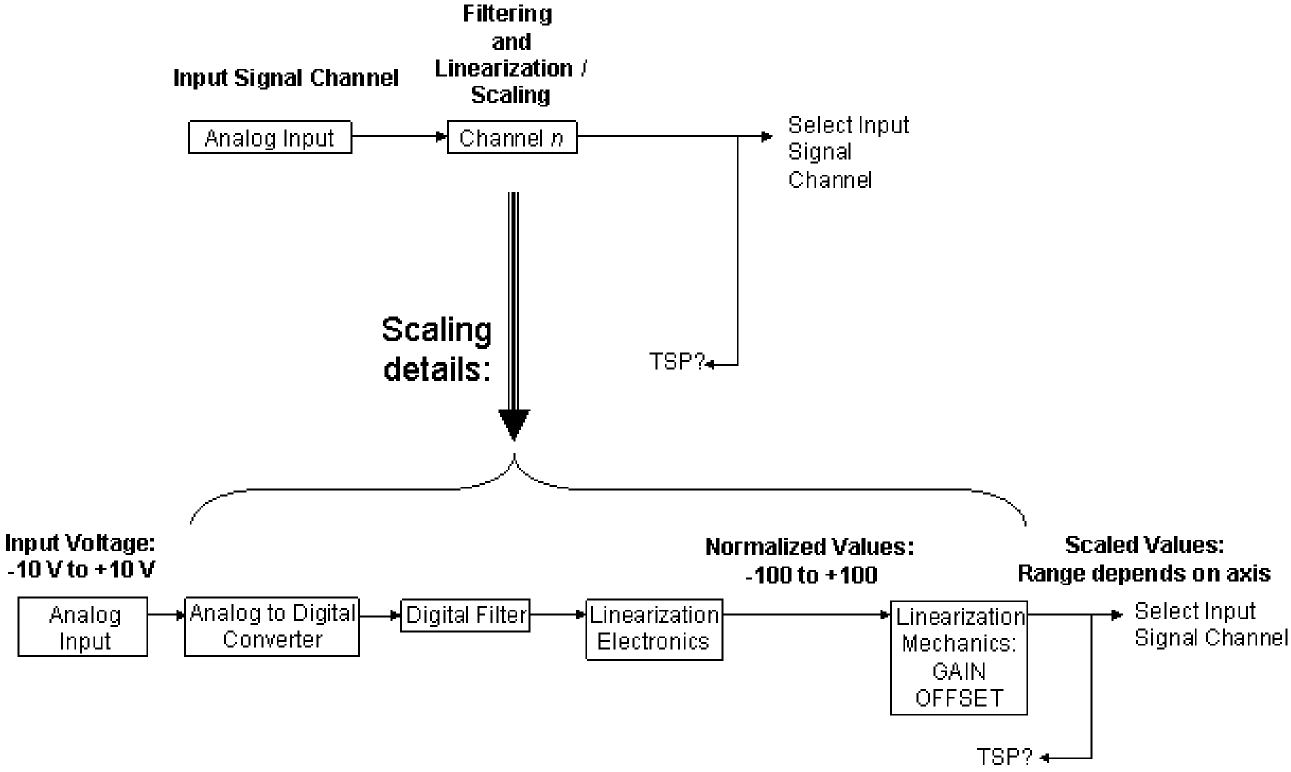


Figure 29: Processing of an analog input signal, detail from the overview figure above

Input Voltage:
The range is -10 to +10 V or -5 to +5V, depending on the value of the ***Sensor Range Factor*** parameter (ID 0x02000100).

Normalized Value:
The polynomial used for electronics linearization (see "Digital processing" for details) converts the analog input voltage to a number in the range of -100 to +100. The minimum input voltage value corresponds to -100, and the maximum input voltage corresponds to +100 respectively. If, for example, an input range of -10 to +10 V is selected, then -10 V correspond to -100, and +10 V correspond to +100.

Scaled Value:
The range depends on the axis and can be set by the coefficients of the polynomial used for Mechanics linearization (see "Digital processing" for details):

ScaledValue = OFFSET + GAIN * NormalizedValue

where
OFFSET corresponds to the ***Sensor Mech. Correction 1*** parameter, ID 0x02000200
GAIN corresponds to the ***Sensor Mech. Correction 2*** parameter, ID 0x02000300

If no linearization is necessary, the other coefficients of the Mechanics linearization polynomial can be set to zero (parameter IDs 0x02000400, 0x02000500, 0x02000600).

Note that in PIMikroMove, these parameters are available in the *Sensor Mechanics 4* to *Sensor Mechanics 7* parameter groups in the ***Device Parameter Configuration*** window.

How to calculate the values to set for OFFSET and GAIN:

GAIN = (MaxScaledValue - MinScaledValue) / (MaxNormalizedValue - MinNormalizedValue)

OFFSET = MaxScaledValue - GAIN * MaxNormalizedValue

The values of "MinScaledValue" and "MaxScaledValue" depend on the travel range of the axis with which the analog input line is to be used:

"MinScaledValue" is given by the TMN? answer (is defined by the ***Range Limit min*** parameter, ID = 0x07000000), and "MaxScaledValue" is given by the TMX? answer (is defined by the ***Range Limit max*** parameter, ID = 0x07000001).

The values of "MinNormalizedValue" and "MaxNormalizedValue" depend on the range of the external signal applied to the analog input line. See the examples below.

For all examples, the following is assumed:

- Input signal channel 4 (pins 2 and 9 of **Analog I/O**) is to be used with axis 1.
- Via the ***Sensor Range Factor*** parameter, a maximum input range of -10 to +10 V is selected for input signal channel 4.
- Axis 1 has the following travel range:
  MinScaledValue = -20 µm
  MaxScaledValue = +120 µm

Example 1:

The full range of -10 V to +10 V is to be used (this is recommended for highest resolution).

MinNormalizedValue = -100

MaxNormalizedValue = +100

GAIN = (120 - (-20)) / (100 - (-100)) = 0.7

OFFSET = 120 - 0.7 * 100 = 50

ScaledValue = 50 + 0.7 * NormalizedValue

So you have to send

```
SPA 4 0x02000200 50
SPA 4 0x02000300 0.7
```

to adjust the GAIN and OFFSET parameters for input signal channel 4.

Example 2:

Only positive input voltages are to be used, i.e. the range is 0 V to +10 V.

MinNormalizedValue = 0

MaxNormalizedValue = +100

GAIN = (120 - (-20)) / (100 - 0) = 1.4

OFFSET = 120 - 1.4 * 100 = -20

ScaledValue = -20 + 1.4 * NormalizedValue

Send:

```
SPA 4 0x02000200 -20
SPA 4 0x02000300 1.4
```

Example 3:

Positions with positive sign shall correspond to positive input voltages, and positions with negative sign shall correspond to negative input voltages.

The positive input voltage ranges to +10 V.

Then, the following is valid provided that the absolute value of the negative positions will never be greater than the positive positions.

MinNormalizedValue = 0

MaxNormalizedValue = +100

GAIN = (120 - 0)) / (100 - 0) = 1.2

OFFSET = 120 - 1.2 * 100 = 0

ScaledValue = 1.2 * NormalizedValue

Send:

```
SPA 4 0x02000200 0
SPA 4 0x02000300 1.2
```

Note that these OFFSET and GAIN values would also be valid if axis 1 had a travel range of 0 to 120 µm and if there were only positive input voltages to +10 V.

Example 4:

The same conditions as in example 3 are valid, but the positive input voltages range to +5 V.

MinNormalizedValue = 0

MaxNormalizedValue = +50

GAIN = (120 - 0)) / (50 - 0) = 2.4

OFFSET = 120 - 2.4 * 50 = 0

ScaledValue = 2.4 * NormalizedValue

Send:

```
SPA 4 0x02000200 0
SPA 4 0x02000300 2.4
```

Use as Control Value Generation Source

To enable the analog control input for an axis, an input signal channel must be connected to that axis. This is done with the ***ADC Channel For Target*** parameter (ID 0x06000500). If the connection of axis and input signal channel is saved as the power-on default, the axis can be commanded via analog input immediately after controller start-up, and no host PC is required. Example: With an E-727.3CDA, the input signal channel 4 (pins 2 and 9 of **Analog I/O**) is to be used to command axis 1. Send:

```
SPA 1 0x06000500 4
```

to enable the connection in volatile memory. Note that in PIMikroMove, this parameter is available in the *Target Manipulation 1* parameter group in the ***Device Parameter Configuration*** window.

When the analog control input is enabled for an axis, then it overwrites the values of all other control sources for that axis except those from the AutoZero procedure. The AutoZero procedure has the highest priority, i.e. it will overwrite the control values given by all other sources. When the analog control input is enabled, it will be disabled automatically at the start of the AutoZero procedure and reenabled again when AutoZero is finished. See "Control Value Generation" (p. 30) for more information.

An offset value can be added to the analog input scaled value for an axis using the AOS command.

When no input signal channel is connected to an axis (i.e. the value of the ***ADC Channel For Target*** parameter is 0), the analog control input is disabled for that axis (including the offset set with AOS).

When the analog input is used as control source and the axis motion is stopped with STP or #24, the behaviour depends on the value of the ***Disconnect Analog Target Input When Stopping*** parameter (ID 0x0E001E00): 1 = the analog input channel is disconnected from the axis; 0 = the analog input channel remains connected to the axis. If the analog input channel is disconnected from the axis: To recommence commanding the axis via the analog input, the corresponding input signal channel must be reconnected to the axis. See the description above.

When the analog input is being used as control source and the servo mode is switched off, the axis motion will continue in open-loop mode.

INFORMATION

The analog input values must be scaled to suitable position values. See "Scaling the Analog Input" (p. 87) for more information.

Make sure that the analog input line which is used to control an axis is not used as external sensor for the same axis. This means that in the input matrix, the coefficient of the appropriate analog input line must be set to zero for that axis.

The coefficients of the analog input lines are represented by the values of the ***Position From Sensor 4*** to ***Position From Sensor 7*** parameters (IDs 0x07000503 to 0x07000506). In PIMikroMove, these parameters are available in the *Axis Definition* parameter groups in the ***Device Parameter Configuration*** window.

Use as External Sensor Input

To let the sensor on the analog input line participate in the position signal of an axis, set the corresponding coefficient in the input matrix to 1 for that axis. Example: With an E-727.3CDA, an external sensor is connected to input signal channel 4 (pins 2 and 9 of **Analog I/O**). This sensor is to be used to measure the position of axis 1, i.e. the ***Position From Sensor 4*** parameter (ID 0x07000503) must be set to 1 for axis 1. Send:

```
SPA 1 0x07000503 1
```

to change the coefficient in volatile memory. In PIMikroMove, this parameter is available in the *Axis Definition 1* parameter group in the ***Device Parameter Configuration*** window.

If only the external sensor on the analog input line is to be used for position measurement of an axis, the signals of all other sensors must be excluded from the position monitoring of that axis, especially the signals of the internal sensors integrated in the mechanics. To do this, set the corresponding coefficients in the input matrix to zero for that axis (with the E-727.3CDA, the internal sensors are represented by the ***Position from Sensor 1*** to ***Position from Sensor 3*** parameters). Example: To deactivate the first internal sensor for axis 1, send:

```
SPA 1 0x07000500 0
```

The position of the axis (i.e. the POS? response) will then be based on the external sensor only, but it is still possible to read the signals of all sensors using the TSP? command.

INFORMATION

The analog input values must be scaled to suitable position values. See "Scaling the Analog Input" (p. 87) for more information.

Make sure that internally, the analog input line used to monitor the position of an axis is not connected to the same axis for control value generation. This means that the value of the ***ADC Channel For Target parameter*** (ID 0x06000500) for an axis must be different from the identifier of the analog input line which is used as external sensor for that axis. In PIMikroMove, you can check this in the *Target Manipulation* parameter groups in the ***Device Parameter Configuration*** window.

Deactivation of Unused Analog Input Lines

Analog input lines which are not used should be deactivated to avoid interferences. To deactivate an unused analog input line, the following settings must be done:

Exclude the analog input line from the calculation of axis positions by setting its coefficients for all axes to zero in the input matrix (***Position from Sensor n*** parameters).

Make sure that the analog input line is not connected to an axis for control value generation. This means that the value of the ***ADC Channel For Target*** parameter, ID 0x06000500, for all axes must be different from the ID of the analog input line.

Example: The input signal channel 7 (pins 5 and 12 of **Analog I/O**) of an E-727.3CDA model is to be deactivated temporarily (i.e. in volatile memory). The input matrix coefficients of that channel correspond to the ***Position from Sensor 7*** parameters (ID 0x07000506) for axis 1 to axis 3.

Send:

```
SPA 1 0x07000506 0 2 0x07000506 0 3 0x07000506 0
```

to exclude the analog input line from the axis position calculation. Then send:

```
SPA? 1 0x06000500 2 0x06000500 3 0x06000500
```

to check the RAM settings for the axis control value generation. The response must be different from 7 for all axes, i.e. if the E-727 replies

```
1 0x06000500=7
2 0x06000500=3
3 0x06000500=0
```

then input signal channel 7 is still connected to axis 1, and you have to send

```
SPA 1 0x06000500 0
```

to disconnect it.

Analog-Input-Related Commands and Parameters

Command	Description	Notes
AOS	Set Analog Input Offset	Adds an offset value to the analog input scaled value (***Analog Target Offset***, ID 0x06000501). This offset is active as long as the analog input is enabled as control source for this axis.
AOS?	Get Analog Input Offset	Reads the current value of ***Analog Target Offset***, parameter ID 0x06000501, from volatile memory
SEP	Set Nonvolatile Memory Parameters	Can be used to set the power-on default configuration for analog input usage.
SEP?	Get Nonvolatile Memory Parameters	Reads the current parameter values from nonvolatile memory
SPA	Set Temporary Memory Parameters	Can be used to set a temporary configuration for analog input usage.
SPA?	Get Temporary Memory Parameters	Reads the current parameter values from volatile memory (RAM)

Command	Description	Notes
TAD?	Get ADC Value Of Input Signal	Reports the current ADC value of the analog input, dimensionless
TNS?	Get Normalized Input Signal Value	Reports the resulting value for the analog input after the electronics linearization, dimensionless
TSP?	Get Input Signal Position Value	Reports the resulting value for the analog input after the mechanics linearization (scaling), the unit is µm
WPA	Save Parameters To Nonvolatile Memory	Can be used to save the currently active configuration (including analog input usage) to nonvolatile memory, where it becomes the power-on default.

See "How to work with the Analog Input - Overview" (p. 85) for more information. For detailed command descriptions see the GCS commands manual PZ281E. For the identifiers of the items which can be addressed with the commands see " Axes, Channels, Functional Elements " (p. 22).

See " Parameters" (p. 185) for more information regarding the controller parameters and their handling.

Parameter ID	CCL for write access	Item Type Concer-ned	Max. No. of Items	Data Type	Parameter Description
0x02000200	1	Input Signal Channel	7	FLOAT	Sensor Mech. Correction 1 (Offset)
0x02000300	1	Input Signal Channel	7	FLOAT	Sensor Mech. Correction 2 (Gain)
0x05000000	1	Input Signal Channel	7	INT	Digital Filter Type
0x05000001	1	Input Signal Channel	7	FLOAT	Digital Filter Bandwidth
0x05000002	1	Input Signal Channel	7	INT	Digital Filter Order
0x06000500	1	Logical Axis	3	INT	ADC Channel for Target; if 0, then the analog control input is disabled for the axis
0x06000501	1	Logical Axis	3	FLOAT	Analog Target Offset
0x07000500	1	Logical Axis	3	FLOAT	Position from Sensor 1
0x07000501	1	Logical Axis	3	FLOAT	Position from Sensor 2
0x07000502	1	Logical Axis	3	FLOAT	Position from Sensor 3
0x07000503	1	Logical Axis	3	FLOAT	Position from Sensor 4
0x07000504	1	Logical Axis	3	FLOAT	Position from Sensor 5
0x07000505	1	Logical Axis	3	FLOAT	Position from Sensor 6
0x07000506	1	Logical Axis	3	FLOAT	Position from Sensor 7
0x0e000b00	3	System	1	INT	Number of input channels
0x0e000b03	3	System	1	INT	Number of sensor channels

Using the Analog Output

How to Work with the Analog Output - Overview

With models E-727.xxxA and .xxxAx, one analog output is available on pin 8 of the **Analog I/O** socket (p. 228). 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. 225 or p. 226).

Output signal channel 4 can therefore be configured via parameters ***Select Output Type*** (ID 0x0a000003) and ***Select Output Index*** (ID 0x0a000004).

Via the value of the ***Select Output Type*** parameter (ID 0x0A000003), you can select the purpose for which the analog output is to be used:

- 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. For an example, see p. 95.
- 5: Control signal for an external amplifier, 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 already scaled to be in the range of -10 to 10 V. For an example, see p. 96.

Via the value of the ***Select Output Index*** parameter, you can connect the analog output with the axis or channel whose signal is to be output:

- If ***Select Output Type*** has the value 1 or 5, the value of ***Select Output Index*** gives the identifier of the output signal channel whose control value is to be used.
- If ***Select Output Type*** has the value 2, the value of ***Select Output Index*** gives the identifier of the axis whose position is to be output.

INFORMATION

If the piezo stage(s) delivered with the E-727 do(es) not use the 4th amplifier, parameters ***Select Output Type*** and ***Select Output Index*** are preset so that the position of axis 1 is 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 is configured for use as output voltage (***Select Output Type*** has value 1, ***Select Output Index*** has value 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. 234).

INFORMATION

To achieve the highest possible resolution and eliminate potential interference that affects the cable used:

- Filter the analog signal in a suitable way, e.g., before you convert it to a digital format. Recommended: low-pass filter with max. 100 kHz cut-off frequency (characteristics: single pole, 6 dB/octave)

INFORMATION

Changing the parameter values for configuration of the analog output requires command level 1. Switch to command level 1 as follows:

- In a terminal program: Send `CCL 1 advanced`
- When prompted to enter a password in any PC software, e.g. in PIMikroMove: Enter `advanced`.

If you want to work in the ***Device Parameter Configuration*** window of PIMikroMove (for an example, see p. 53):

- 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.

For further details, see the description of the output signal channels in „Axes, Channels, Functional Elements“ (p. 20) and the information in „E-727.xxxA, E-727.xxxAx: Analog I/O“ (p. 228).

Use as Position Monitor

Example: The position of axis 3 is to be monitored via the analog output on pin 8 of the **Analog I/O** socket (= output signal channel 4).

1. Select output type 2 = "position monitor of an axis" for the analog output line using the ***Select Output Type*** parameter:

   If you work in a terminal program, send the following command to change the parameter value in volatile memory:

   ```
   SPA 4 0x0A000003 2
   ```

   In PIMikroMove, this parameter is available in the ***DAC 4*** parameter group in the ***Device Parameter Configuration*** window.

2. Connect the axis (axis identifier is 3) to the analog output line using the ***Select Output Index*** parameter:

   If you work in a terminal program, send the following command to change the parameter value in volatile memory:

   ```
   SPA 4 0x0A000004 3
   ```

   In PIMikroMove, this parameter is also available in the ***DAC 4*** parameter group in the ***Device Parameter Configuration*** window.

3. Scale the output value, i.e. associate the axis position values 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 axis.

   ScaledPositionValue = PositionReportScaling * (PositionReportOffset + PositionValue)

   Example:

   The position range of the axis is given by the TMN? answer (is defined by the ***Range Limit min*** parameter, ID = 0x07000000) and by the TMX? answer (is defined by the ***Range Limit max*** parameter, ID = 0x07000001), it is -20 µm to +120 µm in the example. Furthermore, the output range to be used is -10 V to +10 V. The resulting parameter values for the axis position scaling are as follows:

   Position Report Scaling = 0.143

   Position Report Offset = -50

   i.e. you have to send:

   ```
   SPA 3 0x07001005 0.143
   ```

   ```
   SPA 3 0x07001006 -50
   ```

   In PIMikroMove, these parameters for the axis are available in the ***Servo 3*** parameter group in the ***Device Parameter Configuration*** window.

Use as Control Signal

INFORMATION

The control value of an output signal channel results from the output matrix, see "Output Generation" (p. 34) for more information.

Example: An external amplifier is to be controlled via the analog output on pin 8 of the **Analog I/O** socket (= output signal channel 4). To command the external amplifier, the logical axis 3 of the E-727 is to be used. Via the output matrix, the control value of axis 3 is to be transferred into the control voltage for output signal channel 4.

1. Select output type 5 = “control signal for an external amplifier” for the analog output line using the ***Select Output Type*** parameter:

   If you work in a terminal program, send the following command to change the parameter value in volatile memory:

   ```
   SPA 4 0x0A000003 5
   ```

   In PIMikroMove, this parameter is available in the ***DAC 4*** parameter group in the ***Device Parameter Configuration*** window, and the ***ANA_DRV*** entry corresponds to parameter value 5.

2. Connect output signal channel 4 (i.e. the channel whose control voltage is to be output) to the analog output line using the ***Select Output Index*** parameter.

   If you work in a terminal program, send the following command to change the parameter value in volatile memory:

   ```
   SPA 4 0x0A000004 4
   ```

   In PIMikroMove, this parameter is also available in the ***DAC 4*** parameter group in the ***Device Parameter Configuration*** window.

3. For the logical axis 3, set the output-matrix coefficient of output signal channel 4 to a suitable value (***Driving Factor of Piezo 4*** parameter, ID 0x09000003; in PIMikroMove available in the ***Axis Definition 3*** parameter group in the ***Device Parameter Configuration*** window).

Analog-Output-Related Commands and Parameters

Command	Description	Notes
SEP	Set Nonvolatile Memory Parameters	Can be used to set the power-on default configuration for analog output usage.
SEP?	Get Nonvolatile Memory Parameters	Reads the current parameter values from nonvolatile memory
SPA	Set Temporary Memory Parameters	Can be used to set a temporary configuration for analog output usage.
SPA?	Get Temporary Memory Parameters	Reads the current parameter values from volatile memory (RAM)
VOL?	Get Voltage Of Output Signal Channel	Reads output voltage value of the given output signal channel; relevant if the output type is set to "control signal for an external amplifier"
WPA	Save Parameters To Nonvolatile Memory	Can be used to save the currently active configuration (including analog output usage) to nonvolatile memory, where it becomes the power-on default.

See "How to work with the Analog Output - Overview" (p. 94) for more information. For detailed command descriptions see the GCS commands manual PZ281E. For the identifiers of the items which can be addressed with the commands see " Axes, Channels, Functional Elements " (p. 22).

Parameter ID	CCL for write access	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x07001005	1	Logical Axis	3	FLOAT	Position Report Scaling, required if the axis position is to be output (output type = 2)
0x07001006	1	Logical Axis	3	FLOAT	Position Report Offset, required if the axis position is to be output (output type = 2)
0x09000000 to 0x09000003	1	Logical Axis	3	FLOAT	Driving Factor of Piezo 1 to Driving Factor of Piezo 4, give the output matrix
0x0A000003	1	Output Signal Channel	4	INT	Select Output Type; 1 = output voltage for a piezo actuator in the stage 2 = position monitor of an axis 5 = control signal for external amplifier
0x0A000004	1	Output Signal Channel	4	INT	Select Output Index; the selected object can be an axis or an output signal channel (depends on the selected output type)
0x0e000b01	3	System	1	INT	Number of output channels
0x0e000b04	3	System	1	INT	Number of piezo channels

See "Parameters" (p. 185) for more information regarding the controller parameters and their handling.

Wave Generator

How to Work with the Wave Generator

The following subsections describe the wave generator handling in detail. See also "Wave Generator Examples" (p. 104).

Basic Data

The number of wave tables can be queried using the SPA? command, parameter ID 0x1300010A. The E-727 has 40 wave tables for creating and (temporarily) storing arbitrary waveforms (identifiers are 1 to 40).

To ask for the number of wave generators, use the TWG? command. The assignment of wave generators and axes to each other is fixed: wave generator 1 is connected to axis 1, wave generator 2 to axis 2, ..., wave generator n to axis n.

The available wave tables can be flexibly assigned to the wave generators and hence to the axes using the WSL command. A wave table can be used by multiple wave generators at the same time.

A certain amount of the controllers memory space is reserved for the waveform data (ask with the SPA? command, parameter ID 0x13000004). The E-727 provides 262144 data points for waveform definition. This memory space is (temporarily) allocated to the individual wave tables during the waveform definition.

Basic Operation

1. Define the waveform segment-by-segment using the WAV command. The waveform will be written to the selected wave table.
2. Connect the wave generator to the wave table using the WSL command.
3. Start the wave generator output and hence the motion of the axis using the WGO command. You can choose several start options (e.g. start/stop by external trigger, initialization/usage of the Dynamic Digital Linearization (DDL) feature; see the description of the WGO command and "Dynamic Digital Linearization (DDL)" (p. 114) for more information).
   When starting the wave generator, data recording is started automatically.
4. Stop the wave generator output with WGO or #24 or STP.

A simple example for your first steps (using the command entry facilities of PIMikroMove or PITerminal):

Command String to Send	**Action Performed**
WAV 4 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for Wave Table 4; see WAV description for details
WSL 1 4	Connect the Wave Generator 1 (axis 1) to Wave Table 4
WGO 1 1	Start output of Wave Generator 1 immediately (synchronized by servo cycles)
WGO 1 0	Stop output of Wave Generator 1

Additional Steps and Settings

You can calculate the memory space remaining if you ask with WAV? for the current wave table length. To release memory space, delete the content of selected wave tables with the WCL command.

After you send the waveform definition to the wave table (with WAV), it is always a good idea to check it by reading back the waveform sequence from the controller before actually outputting it. This can be done by the GWD? command. Note that the response to GWD? does not contain any offset set with WOS to the wave generator output.

You can add an offset to the output of a wave generator using the WOS command. Thereafter, the output of the specified wave generator is the sum of the offset value and the wave value:
Generator Output = Offset + Current Wave Value
If the wave generator is started with the option "start at the endpoint of the last cycle", the E-727 at the end of each output cycle equates the WOS offset value with the current generator output.
WOS sets the value of the ***Wave Offset*** parameter, ID 0x1300010b, in volatile memory. You can also change this parameter with SPA or SEP and save the value to nonvolatile memory with WPA (switch to command level 1 before with the CCL command).
Deleting wave table content with WCL has no effect on the WOS settings.

For triggering purposes, the wave generator output can be coupled with the digital output lines OUT1 to OUT3 of the controller (see "Digital I/O Socket" (p. 227)). You should first set the signal state of the output lines to "low" for all waveform points ("low" is also the power-on default). This can be done with TWS or TWC. Then use the TWS command to define the trigger actions (high/low level, rising/falling edge) for selected output lines at selected waveform points. To activate the Generator Trigger mode for the selected output lines, use the CTO command. Examples see p. 109.

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

You can limit the duration of the wave generator output by setting the number of output cycles with WGC. The waveform itself remains unchanged.

Using the WTR command, you can lengthen the individual output cycles of the waveform. The duration of one output cycle for the waveform can be calculated as follows:
Output Duration = Servo Update Time * WTR value * Number of Points
where
Servo Update Time is given in seconds by parameter 0x0E000200
WTR value gives the number of servo cycles the output of a waveform point lasts, default is 1
Number of Points is the length of the waveform (i.e. the length of the wave table)

WTR sets the value of the ***Wave Generator Table Rate*** parameter, ID 0x13000109, in volatile memory. You can change this parameter also with SPA or SEP and save the value to nonvolatile memory with WPA (switch to command level 1 before with the CCL command). The value is always valid for the whole system and cannot be set separately for individual wave generators. The value of the parameter in volatile memory can be read with the WTR? command.

WTR also sets the type of interpolation to use for the wave generator output. If ***Wave Generator Table Rate*** is greater than 1, interpolation helps to avoid sudden position jumps of an axis controlled by the wave generator.

With WGR you can restart data recording while the wave generator is running. The recorded data can be read with the DRR? command. See "Data Recording" (p. 65) for more information.

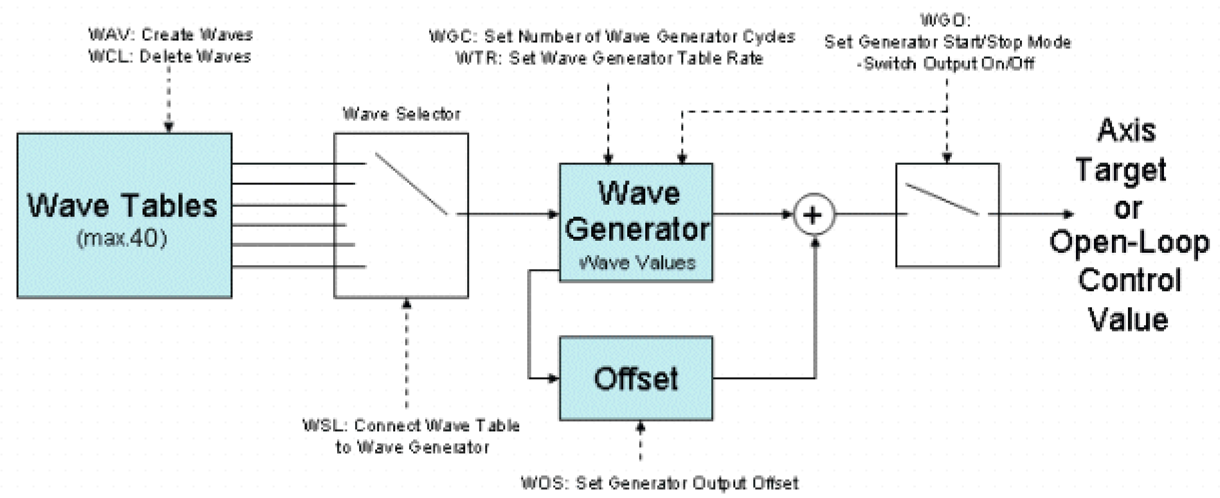


Figure 30: Block diagram of one wave generator

Application Notes

All wave generators can run simultaneously. All waveform output is synchronized because there is a common pulse generator used by all wave generators. For that reasons, wave tables which are supposed to run at the same time (each with one wave generator) should have the same length. If the wave tables have different lengths, an output cycle will comprise only the number of points given by the shortest table. This means that all waveform output is cut to the length of the shortest waveform currently running.

Waveforms cannot be changed while they are being output by a wave generator. If you want to modify a waveform with WAV, first stop any wave generator output from the associated wave table.

The frequency of the wave generator output depends, among other factors, on the wave table length. When you create waveforms, keep in mind that the usable frequency is limited by the available amplifier power. If the frequency is too high, overheating of the amplifier(s) can occur, and the piezo voltage output will be deactivated automatically.

When you use the wave generator to create white noise for a measurement (WAV command with <WaveType> = "NOISE"), make sure that the wave table segment is at least as long as the duration of the measurement. Otherwise, the noise sequence will be repeated during the measurement and will no longer be white noise.

Wave generator output and analog control input:
It is possible to configure an axis for control by an analog input line while the wave generator output is active for that axis. In that case, the wave generator will continue running, but its output will no longer be used for control value generation. As long as the corresponding axis is set up to be commanded by analog control input, you can stop the wave generator output, but not restart it.

Wave generator output and move commands:
When the wave generator output is active, move commands like MOV or SVA are not allowed for the associated axis.

See "Control Value Generation" (p. 30) for details.

When the wave generator is to be started by an external trigger signal (WGO bit 1 is set): For reliable triggering, the pulse width of the input signal has to be at least 2 x the servo update time of the E-727 (the servo update time is given in seconds by parameter 0x0E000200). The value of the ***Wave Multi Start By Trigger*** parameter (ID 0x13000202) determines if the trigger is enabled for only one generator start or for multiple starts. See "Wave Generator Started by Trigger Input" (p. 111) for details and an example.

A wave generator outputs absolute values. In closed-loop operation (servo ON), the output is interpreted as target positions in either case. In open-loop operation (servo OFF), the interpretation of the wave generator output depends on the settings of the output matrix (see "Output Generation" (p. 34) for more information). By default, the matrix is set up so that commanded open-loop control values numerically correspond to axis position values.

Servo cannot be switched off (SVO) while a wave generator is running for the axis.

As long as a wave generator is running, it is not possible to change (WSL) or to delete (WCL) the connected wave table (i.e. the waveform). The wave generator table rate (WTR), the number of output cycles (WGC), the wave offset (WOS) and the output trigger settings (TWS) can be modified while a wave generator is running.

When a wave generator finishes by running through a specified number of cycles completely, the final position will be the first point of the waveform, unless the option "start at the endpoint of the last cycle" was selected. In that case, the final position is the sum of the endpoint of the last output cycle and any offset defined with WAV for the waveform.

When the wave generator is stopped within an output cycle by command, the axis will remain at the last output position until a new position is commanded. If the wave generator is then restarted, it will normally continue with the first point of the waveform, unless started with the option "start wave generator output triggered by external signal", and the digital input line IN2 (see "Digital I/O Socket" (p. 227)) is used for triggering. In that case, the generator starts with the first rising edge which is detected on this input line, and it will be stopped when a falling edge is detected on this line. With the next rising edge, the generator output will continue at the waveform point where it was stopped.

Wave generator output will continue even if the terminal or the program from which it was started is quit or if the high voltage output is deactivated.

See the WGO command for more information.

The following data is always lost when the controller is powered down or rebooted:

- Wave table content (WAV)
- Assignment of wave tables to wave generators (WSL)
- Output trigger settings (TWS)
- Number of cycles for wave generator output (WGC)

The following settings can be saved with WPA to non-volatile memory, where they become the power-on defaults (switch to command level 1 before with the CCL command):

- Wave offset (WOS)
- Wave generator table rate (WTR)

INFORMATION

You can permanently save the settings of the wave generator in the E-727 with the macro functionality of the E-727 (p. 121). You can also use a startup macro to configure the wave generator and start the output each time that the E-727 is switched on or rebooted.

The different software interfaces provided for the controller also support use of the wave generator. Waveforms can be defined, stored and displayed in and by the software in a more user-friendly way than in a terminal using WAV and WGO. If using the wave generator with the GCS DLL, PIMikroMove or drivers for NI LabView software, read the descriptions in the associated software manual first.

INFORMATION

It is recommended to use the PI Frequency Generator Tool or the PI Wave Generator Tool for work with the wave generator (both available in PIMikroMove).

No command knowledge is necessary to work with PIMikroMove. The use of the PI Frequency Generator Tool is described in the A000T0057 technical note. The use of the PI Wave Generator Tool is described in the PIMikroMove manual (SM148E).

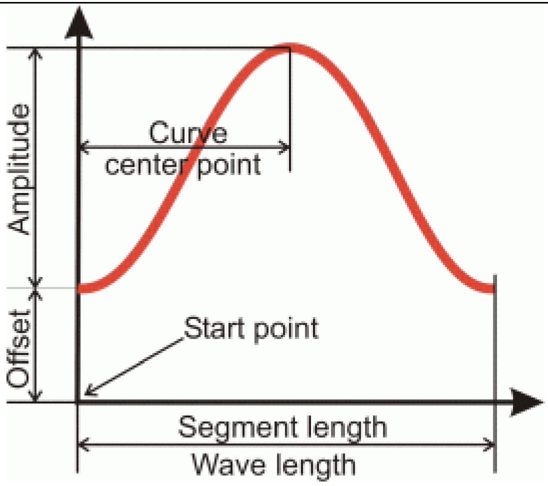
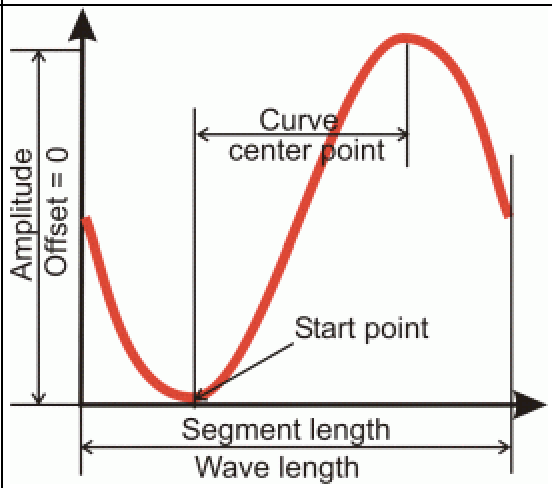
Wave Generator Examples

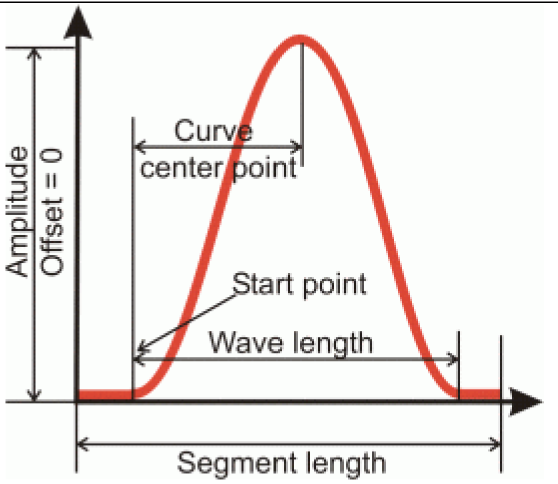
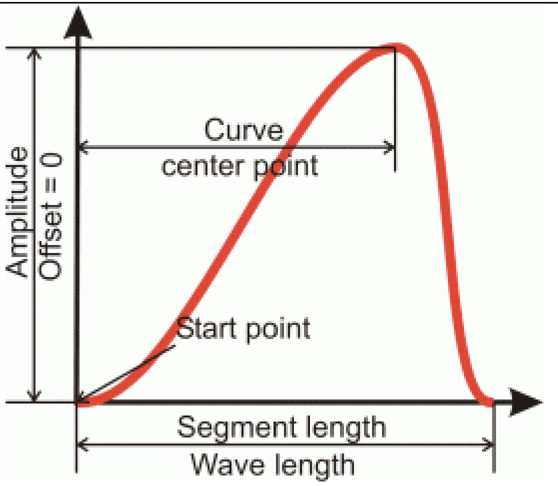
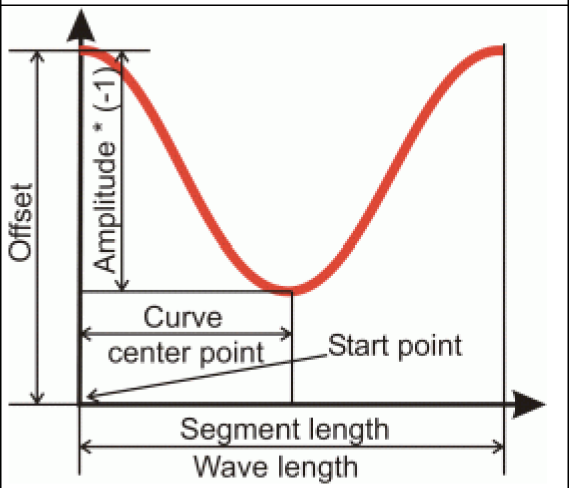
The following examples can be reproduced using the command entry facilities of PIMikroMove or PI Terminal. Note that it might be necessary to adapt them to your hardware configuration.

Defining Waveforms

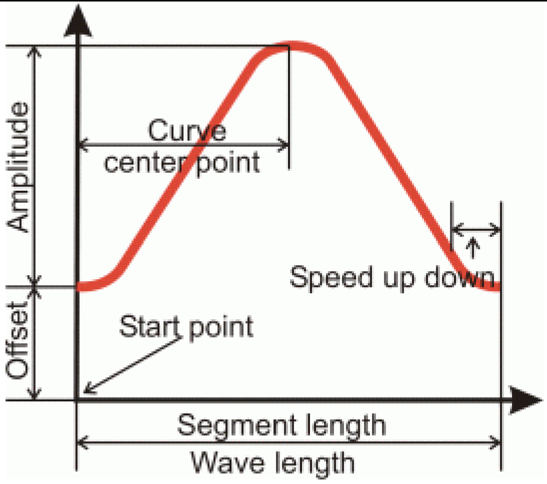
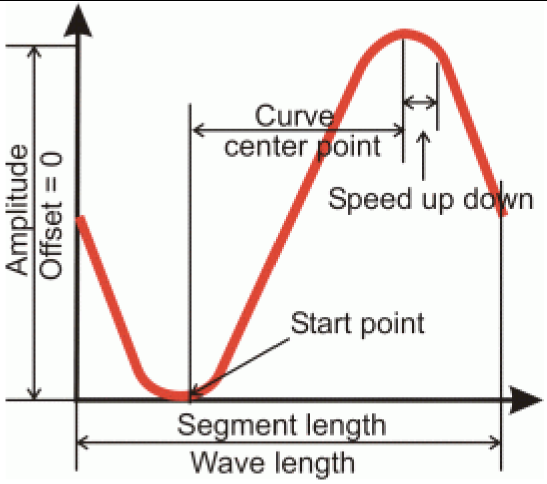
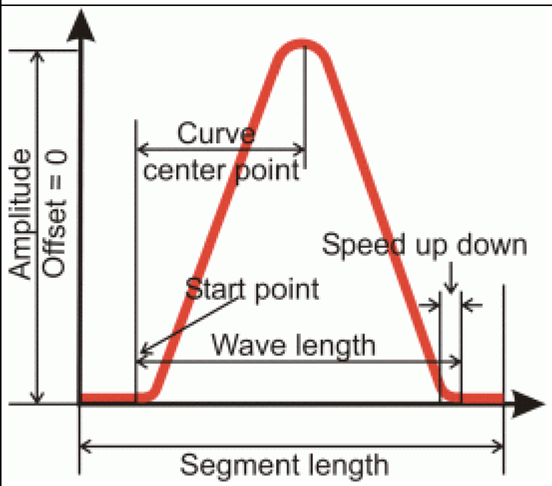
Examples for how to define waveform segments for the wave tables, based on predefined curve shapes (each WAV command defines a waveform segment which either replaces or is appended to the waveform in the specified wave table):

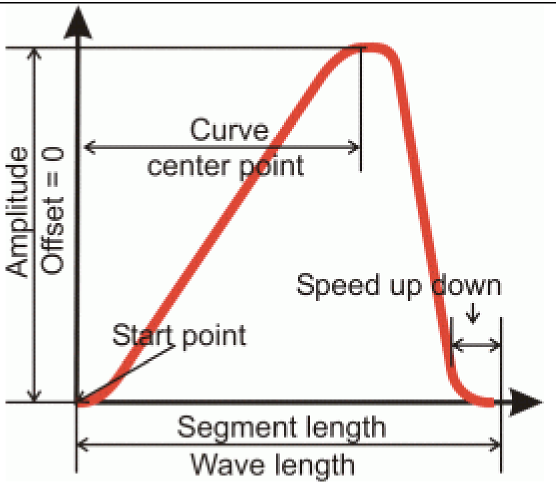
Sine Curves

WAV command	Comments	Waveform Segment
WAV 2 X SIN_P 2000 20 10 2000 0 1000 <WaveTableID> = 2 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 2000 <Amp> = 20 <Offset> = 10 <WaveLength> = 2000 <StartPoint> = 0 <CurveCenterPoint> = 1000	The previous contents of the wave table are overwritten by the new segment, waveform offset = 10 (Do not confuse with the wave generator output offset set with WOS!), symmetric curve	
WAV 2 X SIN_P 2000 30 0 2000 499 1000 <WaveTableID> = 2 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 2000 <Amp> = 30 <Offset> = 0 <WaveLength> = 2000 <StartPoint> = 499 <CurveCenterPoint> = 1000	The previous contents of the wave table are overwritten by the new segment, symmetric curve	

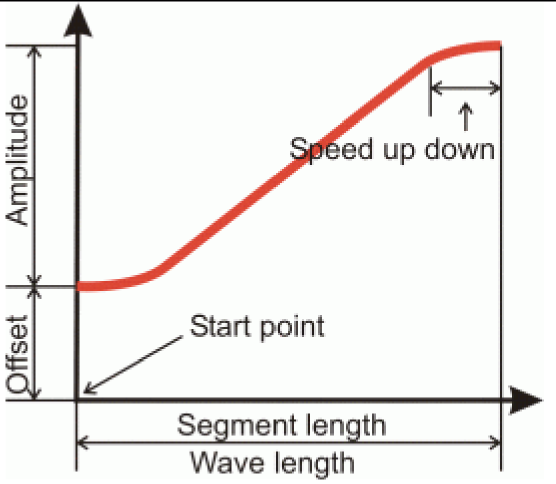
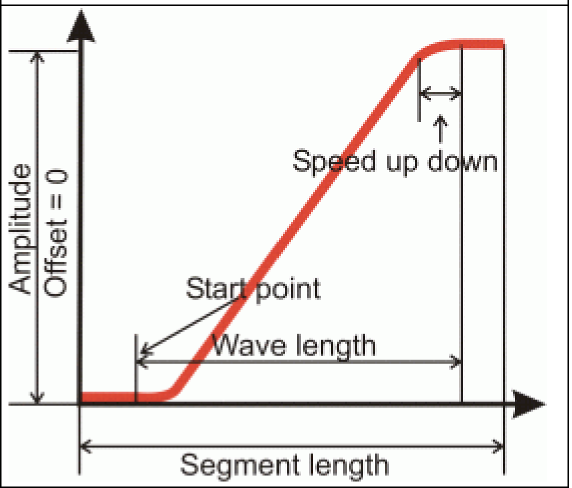
WAV command	Comments	Waveform Segment
WAV 2 & SIN_P 2000 25 0 1800 100 900 <WaveTableID> = 2 <AppendWave> = & <WaveType> = SIN_P <SegLength> = 2000 <Amp> = 25 <Offset> = 0 <WaveLength> = 1800 <StartPoint> = 100 <CurveCenterPoint> = 900	The defined segment will be appended to the existing wave table contents, symmetric curve	
WAV 3 X SIN_P 4000 20 0 4000 0 3100 <WaveTableID> = 3 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 4000 <Amp> = 20 <Offset> = 0 <WaveLength> = 4000 <StartPoint> = 0 <CurveCenterPoint> = 3100	The previous contents of the wave table are overwritten by the new segment, asymmetric curve	
WAV 2 X SIN_P 1000 -30 45 1000 0 500 <WaveTableID> = 2 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 1000 <Amp> = -30 <Offset> = 45 <WaveLength> = 1000 <StartPoint> = 0 <CurveCenterPoint> = 500	The previous contents of the wave table are overwritten by the new segment, negative-amplitude curve, symmetric curve	

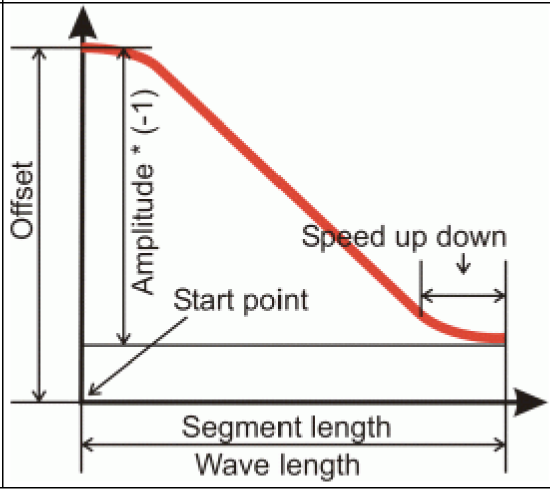
Ramp Curves

WAV command	Comments	Waveform Segment
WAV 4 X RAMP 2000 20 10 2000 0 300 1000 <WaveTableID> = 4 <AppendWave> = X <WaveType> = RAMP <SegLength> = 2000 <Amp> = 20 <Offset> = 10 <WaveLength> = 2000 <StartPoint> = 0 <SpeedUpDown> = 300 <CurveCenterPoint> = 1000	The previous contents of the wave table are overwritten by the new segment, waveform offset = 10 (Do not confuse with the wave generator output offset set with WOS!) symmetric curve	
WAV 4 X RAMP 2000 35 0 2000 499 300 1000 <WaveTableID> = 4 <AppendWave> = X <WaveType> = RAMP <SegLength> = 2000 <Amp> = 35 <Offset> = 0 <WaveLength> = 2000 <StartPoint> = 499 <SpeedUpDown> = 300 <CurveCenterPoint> = 1000	The previous contents of the wave table are overwritten by the new segment, symmetric curve	
WAV 5 X RAMP 2000 15 0 1800 120 150 900 <WaveTableID> = 5 <AppendWave> = X <WaveType> = RAMP <SegLength> = 2000 <Amp> = 15 <Offset> = 0 <WaveLength> = 1800 <StartPoint> = 120 <SpeedUpDown> = 150 <CurveCenterPoint> = 900	The previous contents of the wave table are overwritten by the new segment, symmetric curve	

WAV command	Comments	Waveform Segment
WAV 5 & RAMP 3000 35 0 3000 0 200 2250 <WaveTableID> = 5 <AppendWave> = & <WaveType> = RAMP <SegLength> = 3000 <Amp> = 35 <Offset> = 0 <WaveLength> = 3000 <StartPoint> = 0 <SpeedUpDown> = 200 <CurveCenterPoint> = 2250	The defined segment will be appended to the existing wave table contents, asymmetric curve	

Single Scan Line Curves

WAV command	Comments	Waveform Segment
WAV 1 X LIN 1500 30 15 1500 0 370 <WaveTableID> = 1 <AppendWave> = X <WaveType> = LIN <SegLength> = 1500 <Amp> = 30 <Offset> = 15 <WaveLength> = 1500 <StartPoint> = 0 <SpeedUpDown> = 370	The previous contents of the wave table are overwritten by the new segment, waveform offset = 15 (Do not confuse with the wave generator output offset set with WOS!)	
WAV 2 X LIN 1500 40 0 1100 210 180 <WaveTableID> = 2 <AppendWave> = X <WaveType> = LIN <SegLength> = 1500 <Amp> = 40 <Offset> = 0 <WaveLength> = 1100 <StartPoint> = 210 <SpeedUpDown> = 180	The previous contents of the wave table are overwritten by the new segment	

WAV command	Comments	Waveform Segment
WAV 2 & LIN 3000 -40 50 3000 0 650 <WaveTableID> = 2 <AppendWave> = & <WaveType> = LIN <SegLength> = 3000 <Amp> = -40 <Offset> = 50 <WaveLength> = 3000 <StartPoint> = 0 <SpeedUpDown> = 650	The defined segment will be appended to the existing wave table contents, negative-amplitude curve	

Modifying the Wave Generator Table Rate

An example for how to modify the duration of the wave generator output using the wave table rate:

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for Wave Table 2, the segment length and hence the number of points in the wave table is 2000
SPA? 1 0x0E000200	Ask for the servo update time of the controller (reading the wave table for wave generator output is clocked by servo cycles). The E-727 has a servo update time of 50 µs.
WTR?	Ask for the current wave table rate and interpolation settings, default is wave table rate = 1 (i.e. each wave table point will be output for a duration of one servo cycle). The duration of one wave generator output cycle will be: Servo Update Time (in s) * WTR value * Number of Points = Output Duration (in s) 0.000050 s * 1 * 2000 = 0.1 s

Command String to Send	Action Performed
WTR 0 3 1	Set the wave table rate to 3, tripling the duration of one wave generator output cycle, with linear interpolation (each wave table point will now "occupy" 3 servo cycles, but with linear interpolation applied to smooth the output). Duration of one output cycle will now be: 0.000050 s * 3 * 2000 = 0.3 s Note that the WTR command must always specify all wave generators in the system (<WaveGenID> must be 0).

Trigger Output Synchronized with Wave Generator

Using the digital output lines OUT1 to OUT3 of the E-727, it is possible to trigger external devices. See "Digital I/O Socket" (p. 227) for the availability of the lines (pinout) and "Configuring Trigger Output" (p. 69) for further information.

Example 1: Generate single trigger pulses synchronized with the wave generator in Generator Level Trigger mode:

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for Wave Table 2, the segment length and hence the number of points in the wave table is 2000
TWC	Clears all output trigger settings related to the wave generator by switching the signal state for all points to "low" (the power-on default state is also "low"). It is recommended to use TWC before new trigger actions are defined.
TWS 1 500 1 1 1500 1 1 1900 1 1 2000 1	Set trigger actions for the digital output line OUT1 (identifier is 1): at the waveform points 500, 1500, 1900 and 2000 it is set high; at all other points the state of the line is low (due to the TWC usage).
CTO 1 3 4	The digital output line OUT1 is set to "Generator Level Trigger" mode.
WSL 1 2	Connect Wave Generator 1 (Axis 1) to Wave Table 2
WGO 1 1	Start output of Wave Generator 1 immediately (synchronized by servo cycle). Now the trigger output action will take place as specified.
WGO 1 0	Stop output of Wave Generator 1 and hence also the trigger output.

Example 2: Use Generator Level Trigger mode to switch the digital output line to a certain level for a certain range of the waveform.

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for Wave Table 2, the segment length and hence the number of points in the wave table is 2000
CTO 1 3 4	The digital output line OUT1 is set to "Generator Level Trigger" mode.
TWS 1 1 3 1 750 2 1 1150 3	For all waveform points from point 1 to point 749, the output line is set low. At point 750, there is a rising edge on the output line. Therefore, the output line is set high from point 751 to point 1149. At point 1150, there is a falling edge on the output line, and for all subsequent points the line will therefore be set low.
WSL 1 2	Connect Wave Generator 1 (Axis 1) to Wave Table 2
WGO 1 1	Start output of Wave Generator 1 immediately (synchronized by servo cycle). Now the trigger output action will take place as specified.
WGO 1 0	Stop output of Wave Generator 1 and hence also the trigger output.

INFORMATION

If a phase shift between the trigger output and the actual position is observed when the wave generator is running with the "Use DDL" or "Use and reinitialize DDL" options (actual position measured with external device is ahead of the trigger signal), check the cut-off-frequency f_g of the IIR low-pass filter used for digital filtering and increase it to shorten the duration of the digital filtering. See below for details.

The duration of the signal processing for an internal sensor results from two portions:

- Duration of the analog sensor processing, which takes about 100 µs for the E-727
- Duration of the digital filtering. If the ***Digital Filter Type*** parameter, ID 0x05000000, is set to "IIR low-pass filter, 2nd order", the required time depends on the cut-off-frequency f_g of the IIR low-pass filter which is given by the ***Digital Filter Bandwidth*** parameter, ID 0x05000001. For signal frequencies $f < f_g/2$, the duration of the filtering can be estimated as follows:
  $t \approx 0.216 / f_g$
  If required, increase f_g. Examples:
  f_g = 300 Hz: t = 0.72 ms
  f_g = 1000 Hz: t = 0.216 ms

Wave Generator Started by Trigger Input

Using the digital input lines of the E-727, it is possible to apply start/stop signals for the wave generator output. See the pinout description of the digital I/O socket (p. 227) for the availability of the lines and "Using Digital Input" (p. 68) for an overview.

INFORMATION

For reliable triggering, the pulse width of the input signal has to be at least 2 x the servo update time of the E-727. The servo update time is given in seconds by parameter 0x0E000200.

An example for how to start / stop the wave generator by external trigger signals:

Command String to Send	Action Performed
WAV 2 X SIN_P 2000 20 10 2000 0 1000	Define a sine waveform for Wave Table 2, the segment length and hence the number of points in the wave table is 2000
WSL 1 2	Connect Wave Generator 1 (Axis 1) to Wave Table 2
WGO 1 2	Start output of Wave Generator 1 triggered by external signal. To provide the external signal, the digital input lines IN1 or IN2 can be used. If IN1 is used: The wave generator output starts with the first rising edge which is detected on this input line. If IN2 is used: The generator output starts with the first rising edge which is detected on this input line, and it will be stopped when a falling edge is detected on this line. With the next rising edge, the generator output will continue at the waveform point where it was stopped. Starting and stopping the wave generator this way can be repeated indefinitely. If output cycle limitations were made with WGC: with each generator restart the counting of the output cycles continues, and the generator will be stopped when the given number of cycles are completed, irrespective of any further trigger pulses. It is possible to mix the usage of both digital input lines.
WGO 1 0	Stop output of Wave Generator 1 (any further trigger pulses will be ignored). You can also use #24 or STP.

INFORMATION

When the wave generator is to be started by an external trigger signal (WGO bit 1 is set), the value of the ***Wave Multi Start By Trigger*** parameter (ID 0x13000202) 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.

Wave-Generator-Related Commands and Parameters

Command	Description	Notes
CTO	Set Configuration Of Trigger Output	Activates the Generator Trigger output mode which is required for the triggerline actions set with TWS.
DDL?	Get DDL Table Values	Reads the current content of the DDL table(s). DDL initialization and usage are started by the wave generator start command (WGO).
DRR?	Get Recorded Data Values	Reads the last recorded data. Data recording is triggered by the WGO and WGR commands (among others).
GWD?	Get Wave Table Data	Should be used to check the waveform before the wave generator output is started.
TWC	Clear All Wave Related Triggers	Clears only the TWS settings, but not the CTO settings.
TWG?	Get Number Of Wave Generators	Number of wave generators = number of axes
TWS	Set TriggerLine Action To Waveform Point	In addition, the CTO command must be used to activate the Generator Trigger mode for the desired digital output line.
TWS?	Get Trigger Line Action At Waveform Point	
WAV	Set Waveform Definition	A waveform must be defined before the wave generator output can be started.
WAV?	Get Waveform Definition	Reads the current wave table length.
WCL	Clear Wave Table Data	Clears the wave table content, but not the WSL and WOS settings.
WGC	Set Number Of Wave Generator Cycles	If WGC is not used, the wave generator must be stopped with WGO, #24 or STP.
WGC?	Get Number Of Wave Generator Cycles	
WGO	Set Wave Generator Start/Stop Mode	The WGO command starts the wave generator output. It provides several start options, e.g. "Start wave generator output triggered by external signal", "Use and reinitialize DDL" or "Use DDL".
WGO?	Get Wave Generator Start/Stop Mode	Gets the last commanded start options, but not the activation status (use #9 instead)
WGR	Starts Recording in Sync with Wave Generator	Restarts data recording as long as a wave generator is running.
WOS	Set Wave Generator Output Offset	Sets the value of the ***Wave Offset*** parameter, ID 0x1300010b, in volatile memory.
WOS?	Get Wave Generator Output Offset	Gets the value of the ***Wave Offset*** parameter, ID 0x1300010b, from volatile memory.

Command	Description	Notes
WSL	Set Connection Of Wave Table To Wave Generator	Must be set before the wave generator can be started.
WSL?	Get Connection Of Wave Table To Wave Generator	
WTR	Set Wave Generator Table Rate	Sets the value of the ***Wave Generator Table Rate*** parameter, ID 0x13000109, in volatile memory; determines the interpolation type for wave table rate values > 1.
WTR?	Get Wave Generator Table Rate	Gets the value of the ***Wave Generator Table Rate*** parameter (ID 0x13000109) from volatile memory and the current interpolation type.
#9	Get Wave Generator Status	Gets the current activation status of the wave generator, but not the start options (use WGO? instead)

See "How to Work with the Wave Generator" (p. 99) for more information. For detailed command descriptions see the GCS commands manual PZ281E. For the identifiers of the items which can be addressed with the commands see "Axes, Channels, Functional Elements" (p. 22).

Parameter ID	CCL for write access	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x13000004	3	System	1	INT	Max Wave Points
0x13000109	1	System	1	INT	Wave Generator Table Rate
0x1300010a	3	System	1	INT	Number of Waves
0x1300010b	1	Logical Axis	3 or 4	FLOAT	Wave Offset
0x13000202	1	System	1	INT	Wave Multi Start By Trigger

See "Parameters" (p. 185) for more information regarding the controller parameters and their handling.

Dynamic Digital Linearization (DDL)

Dynamic Digital Linearization (DDL) makes it possible to achieve significantly better position accuracy in dynamic applications with periodic motion. It is used in conjunction with the wave generator output and in addition to the servo algorithm in closed-loop operation of an axis. DDL "observes" axis motion over one or more wave generator output cycles (DDL initialization). The information gathered is written to "DDL tables" and can then be used to refine the control output signals.

"Working Principle" (p. 114) describes the DDL basics, "How to Activate the DDL Licence" (p. 117) gives information on how to get the DDL started the first time, in "How to work with the DDL" (p. 118) you will learn how to use the DDL feature, and "DDL-Related Commands and Parameters" (p. 198) gives a summary.

See also "Wave Generator" (p. 99).

Working Principle

The tracking error of a standard piezo servo-controller is practically zero at very low speeds but increases with the operating frequency of the axis.

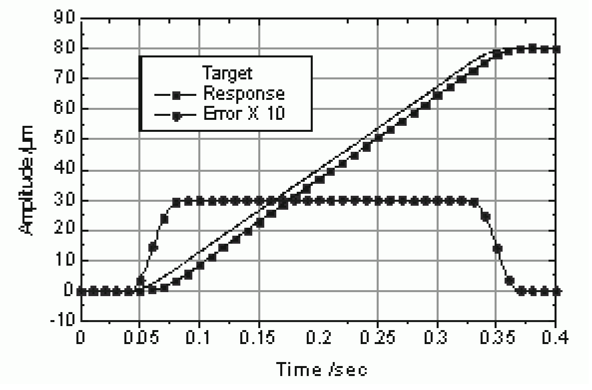


Figure 31: Tracking error for typical linear PID servo system

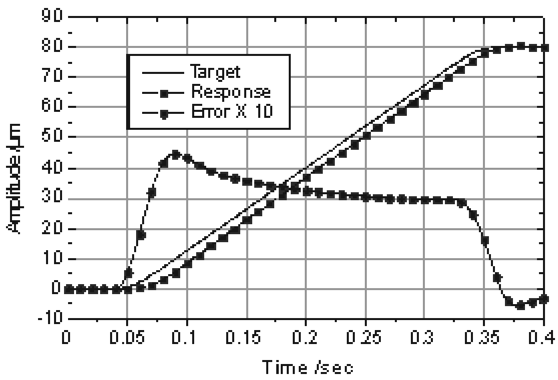


Figure 32: Tracking error for linear bi-directional scanning application

In addition, for linear bi-directional scanning applications, the tracking error is not a constant, as with other types of linear PID servo systems, but changes with the scan position. This is caused by the non-linearity of piezo actuators and the servo-controller's limited dynamic performance. As a result of these factors, the scan position is not exactly proportional to time but exhibits dynamic non-linearity.

For periodic motion it is possible to record the errors for an axis during one or more DDL initialization periods and then compensate for them in all subsequent periods. The principle of DDL is thus the inclusion of an error compensation in addition to the servo-control loop.

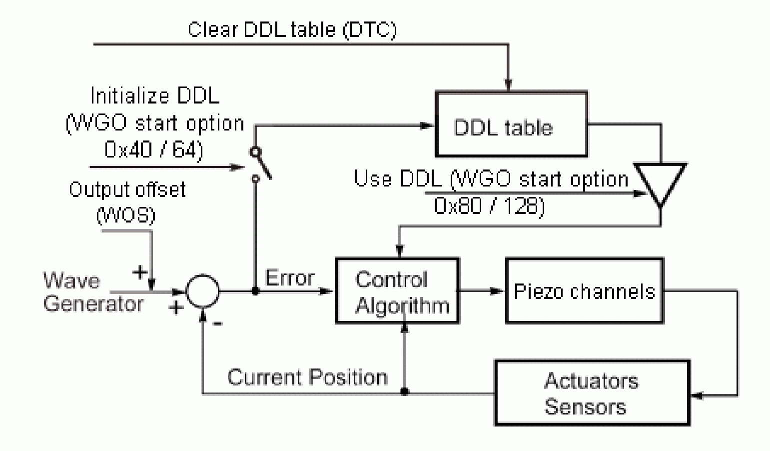


Figure 33: DDL block diagram, for one axis

Example:

A simple 400 µm forward and backward scan at 2.5 Hz. Even though the frequency is not high, there is still a static error of about ±10 µm and a dynamic error of about ±5 µm. The dynamic error indicates that the scan speed is not constant. This error will deteriorate the scanning accuracy.

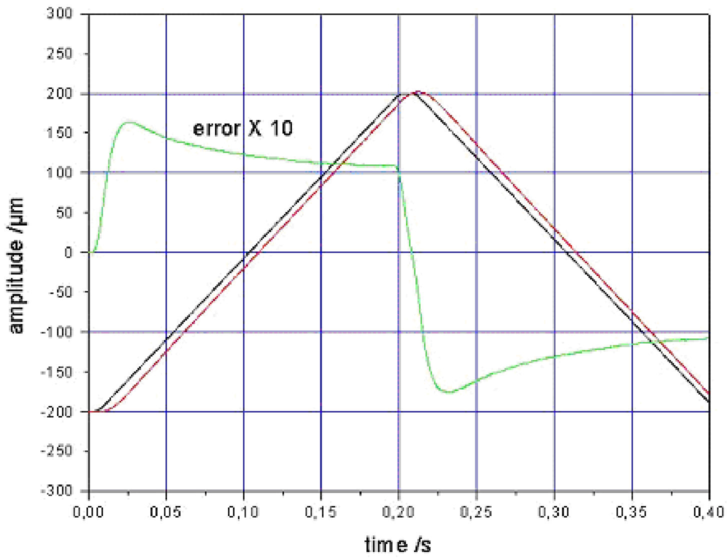


Figure 34: Without DDL

With DDL, after one initialization period the error for constant-speed motion will be reduced by a factor of about 100 and the dynamic error by a factor between 20 and 90.

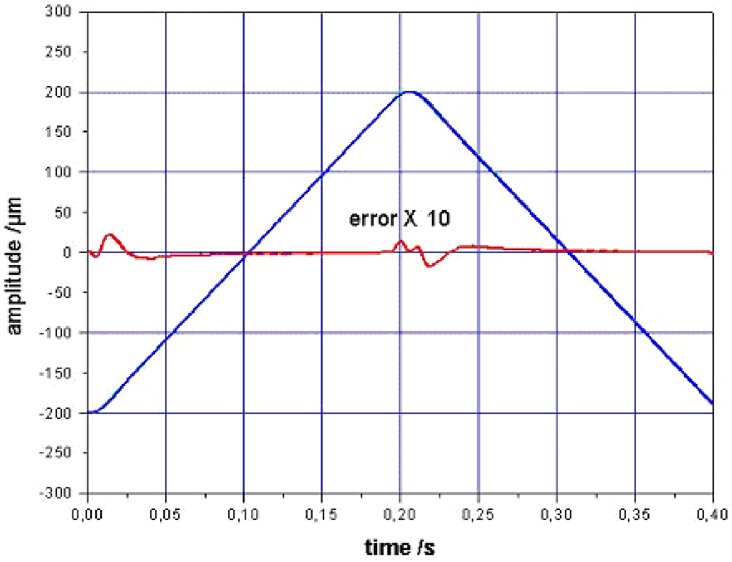


Figure 35: With DDL, one initialization cycle

With more DDL initialization periods, the dynamic error can be reduced by a factor of many hundred.

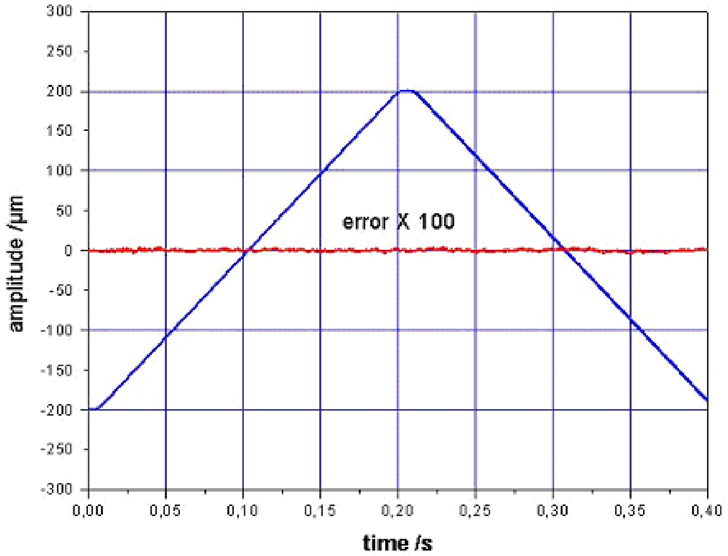


Figure 36: With DDL, after several DDL initialization cycles

How to Activate the DDL License

The Dynamic Digital Linearization (DDL) feature must be expressly ordered (order# E-710.SCN). You can activate DDL after purchase and without opening the device.

To activate the DDL license, proceed as follows:

1. Order DDL for your controller and inform PI of the controller serial number—PI generates your DDL license number based on this information.
   The serial number is saved in the controller as the ***Device S/N*** parameter, ID 0x0D000000.
   You can read it in PIMikroMove or using the SPA? or SEP? command:
   Send: `SPA? 1 0x0D000000` (reads the value from volatile memory)
   or
   Send: `SEP? 1 0x0D000000` (reads the value from nonvolatile memory)
2. PI will provide you with the DDL license number.
3. Switch to command level 1 using the CCL command:
   Send: `CCL 1 advanced`
4. Write the DDL license number to the nonvolatile memory of your controller as the ***DDL License*** parameter, ID 0x0E000400.
   You can do this in PIMikroMove or using the SEP command:
   Send: `SEP 100 1 0x0E000400 licensenumber`
5. Power cycle the controller.
6. Check if the DDL activation was successful. To do this, read the value of the ***DDL License Valid*** parameter, ID 0x0E000401 in PIMikroMove or using SPA? or SEP?.
   Send: `SPA? 1 0x0E000401`
   or
   Send: `SEP? 1 0x0E000401`
   DDL activation was successful if the parameter value is 1. Otherwise the DDL feature is still deactivated.

If the DDL activation fails:

- Check if the correct serial number was used for license number generation.
- Check whether the correct license number was entered into controller memory:
  Send: `SEP? 1 0x0E000400`
- Make sure that command level 1 is active during license activation.

How to work with the DDL

NOTICE

Damage to the stage and the load from oscillations!

Using DDL could be critical if there are any residual oscillations in the system; DDL will then cause the oscillations to build up—the more wave generator cycles are used for DDL initialization, the stronger the effect.

- Before you work with Dynamic Digital Linearization (DDL), eliminate any residual oscillations by adjusting the notch filter(s) and the servo-control parameters (P-term, I-term; see "Servo-Controller Dynamic Tuning" (p. 134)).

INFORMATION

The Dynamic Digital Linearization (DDL) feature must be expressly ordered (order# E-710.SCN). You can activate it after purchase and without opening the device. See "How to activate the DDL License" (p. 117) for more information.

If you are not sure if the DDL feature is activated on your controller, use the SPA? command to check the value of parameter 0x0E000401 (***DDL License Valid***). If the parameter value is 1, the DDL feature is activated, otherwise it is deactivated.

DDL related commands will not provoke an error when the DDL feature is deactivated.

The number of DDL tables can be queried using the TLT? command. The number of DDL tables present in the E-727 is the same as the number of logical axes, and each DDL table is dedicated to one axis.

A certain amount of memory space is reserved for the DDL data (ask with the SPA? command, parameter ID 0x1400000B). 256144 memory points are available. The individual DDL tables may use different portions of the complete memory space—the number of points in a DDL table corresponds to the length of the waveform which was output during the DDL initialization. You can ask for the current DDL table length using the DTL? command.

DDL tables are automatically filled with data when a wave generator is started with the "Use and reinitialize DDL" option activated. It is also possible to write data "manually" with the DDL command—in this case, make sure that the DDL table length is correct (see DDL command description for more information).

INFORMATION

The DDL initialization must be repeated when a new stage is connected, the servo parameters are changed (e.g. due to load changes) or the waveform is changed.

It is recommended to start the DDL initialization for all axes at the same time. Each new initialization will stop all running initialization processes. The initialization process is also stopped by the DDL command.

How to initialize DDL for an axis:

1. Define a waveform using the WAV command.
2. Optionally: Set the number of wave generator cycles to use for DDL initialization. This can be done using the SPA or SEP command with the ***DDL Repeat Number parameter*** (ID 0x14000001). The factory default is 35. (To have write access to the parameter, it might be necessary to switch to command level 1 using CCL).
3. Not essential but recommended: Delete DDL table content which is no longer used. This can be done using the DTC command.
4. When the servo parameters (notch filter frequency, servo-loop P-term, servo-loop I-term and servo-loop slew rate) have changed: Recalculate the internal DDL processing parameters using the DPO command.
5. Assign the waveform to the axis using the WSL command.
6. Switch the servo on for the axis using the SVO command (closed-loop operation).
7. Start the wave generator with the WGO command, with the "Use and reinitialize DDL" option activated.
8. Optionally: Check the content of the DDL table using the DDL? command.

How to use DDL for an axis:

As long as your application does not change, you can use the current DDL table content without new initialization. In this case, start the wave generator with WGO and the "Use DDL" option activated.

INFORMATION

If the DDL performance does not prove satisfactory or if errors occur during DDL use:

- ➢ Recalculate the internal DDL processing parameters using DPO.
- ➢ Reduce the frequency of the wave generator output to make sure that the axis can follow the target values. To do this, define a new, optimized waveform with the WAV command. You can also use the WTR command, which sets the wave generator table rate, to optimize the output frequency.

After such changes DDL initialization must be repeated.

INFORMATION

If a phase shift between the trigger output and the actual position is observed when the wave generator is running with the "Use DDL" or "Use and reinitialize DDL" options (actual position measured with external device is ahead of the trigger signal), check the cut-off-frequency f_g of the IIR low-pass filter used for digital filtering and increase it to shorten the duration of the digital filtering. See below for details.

The duration of the signal processing for an internal sensor results from two portions:

- Duration of the analog sensor processing, which takes about 100 µs for the E-727
- Duration of the digital filtering. If the ***Digital Filter Type*** parameter, ID 0x05000000,

is set to "IIR low-pass filter, 2nd order", the required time depends on the cut-off-frequency f_g of the IIR low-pass filter which is given by the ***Digital Filter Bandwidth*** parameter, ID 0x05000001. For signal frequencies $f < f_g/2$, the duration of the filtering can be estimated as follows:

$t \approx 0.216 / f_g$

If required, increase f_g. Examples:

f_g = 300 Hz: t = 0.72 ms

f_g = 1000 Hz: t = 0.216 ms

INFORMATION

The DDL table content and the calculated processing parameters will be lost when the controller is powered down or rebooted.

DDL-Related Commands and Parameters

Command	Description	Notes
DDL	Set DDL Table Values	Can be used to fill the DDL table(s) "manually" with data
DDL?	Get DDL Table Values	Reads the current content of the DDL tables(s)
DPO	DDL Parameter Optimization	Recalculates the DDL processing parameters (***Time Delay Max***, ID 0x14000006, and ***Time Delay Min***, ID 0x14000007), required if the servo parameters have changed
DTC	Clear DDL Table Data	It is recommended that the content of DDL tables which are not used be deleted. This avoids errors due to unsufficient memory space during the DDL initialization. On controller power down or reboot, the content of the DDL tables will be erased automatically.
DTL?	Get DDL Table Length	Returns the value of the ***Max DDL Points*** parameter, ID 0x1400000B
TLT?	Get Number of DDL Tables	The reported value corresponds to the number of axes available on the controller
WAV	Set Waveform Definition	Since DDL works only in conjunction with the wave generator output, the waveform must first be defined
WGO	Set Wave Generator Start/Stop Mode	The WGO command starts the wave generator output and provides two modes for the DDL: "Use and reinitialize DDL" and "Use DDL"

See "How to Work with the DDL" (p. 118) for more information. For detailed command descriptions see the GCS commands manual PZ281E. For the identifiers of the items which can be addressed with the commands see "Axes, Channels, Functional Elements" (p. 22).

Parameter ID	CCL for write access	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x0e000400	1	System	1	INT	DDL license
0x0e000401	3	System	1	INT	DDL license valid
0x14000001	1	Logical Axis	3 or 4	INT	DDL repeat number
0x14000006	1	Logical Axis	3 or 4	FLOAT	Time Delay Max
0x14000007	1	Logical Axis	3 or 4	FLOAT	Time Delay Min
0x14000008	1	Logical Axis	3 or 4	INT	Time Delay Change Rule
0x1400000a	1	Logical Axis	3 or 4	INT	DDL Zero Gain Number
0x1400000b	3	System	1	INT	Max DDL Points

See "Parameters" (p. 185) for more information regarding the controller parameters and their handling.

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 max. 5 nesting levels.
- Variables (p. 130) 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:

Command	Syntax	Function
ADD	ADD <Variable> <FLOAT1> <FLOAT2>	Adds two values and saves the result to a variable (p. 130). Can only be used for local variables in macros.
CPY	CPY <Variable> <CMD?>	Copies a command response to a variable (p. 130). Can only be used for local variables in macros.
DEL	DEL <uint>	Can only be used in macros. Delays <uint> milliseconds.

Command	Syntax	Function
JRC	JRC <Jump> <CMD?> <OP> <Value>	Can only be used in macros. Triggers a relative jump of the macro execution pointer depending on a condition.
MAC	MAC BEG <macroname>	Starts the recording of a macro with the name *macroname* on the controller. *macroname* can consist of up to 15 characters.
	MAC DEF <macroname>	Defines the given macro as the start-up macro.
	MAC DEF?	Gets the start-up macro.
	MAC DEL <macroname>	Deletes the given macro.
	MAC END	Ends the macro recording.
	MAC ERR?	Reports the last error which occurred during macro execution.
	MAC FREE?	Gets the free memory space for macro recording (unit: number of characters).
	MAC NSTART <macroname> <uint> [<String1> [<String2>]]	Starts the given macro n times in succession (n = number of executions). The values of local variables can be set for the macro with <String1> and <String2>.
	MAC START <macroname> [<String1> [<String2>]]	Starts one execution of the specified macro. The values of local variables can be set for the macro with <String1> and <String2>.
MAC?	MAC? [<macroname>]	Lists all macros or the content of a given macro.
MEX	MEX <CMD?> <OP> <Value>	Can only be used in macros. Stops the macro execution depending on a condition.
RMC?	RMC?	Lists macros which are currently running.
VAR	VAR <Variable> <String>	Sets a variable (p. 130) to a certain value or deletes it. Can only be used for local variables in macros.
VAR?	VAR? [{<Variable>}]	Gets variable values.
WAC	WAC <CMD?> <OP> <Value>	Can only be used in macros. Waits until a condition is met.
#8	-	Tests if a macro is running on the controller.

Working with Macros

Work with macros comprises the following:

- Recording of macros (p. 123)
- Starting macro execution (p. 125)
- Stopping macro execution (p. 128)
- Setting up a start-up macro (p. 128)
- Deleting of macros (p. 129)

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. 181) 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. 130) 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. 130) in macros to make macros more flexible, and give the corresponding variable values when starting macro execution.
- Contact our customer service department (p. 234) 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 ofPIMikroMove: 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 ofPIMikroMove: 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 ofPIMikroMove:
   - 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

INFORMATION

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

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

INFORMATION

A running macro may not be deleted.

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 alphabetic character.
- Names of local variables must not contain alphabetic characters. Possible characters are 0 to 9.
- The variable name can also be given via the value of another variable.

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

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

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

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

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 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, CPY or VAR, 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

```
VAR? 0
```

```
VAR? 1
```

```
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?.

Adjustment Procedures

ID-Chip Support / Stage Replacement

The stage which is connected to the E-727 may contain an ID-chip. 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)

The parameters which are usually stored in ID-chips are marked in the table in "Parameter Overview" (p. 189), but the list can differ slightly among the different mechanics provided by PI.

When a stage with ID-chip is connected to the E-727 for the first time, the stage parameters from the ID-chip will be written to nonvolatile and volatile memory upon power-on or reboot of the E-727. 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. By default, this option is disabled to facilitate maintaining optimized parameter settings in the E-727.

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

Consider the following when replacing stages with ID-chips.

"Simple" Replacement

Normally, when you replace a stage with a new unit and you are using standard factory settings for all parameters, you do not have to adjust anything. The ID-chip holds all information needed. At power-on of the system, the firmware reads the stage type and serial number stored in the ID-chip and compares this data to the data from the last connected stage, stored in the E-727:

- If there is a new stage type connected to the E-727, all the data in the ID-chip will be read and the corresponding parameters in the E-727 overwritten.
- If there is a stage of the same type but with a different serial number connected to the E-727, the calibration data from the ID-chip will be read and only the

corresponding parameters overwritten. The servo-control data will not be read, so those parameters will remain unchanged in the E-727.

If you have optimized some parameters for your application, PI recommends that you repeat your optimization routine with any new stage, because there are variations, e.g. in the stiffness and natural frequency, of stages.

Upgrade or Repair of Stages

If you send your stage to PI, e.g. for upgrade or repair, the calibration data stored in the ID-chip might be changed in the process. However, when you reconnect this stage to the E-727 to which it was connected before, the firmware will detect that the type and serial number are unchanged and will not read the new ID-chip data.

To force the E-727 to read the complete data of the ID-chip of the connected stage when the E-727 is switched on, you can enable the "Power Up Read ID-Chip" option (parameter ID 0X0F000000).

This has to be done for each input signal channel separately. Note that it might be necessary to switch to a higher command level to have write access to that parameter (use CCL or the appropriate facilities of PIMikroMove). Proceed as follows:

1. In the PIMikroMove main window, open the ***Device Parameter Configuration*** window (***E-727… > Parameter Configuration***) and select the *System Mechanics* groups where you can enable the option. When this is done for all input signal channels associated with the stage, press the ***Write selected edit values to default settings*** button in the icon bar of the ***Device Parameter Configuration*** window. Alternatively you can use the following commands in a terminal to enable the option:
   `SEP 100 1 0X0F000000 1` for input signal channel 1
   `SEP 100 2 0X0F000000 1` for input signal channel 2 , etc.
2. Now reboot the E-727 by typing the `RBT` command in the terminal (alternatively you can power-cycle the E-727). This time all data is read from the ID-chip and stored in the E-727.
3. To ensure that at next power-on or reboot the E-727 will not read all data again and overwrite parameters you may have optimized, you will have to disable the "Power Up Read ID-Chip" option, again for each input signal channel separately. In PIMikroMove, proceed as described above in step 1 but make sure that the parameter now has the value "disabled". Alternatively you can use the following commands in a terminal to disable the option:
   `SEP 100 1 0X0F000000 0` for input signal channel 1
   `SEP 100 2 0X0F000000 0` for input signal channel 2, etc.

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

Replacement of Tip/Tilt Platforms with Differential Drive

The following tip/tilt platforms provided by PI are equipped with differential drive:

- S-330 Piezo Tip / Tilt Platform
- S-331 Fast Tip / Tilt Platform
- S-334 Piezo Tip/Tilt Mirror
- S-335 Fast Tip / Tilt Platform
- S-340 Piezo Tip / Tilt Platform

Tip/tilt platforms with differential drive require a fixed voltage (usually 100 V) at the third amplifier channel. For further drive details, see the user manual of the tip/tilt platform.

There are two options for the E-727 to provide the fixed voltage:

- Option 1: Offset voltage
  The ***Voltage Offset of Amplifier*** parameter (ID 0x0b00000a) of output signal channels 3 is set to the fixed voltage value.
- Option 2: Third axis
  In the E-727, a third axis is defined that is only used to provide the fixed voltage.

When you replace a tip/tilt platform with a new stage, you have to observe that the ***Voltage Offset of Amplifier*** parameter (ID 0x0b00000a) is **not** stored in the ID-chip. Therefore, you have to check the parameter value in the E-727 and adjust it manually, if necessary. Before you connect the new stage to the E-727 for the first time, proceed as follows, (here, a terminal program is used):

1. Switch to command level 1 by sending `CCL 1 advanced` to have write access to the ***Voltage Offset of Amplifier*** parameter.
2. Query the value of the ***Voltage Offset of Amplifier*** parameter for output signal channel 3 in nonvolatile memory by sending `SEP? 3 0x0b00000a`
3. If necessary, set the value of the ***Voltage Offset of Amplifier*** parameter for output signal channel 3 to a suitable value in nonvolatile memory:
   - If the new stage has a differential drive **and** option 1 (offset voltage) is to be used, the offset parameter value must be set to the fixed voltage. Send, for example:
     `SEP 100 3 0x0b00000a 100`
     Furthermore, in the output matrix (p. 34), the matrix coefficients of output signal channel 3 must be set to zero for all axes. If necessary, send for axes 1 to 3:
     ```
     SEP 100 1 0x09000002 0
     SEP 100 2 0x09000002 0
     SEP 100 3 0x09000002 0
     ```
     If the controller has a 4th axis, send also:
     `SEP 100 4 0x09000002 0`
   - If the new stage has a differential drive **and** option 2 ("third axis") is to be used, the offset parameter value must be set to zero. Send:
     `SEP 100 3 0x0b00000a 0`
     The settings for the third axis will be loaded from the ID chip of the new stage.
   - If the new stage is **no** tip/tilt platform with differential drive, the offset parameter value must be set to zero. Send:
     `SEP 100 3 0x0b00000a 0`

Servo-Controller Dynamic Tuning

If the E-727 and the attached stage(s) are ordered together and if PI has sufficient knowledge of your application, then the parameters of notch filters and standard control algorithm (servo parameters) will be set to suitable values at the factory, and, if present, saved in the stage's ID-chip. Modifications of those parameters may, however, be necessary if the load applied to the stage is changed.

Parameters to be Modified

The following parameters may need to be modified in the E-727:

- Settings for notch filters 1 and 2:

  Notch frequency 1 (parameter ID 0x08000100), notch frequency 2 (parameter ID 0x08000101)

  To determine the resonant frequencies and set the notch filters properly, observe the system response to an impulse in open-loop operation (p. 136).

  Note that by default, the notch filters are enabled only in closed-loop operation.

- Servo-control parameters:

  P-term (parameter ID 0x07000300), I-term (parameter ID 0x07000301):

  Normally the proper P-term and I-term settings are found by observing the response of the axis to an abrupt change of the control value (step response) in closed-loop operation (p. 140).

General Notes on Servo-Controller Dynamic Tuning

NOTICE

Damage to the stage and the load from oscillations!
Unsuitable settings of the notch filters 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):

1. Immediately switch off the servo mode or disconnect the E-727 from the power source.
2. Only switch on the servo mode after you have modified the settings of the notch filters and the servo-control parameters of the E-727.

- Before you change parameter values of the E-727, create a backup file. See "Creating Backup Files for Controller Parameters" (p. 53) for more information.
- Enter the password “advanced” when prompted to change to command level 1.
- For stages with ID-chip, to make the optimized settings available in the future, the option "Power Up Read ID-Chip" must have "disabled" as its power-on default (value of parameter 0x0F000000 = 0 in nonvolatile memory). See "ID-Chip Support / Stage Replacement" (p. 131) for more information.

- The settling behavior of the axis in closed-loop operation is influenced by the notch filter settings. Set the notch filters before you optimize the servo-control parameters (p. 136).
- If you work with the ***Piezo Dynamic Tuner*** window of PIMikroMove:
  - If you change a parameter value of the E-727 by entering a corresponding value: The value is displayed in a blue font until you press Enter on your keyboard. Pressing Enter sends the value to the E-727 and changes the font color from blue to black. For fields highlighted by a red background, the parameter values in volatile and nonvolatile memory of the E-727 differ.
  - When the ***Notch Frequency 1*** value is set in the ***Parameter Settings*** panel of the ***Piezo Dynamic Tuner*** window, the ***Servo-Loop I-Term*** value can be adjusted automatically in accordance. The adjustment depends on the selection in the ***Automatic I-Term*** calculation drop down menu.

    Default: The I-term is set to a "conservative" value which is calculated with the following formula:

    $$I\ term_{conservative} = \frac{P\ term}{0.05\ \times 4\ \times \pi\ \times \text{Notch Frequency 1}}$$

    Further options:

    "Dynamic" I-term value, calculated with the following formula:

    $$I\ term_{dynamic} = \frac{0.8\ \times P\ term}{0.05\ \times 4\ \times \pi\ \times \text{Notch Frequency 1}}$$

    "Off", i.e. no automatic I-term calculation.
- The settings for slew rate (***Slew Rate / Velocity*** field) and record table rate (***Record Rate*** field) can be changed in the ***Piezo Dynamic Tuner*** window. Entering new values in these fields changes the values of the corresponding parameters in volatile memory:
  - ***Servo Loop Slew-Rate*** (ID 0x07000200) or ***Open Loop Slew-Rate*** (ID 0x07000201), depending on the current operating mode (open-loop or closed-loop operation)
  - ***Data Recorder Table Rate*** (ID 0x16000000).

  The values are **not** saved or reset when you use the ***Save …*** and ***Reset …*** buttons in the ***Parameter Settings*** panel of the ***Piezo Dynamic Tuner*** window.

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.

For further details, see “Notch Filters”, p. 37.

INFORMATION

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

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 the axis (0 = disable notch filter in open-loop operation; 1 = enable notch filter in open-loop operation). You can do this in the ***Servo*** parameter group of the ***Device Parameter Configuration*** window in PIMikroMove.

In addition to the measurement described below, you can create a Bode plot: In the PIMikroMove main window, open the ***Data Recorder*** window via the ***E-727… > Show data recorder …*** menu item. At the bottom of the ***Data Recorder*** window, enter the ***Amplitude*** value and click ***Estimate*** to start the frequency response.

INFORMATION — SCREENSHOTS

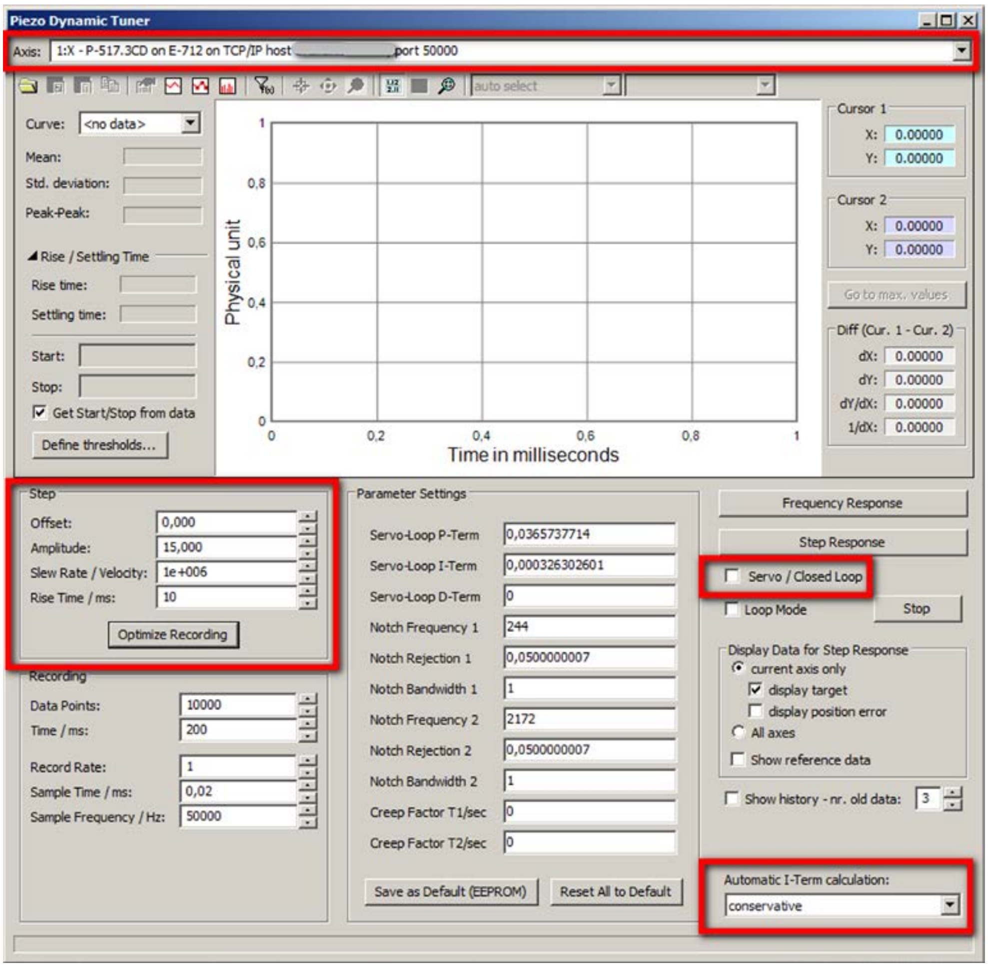
The screenshots in the following instructions were created with an E-712 digital multi-axis controller. With E-727, some values may differ, but the procedure outlined in the screenshots is as with E-712.

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. Read the “General Notes on Servo-Controller Dynamic Tuning” (p. 134).
2. Make sure the stage is mounted in exactly the same way as in the application. The load on the stage is especially important.
3. In the main window of PIMikroMove, open the ***Piezo Dynamic Tuner*** window via the ***E-727… > Dynamic Tuner …*** menu item.
4. Configure the frequency response in the ***Piezo Dynamic Tuner*** window:
   a) Make sure that the correct axis is selected (***Axis*** drop down list).
   b) Enter suitable values for the start value (***Offset:***) and the amplitude (***Amplitude:***) of the impulse in the ***Step*** panel. The start value should be 0, and the amplitude should be about 10 % of the axis travel range.
   c) Make sure that the axis is in open-loop operation (***Servo / Closed Loop*** box is **not** checked).

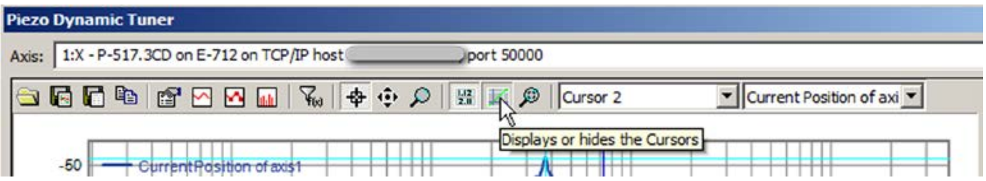
d) Select if and how the ***Servo-Loop I-Term*** is to be adjusted automatically when ***Notch Frequency 1*** is changed (***Automatic I-Term calculation***).

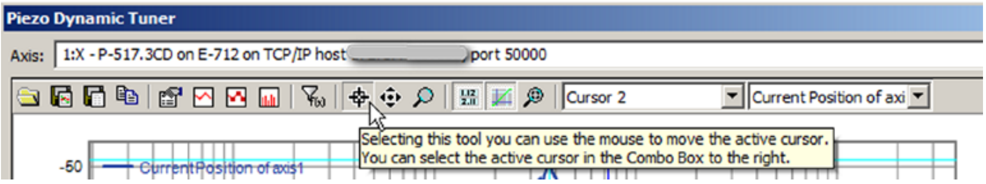


5. Perform the frequency response measurement by clicking the ***Frequency Response*** button in the ***Piezo Dynamic Tuner*** window.

6. Identify the resonant frequency in the ***Piezo Dynamic Tuner*** window:

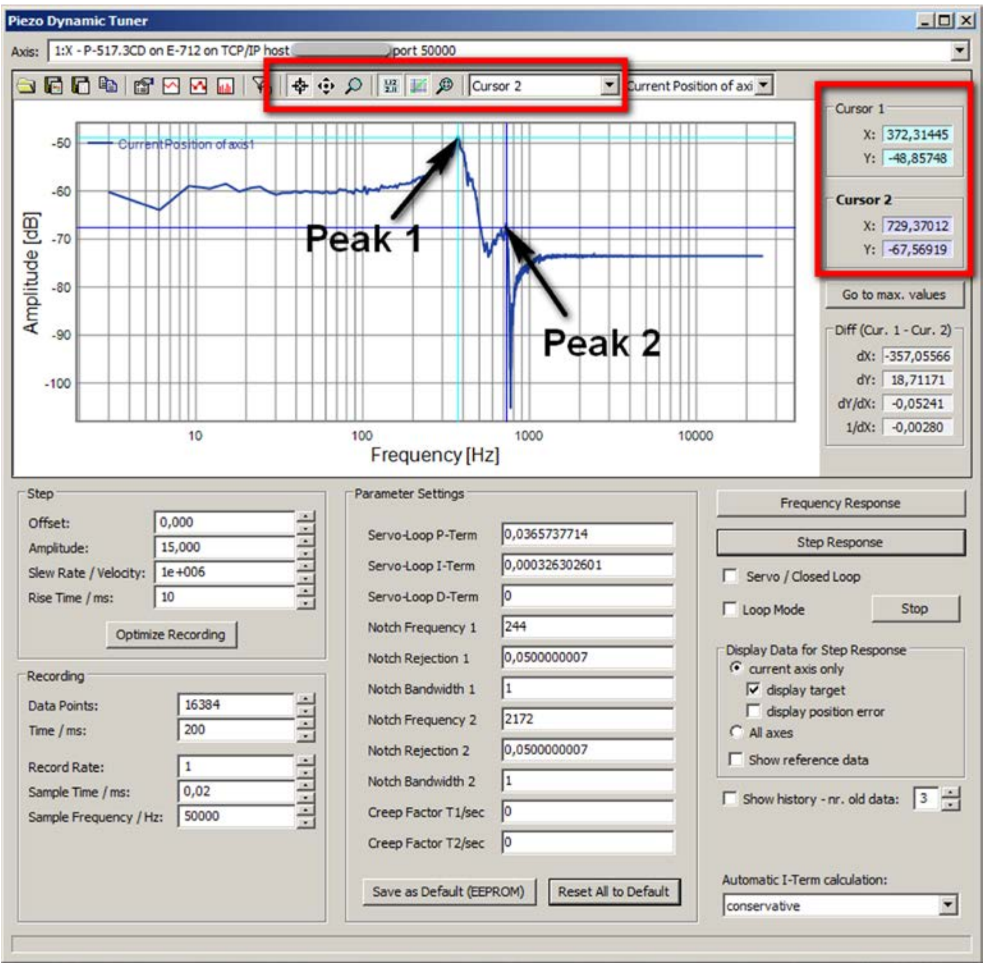
a) Show cursors and enable cursor motion using the buttons shown in the figures below.





b) 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, cursor 1 is at the first resonance peak (372.31445 Hz), and cursor 2 is at the second (next higher) resonance peak (729.37012 Hz).

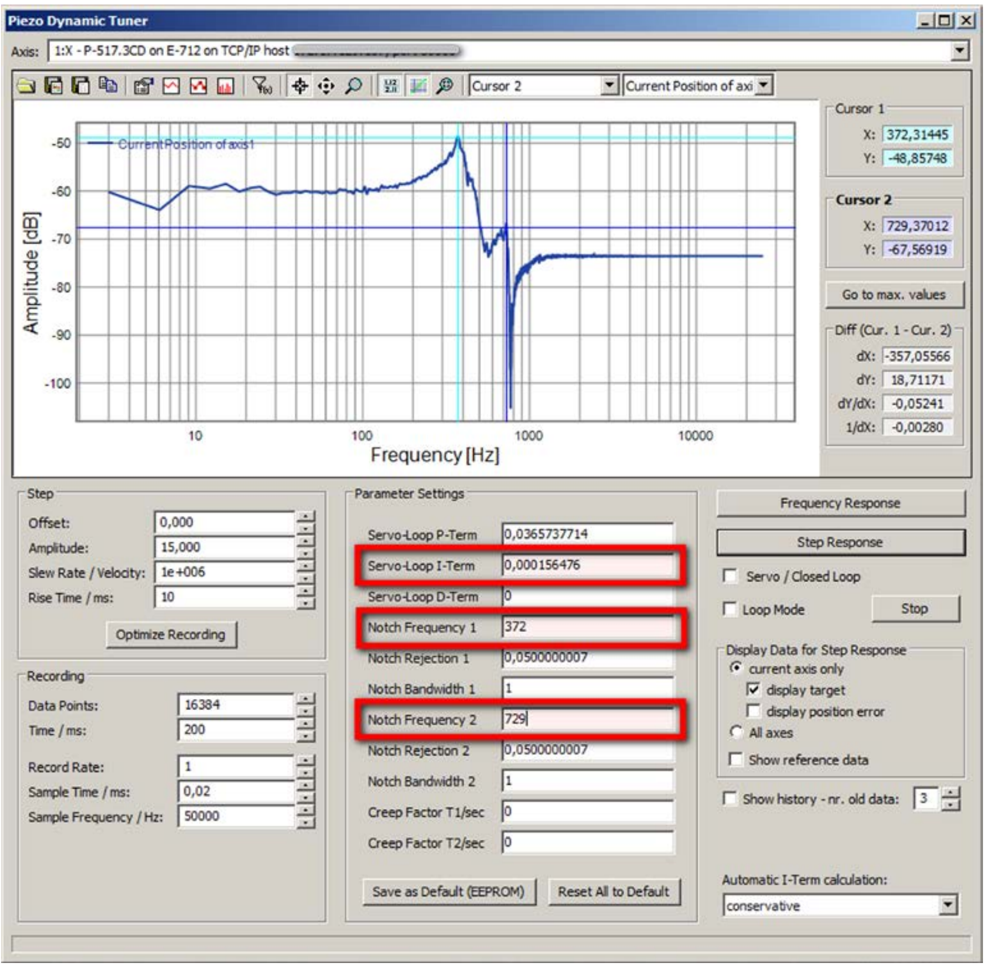


7. If necessary, adjust the notch filter settings in the ***Parameter Settings*** panel to the measured resonant frequencies (adjustment is necessary if the values significantly differ).

   a) Enter the frequency value of the first resonance peak in the ***Notch Frequency 1*** field (in Hz). You can either right-click the field with the mouse and select the value from a menu, or type the value in the field.

Note that depending on the selection for ***Automatic I-Term calculation***, the ***Servo-Loop I-Term*** value is changed too automatically when you change the ***Notch Frequency 1*** value (for details, see p. 134).

b) If you have measured a second resonance peak, enter the frequency value of the second resonance peak in the ***Notch Frequency 2*** field (in Hz). If the second notch filter is deactivated (rejection value = 1), change the rejection value to 0.05 in the ***Notch Rejection 2*** field.



8. Save or discard the new settings in the ***Parameter Settings*** panel:

   – If you want to keep the new settings, save them to the nonvolatile memory of the E-727 by clicking the ***Save as Default (EEPROM)*** button.

   – If you want to discard the new settings and reset the parameter values to their defaults (i.e. to their values from nonvolatile memory), click the ***Reset All to Default*** button.

Checking and Optimizing the Servo-Control Parameters

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

For further details regarding the PID algorithm, see “Control Algorithms”, p. 35.

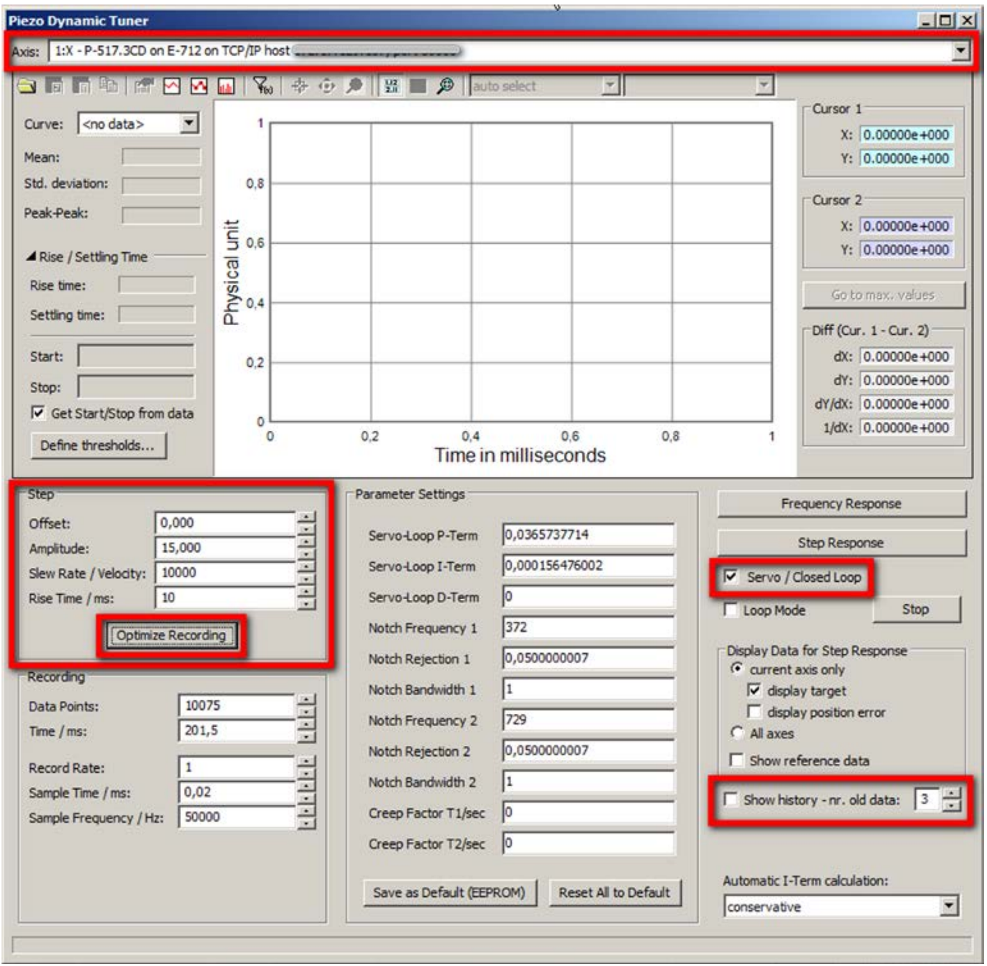
The optimization of the servo-control parameters is typically done empirically: The response of the axes to a step (“step response”) is analyzed under various values in closed-loop operation.

INFORMATION — SCREENSHOTS

The screenshots in the following instructions were created with an E-712 digital multi-axis controller. With E-727, some values may differ, but the procedure outlined in the screenshots is as with E-712.

Proceed as follows for each axis:

1. Read the “General Notes on Servo-Controller Dynamic Tuning” (p. 134).
2. Make sure the stage is mounted in exactly the same way as in the application. The load on the stage is especially important.
3. Make sure that the notch filter(s) are properly adjusted. For details, see “Adjusting the Notch Filter(s) in Open-Loop Operation” (p. 136).
4. In the main window of PIMikroMove, open the ***Piezo Dynamic Tuner*** window via the ***E-727… > Dynamic Tuner …*** menu item.
5. Configure the step response in the ***Piezo Dynamic Tuner*** window:
   a) Make sure that the correct axis is selected (***Axis*** drop down list).
   b) Make sure that the axis is in closed-loop operation (***Servo / Closed Loop*** box is checked).
   c) Enter suitable values for the start value (***Offset:***) and the amplitude (***Amplitude:***) of the step in the ***Step*** panel. The start value should be 0, and the amplitude should be about 10 % of the axis travel range.
   d) By clicking the ***Optimize Recording*** button in the ***Step*** panel, optimize the number of data recorder points that will be read from the controller when the step response has been performed.
   e) If you want to compare the results of multiple step resonse measurements, check the ***Show history*** box and select the number of old recordings to be displayed.

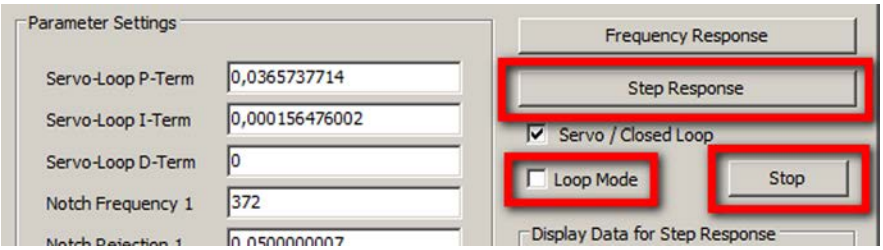


6. Perform and analyze the step response measurement in the ***Piezo Dynamic Tuner*** window:

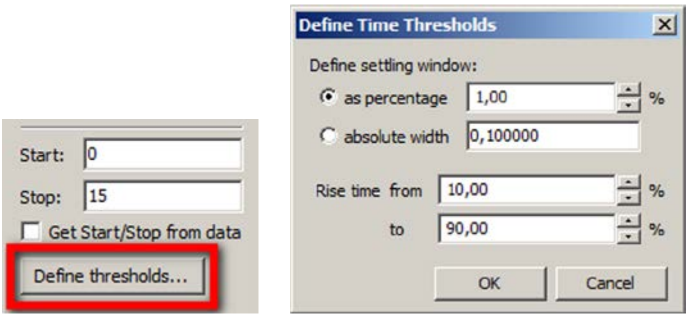
   a) Optional: Check the ***Loop Mode*** box to move the axis in a permanent loop.

      The loop mode is useful if you want to do the adjustment of the servo-control parameters during the motion. (The loop motion can be stopped at any time by clicking the ***Stop*** button.)

   b) Start the step response by clicking the ***Step Response*** button.

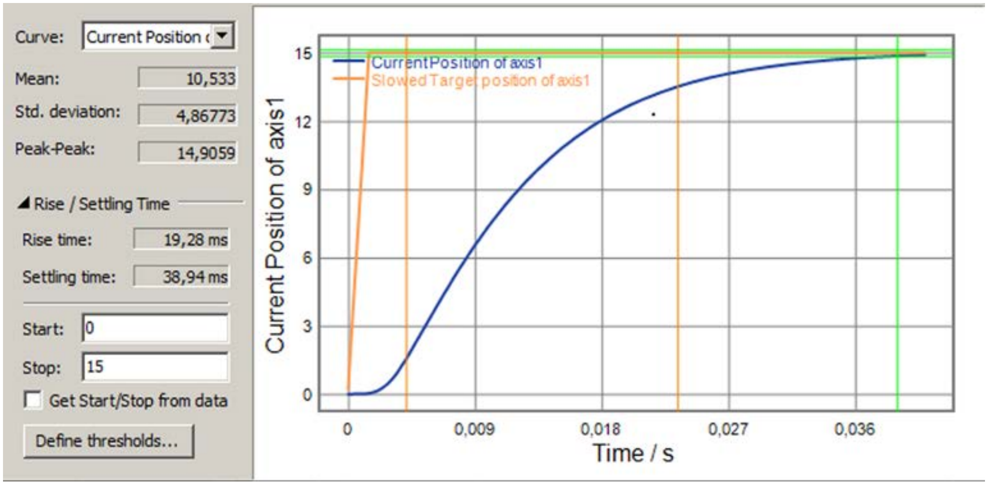


   c) Optional: Click the ***Define thresholds…*** button to open the ***Define Time Thresholds*** window. In the ***Define Time Thresholds*** window, you can adjust the thresholds which are used by the ***Piezo Dynamic Tuner*** window to calculate and display the rise time and settling time of the axis, based on the recorded step response measurement.

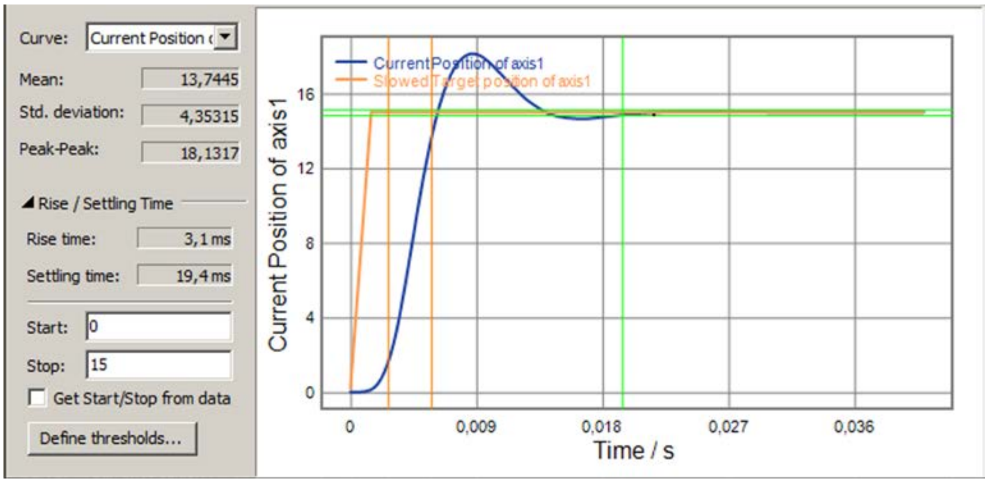


d) Check the step response result and compare it with the examples shown in the figures below.

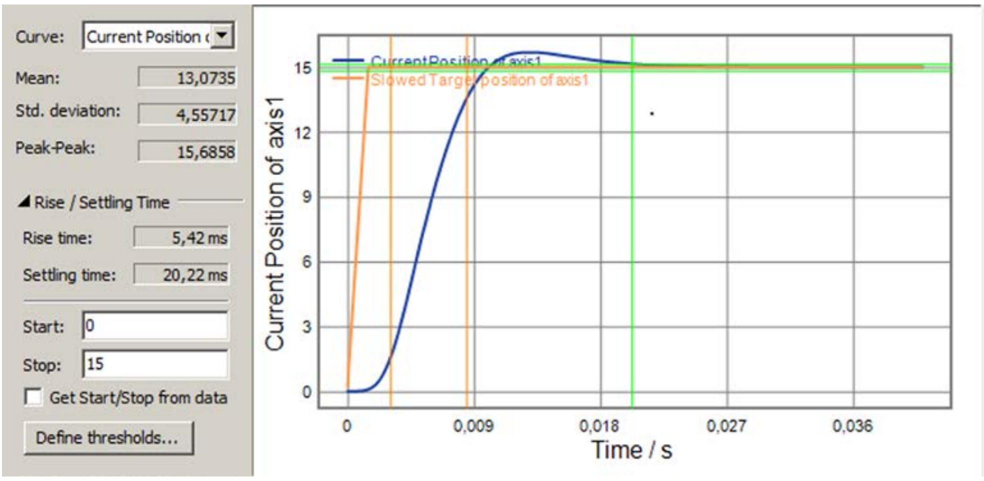
The rise rate of the step response is very low in the figure below. This means that the P term is too low and has to be increased.



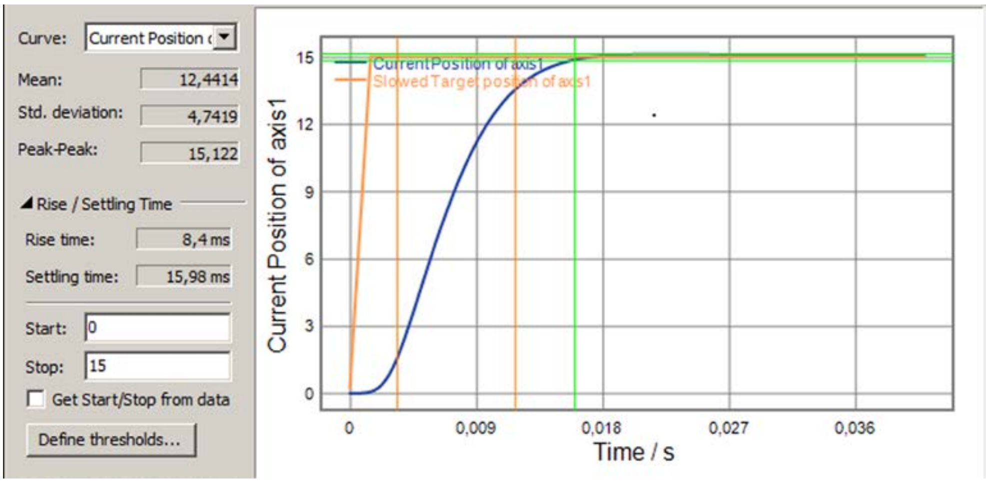
The figure below shows a step response with a high overshoot which means that the P term is too high and has to be decreased.



The figure below shows a step response with a small overshoot which means that the P term is still 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.



7. Save or discard the new settings in the ***Parameter Settings*** panel:
   - If you want to keep the new settings, save them to the nonvolatile memory of the E-727 by clicking the ***Save as Default (EEPROM)*** button.
   - If you want to discard the new settings and reset the parameter values to their defaults (i.e. to their values from nonvolatile memory), click the ***Reset All to Default*** button.

SPI Interface

Definition of Terms

PI-controller: E-727, the SPI slave.

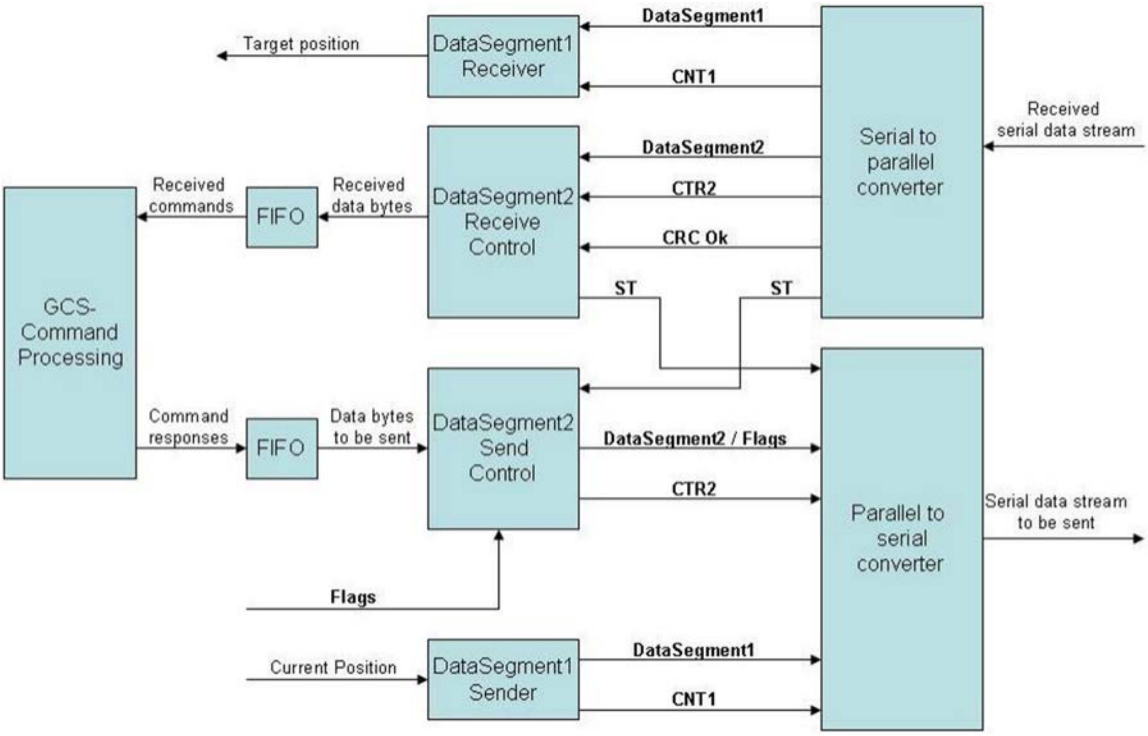
Host: SPI master device which sends commands to the PI-controller.

Overview

The serial data transfer via SPI is primarily designed for transferring position data from and to the PI-controller with minimum latency and update rates as high as the servo frequency of the PI-controller.

It is also possible to send and receive ASCII data so that the host has full access to the PI General Command Set (GCS).

The following block diagram shows the architecture on PI-controller side. The signals shown in this diagram are described in the following sections.



The PI-controller sends current position data to the host and receives new target position data from the host (for configuration options, see p. 145). For that purpose data segment 1 is used.

For transferring GCS commands data segment 2 is used.

As long as no responses to GCS commands are sent the PI-controller can send flags, for example OnTarget-flags.

All transfers are initiated by the host.

Configuration Parameters

For the axes of the E-727, the usage of data segment 1 of the SPI interface can be configured via parameters:

- Usage of the data received from the host
- Limits for position scaling

See "Parameters" beginning on p. 185) for more information on the parameters and their handling.

Configuring the Usage of Received Data

For the individual axes of the E-727, the ***FastIF Axis Input Usage*** parameter (ID 0x10000500) determines the usage of the data received from the host via data segment 1 of the SPI interface. The value of the ***FastIF Axis Input Usage*** parameter can be as follows:

0 = The data received from the host is ignored for the axis.

1 (default) = The data received from the host is used as target value for the axis (closed-loop or open-loop, depending on the servo mode).

2 = The data received from the host is used as current axis position. If this option is selected, the E-727 ignores the current axis position provided by the sensor channels via the input matrix (p. 29).

Configuring the Limits for Position Scaling

Note that position scaling is **not** necessary when the 32-bit floating point data type is set via the ***FastIF Data Type*** parameter (ID 0x10000501). In this case, the data can be read and written directly in axis units. Scaling is only necessary if any integer data type is set. Using the ***Used Range for Fast IF*** parameter (ID 0x10000506), you can select which limits are to be used for scaling the data sent via data segment 1 of the SPI interface:

0 = Fast IF data limits (parameters 0x10000502 and 0x10000503)

1 = Range limits of stage (parameters 0x07000000 and 0x07000001; provided by the ID-chip of the stage)

By default, the parameter is set to 1, i.e. the range limits of the stage are used for all axes.

Closed-Loop Operation

In closed-loop operation, the ***FastIFUsedLowLimit*** and ***FastIFUsedHighLimit*** parameters (0x10000504, 0x10000505) give the currently used limits for the data sent via data segment 1 of the SPI interface. Depending on the setting of ***Used Range for Fast IF*** (0x10000506), the currently used limits are identical to the range limits of the stage or to the Fast IF data limits.

Open-Loop Operation

In open-loop operation, it is necessary to extend data input range to cover the whole travel range. The limits for the input data sent via data segment 1 are calculated from the ***FastIFUsedLowLimit***, ***FastIFUsedHighLimit*** and ***FastIFOpenLoopLimitExtend*** parameters as follows:

FastIFOpenLoopLowLimit = FastIFUsedLowLimit - (FastIFUsedHighLimit - FastIFUsedLowLimit) * FastIFOpenLoopLimitExtend

FastIFOpenLoopHighLimit = FastIFUsedHighLimit + (FastIFUsedHighLimit - FastIFUsedLowLimit) * FastIFOpenLoopLimitExtend

FastIFOpenLoopLimitExtend must be ≥ 0. Normally 0.1 should be sufficient, providing 10% reserve in both directions.

Data Packet Definition

Data Frame

The following data packet definition is valid for both directions.

PID/ST	CTR2/CNT1	DS2/Flags	DS1	CRC-16
1 Byte	1 Byte	2 Bytes	CNT1 x 4 Bytes	2 Bytes

The data frame consists of the following fields:

PID/ST Status bits, explaned later

CTR2/CNT1 CTR2 = control bits for data segment 2,
CNT1 = data word counter for data segment 1 (1 word = 4 bytes)

DS2/Flags Data segment 2, for slow data transfer or for flag bits

DS1 Data segment 1, for fast position value transfer

CRC-16 Cyclic redundancy check of PID/ST, CTR2/CNT1, DS2/Flags and DS1

The data frame is for transportation of user data and consist of two data segments DS1 and DS2.

Data segment 1 (DS1) is used to transfer data which can be updated each servo-loop cycle. For the direction from host to PI-controller DS1 is used to transfer axis target position data (closed-loop operation) or open-loop control values (open-loop operation). For the direction from PI-controller to host DS1 is used to transfer the current axis position.

Whenever the PI-controller receives new target position data transferred by data segment 1, the PI-controller latches this new data and overwrites the old data. In case that a CRC error occurs and no new target position data could be received, the PI-controller uses the latched target position data which was most recently received.

Data segment 2 (DS2) is used for slow data transfer, for example for GCS-commands. DS2 transfers fragments of data from transmit fifo to the receiver fifo where data is collected again. Therefore the transfer of a GCS command normally will take several servo-loop cycles.

When data segment 2 transfers no slow data then flag bits are transferred. This is a cyclic transfer like with data segment 1.

CRC-16 Generation

The CRC-16 of PID/ST, CTR2/CNT1, DS2/Flags and DS1 is appended after packet data.

The following polynomial is used to generate the code (Start value 0xFFFF): $1 + x^2 + x^{15} + x^{16}$

The CRC-16 word is transferred with the highest bit first and the lowest bit last.

Transport Layer

Data Segment 1

Data segment 1 transports the fast 32-bit data. The number of data words which is currently sent is coded by the CTR2/CNT1 byte:

CTR2				CNT1			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SToggle	TwoBytes	DataCtrl		DS1 Data Word Counter			

The upper part CTR2 (bit 4 through bit 7) of the CTR2/CNT1 byte controls the transmission through data segment 2 and will be explained in the next section.

The lower part CNT1 (bit 0 through bit 3) counts the number N of currently transferred 32-bit data words. The value N can be any number between 0 and 15 so that the position data of up to 15 axes can be transferred. Note that with E-727, the behaviour is as follows:

- In the first data packet sent by the slave (E-727) after power-on, the number of axes is 0.
- In all subsequent data packets, the slave sends data for the same number of axes as used by the master in the previous cycle, provided that the number of axes is less than or equal to 4. Should values for more than 4 axes be sent by the master, the slave limits the response to 4 axes.

Data Word 1	Data Word 2	.....	Data Word N
4 Byte	4 Byte	4 Byte	4 Byte

The 32-bit data words are send with the highest bit first and the lowest bit last. The following table shows one 32-bit data word of the data stream:

Data Word i						
b31	b30	b29	b28	……	b1	b0

Data Segment 2

Data segment 2 is used for slow transfer of GCS commands and their responses. The length of data segment 2 is 2 bytes. When no data is transmitted by data segment 2 then it is used for the cyclic transfer of flag bits. In general a GCS command and its response is longer than 2 bytes and so multiple transfer cycles are needed to transfer a complete command or a command response. In the following, GCS commands or command responses which are transferred by data segment 2 are referred to as “data stream” whereas the part of a GCS command or of a command response which is transferred during one transfer cycle is referred to as “data fraction”.

Both the host and the PI-controller need an additional logic to process data segment 2. In the following, these control units are called sender and receiver. The two transfer directions from host to PI-controller and from PI-controller to host are working completely independent from each other. The architecture of sender and receiver can be the same on host and PI-controller side.

For controlling the transmission via data segment 2 the following control and status bytes are transferred. See also next section for the transport layer flow charts (p. 150):

CTR2				CNT1			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SToggle	TwoBytes	DataCtrl		DS1 Data Word Counter			

The bits of CTR2 (bit 4 through bit 7) are used by the sender of a data stream. The bits have the following meaning:

DataCtrl (bit 5, bit 4):

(0, 0) = No slow data transferred. The values of bits 6 und 7 (TwoBytes-Flag und SToggle-Flag) are invalid. The bytes of data segment 2 ares used for cyclic flag transfer, for example OnTarget flags from PI-controller to host or Servo mode change bits from host to PI-controller.

(0, 1) = Initialize stream transmission on receiver side. SToggle-Flag is transferred so that the RToggle-Flag on receiver has a definite initial value. No slow data transferred. The bytes of data segment 2 are used for cyclic flag transfer.

(1, 0) = A data fraction of the stream is transmitted by data segment 2. The TwoBytes-Flag und SToggle-Flag are valid and must be evaluated by the receiver.

(1, 1) = Last data fraction of a stream is transmitted by data segment 2. The receiver has the possibility to generate an interrupt so that the receiver FIFO can be read out. The E-727 will also set this control value when large data blocks are transferred or when the speed is high before the complete data block was transferred (before terminating line feed) so that one large block is split up in a number of smaller data blocks.

TwoBytes (bit 6):

0 = One data byte is transferred by data segment 2. The second byte of data segment 2 is invalid.

1 = Two bytes are transferred by data segment 2.

The TwoBytes-Flag is valid only when bit 5 is “1” (DataCtrl = (1, 0) or DataCtrl = (1, 1)).

SToggle (bit 7):

In principle the SToggle bit is just the lowest bit of a data counter.

The toggle bit is used to show the receiver that a new fraction of data is sent. The sender shows this by inverting the SToggle value each time when a new data fraction is loaded from the sender FIFO to the transmission register. When the receiver detects that the SToggle changed its value compared with the last data segment 2 transfer cycle then it must save the received data.

When the receiver sends an acknowledge it will reflect the very last SToggle value. For this purpose the the RToggle bit of the PID/ST byte is used (see description of the PID/ST byte).

The SToggle-Flag is only valid when DataCtrl ≠ (0, 0).

For the acknowledge response the lower bits of the PID/ST byte are used.

PID				ST			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	1	0	CRCError	RToggle	ACK

The bits of ST (bit 0 through bit 2) are set by the receiver of a data stream. The bits have the following meaning (bit 3 is currently not used and reserved for future applications):

ACK (bit 0):

0 = No data and no stream initialization received through data segment 2, or data processing is not yet finished, or CRC error occured.

1 = Acknowledge: receiver tells that valid data was received through data segment 2 and has been processed, or a stream initialization has been processed.

When this bit is zero then the RToggle bit has no meaning.

RToggle (bit 1):

When the ACK bit is non-zero then this bit reflects the most recently received SToggle bit. With this the sender can check if the acknowledge response matches the sent data fraction.

When the sender receives an acknowledge and the RToggle bit matches the SToggle bit then the sender transmits the next data fraction.

CRCError (bit 2):

Tells the sender that the last data packet transmission failed. This bit is always set when an CRC error occurred, regardless if data segment 2 of the data packet contained a data stream or not.

The bits of PID (bit 4 through bit 7) determine the interpretation of data segment 1 (DS1) and data segment 2 (DS2). At present, the PID bits always have to be set as shown above which means that DS1 is used for axis position data transfer and DS2 for transfer of GCS commands or flag bits.

Transfer of a fraction of a data stream (maximum 2 bytes):

Only one byte is transferred (DataCtrl ≠ (0, 0) and DataCtrl ≠ (0, 1) and TwoBytes = 0). In this case only byte 1 of data segment 2 is used whereas the unused higher byte is set zero:

Data segment 2	
0	Byte 1

Two bytes are transferred (DataCtrl ≠ (0, 0) and DataCtrl ≠ (0, 1) and TwoBytes = 1). Both bytes of data segment 2 are used:

Data segment 2	
Byte 2	Byte 1

When no bytes are transferred (DataCtrl = (0, 0) or DataCtrl = (0, 1)) then the PI-controller uses the two bytes of data segment 2 to send controller flags (OnTarget flags, for example). The PI-controller always sends the maximum possible number of flags. The CNT1 value does not influence the number of transferred flags.

Data segment 2					
Bit 15	Bit 14	…	Bit 2	Bit 1	Bit 0
Flag 16	Flag 15	…	Flag 3	Flag 2	Flag 1

The data segment 2 bytes are transferred with the highest bit first and the lowest bit last.

Connection Lost

When the connection is lost during the transfer of a data stream then the transmission must be aborted. For this purpose the number of CRC-errors should be counted which occur during the transmission of a data stream. When the CRC-error counter reaches a threshold then the transmission will be aborted.

It is up to the sender to set the threshold value where it aborts the transmission.

Because the sender tells the start of the transmission of a data stream the receiver will always be synchronized with the sender. So it is not necessary for the receiver to check the connection.

Transport Layer Flow Charts

The following flow charts show the transfer of data streams via data segment 2. It is shown for sender and receiver side separately.

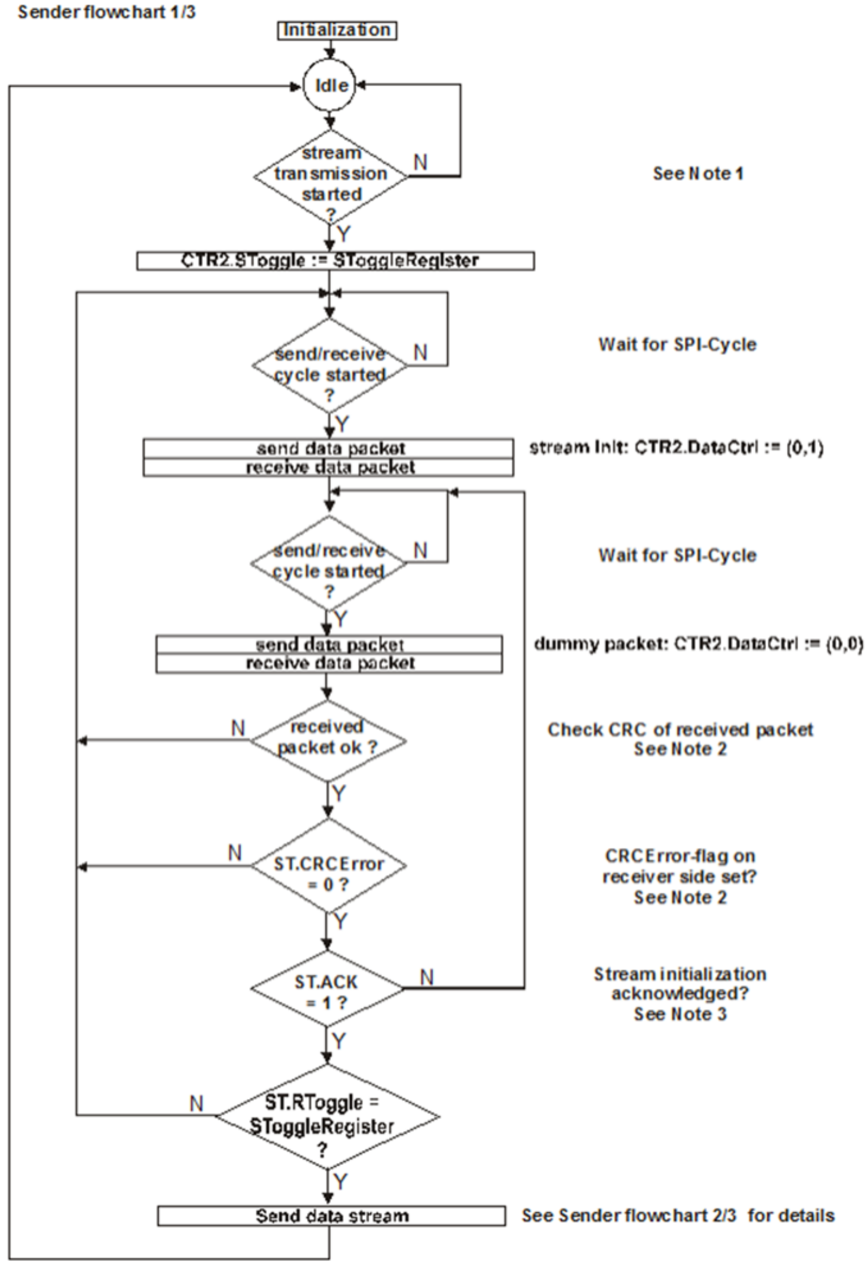
The flow charts do not show the transfer of the flag bits. Note that these bits are always transferred when data segment 2 is not used for data stream transfer.

Nomenclature

The following names are used in the flow charts:

- CTR2.SToggle, CTR2.TwoBytes, CTR2.DataCtrl: Control bits for data segment 2 as they are sent with the data packet
- DataCtrlReg: Buffer of DataCtrl bits which are set for data transfer. When data is sent then these bits are copied to CTR2.DataCtrl.
- ST.CRCError, ST.RToggle, ST.ACK: Status bits which are set and sent by the receiver.
- SToggleRegister: Toggle register implementation on sender side
- RToggleRegister: Local copy of toggle register on sender side.
- DataSegment2.Byte1, DataSegment2.Byte2: Data bytes transferred by data segment 2. Loading data segment 2 bytes with current flag bits while dummy packets are sent is not shown by the flow charts.

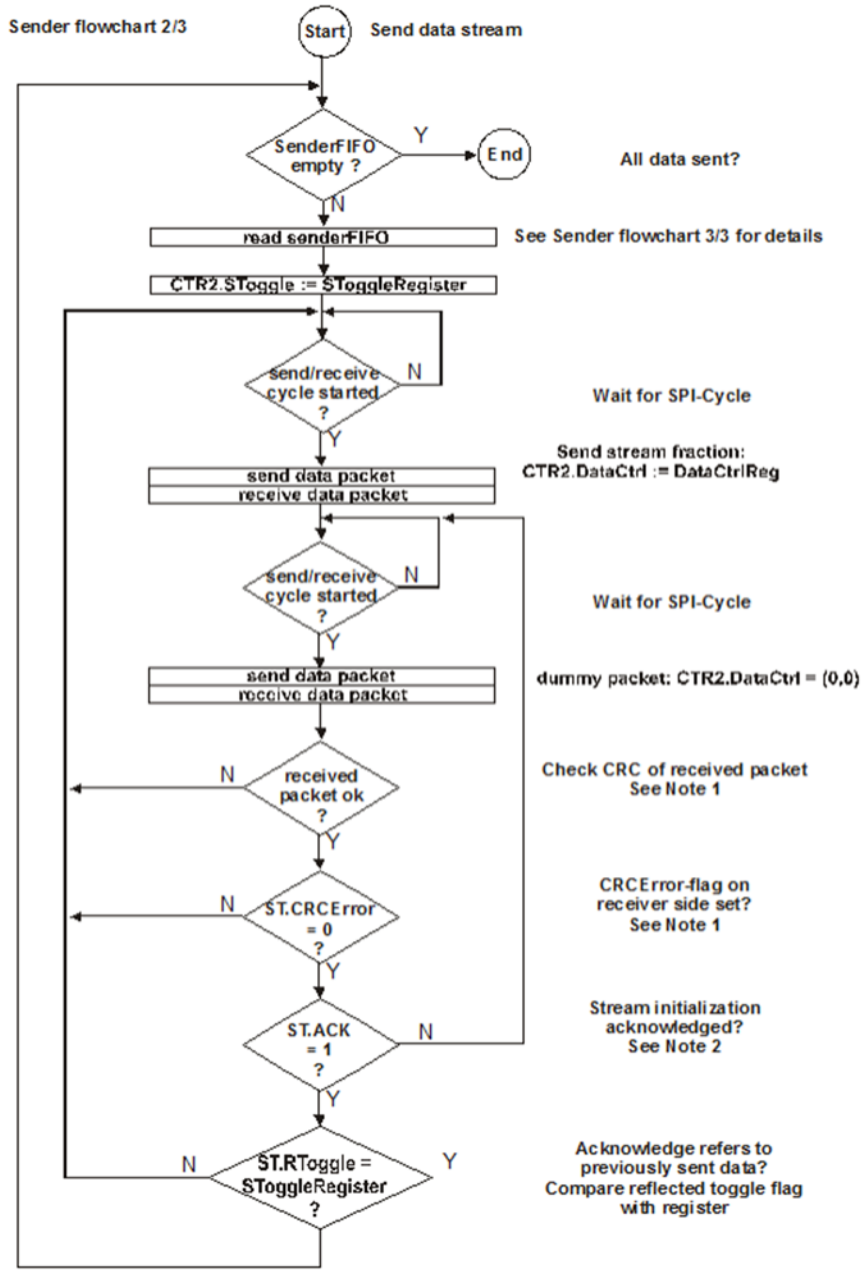
Sender Flow Charts



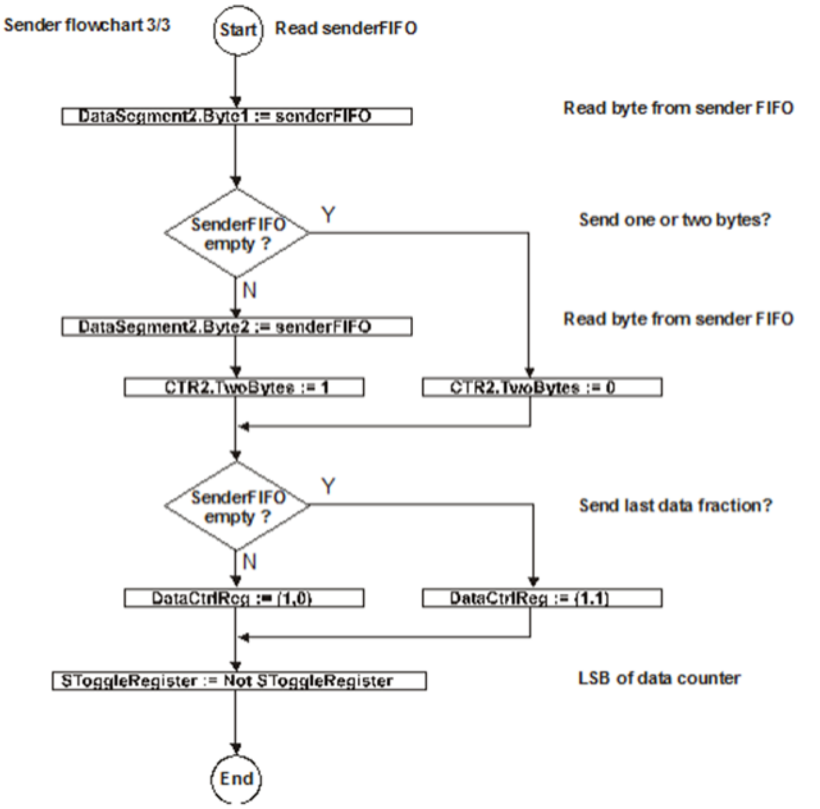
Note 1: Transmission is started after writing the sender FIFO, or when FIFO is full (depends on implementation). Writing of data to sender FIFO is not shown.

Note 2: To check if the connection is lost an error counter can be added. When too many errors occur the state machine returns to the idle state.

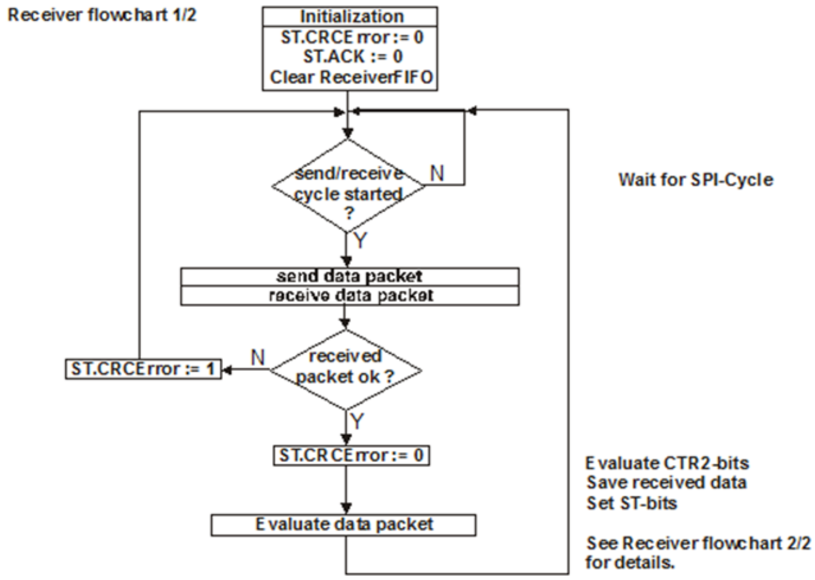
Note 3: Especially for slow systems a time out for acknowledge is useful, because acknowledges could be lost during the system evaluates the received data packets. In this case data should be sent again to force the receiver to sent acknowledge again.

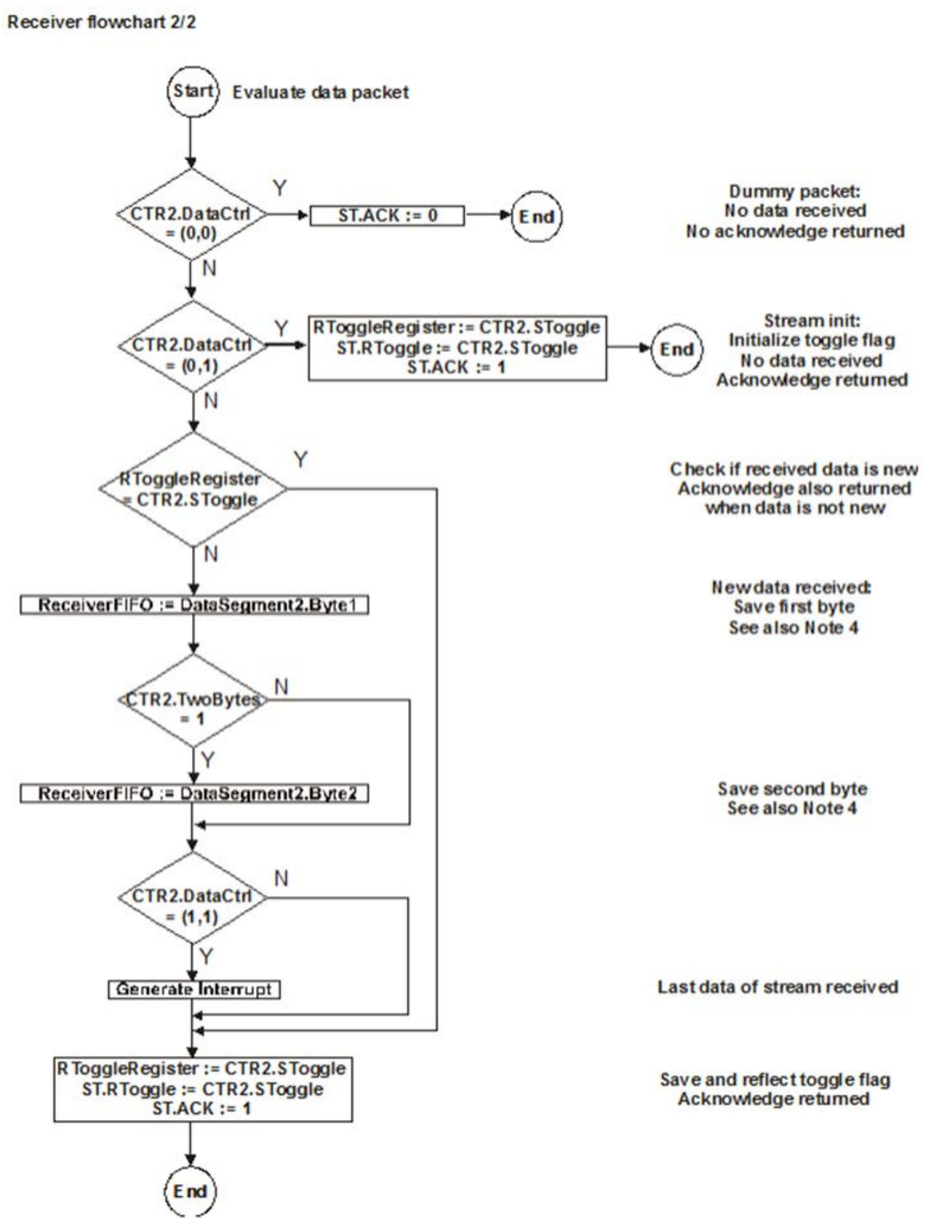


Sender flowchart 2/3
Start
Send data stream
SenderFIFO empty ?
Y
N
End
All data sent?
read senderFIFO
See Sender flowchart 3/3 for details
CTR2.SToggle := SToggleRegister
send/receive cycle started ?
N
Y
Wait for SPI-Cycle
send data packet
receive data packet
Send stream fraction:
CTR2.DataCtrl := DataCtrlReg
send/receive cycle started ?
N
Y
Wait for SPI-Cycle
send data packet
receive data packet
dummy packet: CTR2.DataCtrl = (0,0)
received packet ok ?
N
Y
Check CRC of received packet
See Note 1
ST.CRCError = 0 ?
N
Y
CRCError-flag on receiver side set?
See Note 1
ST.ACK = 1 ?
N
Y
Stream initialization acknowledged?
See Note 2
ST.RToggle = SToggleRegister ?
N
Y
Acknowledge refers to previously sent data?
Compare reflected toggle flag with register



Receiver Flow Charts





Note 4: Before writing the received data bytes to FIFO the receiver should check if the FIFO is not full. This is not shown by the flow chart because the implementation might be different, depending on the PI-controller type or host implementation. However, the following procedure is recommended

- Save the received DataSegmen2.Byte1, DataSegmen2.Byte2 and CTR2 bits temporarily to a buffer.
- Set the ST.ACK bit zero as long as the receiver FIFO is full. This tells the sender that the receiver is not ready to receive any more DataSegmen2 data bytes.
- When the FIFO is not full any more save the received data bytes from buffer to the FIFO.
- After saving the data bytes to FIFO the ST.ACK bit should be set to one and the received toggle bit should be returned to the sender (shown at the end of the flow

chart above). This tells the sender that the next data fraction can be sent.

Application Layer

In the following, some examples show how the fast axis data and the slow GCS command transfer is realized.

Axis Position Data Transfer

Data segment 1 is used for axis position data transfer.

The transfer of target positions from host to the PI-controller and the transfer of current position values from the PI-controller to host is done in the same way. Only the interpretation of the target values might be different, depending on the controller type (piezo controller, piezo motor controller, DC motor controller…) and the servo mode (closed-loop operation, open-loop operation). With E-727, both open-loop and closed-loop target values are interpreted as axis positions, due to the settings of the output matrix (p. 34).

In any case for each axis 32-bit values are used.

The number of axes which are sent depends on the controller type and on its settings.

When less target positions are sent than axes are available in the system then only the first CNT1 axes are updated. For example when the system has 4 axes and the host sends target position data for 3 axes than only axes 1 through 3 are permanently updated by the interface. Axis 4 remains at its initial position.

When the number of target positions which are sent is higher than the number of axes which are available then the controller will update the axis positions which are available and ignore the rest of the received data. For example when the system has 3 axes but the host sends target positions for 4 axes then the controller will update the 3 present axes and ignore the 4th target value.

Generally, the E-727 responds with the number of axes (CNT1) it has received from the host in the previous cycle, provided that this number is equal to or less than 4. Exception: In the first data packet sent by the E-727 after power-on, the number of axes is zero (CNT1 = 0).

In the following example data positions for two axes are sent and no GCS command is sent (no data sent through data segment 2). Instead of this servo flags are transferred.

PID/ ST	CTR2/ CNT1	DS2	Axis 1				Axis 2				CRC-16	
0x10	0x02	Flags	B11	B12	B13	B14	B21	B22	B23	B24	BC1	BC2

where B11 through B14 are the bytes of axis 1 position value, B21 through B24 are the bytes of axis 2 position value and BC1 and BC2 are the CRC-16 bytes.

The highest bits of the data words and the CRC-16 are send first and the lowest bits are send last, that is, bytes B11, B21, and BC1 are the highest bytes and bytes B14, B24, and BC2 are the lowest bytes.

OnTarget-Flag Transfer

As long as data segment 2 is not used for GCS command transfer (DataCtrl = 0,0) the two bytes are used to transfer flag bits. When the sender is the PI-controller then the OnTarget flags (ONT), ServoOn flags (SON), and Overflow flags (OVF) for axis 1 to axis 4 (A1 to A4) are sent:

Data segment 2												
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7 to Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OVF A4	OVF A3	OVF A2	OVF A1	SON A4	SON A3	SON A2	SON A1	0	ONT A4	ONT A3	ONT A2	ONT A1

Transfer of Servo-Mode Changing Commands

As long as data segment 2 is not used for GCS command transfer (DataCtrl = 0,0) the two bytes are used to transfer flag bits. When the sender is the host then command bits to change the servo mode (SMC) are transferred for axis 1 to axis 4 (A1 to A4).

Data segment 2				
Bit 15 to Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SMC A4	SMC A3	SMC A2	SMC A1

Changing the SMC bit changes the servo mode of the addressed axis as follows:

- A change from 0 to 1 switches the servo mode of the addressed axis on (closed-loop operation).
- A change from 1 to 0 switches the servo mode of the addressed axis off (open-loop operation).

Only changes of the SMC bits have an effect on the servo mode. Therefore the first servo-mode changing command which is received by the E-727 after power-on is saved internally by the E-727 and is only used to detect changes of the SMC bits.

GCS Command Transfer

GCS commands are transferred in ASCII format.

The following example shows the transfer of a command to switch on the servo mode for of axis 2. The GCS representation is:

SVO 2 1

The representation as ASCII numbers, including spaces and terminating line feed, is:

S	V	O	Space	2	Space	1	Line feed
0x53	0x56	0x4F	0x20	0x32	0x20	0x31	0x0A

For this command eight characters must be transferred and so four transfer packets are needed. In this example only one axis is transferred in data segment 1 (CNT1 = 1). Before the transmission of the actual data can be started the initialization packet must be sent to initialize the receiver's toggle bit. As initial value for SToggle "1" is assumed.

Initialization packet:

PID/ST	CTR2/CNT1	DS2		Axis 1				CRC-16	
0x10	0x91	0x00	0x00	B11	B12	B13	B14	BC1	BC2

DataCtrl = (0,1)

TwoBytes = 0

SToggle = 1

1st packet:

PID/ST	CTR2/CNT1	DS2		Axis 1				CRC-16	
0x10	0x61	0x56	0x53	B11	B12	B13	B14	BC1	BC2

DataCtrl = (1,0)

TwoBytes = 1

SToggle = 0

2nd packet:

PID/ST	CTR2/CNT1	DS2		Axis 1				CRC-16	
0x10	0xE1	0x20	0x4F	B11	B12	B13	B14	BC1	BC2

DataCtrl = (1,0)

TwoBytes = 1

SToggle = 1

3rd packet:

PID/ST	CTR2/CNT1	DS2		Axis 1				CRC-16	
0x10	0x61	0x20	0x32	B11	B12	B13	B14	BC1	BC2

DataCtrl = (1,0)

TwoBytes = 1

SToggle = 0

4th packet:

PID/ST	CTR2/CNT1	DS2		Axis 1				CRC-16	
0x10	0xF1	0x0A	0x31	B11	B12	B13	B14	BC1	BC2

DataCtrl = (1,1)

TwoBytes = 1

SToggle = 1

Data Transmission via SPI

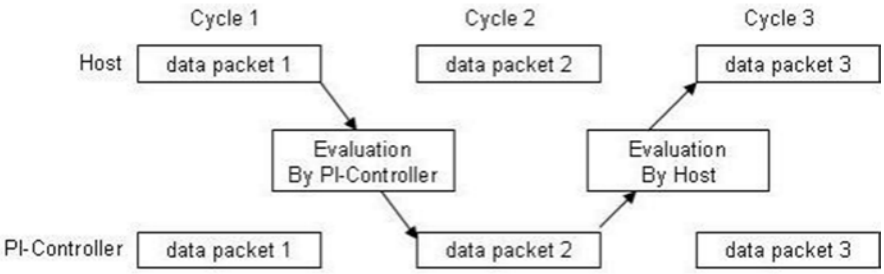
Data Line Definition

Name	Description	Host	PI-Controller
SCLK	Serial data clock from host	Output	Input
CS	Data word latch	Output	Input
MISO	Master Input Slave Output Date line. Data output changes with positive edge of SCLK	Input	Output
MOSI	Master Output Slave Input Date line. Data input is latched with falling edge of SCLK	Output	Input
LDAT	Latch / Load data: Falling edge latches sender data Rising edge loads receiver data	Output	Input
DCLK	Serial data clock from PI-controller	Input	Output

Packet Transmission

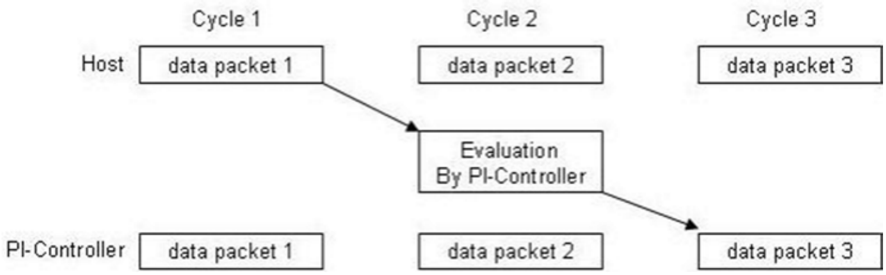
Because the SPI interface has only one clock the data packet transmission of host and PI-controller runs simultaneously. The PI-controller responds at least one cycle after the host command was received. This applies to the GCS commands and their responds which are transferred by data segment 2 and which need to be acknowledged by the receiver.

In the following example, data packet 2 sent by the PI-controller contains the response to the command contained in data packet 1, sent by the host: For example, the host sends the first data fraction with data packet 1, the PI-controller sends an acknowledge to this with data packet 2 and then the host can send the next fraction with data packet 3.



The data packets have the same format as described in the data packet definition section (p. 145).

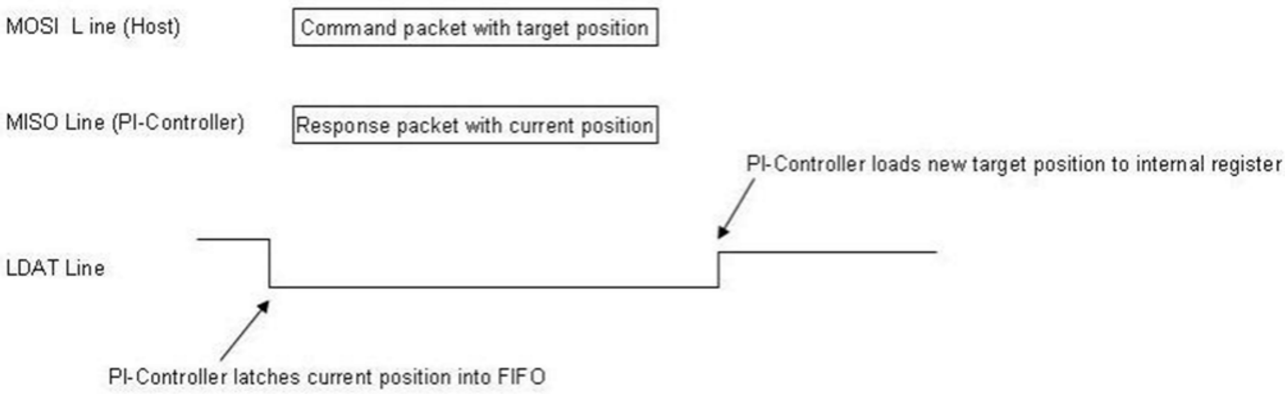
It is also possible that the PI-controller needs more than one cycle for processing of data which was sent by the host. In the following example the PI-controller needs one extra cycle for command data processing so that data packet 3 from PI-controller is the answer to data packet 1 from the host side.



LDAT: Latch Data Output / Load Data Input

With the LDAT command the input and output data of the PI-controller is synchronized with the command interface. The falling edge of the LDAT line latches the current position values of all axis into the output FIFO. With the next response packet the position values will be sent to the host.

With the rising edge of the LDAT line the most recently received data is loaded into the target position register of the PI-controller.



CS: Data Word Latch

The CS-signal divides the command packet into 16-bit data words. While the LDAT is active low throughout a command packet transmission the CS-signal becomes low while a 16-bit data word is transferred. With the rising edge of CS the controller internally saves the last 16-bit data word.

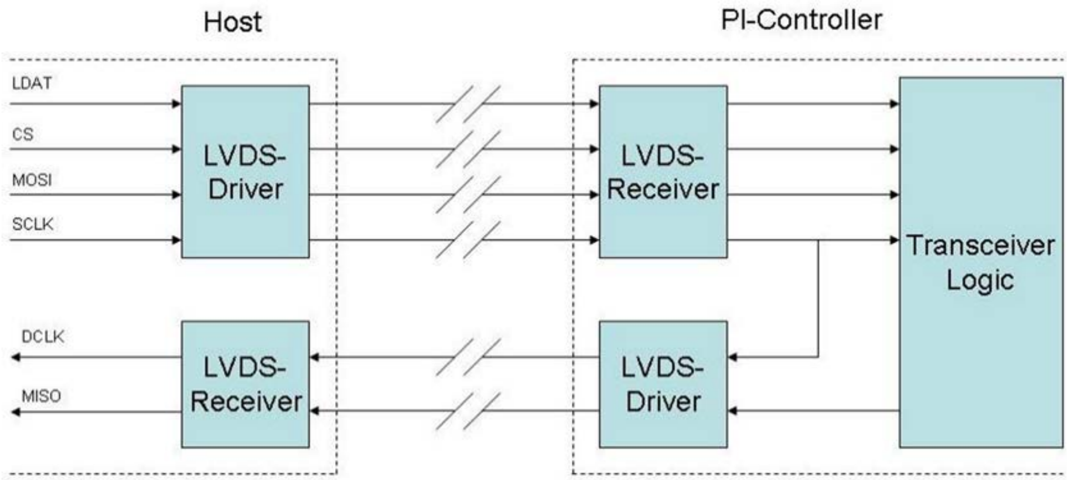
SCLK: Serial Data Clock from Host

The serial clock is active only while data bits are transferred. The reason for this is that some devices do not have enable inputs for the serial clock.

DCLK: Data Output Clock

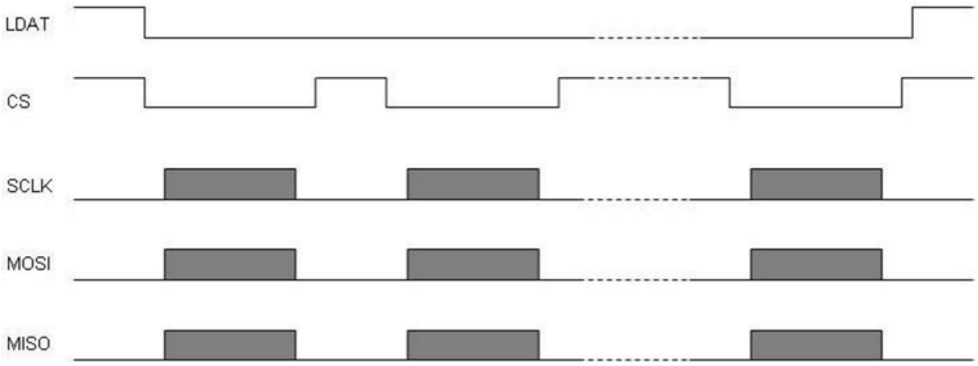
The PI-controller returns the SCLK as DCLK line (see next figure). The host can use the DCLK line for the reception of the MISO line. With this method the delays caused by the transmission line, the LVDS-divers and receivers can be compensated. For this reason higher data rates are possible with DCLK.

It is possible to run the interface without using the DCLK line. But in this case the delays on the transmission lines and the driver and receiver components become the limiting factor for the transmission rate. For proper data transfer it is therefore recommended to use the DCLK line in every case with data rates higher than 15 MHz.

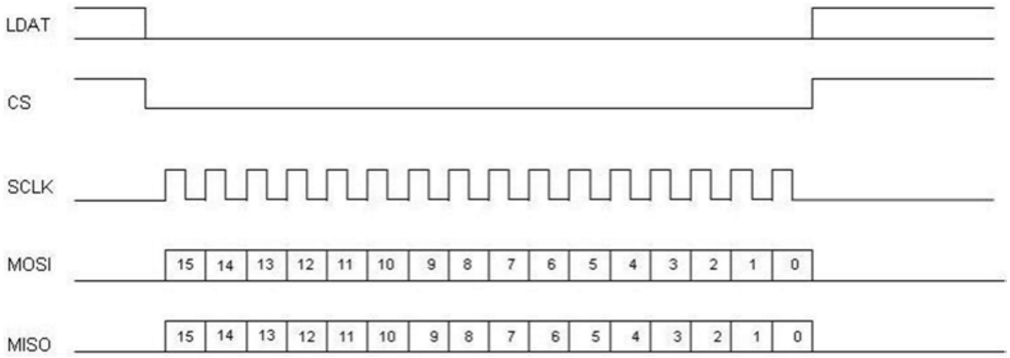


Serial Transmission

The following diagram shows the transfer of a complete command packet. LDAT is active low throughout a command packet transmission. CS is set low before a new word is transferred and set high when a word transfer is completed. SCLK is active only when data bits are transferred.



The following diagram shows the transfer of one single 16-bit data word. All bytes of a packet are transferred serially with the highest bit first. The data bits of the MISO line change with the rising edge of SCLK. The data bits of the MOSI line are latched with the negative edge of SCLK. The SCLK line is active only while data bits are transferred.



Physical Layer

Transmission Lines

Each signal is transmitted by a differential data line pair. For the pin assignment, see p. 233.

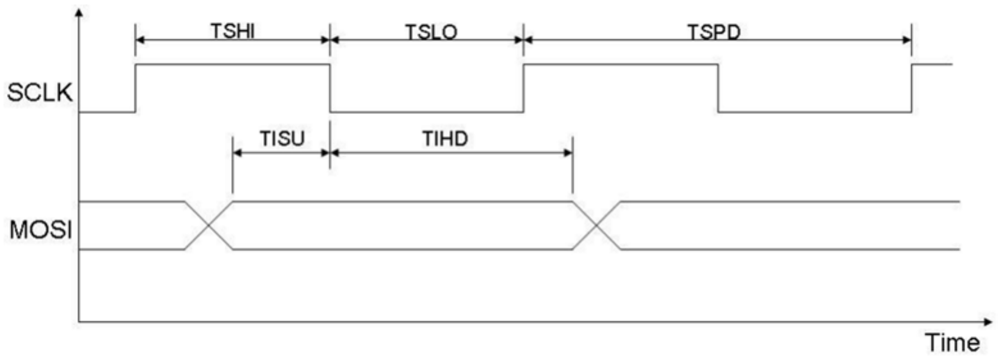
The electrical level is LVDS. On receiver side each line must be terminated with 100 ohm.

Timing Diagrams

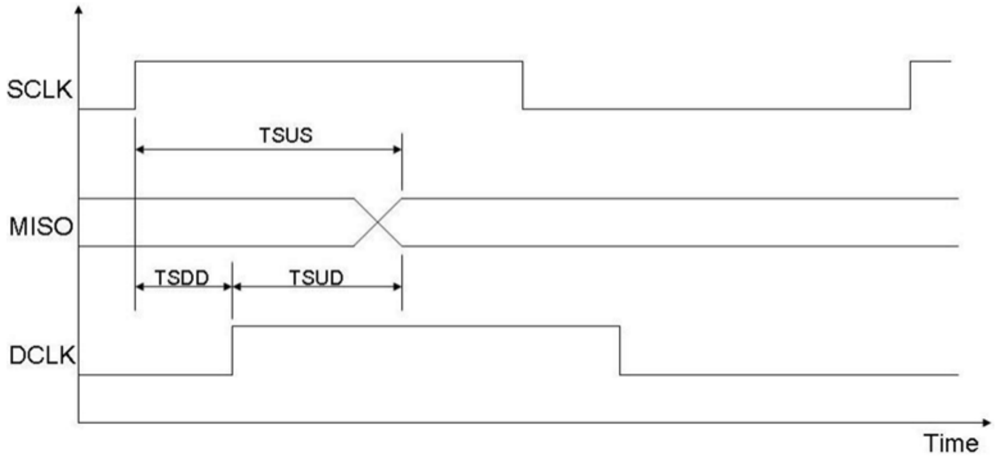
Note that the time values depend on controller type. The time values listed below were measured at the connector of E-727.

Symbol	Description	Min	Max
TSHI	SLCK high time	7 ns	
TSLO	SCLK low time	7 ns	
TSPD	SCLK period time	15 ns	
TISU	Input setup time: MOSI-bit valid before falling edge of SCLK	5 ns	
TIHD	Input hold time: MOSI-bit valid after falling edge of SCLK	0 ns	
TSUS	Output setup time: MISO-bit valid after rising edge of SLCK		15 ns
TSDD	SCLK to DCLK delay time		10 ns
TSUD	Output setup time: MISO-bit valid after rising edge of DLCK		6 ns
TFLS	Falling edge of LDAT before first rising edge of SCLK	0 ns	
TSRL	Last falling edge of SCLK before rising edge of LDAT	0 ns	
TLDH	LDAT high: Time between two commands	220 ns	
TFCS	Falling edge of CS before rising edge of SCLK	N/A	N/A
TSRC	Falling edge of SCLK before rising edge of CS	N/A	N/A
TCSH	CS high: Time between two 16-bit data words	N/A	N/A

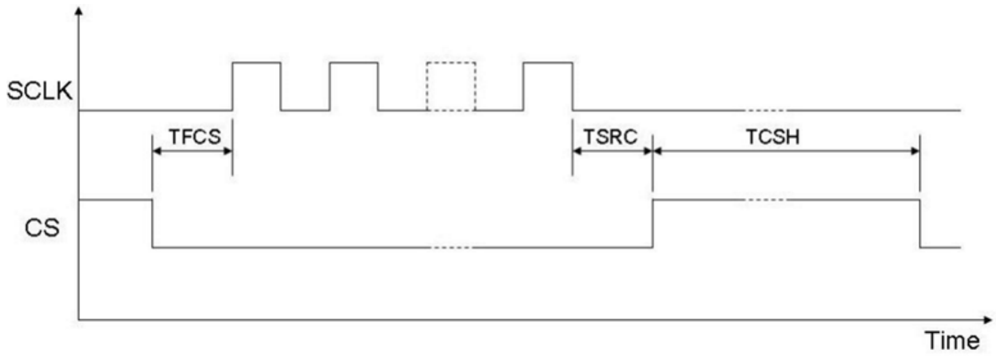
SCLK and data input:



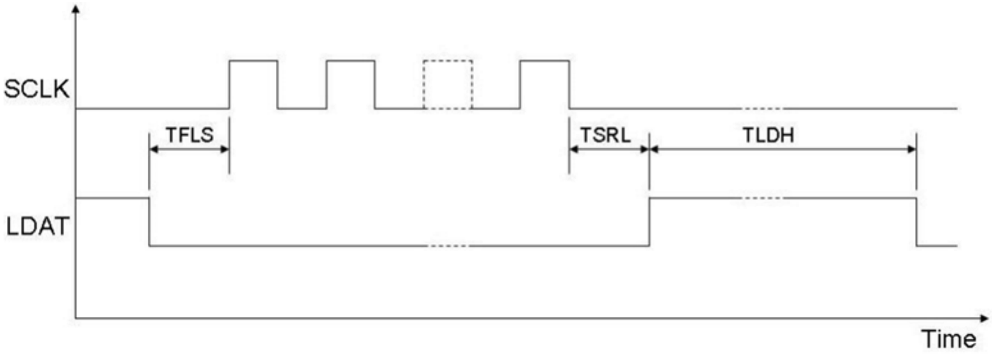
SCLK, DCLK, and data output:



SCLK and CS:



SCLK and LDAT:



EtherCAT Interface

Overview

E-727.xxxF and .xxxAF models are equipped with an EtherCAT interface for control by an EtherCAT master. The E-727 with the connected piezo nanopositioning system behaves like an intelligent multi-axis drive according to the CiA402 drive profile. The EtherCAT master specifies target positions and velocities for the axes of the E-727 and receives the position feedback from the axes.

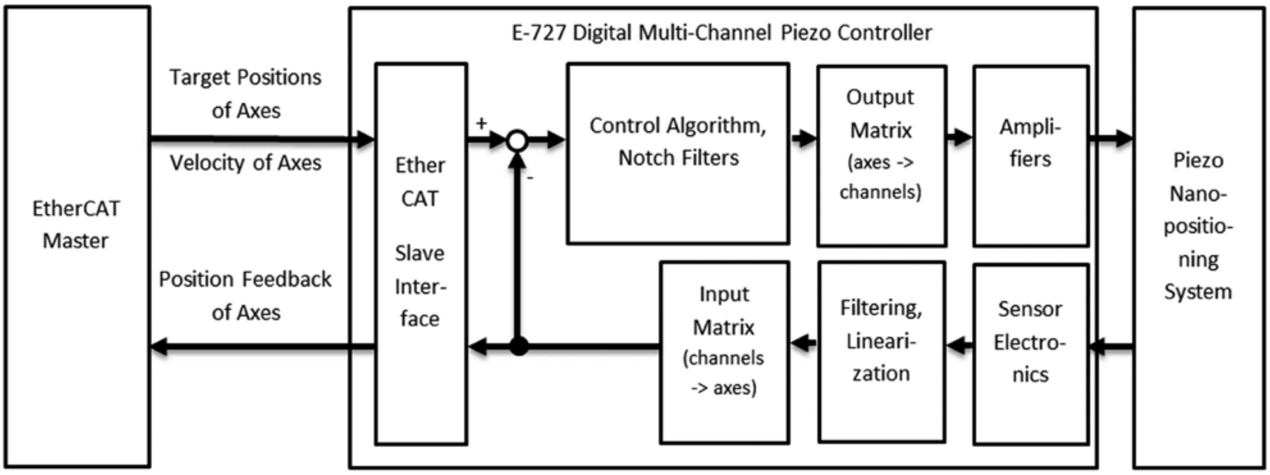


Figure 37: E-727 with piezo nanopositioning system controlled by EtherCAT master. EtherCAT communication state: OPERATIONAL; drive state: Operation Enabled.

The table below gives an overview over the most important characteristics of control via EtherCAT master for E-727:

Field bus protocol:	EtherCAT (CoE = CANopen over EtherCAT)
Drive profile:	CiA402 Drive Profile (IEC 61800-7-201)
Cycle time for specification of data (e.g., target positions) by the EtherCAT master	2 ms
Supported modes of operation according to IEC 61800-7-201:	Cyclic Synchronous Position (CSP) Profile Position (PP) Homing (performs AutoZero)
Supported synchronization modes:	SyncManager (SM)
Connector for connection to field bus:	RJ45 socket

For basic information on EtherCAT networks and the CiA402 drive profile, see the documents listed on p. 9.

For implementing a PI controller in TwinCAT 3.1, see the examples in the \Samples\TwinCAT directory of the E-727 CD which is in the scope of delivery.

Configuring the EtherCAT Network

Connecting the EtherCAT Master

NOTICE

Malfunction due to incorrect connection!

Incorrect cable connections can lead to incorrect addressing and communication failure.

- Use the **IN** RJ45 socket (on top) to connect the E-727 with the EtherCAT master (the **OUT** RJ45 socket (below) is intended for the connection of the next EtherCAT slave).
- Do not use EtherCAT and standard Ethernet together in a physical network. If possible, use cables with different colors for EtherCAT and standard Ethernet connections.

- Connect the EtherCAT master with the **IN** RJ45 socket of the E-727 via a suitable cable:
  - CAT 5 patch cable or higher, straight-through or crossover
  - Cable length: 0.3 to 100 m

Configuring the EtherCAT Master

The steps of configuration, start-up and operation of the EtherCAT master depend on the device used. See the documentation of your EtherCAT master for details.

For the integration of the E-727 into the EtherCAT network, the XML file delivered by PI (Physik_Instrumente_Drives.xml) must be saved on the EtherCAT master. The XML file is available in the /EDS directory of the E-727 CD which is in the scope of delivery. Follow the installation instructions in the documentation of your EtherCAT master.

Furthermore, the following settings must be changed on the EtherCAT master to adapt it to the E-727.

To be set per axis:

- Scaling factor numerator: Must be set to the same value as the ***FieldbusAxis Resolution*** parameter (ID 0x10000800) of the E-727. For further details, see "Scaling Position and Velocity Values" (p. 165).
- Scaling factor denominator: Must remain 1.

To be set once for the whole E-727 system:

- Synchronization mode: SyncManager (SM)

Scaling Position and Velocity Values

Position and velocity values (target position, actual position, profile velocity) are transferred between the EtherCAT master and E-727 as integer values.

The scaling of the transferred values is determined by the ***FieldbusAxis Resolution*** parameter (ID 0x10000800) of the E-727 in combination with the scaling factor numerator of the EtherCAT master. Both values must be identical.

The default value of the ***FieldbusAxis Resolution*** parameter is 0.0001 which results in the following scaling of position values:

Min. value EtherCAT master	Min. value E-727 with axis unit µm	Min. value E-727 with axis unit µrad	Min. value E-727 with axis unit mrad
1	0.0001 µm = 0.1 nm	0.0001 µrad = 0.1 nrad	0.0001 mrad = 0.1 µrad

You can change the scaling to make optimum use of the position resolution of the E-727. In order to change the scaling, adapt both the ***FieldbusAxis Resolution*** parameter of the E-727 and scaling factor numerator of the EtherCAT master.

You can read or adjust the ***FieldbusAxis Resolution*** parameter value via the TCP/IP or USB interface only; for details, see “Parameters” (p. 185). Parameter changes must be done in the nonvolatile memory and require a reboot of the E-727 to become effective in the EtherCAT bus.

The conversion between values of the EtherCAT master and E-727 (EtherCAT slave) is as follows:

SlaveTargetValue = MasterTargetValue * FieldbusAxisResolution
MasterCurrentValue = SlaveCurrentValue / FieldbusAxisResolution

Changing the scaling also changes the value of the position range limit (object 0x607B) of the axis.

EtherCAT Communication

Communication State Machine

The states of the connection to the EtherCAT network are described by a communication state machine. The communication state machine is independent of the drive state machines which are described on p. 169.

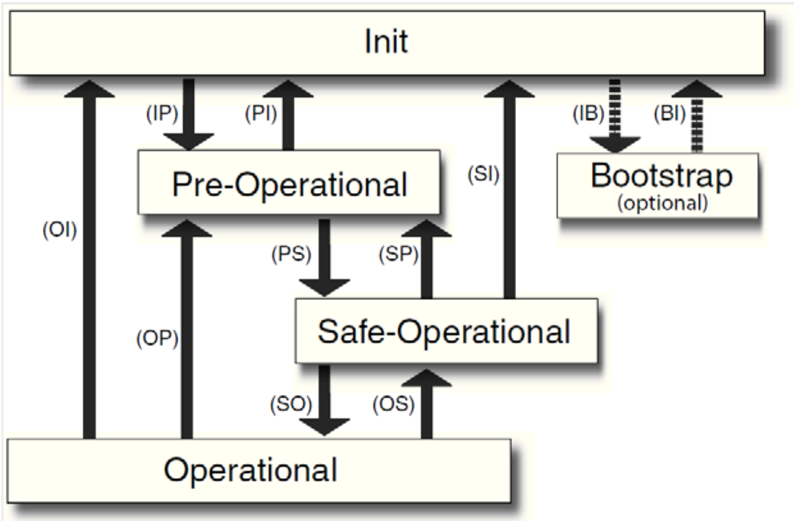


Figure 38: EtherCAT communication state machine

State	Description
Init	Initialization, the device is starting
Pre-Operational	The device is ready for parametrization. Mailbox communication is possible.
Safe-Operational	PDO input data (TxPDO device) is read. PDO output data (RxPDO device) is ignored.
Operational	Cyclic I/O communication PDO output data (RxPDO device) is processed.

Transition	Description
IP	Start mailbox communication
PI	Stop mailbox communication
PS	Start input update
SP	Stop input update

Transition	Description
SO	Start output update
OS	Stop output update
OP	Stop output update / stop input update
SI	Stop input update / stop mailbox communication
OI	Stop output update / stop input update / stop mailbox communication

Communication Details with E-727

EtherCAT communication can be established when the power-on/reboot sequence of the E-727 is finished (**Power** LED is continuously lit). The power-on/reboot sequence for E-727 models with EtherCAT interface takes about 30 seconds.

The PC interfaces of the E-727 (TCP/IP, USB) are disabled when EtherCAT communication is running (i.e., when E-727 is in OPERATIONAL communication state). They will be enabled again after a delay of 100 ms when the E-727 has left the OPERATIONAL communication state.

The **RUN** LED of the E-727 indicates the current state of the EtherCAT communication state machine. For details, see the description in “Product View” (p. 14).

INFORMATION

EtherCAT communication must not run when you attempt to change parameter settings of the E-727 (p. 185) or update the firmware of the E-727 (p. 204).

INFORMATION

If you want to use the TCP/IP interface of the E-727 in a network with DHCP server when the EtherCAT communication does not run, the following is recommended:

➢ Connect the network cable for TCP/IP communication while the EtherCAT communication is still running or even before you switch on the E-727.

Controlling the Axes

Necessary Adjustments

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.

➢ Only control the E-727 via EtherCAT master after you have optimized 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. 136) and "Checking and Optimizing the Servo-Control Parameters" (p. 140). Optimization must be done via the TCP/IP or USB interface.

Cycle Time

During control via EtherCAT master, the minimum cycle time for specifying data for the E-727 (2 ms) must be observed. If the actual cycle time is shorter than the minimum cycle time of the E-727, the axes do not move, and an error occurs.

Object Dictionary

The object dictionary is a list of variables and parameters of the device to be controlled, in this case an axis of the E-727. For details, see the appendix (p. 236). Each entry is addressed via an index of its own and possibly via a sub-index. The entire index space is subdivided into several ranges.

The object range 0x6000 to 0x67FF is dedicated to the CiA402 drive profile. For multi-axis devices like the E-727, the object range 0x6000 to 0x67FF is shifted by 0x800 per axis as described in the IEC 61800-7-201. This way, each axis has its own objects in this object range.

See „Homing Mode“ (p. 173), „Profile Position Mode“ (p. 174) and „Cyclic Synchronous Position Mode“ (p. 177) for the CiA402-specific objects supported in the individual modes of operation.

Drive State Machine

For each axis of the E-727, a separate drive state machine is used. Each drive state machine has its own control word (p. 170) and status word (p. 172).

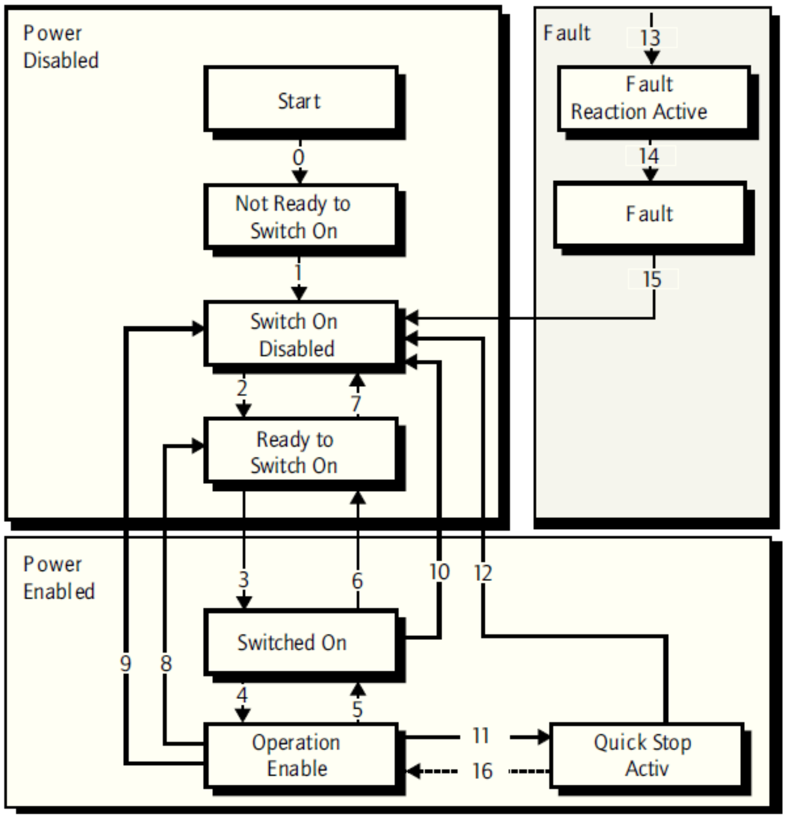


Figure 39: Drive state machine details

The individual drive state machines for the axes of the E-727 can have different states.

State of drive state machine	Characteristics
NOT READY TO SWITCH ON	This state is performed automatically by the axis (initialization). Drive function disabled.
SWITCH ON DISABLED	This state can be passed through automatically by local signals of the axis. Drive function disabled.
READY TO SWITCH ON	Transition 2 into this state is to be performed by the EtherCAT master only if necessary. Drive function disabled.

State of drive state machine	Characteristics
SWITCHED ON	Transition 3 into this state can only be requested by the EtherCAT master. Drive function disabled. Target and set-point values are ignored.
OPERATION ENABLED	Transition 4 into this state can only be requested by the EtherCAT master. If the EtherCAT master sets bits 0, 1 and 3 simultaneously in the control word, the drive can pass from **Ready to switch on** via **Switched on** to **Operation enabled** with one control command. Drive function enabled, drive is ready to follow target and set-point values. Target and set-point values are processed.
QUICK STOP ACTIVE	Transition 11 into this state, and transitions 12 and 16 out of this state can only be requested by the EtherCAT master. Drive function enabled, drive is ready to follow target and set-point values. Target and set-point values are ignored.
FAULT REACTION ACTIVE	An error occured. Drive function enabled, drive is ready to follow target and set-point values. Target and set-point values are ignored.
FAULT	An error occured, and the **Fault reaction active** state has been passed. Drive function disabled.

All axes of the E-727 are in the **Switch on disabled** state after switching on or rebooting the E-727.

In order for the EtherCAT master to be able to initiate a transition to the **Ready to switch on** state, the following conditions must be met by the axes:

- The AutoZero procedure must be finished (takes about 3 seconds).
- No macro is running.
- The axis definition must allow closed-loop operation, i.e., at least one sensor and one piezo amplifier must be assigned via the input and output matrices of the E-727 (p. 29 and p. 34). Note: Normally, the matrices are set properly by PI before delivery.
- No wave generator is running.
- No analog control input is enabled.

The E-727 does not provide information on the status of the power-on/reboot sequence (p. 166) and the above listed procedures / control sources via the EtherCAT interface. For that reason, sufficient waiting time is required before a state transition to **Ready to switch on** is initiated.

A transition to the **Operation Enabled** state switches the axis to closed-loop control (p. 30). The axis remains in closed-loop operation until it is switched to open-loop control due to overheating (p. 173) or by an SVO command sent via TCP/IP or USB interface.

Control Word

The control word (object 0x6040) contains bits which trigger transitions between the states of the drive state machine, bits whose function depends on the selected mode of operation, and bits for manufacturer specific options.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Manufacturer specific					reserved		Halt	Fault reset	Operation mode specific			Enable operation	Quick stop	Enable voltage	Switch on

Table 1: All bits of the CiA402 control word

The transitions between the drive states are triggered by combinations of bits 0, 1, 2, 3 and 7 of the control word. If the **Quick stop active** state is not supported by the EtherCAT master, bit 2 of the control word must always be set.

Commanding the drive state transitions with combinations of bits 0, 1, 2, 3 and 7:

Command	Bit 7 Fault reset	Bit 3 Enable operation	Bit 2 Quick stop	Bit 1 Enable voltage	Bit 0 Switch on	Transition
Shutdown	0	x	1	1	0	2, 6, 8
Switch on	0	0	1	1	1	3
Disable voltage	0	x	x	0	x	7, 9, 10, 12
Quick stop	0	x	0	1	x	11
Disable operation	0	0	1	1	1	5
Enable operation	0	1	1	1	1	4
Fault reset	0 → 1	x	x	X	x	15

Table 2: Bits 0 to 3, and 7 of the CiA402 control word

The function of bits 4 to 6 and bit 8 of the control word depends on the selected mode of operation. For the modes of operation supported by the E-727, these bits are used as follows:

Bit / Operation mode	8	6	5	4
Homing	Halt	reserved	Reserved	Homing operation start
Profile Position	Halt	reserved	Change set immediately	New setpoint
Cyclic Synchronous Position	reserved	reserved	Reserved	reserved

Table 3: Bits 4 to 6, and 8 of the CiA402 control word

With the E-727, the manufacturer specific bits of the control word (bits 11 to 15) are not used.

Status Word

The status word (object 0x6041 for the first axis, object 0x6841 for the second axis, ...) contains bits for the following information:

15	14	13	12	11	10	9	8
Manufacturer specific		Operation mode specific		Internal limit active	Target reached	Reserved	Manufacturer specific

7	6	5	4	3	2	1	0
Reserved	Switch on disabled	Quick stop	Voltage enabled	Fault	Operation enabled	Switched on	Ready to switch on

Table 4: All bits of the CiA402 status word

Bits 0 to 6 show the state of the drive state machine:

State	Bit 6 Switch on disabled	Bit 5 Quick stop	Bit 4 Voltage enabled	Bit 3 Fault	Bit 2 Operation enabled	Bit 1 Switched on	Bit 0 Ready to switch on
Not ready to switch on	0	x	0	0	0	0	0
Switch on disabled	1	x	0	0	0	0	0
Ready to switch on	0	1	1	0	0	0	1
Switched on	0	1	1	0	0	1	1
Operation enabled	0	1	1	0	1	1	1
Fault	0	x	x	1	0	0	0
Fault reaction active	0	x	x	1	1	1	1
Quick stop active	0	0	1	0	1	1	1

Table 5: Bits 0 to 6 of the CiA402 status word

Bit 10 shows the on-target state. Details differ depending on the selected mode of operation.

The meaning of bits 12 and 13 depends on the selected mode of operation. For the modes of operation supported by the E-727, these bits show the following:

Bit / Operation mode	13	12
Homing	Homing error	Homing attained
Profile Position	Reserved*	Setpoint acknowledge
Cyclic Synchronous Position	Reserved*	0 = Target position ignored 1 = Target position shall be used as input to position control loop

*Bit 13 is always 0.

Table 6: Bits 12 and 13 of the CiA402 status word

Bits 7, 8, 9, 11, 14 and 15 are not used at present.

Modes of Operation

The mode of operation can be selected via the Modes of Operation object of the axis (0x6060 for the first axis, 0x6860 for the second axis, …). The selection is possible in the **Operation enabled** state. The current mode of operation is displayed via the Modes of Operation Display objects (0x6061 for the first axis, 0x6861 for the second axis, …).

E-727 supports the following modes of operation according to 61800-7-201:

Code	Mode
1	Profile Position (PP, p. 174)
6	Homing (performs AutoZero, p. 173)
8	Cyclic Synchronous Position (CSP, p. 177)

Behavior when an invalid mode of operation is selected for an axis:

- Mode of Operation Display of the axis shows 0.
- An error occurs for the axis.

Error Handling

The Error code object (0x603F) provides the code of the last error which occurred in the E-727.

Possible error code with E-727:

0x4310 “Excess temperature drive”
The error code indicates that the amplifier output is switched off because the internal temperature of the E-727 exceeds the switch-off threshold (p. 39). When the amplifier output is switched off, in addition all axes are switched to open-loop control.

When error 0x4310 occurs, all axes of the E-727 change in the **Fault reaction active** state. When the E-727 has cooled down and the error does no longer exist, all axes change in the **Fault** state.

Homing Mode

The Homing mode is used to perform an AutoZero procedure of the axis. The AutoZero procedure is the only homing method supported by the E-727 (for AutoZero details, see p. 62). AutoZero is a manufacturer-specific homing method (object 0x6098 = -1).

Supported objects (per axis):

Index*	Name
0x6040	Control word
0x6041	Status word
0x6060	Modes of operation
0x6061	Modes of operation display
0x6064	Position actual value
0x6098	Homing method

* Index for the first axis. Note that the object index is shifted by 0x800 per axis, e.g. the index of the status word for the second axis is 0x6841.

Homing is started using bit 4 of the control word (p. 170). Start options:

- Start axes individually, one after the other. Requirement: The current homing procedure must be finished before the homing procedure for the next axis can be started.
- Start multiple axes at the same time. Requirements: Bit 4 of the control word must be set to 1 for all axes, and the settings must be sent to the E-727 within one cycle.

Homing is stopped using bit 8 of the control word. To start homing again, bit 4 has to be toggled.

Homing-mode specific bits of the control word:

Bit	Name	Value	Desciption
4	Homing operation start	0 → 1	Start homing
8	Halt	0 1	Homing mode active Stop axis with profile deceleration

The status word (p. 172) shows the success of the homing procedure:

- Homing successful: Bit 12 is set
- Homing not successful: Bit 13 is set

Homing-mode specific bits of the status word:

Bit	Name	Value	Desciption
10	Target reached	0	Halt = 0: Home position not reached Halt = 1: Axis decelerates
		1	Halt = 0: Home position reached Halt = 1: Axis has velocity 0
12	Homing attained	0	Homing mode not yet completed
		1	Homing mode carried out successfully
13	Homing error	0	No homing error
		1	Homing error occurred Homing mode carried out not successfully The error cause can be found by reading the error code

Profile Position Mode

In Profile Position mode, an axis performs motions to absolute target positions. The dynamics profile is calculated by the E-727 based on target position and profile velocity given by the EtherCAT master.

Supported objects (per axis):

Index*	**Name**
0x6040	Control word
0x6041	Status word
0x6060	Modes of operation
0x6061	Modes of operation display

Index*	Name
0x6064	Position actual value
0x607A	Target position
0x6081	Profile velocity
0x6083	Profile acceleration

* Index for the first axis. Note that the object index is shifted by 0x800 per axis, e.g. the index of the status word for the second axis is 0x6841.

The profile velocity (object 0x6081) specifies the slew rate of the axis (p. 35).

The profile acceleration (object 0x6083) can be specified but is not evaluated.

Profile-Position-mode specific bits of the control word (p. 170):

Bit	Name	Value	Desciption
4	New setpoint	0	Does not assume target position
		1	Assume target position
5	Change set immediately	0	Finish the actual positioning and then start the next positioning
		1	Interrupt the actual positioning and start the next positioning
6	Reserved		
8	Halt	0	Execute positioning
		1	Stop axis with profile deceleration

Profile-Position-mode specific bits of the status word (p. 172):

Bit	Name	Value	Desciption
10	Target reached	0	Halt = 0: Target position not reached Halt = 1: Axis decelerates
		1	Halt = 0: Target position reached Halt = 1: Axis has velocity 0
12	Setpoint acknowledge	0	Previous target position already processed, waiting for new target
		1	Previous target position still in process, target overwriting shall be accepted
13	Reserved	The value is always 0.	

The setting of target positions is controlled by the timing of the ***new setpoint*** bit (bit 4) and the ***change set immediately*** bit (bit 5) in the control word as well as the ***setpoint acknowledge*** bit (bit 12) in the status word.

If the ***change set immediately*** bit of the control word is set to 1, the last sent target position is processed immediately (previous target positions are discarded). If the ***change set immediately*** bit of the control word is set to 0, the target positions are processed in the order they were received (i.e., processed as a “set of target positions”).

After a target position is applied to the axis, the EtherCAT master signals that the target position is valid by a rising edge of the ***new setpoint*** bit in the control word. The axis sets the ***setpoint acknowledge*** bit in the status word to 1, and afterwards, the axis signals with the ***setpoint acknowledge*** bit set to 0 its ability to accept new target positions.

For further details, see the description of the Profile position mode in the IEC 61800-7-201 standard.

When the EtherCAT communication is interrupted, any motion is stopped.

INFORMATION

The units of the position and velocity values are set via the ***FieldbusAxis Resolution*** parameter (ID 0x10000800) of the E-727 and the scaling factor numerator of the EtherCAT master. The values must be identical. For further details, see p. 165.

INFORMATION

For each axis, the E-727 provides an input buffer for value sets consisting of target position (object 0x607A) and profile velocity (object 0x6081). Maximum size of the input buffer: 10 value sets.

The maximum buffer size of the E-727 is given by the ***Fieldbus Set-Point List Size*** parameter (ID 0x10000801). You can read this parameter via the TCP/IP or USB interface only; for details, see "Parameters" (p. 185)

INFORMATION

Bit 10 of the status word indicates if the target position is reached. The value of bit 10 is influenced by two parameters of the E-727: settling window (***On Target Tolerance***, ID 0x07000900) and settling time (***Settling Time***, ID 0x07000901). Bit 10 has the value 1 ("Target position reached") when the current position is inside the settling window and stays there for at least the settling time. The settling window is centered around the target position.

Settling window and settling time are also applied if a set of target positions is specified. The next target position is activated not until the previous target position has been reached accordingly.

The value of bit 10 of the status word corresponds to the on-target status flag of the E-727 which can be output by a digital output line ("On Target" trigger mode, p. 72) or read by the ONT? command via TCP/IP or USB.

You can read or adjust the values of the settling window and settling time parameters via the TCP/IP or USB interface. For details, see "Parameters" (p. 185).

INFORMATION

Optimization of digital filter settings (p. 27) and tuning of the servo parameters (p. 134) can improve the settling behaviour and reduce position noise of the axes. Optimization must be done via the TCP/IP or USB interface.

Cyclic Synchronous Position Mode

In Cyclic Synchronous Position mode (CSP), the dynamics profile is calculated by the EtherCAT master.

Supported objects (per axis):

Index*	Name
0x6040	Control word
0x6041	Status word
0x6064	Position actual value
0x607A	Target position
0x6060	Modes of operation
0x6061	Modes of operation display
0x60F4	Following error actual value

* Index for the first axis. Note that the object index is shifted by 0x800 per axis, e.g. the index of the status word for the second axis is 0x6841.

The operation-mode-specific bits 4 to 6 and bit 8 of the control word are ignored in CSP.

The following error actual value (object 0x60F4) is calculated by the E-727 as the difference of target and actual position.

In CSP the new target position is given cyclically by the EtherCAT master. Bit 10 of the status word is therefore used as status-toggle information to indicate if the EtherCAT master provides updated input data. The bit is toggled between 0 and 1 with every update of the input process data.

Bit 12 of the status word in CSP is set if the axis is in the **Operation enabled** state and follows the target values of the EtherCAT master. In all other cases it is 0 (= target position ignored).

Bit 13 of the status word is always 0.

INFORMATION

The unit of the position values is set via the ***FieldbusAxis Resolution*** parameter (ID 0x10000800) of the E-727 and the scaling factor numerator of the EtherCAT master. The values must be identical. For further details, see p. 165.

INFORMATION

Optimization of digital filter settings (p. 27) and tuning of the servo parameters (p. 134) can improve the settling behaviour and reduce position noise of the axes. Optimization must be done via the TCP/IP or USB interface.

GCS Commands

The E-727 supports the PI General Command Set (GCS). GCS commands are accessible via the TCP/IP, RS-232, and USB interfaces, and via data segment 2 of the SPI interface.

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.
LF	LineFeed (ASCII char #10), is the default termination character (character at the end of a command line)
SP	Space (ASCII char #32), indicates a space character
"..."	Quotation marks indicate that the characters enclosed are returned or to be entered.

GCS Syntax for Syntax Version 2.0

A GCS command consists of 3 characters, e.g. CMD. The corresponding query command has a question mark added to the end, e. g. CMD?.

Command mnemonic:

CMD ::= character1 character2 character3 [?]

Exceptions:

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

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

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

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

Exception:

- Single-character commands are not followed by a termination character. The response to a single-character commands is followed by a termination character, however.

The argument <AxisID> is used for the logical axes of the controller. Depending on the controller, an axis identifier can consist of up to 16 characters. All alphanumeric characters and the underscore are allowed. See "Axes, Channels, Functional Elements" (p. 22) for the identifiers supported by the E-727.

Example 1:

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

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

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

Example 2:

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

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

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

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

Example 3:

All parameters in the volatile memory are to be reset.

Send: RPA LF

Example 4:

The position of all axes is to be queried.

Send: `POS?` LF

The response syntax is as follows:

[<Argument>[{SP<Argument>}]"="]<Value>LF

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

{[<Argument>[{SP<Argument>}]"="]<Value>SPLF}

[<Argument>[{SP<Argument>}]"="]<Value>LF for the last line!

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

Query command:

CMD?SP<Arg3>SP<Arg1>SP<Arg2>LF

Response to this command:

<Arg3>"="<Val3>SPLF

<Arg1>"="<Val1>SPLF

<Arg2>"="<Val2>LF

Example 5:

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

Receive: `2=-1158.4405`SPLF

`1=+0000.0000`LF

Limitations for GCS Commands

More than one command mnemonic per line is not allowed.

The number of characters per line is limited to 256 byte (1 character = 1 byte). This means that the number of arguments following a command mnemonic is limited to 32.

Example:

If you send

```
TWS 1 100 1 1 200 1 1 300 1 1 400 1 1 500 1 1 600 1 1 700 1 1 800 1 1 900 1 1 1000 1 1 1100 1
```

the controller will return error 24 ("Incorrect number of parameters") when you ask with the ERR? command afterwards because the number of arguments is 33.

Command Overview

The table below lists the GCS commands supported by the E-727 in alphabetical order. For detailed command descriptions see the GCS commands manual (PZ281E).

Command	Format	Short Description
#5	#5	Request Motion Status
#7	#7	Request Controller Ready Status
#8	#8	Query If Macro Is Running
#9	#9	Get Wave Generator Status
#24	#24	Stop All Axes
*IDN?	*IDN?	Get Device Identification
ADD	ADD <Variable> <FLOAT1> <FLOAT2>	Add and Save To Variable
AOS	AOS {<AxisID> <Offset>}	Set Analog Input Offset
AOS?	AOS? [{<AxisID>}]	Get Analog Input Offset
ATZ	ATZ [{<AxisID> <LowValue>}]	Set Automatic Zero Point Calibration
ATZ?	ATZ? [{<AxisID>}]	Get State Of Automatic Zero Point Calibration
CCL	CCL <Level> [<PSWD>]	Set Command Level
CCL?	CCL?	Get Command Level
CPY	CPY <Variable> <CMD?>	Copy Into Variable
CST?	CST? [{<AxisID>}]	Get Assignment Of Stages To Axes
CSV?	CSV?	Get Current Syntax Version
CTO	CTO {<TrigOutID> <CTOPam> <Value>}	Set Configuration Of Trigger Output
CTO?	CTO? [{<TrigOutID> <CTOPam>}]	Get Configuration Of Trigger Output
DDL	DDL <DDLtableID> <StartPoint> {<ValueN>}	Set DDL Table Value(s)
DDL?	DDL? [<StartPoint> [<NumberOfPoints> [{<DDLtableID>}]]]	Get DDL Table Value(s)
DEL	DEL <uint>	Delay The Command Interpreter
DIA?	DIA? [{<MeasureID>}]	Get Diagnosis Information
DIO?	DIO? [{<DIOID>}]	Get Digital Input Lines
DPO	DPO [{<AxisID>}]	DDL Parameter Optimization
DRC	DRC {<RecTableID> <Source> <RecOption>}	Set Data Recorder Configuration
DRC?	DRC? [{<RecTableID>}]	Get Data Recorder Configuration
DRL?	DRL? [{<RecTableID>}]	Get Number Of Recorded Points

Command	Format	Short Description
DRR?	DRR? [<StartPoint> <NumberOfPoints> [{<RecTableID>}]]	Get Recorded Data Values
DRT	DRT {<RecTableID> <TriggerSource> <Value>}	Set Data Recorder Trigger Source
DRT?	DRT? [{<RecTableID>}]	Get Data Recorder Trigger Source
DTC	DTC {<DDLtableID>}	Clear DDL Table Data
DTL?	DTL? [{<DDLtableID>}]	Get DDL Table Length
ERR?	ERR?	Get Error Number
GWD?	GWD? [<StartPoint> <NumberOfPoints> [{<WaveTableID>}]]	Get Wave Table Data
HDI?	HDI?	Get Help For Interpretation Of DIA? Response
HDR?	HDR?	Get All Data Recorder Options
HLP?	HLP?	Get List Of Available Commands
HLT	HLT [{<AxisID>}]	Halt Motion Smoothly
HPA?	HPA?	Get List Of Available Parameters
HPV?	HPV?	Get Parameter Value Description
IDN?	IDN?	Get Device Identification
IFC	IFC {<InterfacePam> <PamValue>}	Set Interface Parameters Temporarily
IFC?	IFC? [{<InterfacePam>}]	Get Current Interface Parameters
IFS	IFS <Pswd> {<InterfacePam> <PamValue>}	Set Interface Parameters As Default Values
IFS?	IFS? [{<InterfacePam>}]	Get Interface Parameters As Default Values
IMP	IMP <AxisID> <Amplitude>	Start Impulse And Response Measurement
JOG	JOG {<AxisID> <Velocity>}	Start Motion With Given Velocity
JOG?	JOG? [{<AxisID>}]	Get Velocity For Motion Caused By JOG
JRC	JRC <Jump> <CMD?> <OP> <Value>	Jump Relatively Depending On Condition
MAC	MAC <keyword> {<parameter>} Especially: 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>]]	Call Macro Function

Command	Format	Short Description
MAC?	MAC? [<macroname>]	List Macros
MEX	MEX <CMD?> <OP> <Value>	Stop Macro Execution Due To Condition
MOV	MOV {<AxisID> <Position>}	Set Target Position
MOV?	MOV? [{<AxisID>}]	Get Target Position
MVR	MVR {<AxisID> <Distance>}	Set Target Relative To Current Position
ONT?	ONT? [{<AxisID>}]	Get On-Target State
OVF?	OVF? [{<AxisID>}]	Get Overflow State
POS?	POS? [{<AxisID>}]	Get Real Position
PUN?	PUN? [{<AxisID>}]	Get Axis Unit
RBT	RBT	Reboot System
RMC?	RMC?	List Running Macros
RPA	RPA [{<ItemID> <PamID>}]	Reset Volatile Memory Parameters
RTR	RTR <RecordTableRate>	Set Record Table Rate
RTR?	RTR?	Get Record Table Rate
SAI?	SAI? [ALL]	Get List Of Current Axis Identifiers
SEP	SEP <Pswd> {<ItemID> <PamID> <PamValue>}	Set Nonvolatile Memory Parameters
SEP?	SEP? [{<ItemID> <PamID>}]	Get Nonvolatile Memory Parameters
SPA	SPA {<ItemID> <PamID> <PamValue>}	Set Volatile Memory Parameters
SPA?	SPA? [{<ItemID> <PamID>}]	Get Volatile Memory Parameters
SSN?	SSN?	Get Device Serial Number
STE	STE <AxisID> <Amplitude>	Start Step And Response Measurement
STP	STP	Stop All Axes
SVA	SVA {<AxisID> <Amplitude>}	Set Open-Loop Axis Value
SVA?	SVA? [{<AxisID>}]	Get Open-Loop Axis Value
SVO	SVO {<AxisID> <ServoState>}	Set Servo Mode
SVO?	SVO? [{<AxisID>}]	Get Servo Mode
SVR	SVR {<AxisID> <Difference>}	Set Relative Open-Loop Axis Value
TAD?	TAD? [{<InputSignalID>}]	Get ADC Value Of Input Signal
TIO?	TIO?	Tell Digital I/O Lines
TLT?	TLT?	Get Number of DDL Tables
TMN?	TMN? [{<AxisID>}]	Get Minimum Commandable Position
TMX?	TMX? [{<AxisID>}]	Get Maximum Commandable Position
TNR?	TNR?	Get Number Of Record Tables
TNS?	TNS? [{<InputSignalID>}]	Get Normalized Input Signal Value

Command	Format	Short Description
TPC?	TPC?	Get Number of Output Signal Channels
TSC?	TSC?	Get Number of Input Signal Channels
TSP?	TSP? [{<InputSignalID>}]	Get Input Signal Value
TWC	TWC	Clear All Wave Related Triggers
TWG?	TWG?	Get Number of Wave Generators
TWS	TWS {<TrigOutID> <PointNumber> <Switch>}	Set Trigger Line Action To Waveform Point
TWS?	TWS? [<StartPoint> [<NumberOfPoints> [{<TrigOutID>}]]]	Get Trigger Line Action At Waveform Point
VAR	VAR <Variable> <String>	Set Variable Value
VAR?	VAR? [{<Variable>}]	Get Variable Value
VCO	VCO {<AxisID> <VelCtrlState>}	Set Velocity Control Mode
VCO?	VCO? [{<AxisID>}]	Get Velocity Control Mode
VEL	VEL {<AxisID> <Velocity>}	Set Closed-Loop Velocity
VEL?	VEL? [{<AxisID>}]	Get Closed-Loop Velocity
VOL?	VOL? [{<OutputSignalID>}]	Get Voltage Of Output Signal Channel
WAC	WAC <CMD?> <OP> <Value>	Wait For Condition
WAV	WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>	Set Waveform Definition
WAV?	WAV? [{<WaveTableID> <WaveParameterID>}]	Get Waveform Definition
WCL	WCL {<WaveTableID>}	Clear Wave Table Data
WGC	WGC {<WaveGenID> <Cycles>}	Set Number Of Wave Generator Cycles
WGC?	WGC? [{<WaveGenID>}]	Get Number Of Wave Generator Cycles
WGO	WGO {<WaveGenID> <StartMode>}	Set Wave Generator Start/Stop Mode
WGO?	WGO? [{<WaveGenID>}]	Get Wave Generator Start/Stop Mode
WGR	WGR	Starts Recording In Sync With Wave Generator
WOS	WOS {<WaveGenID> <Offset>}	Set Wave Generator Output Offset
WOS?	WOS? [{<WaveGenID>}]	Get Wave Generator Output Offset
WPA	WPA <Pswd> [{<ItemID> <PamID>}]	Save Parameters To Nonvolatile Memory
WSL	WSL {<WaveGenID> <WaveTableID>}	Set Connection Of Wave Table To Wave Generator
WSL	WSL? [{<WaveGenID>}]	Get Connection Of Wave Table To Wave Generator
WTR	WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}	Set Wave Generator Table Rate
WTR?	WTR? [{<WaveGenID>}]	Get Wave Generator Table Rate

Parameters

Settings of the E-727

Parameters reflect the properties of the E-727 and the connected stage and define the behavior of the system (e.g. settings for the control algorithm (p. 35) and notch filters (p. 37)).

The parameters can be divided into the following categories:

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

The write permission for the parameters is determined by command levels.

Every parameter is present in the volatile as well as in the nonvolatile memory of the E-727. The values in the nonvolatile memory are loaded to the volatile memory as default values when switching on or rebooting the E-727. The values in the volatile memory determine the current behavior of the system.

The values of some parameters are stored on the ID chip of the stage. They are loaded to the volatile and nonvolatile memory when the E-727 is switched on or rebooted. You cannot overwrite the parameters in the ID-chip (this can only be done by PI). See "ID-Chip Support / Stage Replacement" (p. 131) for more information.

Parameter Handling

NOTICE

Improper parameter settings!

Incorrect parameter values may lead to improper operation or damage to your hardware.

- ➢ Save the current parameter values of the E-727 to the PC (p. 53) before you make changes in the nonvolatile memory.
- ➢ It is recommended that any modifications be first made in the volatile memory, and when the E-727 runs well, saved to the nonvolatile memory.
- ➢ Do not change the current interface settings of the E-727 in the volatile memory—except of the baud rate—because it will not be possible to maintain communication afterwards.

INFORMATION

While parameter values are being read from or written to the nonvolatile memory of the E-727, the servo mode is temporarily switched off. This may lead to faulty control or large position errors.

- ➢ Observe the following when you use the commands SEP, SEP?, RPA, WPA, or the corresponding operating elements in the PC software:
  - Avoid motions of the axes.
  - Do not carry out measurements with the axes.

INFORMATION

To change parameter values, you can use the commands listed in "General Commands for Parameters" (p. 186) and "Commands for Fast Access to Individual Parameters" (p. 187). But if you want to access the parameters in a more convenient way, you should use the ***Device Parameter Configuration*** window of PIMikroMove (for an example, see p. 53).

- 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.

INFORMATION

If changing the parameter value requires command level 1, switch to command level 1 as follows:

- In a terminal program: Send `CCL 1 advanced`
- When prompted to enter a password in any PC software, e.g. in PIMikroMove:
  Enter `advanced`.

On command levels > 1, write access is only available to PI service personnel.

- Contact the customer service department if there seem to be problems with parameters of command level 2 or higher (p. 234).

INFORMATION

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

- Overwrite the default values only when it is necessary.
- Save the current parameter values to the PC (p. 53) before you make changes in the nonvolatile memory.
- Contact our customer service department (p. 234), if the E-727 exhibits unexpected behavior.

INFORMATION

E-727 models with EtherCAT interface (p. 164):

- You can read or write parameter values via the TCP/IP or USB interface only. Before you access parameter settings, make sure that these interfaces are enabled: TCP/IP and USB will be enabled again after a delay of 100 ms when the E-727 has left the OPERATIONAL EtherCAT communication state.
- Change parameter values in the nonvolatile memory and reboot the E-727 afterwards to make the changes effective in the EtherCAT bus.

General Commands for Parameters

The following general commands are available for parameters:

Command	Function
CCL	Change to a higher command level, e.g., to obtain write permission for particular parameters.
CCL?	Get active command level.
HPA?	Get a help string which contains all available parameters with short descriptions.
RPA	Copy a parameter value from the nonvolatile to the volatile memory.
SEP	Change parameters in the nonvolatile memory.
SEP?	Get parameter values from the nonvolatile memory.
SPA	Change parameters in the volatile memory.
SPA?	Get parameter values from the volatile memory.
WPA	Copy a current parameter value from the volatile to the nonvolatile memory.

For detailed command descriptions see the GCS commands manual PZ281E.

Commands for Fast Access to Individual Parameters

The following special commands only change the corresponding parameters in the volatile memory. To make them the new default values, the changed values must be written to the nonvolatile memory (with the WPA command or the corresponding controls of the PC software).

Command	Function
AOS	Adds an offset to the analog control input for an axis. For the parameter concerned, see p. 92.
AOS?	Reads the current valid offset value.
ATZ	Starts the AutoZero procedure which changes the sensor offset settings for an axis. For the parameters concerned, see p. 63.
CTO	Programs the trigger settings for the digital output lines of the E-727. For the parameters concerned, see p. 69.
CTO?	Reads the current valid trigger settings.
DRC	Defines the data to be recorded by the data recorder. For the parameters concerned, see p. 67.
DRC?	Reads current data recorder settings.
DPO	Recalculates the processing parameters of the Dynamic Digital Linearization. For the parameters concerned, see p. 120.
IFC	Changes the current baud rate of the E-727. ➢ Do not change other interface settings of the E-727 in the volatile memory because it will not be possible to maintain communication afterwards.
IFC?	Reads the current valid interface configuration.

Command	Function
`RTR`	Gives the data recorder table rate. For the parameter concerned, see p. 67.
`RTR?`	Reads the current setting of the data recorder table rate.
`VEL`	Sets the value of the ***Servo Loop Slew-Rate*** parameter, ID 0x07000200.
`VEL?`	Reads the current value of the ***Servo Loop Slew-Rate*** parameter, ID 0x07000200.
`WOS`	Adds an offset to the wave generator output. For the parameter concerned, see p. 112.
`WOS?`	Reads the current valid offset value.
`WTR`	Gives the wave generator table rate. For the parameter concerned, see p. 112.
`WTR?`	Reads the current setting of the wave generator table rate.

The following special command only changes the corresponding parameter in the nonvolatile memory.

Command	Function
`IFS`	Defines the default settings for communication via the RS-232 interface (baud rate; p. 61) and TCP/IP interface (IP address, IP mask, IP start; p. 56).

For detailed command descriptions see the GCS commands manual PZ281E.

INFORMATION

The parameters listed above can also be changed with the general commands (p. 186).

Parameter Overview

The table below gives short descriptions of some parameters. For the parameters highlighted in grey, possible values can be queried using the HPV? command. The parameters stored in the ID-chip are marked in the "Description" column.

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x02000100	INT	1	Input signal channel	Sensor Range Factor	ID-chip 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. 19) and the information in „E-727.3CDA, E-727.3RDA, E-727.3SDA: Analog I/O“ (p. 228).
0x02000102	INT	1	Input signal channel	Sensor Offset Factor	ID-chip Only present with E-727 models for piezoresistive and strain gauge sensors; changed by the AutoZero procedure (see p. 62).
0x02000103	INT	1	Input signal channel	Sensor Cable Compensation	ID-chip Only present with E-727 models for capacitive sensors
0x02000200	FLOAT	1	Input signal channel	Sensor Mech. Correction 1	ID-chip Coefficients of the polynomial for mechanics linearization. For details, see “Digital Processing”, p. 27.
0x02000300	FLOAT	1	Input signal channel	Sensor Mech. Correction 2	
0x02000400	FLOAT	1	Input signal channel	Sensor Mech. Correction 3	
0x02000500	FLOAT	1	Input signal channel	Sensor Mech. Correction 4	
0x02000600	FLOAT	1	Input signal channel	Sensor Mech. Correction 5	
0x02000a00	INT	1	Input signal channel	Sensor Reference Mode	1 = absolute measuring sensor, i.e. no referencing required

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x02001900	INT	3	Input signal channel	Lim, Ref Signals Detectable	Flag that indicates if signals of limit and reference point switches are evaluated by the firmware (does not indicate if switches are actually present) 0 = no evaluation 1 = switch signals are evaluated
0x02001B00	FLOAT	1	Input Signal Channel	Maximum Position Of Sensor	For future use. The parameters are only available if the plausibility check is enabled with the ***Check Sensor Position Plausibility*** parameter (ID 0x0e001f00)
0x02001C00	FLOAT	1	Input Signal Channel	Minimum Position Of Sensor	
0x02001D00	INT	1	Input Signal Channel	Validate Sensor Position Range	
0x03000*n*00	FLOAT	2	Input signal channel	Sensor Elec. Correction 1 *n*	Coefficients of the polynomial for electronics linearization. *n* = 1 to 5 For details, see "Digital Processing", p. 27.
0x03000*n*01	FLOAT	2	Input signal channel	Sensor Elec. Correction 2 *n*	
0x03000*n*02	FLOAT	2	Input signal channel	Sensor Elec. Correction 3 *n*	
0x03000*n*03	FLOAT	2	Input signal channel	Sensor Elec. Correction 4 *n*	
0x03000*n*04	FLOAT	2	Input signal channel	Sensor Elec. Correction 5 *n*	
0x03000*n*05	FLOAT	2	Input signal channel	Sensor Elec. Correction 6 *n*	
0x03000*n*06	FLOAT	2	Input signal channel	Sensor Elec. Correction 7 *n*	
0x03000*n*07	FLOAT	2	Input signal channel	Sensor Elec. Correction 8 *n*	
0x03000*n*08	FLOAT	2	Input signal channel	Sensor Elec. Correction 9 *n*	
0x03000*n*09	FLOAT	2	Input signal channel	Sensor Elec. Correction 10 *n*	
0x03000*n*0a	FLOAT	2	Input signal channel	Sensor Elec. Correction 11 *n*	
0x03000*n*0b	FLOAT	2	Input signal channel	Sensor Elec. Correction 12 *n*	

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x03000*n*0c	FLOAT	2	Input signal channel	Sensor Elec. Correction 13 *n*	
0x03000*n*0d	FLOAT	2	Input signal channel	Sensor Elec. Correction 14 *n*	
0x03000*n*0e	FLOAT	2	Input signal channel	Sensor Elec. Correction 15 *n*	
0x03000*n*0f	FLOAT	2	Input signal channel	Sensor Elec. Correction 16 *n*	
0x03000*n*10	FLOAT	2	Input signal channel	Sensor Elec. Correction 17 *n*	
0x03000*n*11	FLOAT	2	Input signal channel	Sensor Elec. Correction 18 *n*	
0x03000*n*12	FLOAT	2	Input signal channel	Sensor Elec. Correction 19 *n*	
0x03001000	FLOAT	2	Input signal channel	Sensor Offset Correction 1	Coefficients of a polynomial for compensation of sensor offset non-linearity Only present with E-727 models for piezoresistive and strain gauge sensors
0x03001100	FLOAT	2	Input signal channel	Sensor Offset Correction 2	
0x03001200	FLOAT	2	Input signal channel	Sensor Offset Correction 3	
0x03001300	FLOAT	2	Input signal channel	Sensor Offset Correction 4	
0x03001400	FLOAT	2	Input signal channel	Sensor Offset Correction 5	
0x03001500	FLOAT	2	Input signal channel	Sensor Offset Correction 6	
0x03003400	INT	1	Input signal channel	Input Numerical Format	Gives the data type for input values. 6 = SIGN32
0x03003700	INT	1	Input signal channel	Sensor Autoscaling Enable	With E-727.3RD, .3RDA, .3SD and .3SDA models for piezoresistive and strain gauge sensors: Sensor autoscaling can be included in the AutoZero procedure. For details, see “Special Function: Sensor Autoscaling”, p. 65.
0x03003701	FLOAT	1	Input signal channel	Sensor Autoscaling Gain	

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x03003800	FLOAT	2	Input signal channel	CBI	Calibration parameters required with models for piezoresistive sensors and strain gauge sensors; SBI = Sensor Bridge Impedance of the connected sensor (94000 if no mechanics is connected)
0x03003801	FLOAT	3	Input signal channel	SBI	
0x03003803	FLOAT	3	Input signal channel	OAI	
0x05000000	INT	1	Input signal channel	Digital Filter Type	ID-chip Digital filtering of the input signals. For details, see “Digital Processing”, p. 27.
0x05000001	FLOAT	1	Input signal channel	Digital Filter Bandwidth	
0x05000101	FLOAT	1	Input signal channel	User Filter Param. 1	Coefficients for the USER filter type. For further details, see “Digital Processing”, p. 27.
0x05000102	FLOAT	1	Input signal channel	User Filter Param. 2	
0x05000103	FLOAT	1	Input signal channel	User Filter Param. 3	
0x05000104	FLOAT	1	Input signal channel	User Filter Param. 4	
0x05000105	FLOAT	1	Input signal channel	User Filter Param. 5	
0x06000500	INT	1	System	ADC Channel for Target	Enable the analog control input for an axis. For details, see “Use as Control Value Generation Source”, p. 90
0x06000501	FLOAT	1	System	Analog Target Offset	
0x07000000	FLOAT	1	Axis	Range Limit min	ID-chip
0x07000001	FLOAT	1	Axis	Range Limit max	ID-chip
0x07000200	FLOAT	1	Axis	Servo Loop Slew-Rate	ID-chip For details, see p. 35.
0x07000201	FLOAT	1	Axis	Open Loop Slew-Rate	
0x07000300	FLOAT	1	Axis	Servo-loop P-Term	ID-chip Parameters of the PID algorithm (position control). For details, see p. 35.
0x07000301	FLOAT	1	Axis	Servo-loop I-Term	
0x07000302	FLOAT	1	Axis	Servo-loop D-Term	

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x07000500	FLOAT	1	Axis	Position from Sensor 1	ID-chip Coefficients of the input matrix. For details, see "Allocation of input signal channels to axes", p. 29.
0x07000501	FLOAT	1	Axis	Position from Sensor 2	
0x07000502	FLOAT	1	Axis	Position from Sensor 3	
0x07000503	FLOAT	1	Axis	Position from Sensor 4	
0x07000504	FLOAT	1	Axis	Position from Sensor 5	
0x07000505	FLOAT	1	Axis	Position from Sensor 6	
0x07000506	FLOAT	1	Axis	Position from Sensor 7	
0x07000600	CHAR	1	Axis	Axis Name	ID-chip
0x07000601	CHAR	1	Axis	Axis Unit	ID-chip
0x07000800	INT	1	Axis	Power Up Servo ON Enable	For details, see "Control Value Generation", p. 30.
0x07000802	INT	1	Axis	Power Up AutoZero Enable	For details, see "AutoZero Procedure", p. 62.
0x07000900	FLOAT	1	Axis	ON Target Tolerance	ID-chip [µm] Gives the settling window. The settling window is centered around the axis target position. The on-target status of an axis is true when the current axis position is inside the settling window and stays there for at least the settling time (Settling Time parameter, ID 0x07000901).
0x07000901	FLOAT	1	Axis	Settling Time	[s]
0x07000a00	FLOAT	1	Axis	AutoZero Low Voltage	ID-chip For details, see "AutoZero Procedure", p. 62.
0x07000a01	FLOAT	1	Axis	AutoZero High Voltage	
0x07000c01	FLOAT	1	Axis	Default voltage	ID-chip

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x07001005	FLOAT	1	Axis	Position Report Scaling	For details, see "Using the Analog Output" (p. 94).
0x07001006	FLOAT	1	Axis	Position Report Offset	
0x07020000	INT	1	Axis	Adv. Piezo Ctrl Aperiodic Flag	Parameters of the Advanced Piezo Control option (p. 19). See the separate E712T0007 Technical Note for more information.
0x07020100	FLOAT	1	Axis	Adv. Piezo Ctrl Stage Damping	
0x07020200	FLOAT	1	Axis	Adv. Piezo Ctrl Resonance	
0x07020500	FLOAT	1	Axis	Adv. Piezo Ctrl Stage Substituation Time	
0x07020a00	FLOAT	1	Axis	Adv. Piezo Ctrl I-Term - Relation (0.1 to 1)	
0x07020b00	FLOAT	1	Axis	Adv. Piezo Relative Observer Speed (1 to 10)	
0x07030100	INT	1	Axis	Closed-Loop Control Mode	Gives the control mode to be used for closed-loop operation. For details, see "Control Algorithms" (p. 35).
0x07030600	FLOAT	1	Axis	Feed Forward Gain	For details, see "Feedforward" on p. 37.
0x07030900	INT	1	Axis	Feed Forward Input Channel Index	
0x07030a00	FLOAT	1	Axis	On-Target Tolerance To Fix I-Term	Not relevant for E-727.
0x07030c00	INT	1	Axis	ID-Chip Chan. to Axis Map Ctrl	Mapping of axis IDs when stages with ID chip are connected to the E-727.
0x07030d00	INT	1	Axis	Zeroing Control Value If I-Term Is Fixed	Not relevant for E-727.

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x08000100	FLOAT	1	Axis	Notch frequency 1	ID-chip (excluding 0x08000500 and 0x08000600) Notch filter settings. For details, see “Notch Filters”, p. 37.
0x08000101	FLOAT	1	Axis	Notch frequency 2	
0x08000200	FLOAT	1	Axis	Notch Rejection 1	
0x08000201	FLOAT	1	Axis	Notch Rejection 2	
0x08000300	FLOAT	1	Axis	Notch Bandwidth 1	
0x08000301	FLOAT	1	Axis	Notch Bandwidth 2	
0x08000400	FLOAT	1	Axis	Creep factor T1/sec	
0x08000401	FLOAT	1	Axis	Creep factor T2/sec	
0x08000500	INT	1	Axis	Enable Notch In Open-Loop	
0x08000600	INT	1	Axis	Notch Filter Calculation Method	
0x09000000	FLOAT	1	Axis	Driving Factor of Piezo 1	ID-chip Coefficients of the output matrix. For details, see “Output Generation”, p. 34.
0x09000001	FLOAT	1	Axis	Driving Factor of Piezo 2	
0x09000002	FLOAT	1	Axis	Driving Factor of Piezo 3	
0x09000003	FLOAT	1	Axis	Driving Factor of Piezo 4	
0x0A000003	INT	1	Output signal channel	Select Output Type	Usage of the output signal channels; can only be changed for output signal channel 4. For details see „Using the Analog Output“ (p. 94).
0x0A000004	INT	1 or 3	Output signal channel	Select Output Index	

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x0b000007	FLOAT	1	Output signal channel	Min Output Voltage of Amplifier	Settings valid when used as output voltage for a piezo actuator in the stage (output on the socket for piezo stages). Limited by Soft Voltage Low Limit (ID 0x0c000000) and Soft Voltage High Limit (ID 0x0c000001).
0x0b000008	FLOAT	1	Output signal channel	Max Output Voltage of Amplifier	
0x0b000009	FLOAT	2	Output signal channel	Voltage of Amplifier with zero to Dac	
0x0b00000a	FLOAT	1	Output signal channel	Voltage Offset of Amplifier	
0x0b000500	INT	1	Output signal channel	Output Numerical Format	7 = raw format (used for driver/DAC issues)
0x0b000800	INT	1	Output signal channel	Input Index To Feed Back	Not relevant for E-727
0x0b000a00	FLOAT	2	Output signal channel	Min Output Voltage of 2nd Output	Settings valid when used as control signal for an external amplifier (output on pin 8 of the **Analog I/O** socket). Only present with E-727 models equipped with an **Analog I/O** socket (E-727.xxxA, .xxxAx).
0x0b000a01	FLOAT	2	Output signal channel	Max Output Voltage of 2nd Output	
0x0b000a02	FLOAT	2	Output signal channel	Voltage of 2nd Output with zero to Dac	
0x0c000000	FLOAT	1	Output signal channel	Soft Voltage Low Limit	ID-chip Limits the output voltage for a piezo actuator in the stage (output on the socket for piezo stages).
0x0c000001	FLOAT	1	Output signal channel	Soft Voltage High Limit	
0x0d000000	CHAR	2	System	Device S/N	
0x0d000100	CHAR	2	System	Hardware S/N	
0x0d000200	CHAR	2	System	Hardware Name	
0x0d000300	INT	3	System	Fieldbus Product Code	Identification of the E-727 in the EtherCat network
0x0e000100	FLOAT	2	System	Sensor Sampling Time	[s]

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x0e000200	FLOAT	3	System	Servo Update Time	[s]
0x0e000301	INT	1	System	Disable Error 10	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.)
0x0e000400	INT	1	System	DDL license	
0x0e000401	INT	3	System	DDL license valid	
0x0e000b00	INT	3	System	Number of input channels	
0x0e000b01	INT	3	System	Number of output channels	
0x0e000b02	INT	2	System	Number of system axes	
0x0e000b03	INT	3	System	Number of sensor channels	
0x0e000b04	INT	3	System	Number of piezo channels	
0x0e000b05	INT	3	System	Number of trigger outputs	
0x0e000b06	INT	3	System	Number Of PiezoWalk Channels	Not relevant for E-727.
0x0e000b07	INT	3	System	Number of conf. PiezoWalk chan.	Not relevant for E-727.
0x0E000C01	INT	1	System	OverSampling Filter	Usage of this filter is recommended with strain gauge sensors and/or with servo sampling rates ≤ 10 kHz. Recommended parameter setting for processing of strain gauge sensors: *Cutoff frequency of the filter ≤ Servo sampling rate / 2* See response to HPV? for possible values.

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x0e000e00	INT	1	System	Adv. Piezo Control License	Parameters of the Advanced Piezo Control option (p. 19). See the separate E712T0007 Technical Note for more information.
0x0e000f00	INT	3	System	Adv. Piezo Control License-Valid Flag	
0x0E001500	INT	1	System	Reboot on DIO Input	The value of the parameter enables/disables the Reset input on pin 2 of the MDR14 socket (details see p. 227).
0x0E001D00	INT	1	System	Trigger Input Filter Enable	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.
0x0E001E00	INT	1	System	Discon. Target Man. In With Stop	Determines the behaviour when the analog input is used as a control source. See „Using the Analog Input" (p. 85) for details.
0x0e001f00	INT	1	System	Check Sensor Position Plausibility	For future use. Enables oder disables plausibility check of sensor position.
0x0e002000	INT	1	System	Move To Last Commanded Position	Determines the target positions to be set when switching on the servo mode: 0 = The target positions are set to the current positions of the axes (default). 1= The target positions are set to the last commanded target positions for the axes. This can cause an abrupt motion.
0x0e002400	INT	1	System	Trigger Input Filter Samples	
0x0f000000	INT	1	System	Power Up Read ID-Chip	For details, see "ID-Chip Support / Stage Replacement" (p. 131).

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x0f000100	CHAR	1	Input signal channel	Stage Type	ID-chip
0x0f000200	CHAR	1	Input signal channel	Stage Serial Number	ID-chip
0x0f000300	CHAR	1	Input signal channel	Stage assembly Date	ID-chip
0x0f000700	INT	3	Input signal channel	ID-Chip Data Valid	ID-chip present? Valid data read?
0x10000500	INT	1	Axis	FastIF Axis Input Usage	Determines the usage of the data received from the host via data segment 1 of the SPI interface. For details, see "Configuring the Usage of Received Data" (p. 145).
0x10000501	INT	1	System	FastIF Data Type	Selects the data type for the data sent via data segment 1 of the SPI interface. When the 32 bit floating point data type is selected then the data can be read and written directly in axis units. In this case no scaling is necessary. When any of the integer data types is selected then scaling is necessary. In this case the lower and upper limit parameters must also be set (Fast IF Data Low Limit 0x10000502, Fast IF Data High Limit 0x010000503). 0 = FLOAT 32 bit 1 = UINT 16 bit 2 = UINT 24 bit 3 = UINT 32 bit 4 = SIGNED 16 bit 5 = SIGNED 24 bit 6 = SIGNED 32 bit
0x10000502	FLOAT	1	Axis	FastIF Data Low Limit	Lower limit for the data sent via data segment 1 of the SPI interface. Must be set when any integer data type is selected (Fast IF Data Type, 0x10000501) so that the scaling can be calculated internally.

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
					When the selected data type is floating point (IEEE 754) then this parameter has no effect.
0x10000503	FLOAT	1	Axis	FastIF Data High Limit	Upper limit for the data sent via data segment 1 of the SPI interface. Must be set when any integer data type is selected (Fast IF Data Type, 0x10000501) so that the scaling can be calculated internally. When the selected data type is floating point (IEEE 754) then this parameter has no effect.
0x10000504	FLOAT	3	Axis	FastIF Used Low Limit	Currently used lower limit for the data sent via data segment 1 of the SPI interface. For further details, see "Configuring the Limits for Position Scaling" (p. 145).
0x10000505	FLOAT	3	Axis	FastIF Used High Limit	Currently used upper limit for the data sent via data segment 1 of the SPI interface. For further details, see "Configuring the Limits for Position Scaling" (p. 145).
0x10000506	INT	1	System	Used Range For Fast IF	Determines which limits are used for scaling the data sent via data segment 1 of the SPI interface. For further details, see "Configuring the Limits for Position Scaling" (p. 145).
0x10000507	FLOAT	3	Axis	FastIF Open Loop Low Limit	Currently used lower limit for the input data sent via data segment 1 of the SPI interface, applied in open-loop operation. For further details, see "Configuring the Limits for Position Scaling" (p. 145).
0x10000508	FLOAT	3	Axis	FastIF Open Loop High Limit	Currently used upper limit for the input data sent via data segment 1 of the SPI interface, applied in open-loop operation.

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
					For further details, see "Configuring the Limits for Position Scaling" (p. 145).
0x10000509	FLOAT	1	System	FastIF Open Loop Limit Extend	Necessary to cover the whole travel range in open-loop operation. For further details, see "Configuring the Limits for Position Scaling" (p. 145).
0x10000800	FLOAT	1	Axis	FieldbusAxis Resolution	Scaling factor for target and current values transmitted via EtherCAT interface For further details, see "EtherCAT Interface" (p. 164).
0x10000801	INT	3	Axis	Fieldbus Set-Point List Size	Size of input buffer for profile position mode (max. number of value sets) For further details, see "EtherCAT Interface" (p. 164).
0x11000400	INT	1	System	Uart Baudrate	For details, see "Communication" (p. 56).
0x11000600	CHAR	1	System	IP-Address	
0x11000700	CHAR	1	System	IP-Mask	
0x11000800	INT	1	System	IP-Configura-tion	
0x11000b00	CHAR	2	System	MAC-Address	
0x13000004	INT	3	System	Max Wave Points	For details, see "Wave Generator" (p. 99).
0x13000109	INT	1	System	Wave Generator Table Rate	
0x1300010a	INT	3	System	Number of Waves	
0x1300010b	FLOAT	1	Axis	Wave Offset	
0x13000202	INT	1	System	Wave Multi Start By Trigger	

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x14000001	INT	1	Axis	DDL repeat number	For details, see "Dynamic Digital Linearization (DDL)" (p. 114).
0x14000006	FLOAT	1	Axis	Time Delay Max	
0x14000007	FLOAT	1	Axis	Time Delay Min	
0x14000008	INT	1	Axis	Time Delay Change Rule	
0x1400000a	INT	1	Axis	DDL Zero Gain Number	
0x1400000b	INT	3	System	Max DDL Points	
0x16000000	INT	1	System	Data Recorder Table Rate	For details, see „Data Recording" (p. 65).
0x16000100	INT	3	System	Max Number of Data Recorder Channels	
0x16000200	INT	3	System	Data Recorder Max Points	
0x16000300	INT	1	System	Data Recorder Chan Number	
0x16000700	INT	1	Data rec. table	DRC Data Source	
0x16000701	INT	1	Data rec. table	DRC Record Option	
0x18000201	FLOAT	1	Digital output	CTO Trigger Step	Trigger settings. For details, see „Configuring Trigger Output" on p. 69.
0x18000202	INT	1	Digital output	CTO Axis	
0x18000203	INT	1	Digital output	CTO Trigger Mode	
0x18000205	FLOAT	1	Digital output	CTO Min.Threshold	
0x18000206	FLOAT	1	Digital output	CTO Max.Threshold	
0x18000207	INT	1	Digital output	CTO Polarity	
0x18000208	FLOAT	1	Digital output	CTO Start Threshold	

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0x18000209	FLOAT	1	Digital output	CTO Stop Threshold	
0x18000210	INT	1	Digital output	CTO Trigger Out Mask	
0x18000300	INT	1	Digital Output Line	Pos. Distance Trig. Single Direction	Additional settings for the “Position Distance” trigger mode. For details, see „Example—"Position Distance" Trigger Mode“ on p. 71.
0x18000301	FLOAT	1	Digital Output Line	Pos. Distance Trig. Filter Time	
0x18000302	FLOAT	1	Digital Output Line	Pos. Distance Trig. Filter Level	
0x18000400	INT	1	Digital Output Line	Pos. Distance Trig. High Time Definition	
0x18000401	FLOAT	1	Digital Output Line	Pos. Distance Trig. High Time	
0xffff0001	INT	2	Firmware Unit	firmware mark	
0xffff0002	INT	2	Firmware Unit	firmware CRC	
0xffff0003	INT	2	Firmware Unit	firmware desc. CRC	
0xffff0004	INT	2	Firmware Unit	firmware desc. version	
0xffff0006	CHAR	2	Firmware Unit	firmware matchcode	
0xffff0007	CHAR	2	Firmware Unit	hardware matchcode	
0xffff0008	INT	2	Firmware Unit	firmware version	
0xffff000b	INT	2	Firmware Unit	firmware max. size	
0xffff000c	CHAR	2	Firmware Unit	firmware device	
0xffff000d	CHAR	2	Firmware Unit	short description of firmware	

Parameter ID (hexa-decimal)	Data Type	Command Level for Write Access	Item Type	Parameter Name	Description
0xffff000e	CHAR	2	Firmware Unit	date of firmware	
0xffff000f	CHAR	2	Firmware Unit	firmware developer	
0xffff0010	INT	2	Firmware Unit	firmware length	
0xffff0011	INT	2	Firmware Unit	firmware compatibility	
0xffff0012	INT	2	Firmware Unit	firmware rel. address	
0xffff0013	CHAR	2	Firmware Unit	firmware device type	
0xffff0014	INT	2	Firmware Unit	hardware revision	
0xffff0015	INT	2	Firmware Unit	firmware dest. addr.	
0xffff0016	INT	2	Firmware Unit	firmware configuration	

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,
13.12.00.00
```

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

To update the firmware of E-727, the USB interface must be used with the PI Firmware Update Wizard PC software.

Requirements

✓ You have installed the PI Firmware Update Wizard on the PC (p. 43).

✓ You have obtained the current firmware file from our customer service department (p. 234)

and copied the file to a directory on the PC. Make sure that this directory only contains the current firmware file.
Example for a file name: E-727_DSP_FIRMWARE_HW_00000_FW_01008.hex.

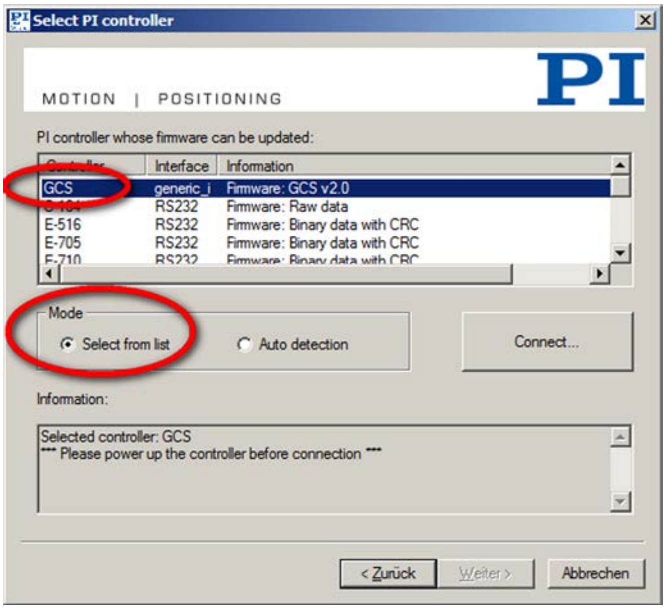
Updating the firmware

Proceed as follows to update the firmware:

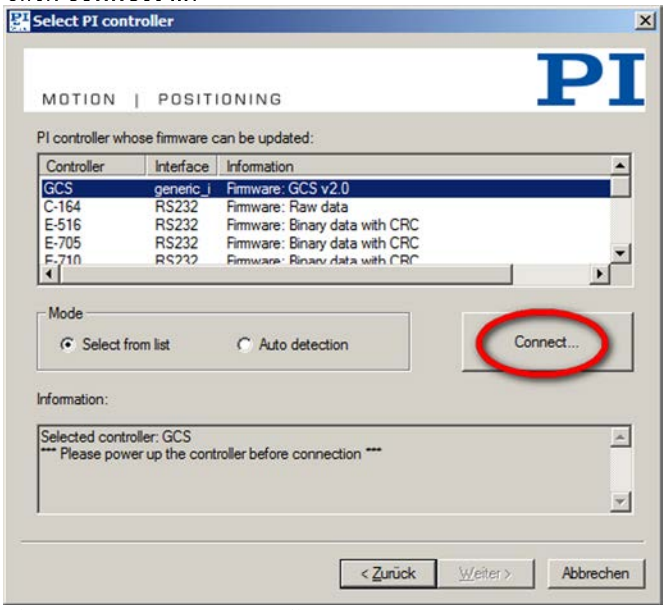
1. Make sure that the **Power** switch of the E-727 is in the OFF position (0).
2. Connect the included wide-range-input power supply to the 24 VDC connection of the E-727 via the included adapter.
3. Connect the E-727 to the PC via USB interface (USB type B).
4. Switch on the E-727:
   a) Connect the 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 (|).

   The power-on/reboot sequence is finished when the **Power** LED is continuously lit. Only then the E-727 is ready for the update.
5. Start the PI Firmware Update Wizard on the PC.
   – Use ***Start > All programs > PI > PIFirmwareWizard***, or navigate to the folder, where the PIFirmwareWizard is installed on your PC, and start it by double-clicking on PIFirmwareWizard.exe
6. In the ***Welcome*** window, click ***Next*** (“Weiter” means “Next”, “Abbrechen” means “Abort”).

   The ***Select PI controller*** window will open.
7. Make sure that ***Select from list*** is selected. Then select ***GCS*** as controller from the list:



8. Click ***Connect …***.

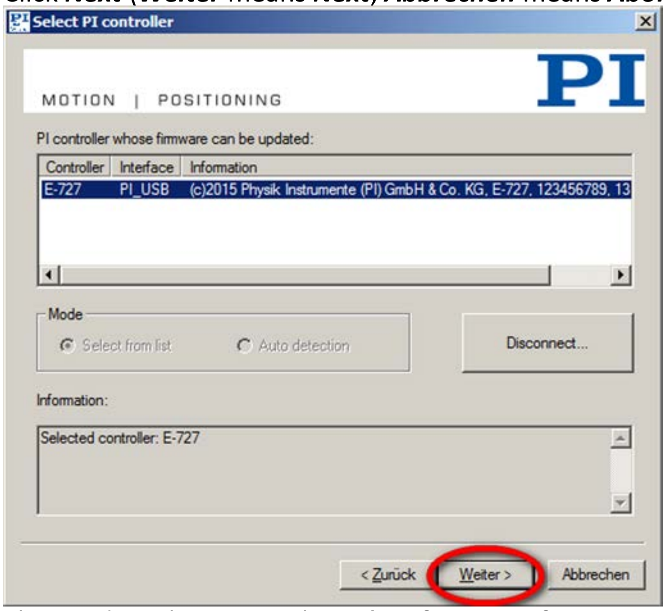


The ***Connect*** window will open.

9. Select the ***PI USB*** tab. When the controller description is displayed as in the example, click ***OK*** to establish the connection.

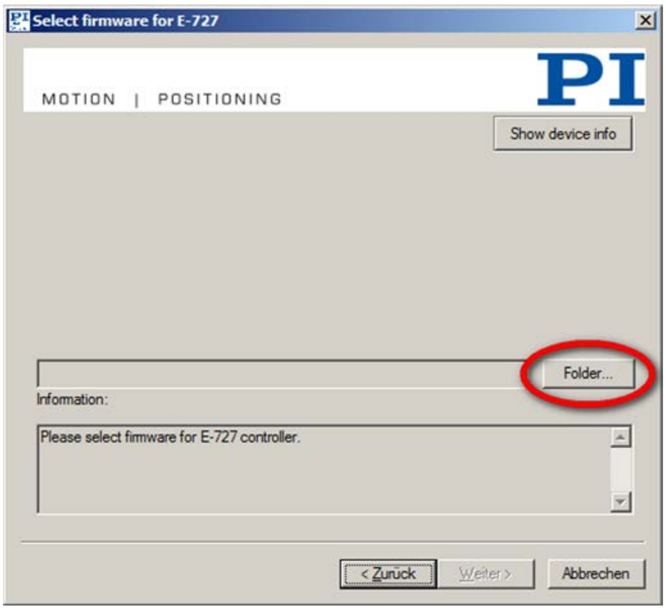
The ***Connect*** window will be closed. In the ***Select PI controller*** window, now information on the connected E-727 should be displayed as shown below.

10. Click ***Next*** (***Weiter*** means ***Next***, ***Abbrechen*** means ***Abort***, ***Zurück*** means ***Back***).



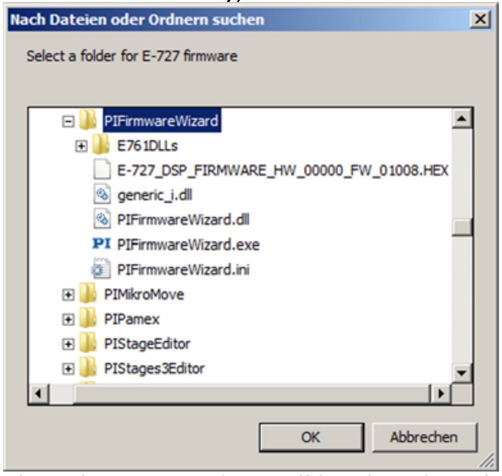
The window changes to the ***Select firmware for E-727*** step.

11. Click ***Folder...*** to select the directory where you have saved the firmware file on the PC:



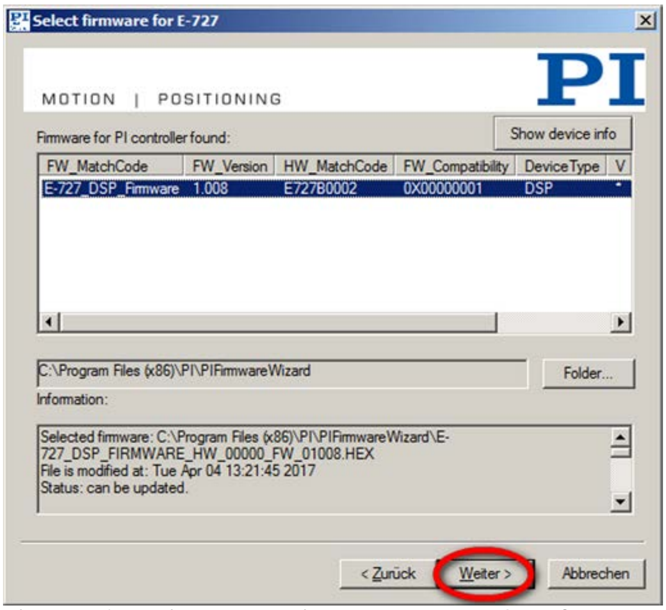
A selection window opens.

12. Select the directory, where the file is located (do **not** select the file!), and click ***OK***:



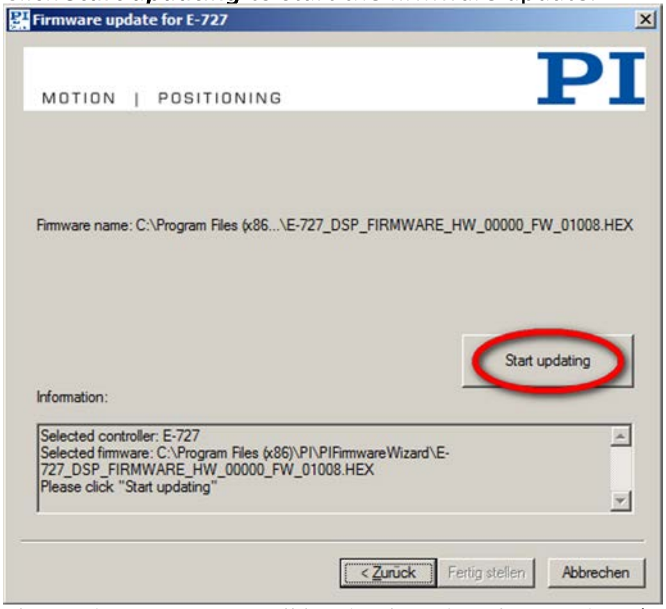
The selection window will be closed. In the ***Select firmware for E-727*** window, now the firmware is displayed as shown below.

13. Click ***Next*** (***Weiter*** means ***Next***):

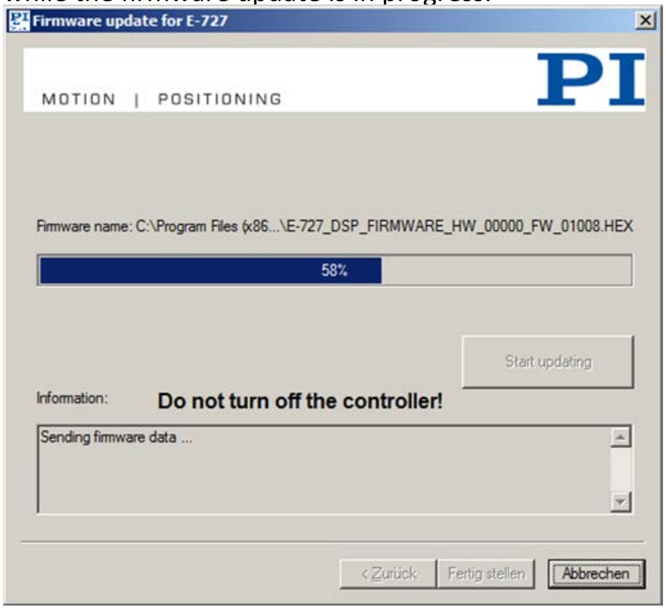


The window changes to the ***Firmware update for E-727*** step.

14. Click ***Start updating*** to start the firmware update.

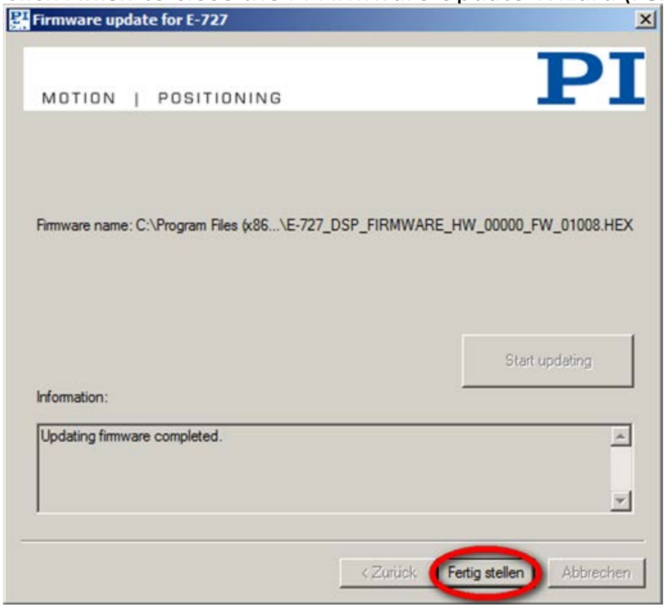


The update progress will be displayed in the window (see below). Do **not** turn off the E-727 while the firmware update is in progress.



When the firmware update was successful, the E-727 will be rebooted automatically with the new firmware. After the reboot, the ***Firmware update for E-727*** window should display a "completed" message as shown below.

15. Click ***Finish*** to close the PI Firmware Update Wizard (***Fertig stellen*** means ***Finish***).



Troubleshouting

Case 1: The E-727 does not reboot automatically:

- Do **not** power cycle the E-727 but proceed as follows:
  a) Close the PI Firmware Update Wizard (if it is still open).
  b) Repeat the firmware update beginning with step 5 of “Updating the firmware“ (p. 205).

Case 2: The PI Firmware Update Wizard ends with an error message, and the E-727 does not reboot automatically:

- Follow the procedure described for case 1.

Case 3: After reboot or power cycling of the E-727, at least one of the following LEDs indicates a failed firmware update (e.g. due to power failure or communication error):

- **Ch1/2 OFL** is permanently flashing while all other LEDs are off: FPGA firmware does not work
- **Ch3/4 OFL** is permanently flashing while all other LEDs are off: DSP firmware does not work

Proceed as follows for every firmware that does not work:

1. Close the PI Firmware Update Wizard (if it is still open).
2. Repeat the firmware update beginning with step 5 of “Updating the firmware“ (p. 205).

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

The E-727 is equipped with a cartridge fuse. The fuse holder is located on the front panel of the case (p. 14). 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 a fuse of the suitable type:

   E-727.xxx, .xxxA, .xxxF, .xxxAF: T3.15 AH, 5 x 20 mm

   E-727.xxxP, .xxxAP: T4 AH, 5 x 20 mm
5. Insert the fuse holder in the case and screw it in clockwise using a suitable tool.

Troubleshooting

Fault: Communication with the controller does not work	
Possible Causes	**Solution**
The wrong communication cable is used or it is defective	➢ If necessary, check whether the cable works on a fault-free system.
The power-on/reboot sequence had not finished	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.
The interface is not configured correctly	➢ With the RS-232 interface, check port and baud rate. ➢ With the TCP/IP connection, connect the controller to a network access point **before** you power it on. Check IP address and IP mask (the settings of the devices in the network must be compatible, e.g. the IP address of each device must be unique). Make sure that your network administrator has not set the network to forbid unknown devices like the E-727 to log on. Note that if the controller is already connected to your or another host PC via TCP/IP, a second TCP/IP session cannot be established.
Another program is accessing the interface.	➢ Close the other program.
Problems with special software	➢ Check whether the system works with other software, such as a terminal program or a development environment. ➢ Test the communication by sending the `*IDN?` or `HLP?` command. ➢ Make sure that you end the commands with an LF (line feed). Exception: Single-character commands are not followed by a terminating character; see "GCS Syntax for Syntax Version 2.0" (p. 178).
E-727 models with EtherCAT interface (p. 164): TCP/IP and USB interface are disabled	➢ Before you use the TCP/IP and USB interfaces, make sure that these interfaces are enabled: TCP/IP and USB will be enabled again after a delay of 100 ms when the E-727 has left the OPERATIONAL EtherCAT communication state.

Fault: Stage does not move	
Possible Causes	**Solution**
Cable not connected correctly	➢ Check the cable connections. ➢ If a connection assignment is given on the labels of the E-727 and/or stage, observe this assignment when connecting the stage.

Fault: Stage does not move	
Possible Causes	**Solution**
The stage has been connected to the switched-on E-727	The ID chip of the stage has not been read out. ➢ Switch the E-727 off and on again, or reboot the E-727 with the RBT command or with the corresponding functions of the PC software.
E-727 is defective	➢ Send the ERR? command and check the error code this returns.
Stage or stage cable is defective	➢ If available, replace the defective stage with a different stage of the same type and test the new combination (only possible with stages which are equipped with ID-chips). Prevent damage to the stage as follows: ➢ Prevent the stage from exceeding the maximally permissible operating frequency, e.g. by using suitable waveforms with the wave generator output. ➢ Prevent the stage from oscillating by suitably setting the control parameters.
High-voltage output of the E-727 is deactivated due to overheating	Overheating of the E-727 ➢ See “Overtemp Protection” (p. 39) for possible measures.
Incorrect configuration	➢ Check the parameter settings of the E-727 with the SPA? (volatile memory) and SEP? (nonvolatile memory) commands or in the ***Device Parameter Configuration*** window of PIMikroMove.
Incorrect command or incorrect syntax	➢ Send the ERR? command and check the error code this returns. ➢ Make sure that the used motion commands match the operating mode (p. 30).
Motion commands or wave generator output are ignored.	➢ Send the ERR? command and check the error code this returns. ➢ Observe that only one control source can be used at a time; see "Control Value Generation" (p. 30).
Incorrect axis commanded	An axis identifier is even required in commands on systems with only one axis. ➢ Make sure that the correct axis identifier (p. 22) is used and that the commanded axis belongs to the correct stage.
In the case of analog control, there is no connection between the axis and the analog input.	When the analog input is used as control source and the axis motion is stopped with STP or #24, the behaviour depends on the value of the ***Disconnect Analog Target Input When Stopping*** parameter (ID 0x0E001E00): 1 = the analog input channel is disconnected from the axis; 0 = the analog input channel remains connected to the axis. ➢ If necessary: Reconnect the corresponding input signal channel to the axis. Further information see "Using the Analog Input" (p. 85).
Sensor zero point is not set correctly	Changes in temperature or changes in the mechanical load can cause small deviations of the sensor zero point. When the sensor zero point is not set correctly, the complete output voltage range of the amplifier cannot be used in closed-loop operation. ➢ Start an AutoZero procedure (p. 62).

Fault: Stage does not move	
Possible Causes	**Solution**
E-727 models with EtherCAT interface (p. 164): Transition to the **Ready to switch on** state not possible	➢ Make sure that the conditions for a transition to **Ready to switch on** are fulfilled (p. 169).

Fault: Stage executes unintentional motion	
Possible Causes	**Solution**
Configuration of the E-727	The E-727 can be configured with parameter settings so that the AutoZero procedure (p. 62) is automatically executed after switching on or rebooting. ➢ Check the setting of the ***Power Up AutoZero Enable*** parameter (ID 0x07000802) and adjust it if necessary.
Start-up macro is executed	➢ Check whether a macro is specified as the start-up macro and cancel the selection of the start-up macro if necessary (p. 123).

Fault: Stage is oscillating or positions inaccurately	
Possible Causes	**Solution**
The load was changed.	Unsuitable settings of the notch filter and the servo-control parameters of the E-727 can cause the stage to oscillate or to position inaccurately. 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 switch off the E-727. ➢ 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. 136) and "Checking and Optimizing the Servo-Control Parameters" (p. 140).
Thermal instability of the system	The E-727 performance can be reduced directly after power on due to thermal instability. ➢ Switch the E-727 on at least one hour before starting work. ➢ If the E-727 is not used, but should remain switched on to ensure the temperature stability: Make sure that the servo mode is switched off (open-loop operation) and the piezo output voltage is set to 0 V. To set the piezo output voltage to 0 V, set the axis position to a corresponding value with the SVA command.

Fault: Stage is oscillating or positions inaccurately	
Possible Causes	**Solution**
Electromagnetic signal causes noise of the sensor signal.	➢ Check the sensor signal. If the sensor signal seems to be abnormal: ➢ Avoid interfering signals. ➢ Take particular care to ensure suitable shielding and grounding. For more information, download the “Guide to Grounding ans Shielding” from our website: 1) Open the website www.pi.ws. 2) Search for A000T0074. 3) In the search results, click the ***Downloads*** tab. 4) Download the Technical Note A000T0074 “Guide to Grounding and Shielding”.
Reading/writing parameters from/to nonvolatile memory	While parameter values are being read from or written to the nonvolatile memory of the E-727, the servo mode is temporarily switched off. This may lead to faulty control or large position errors. ➢ Observe the following when you use the commands SEP, SEP?, RPA, WPA, or the corresponding operating elements in the PC software: - Avoid motions of the axes. - Do not carry out measurements with the axes.

Fault: The customer software does not run with the PI drivers	
Possible Causes	**Solution**
Incorrect combination of driver routines/Vis	➢ Check whether the system works with a terminal program. If so: ➢ Read the information in the corresponding software manual and compare the sample code on the product CD with your program code.

Fault: The Device Parameter Configuration window is not available in PIMikroMove.	
Possible Causes	**Solution**
NI LabVIEW Run-Time Engine has not been installed	➢ Install the NI LabVIEW Run-Time Engine, see "Performing the Initial Installation" (p. 43).

If the problem that occurred with your system is not listed in the table above or cannot be solved as described, contact our customer service department (p. 234).

Technical Data

Specifications

Data Table

	E-727
Function	Digital controller for multi-axis piezo nanopositioning systems. Additional functions: .xxxA: Analog interfaces .xxxF: EtherCAT interface .xxxAF: Analog interfaces, EtherCAT interface .xxxP: Higher output current .xxxAP: Analog interfaces, higher output current
Axes	E-727.3x: 3 E-727.4x: 4
Processor	DSP 32/64-bit floating point, 375 MHz
Sampling rate, servo control	20 kHz
Sampling rate, sensor	100 kHz

Sensor	**E-727**
Servo characteristics	P-I, two notch filters Optional: Advanced Piezo Control
Sensor type	E-727.xCxxx: Capazitive E-727.xSxxx: Strain gauge sensors E-727.xRxxx: Piezoresistive
Sensor channels	E-727.xCxxx: 3 E-727.xSxxx, E-727.xRxxx: 4
Sensor bandwidth (-3 dB)	10 kHz
Sensor resolution (at 1 kHz oversampling)	20 bit

Amplifier	**E-727.xxx, E-727.xxxA, E-727.xxxF, E-727.xxxAF**	**E-727.xxxP, E-727.xxxAP**
Output voltage	-30 to 130 V (±3 V)	-30 to 130 V (±3 V)
Amplifier channels	4	4
Peak output power / channel	28 W max. 30 ms	270 W max. 10 ms
Continuous output power / channel	14 W	30 W
Peak output current / channel	180 mA max. 30 ms	1500 mA max. 10 ms

Amplifier	**E-727.xxx, E-727.xxxA, E-727.xxxF, E-727.xxxAF**	**E-727.xxxP, E-727.xxxAP**
Continuous output current / channel	75 mA	200 mA
Current limitation	Short-circuit proof	Short-circuit proof
Resolution DAC	20 bit	20 bit
Amplifier bandwidth	6.5 kHz	6.5 kHz

Communication	**E-727.xxx, E-727.xxxA, E-727.xxxP, E-727.xxxAP**	**E-727.xxxF, E-727.xxxAF**
PC	TCP/IP, USB, RS-232	TCP/IP, USB
SPI	Connection for SPI master for fast serial transmission of target position and current position	Connection for SPI master for fast serial transmission of target position and current position
Fieldbus	-	EtherCAT (CoE = CANopen over EtherCAT)

Interfaces	**E-727.xxx, E-727.xxxF, E-727.xxxP**	**E-727.xxxA, E-727.xxxAF, E-727.xxxAP**
Piezo / sensor connection	E-727.xCxx: D-Sub 25W3 (f) E-727.xSxx, E-727.xRxx: D-Sub 37 (f)	E-727.xCxxx: D-Sub 25W3 (f) E-727.xSxxx, E-727.xRxxx: D-Sub 37 (f)
Analog inputs	-	D-Sub 15 (f) 4 inputs ±5 V or ±10 V 18-bit A/D converter
Analog output	-	D-Sub 15 (f) ±10 V 20-bit D/A converter
Sensor monitor	-	D-Sub 15 (f) Sensor channels 1 to 3
Digital input / output	MDR14; 4 inputs, 4 outputs	MDR14; 4 inputs, 4 outputs
Separate protective earth connection	Yes	Yes

Operation	**E-727**
Communication via PC	Command set: PI General Command Set (GCS) User software: PIMikroMove® Software drivers: Drivers for use with NI LabVIEW and MATLAB, shared libraries for Windows and Linux; extensive sample code
Communication via fieldbus (EtherCAT)	Drive profile CiA402 (IEC 61800-7-201) Modes of operation: Cyclic Synchronous Position (CSP), Profile Position (PP), Homing Min. cycle time: 2 ms

Operation	E-727
Supported functions	Auto Zero, ID-chip detection Only accessible via GCS commands: Wave generator, data recorder, macros
Display and indicators	LEDs for Power, Servo, Error, Overflow, EtherCAT communication
Linearization	4th order polynomials, DDL (Dynamic Digital Linearization)

Miscellaneous	E-727.xxx, E-727.xxxA, E-727.xxxF, E-727.xxxAF	E-727.xxxP, E-727.xxxAP
Operating temperature range	5 to 40 °C	5 to 40 °C
Overheat protection	Deactivation of the voltage output at 72 °C	Deactivation of the voltage output at 72 °C Alert threshold at 66 °C
Mass	2.4 to 2.6 kg	3.3 kg
Fuse	1 x T3.15 AH, 5 x 20 mm	1 x T4 AH, 5 x 20 mm
Max. power consumption	80 W	84 W
Max. power consumption without load	24 W	40 W
Operating voltage	24 V DC (external power supply in the scope of delivery)	24 V DC (external power supply in the scope of delivery)

Maximum Ratings

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

Input on:	Maximum Operating Voltage ⚠	Operating Frequency ⚠	Maximum Power Consumption ⚠
M8 panel plug	24 V	⎓	80 W

Ambient Conditions and Classifications

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

Area of application	For indoor use only
Maximum altitude	2000 m
Relative humidity	Highest relative humidity 80% for temperatures up to 31°C Decreasing linearly to 50% relative humidity at 40°C
Storage temperature	0°C to 70°C
Transport temperature	–25°C to +85°C

Overvoltage category	II
Protection class	I
Degree of pollution	2
Degree of protection according to IEC 60529	IP20

Operating Limits

In order to achieve minimum distortion of the output waveform, it is important to ensure that the amplitude of higher-frequency control input is reduced in proportion to the fall-off of the output voltage at these frequencies. For exact information on maximum operating frequency with a given piezo load (capacitance), refer to the individual operating limit graphs in the figure below.

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

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

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

$Pa \approx C \cdot Usupply \cdot Up\text{-}p \cdot f$

Peak power for sinusoidal operation is:

$Pmax \approx \pi \cdot C \cdot Usupply \cdot Up\text{-}p \cdot f$

Where:

Pa = average power [W]

Pmax = peak power [W]

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

f = operating frequency [Hz]

Usupply = nominal voltage of the amplifier [V]; 182 V with E-727

Up-p = peak-peak drive voltage [V]

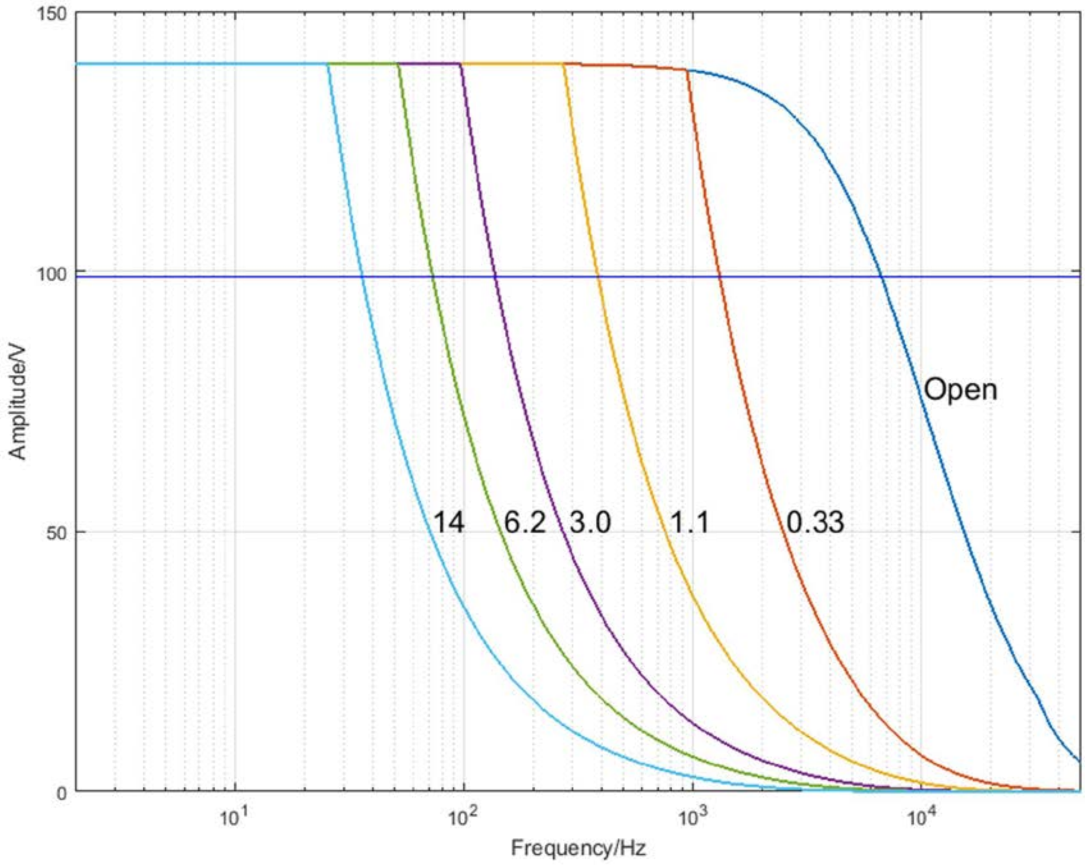


Figure 40: E-727.xxx, E-727.xxxA operating limits with various piezo loads (open-loop operation). Capacitance values in µF

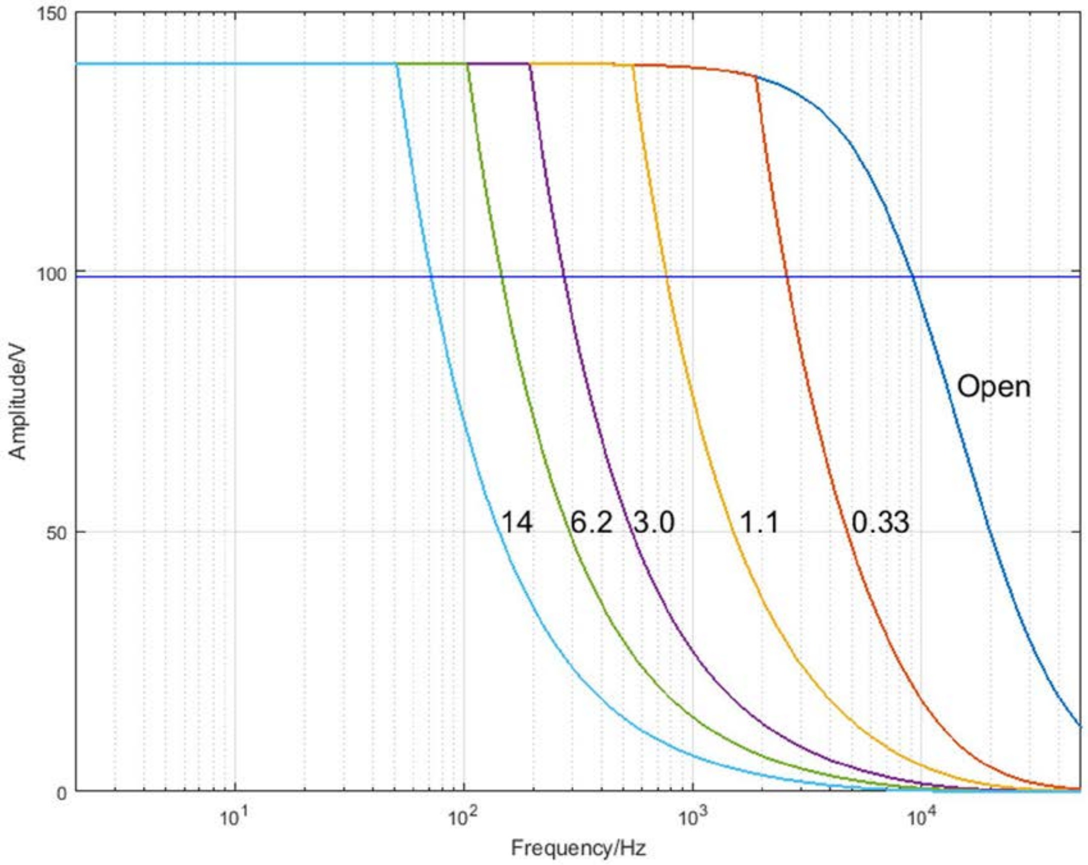


Figure 41: E-727.xxxP, E-727.xxxAP operating limits with various piezo loads (open-loop operation). Capacitance values in µF

System Requirements

The following system requirements must be met to operate the E-727:

- PC with Windows (Windows 7, 8, and 10 (32 bit, 64 bit)) or Linux operating system and at least 30 MB free memory
- Communication interface and corresponding cable:
  - Free COM port on the PC (via RS-232 null-modem cable) or
  - USB socket on the PC (via USB cable, type A to type B) or
  - A free access point on a network to which the PC is connected, or an Ethernet card in the PC (via straight-through or cross-over network cable)
- E-727 with power supply
- The mechanics (piezo stage) with which the E-727 was delivered as a pre-configured system , or any suitable Physik Instrumente mechanics with ID-chip
- Product CD with PC software

Dimensions

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

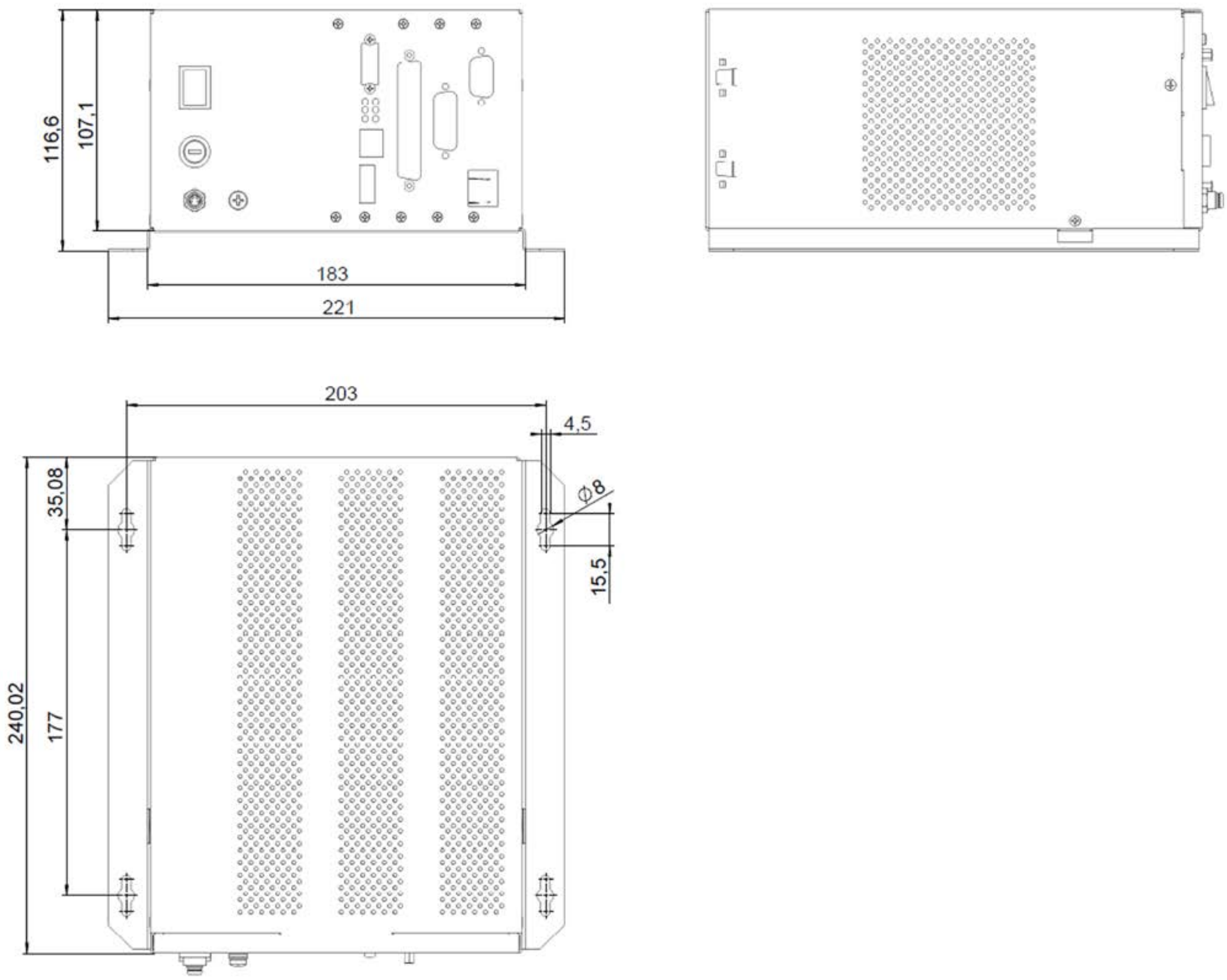


Figure 42: Dimensions of models E-727.xxx, .xxxA, .xxxF, .xxxAF

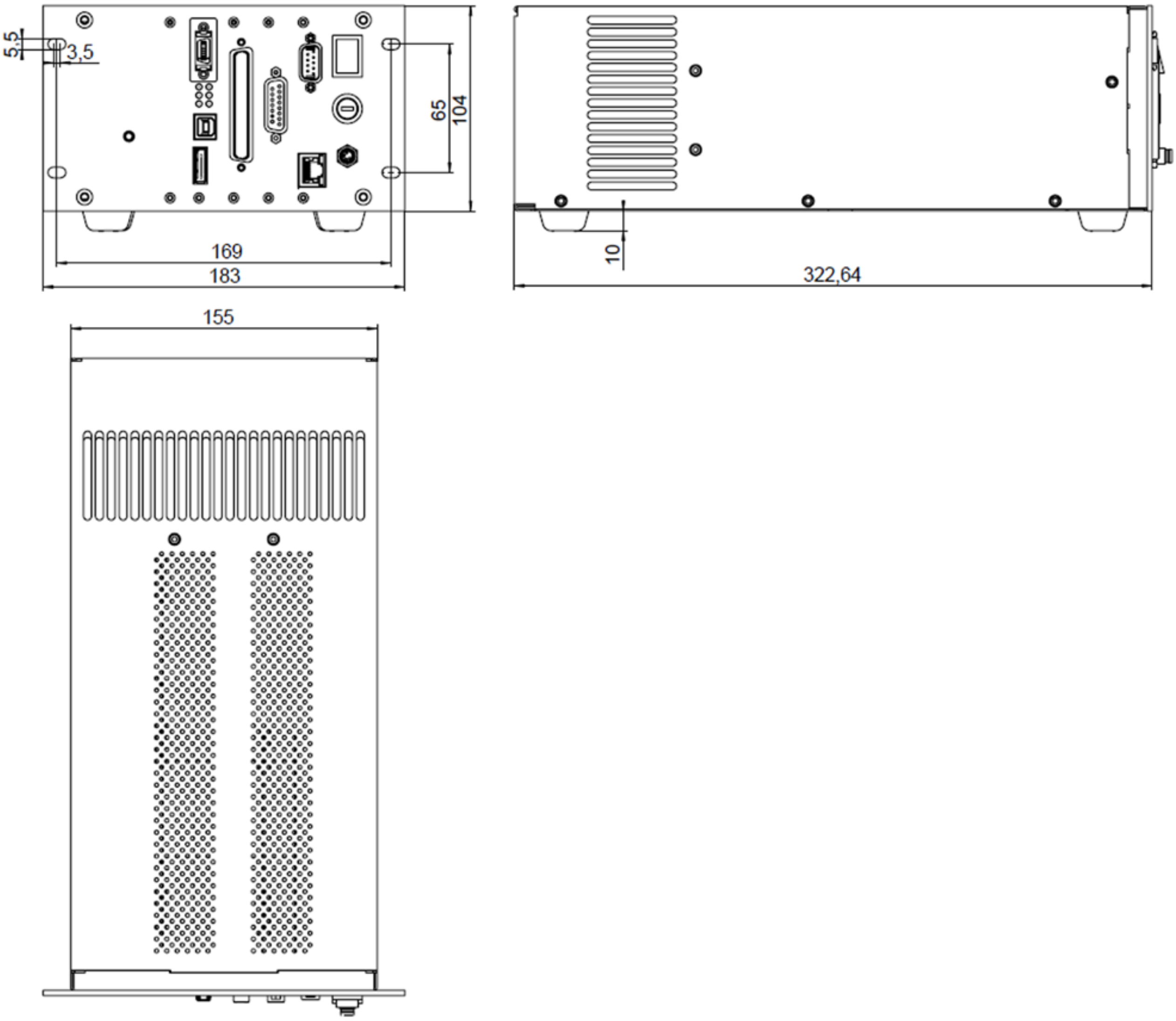


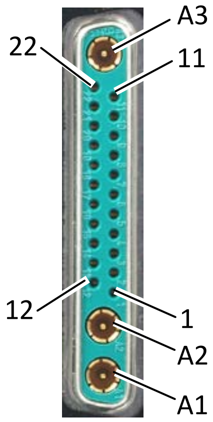
Figure 43: Dimensions of E-727.xxxP, .xxxAP models for higher output current

Pin Assignment

E-727 for Capacitive Sensors: Socket for Piezo Stages

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

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



With E-727.3CDA and .3CDAx 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:

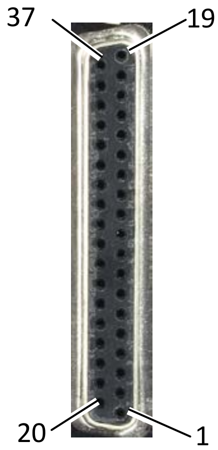
- 1: Output voltage for a piezo actuator in the stage, output as Piezo Ch 4 (pins 7 and 18)
- 2: Position monitor of an axis. The value of the ***Select Output Index*** parameter (ID 0x0A000004) determines the axis whose position is to be output. Output on pin 8 of the **Analog I/O** socket (p. 228).
- 5: Control signal for an external amplifier. 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. 234).

E-727 for Piezoresistive Sensors and SGS: Socket for Piezo Stages

„-30 to +130 V“ – D-Sub 37 (f)

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



Either the lines for the PT1000 temperature sensor or the lines for the 4th piezoresistive or strain gauge 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 piezoresistive or strain gauge sensor (pins 6, 24, 25)
- 2: Use with a PT1000 temperature sensor (pins 1, 20)

With E-727.3RDA, .3RDAx, .3SDA and E-727.3SDAx 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 piezoresistive or strain gauge 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. 228))
- 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-727.3RDA, .3RDAx, .3SDA and .3SDAx 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. 228).
- 5: Control signal for an external amplifier. 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. 234).

Digital I/O

„Digital I/O“ - MDR14

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

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).

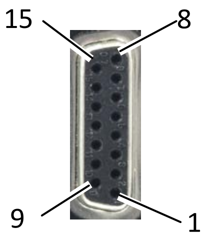
Digital outputs (pins 3, 4, 5, 8):

- 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.xxxA, E-727.xxxAx: Analog I/O

„Analog I/O“ – D-Sub 15 (f)

Pin		Function	Channel identifier*
1		GND	-
	9	-Analog In 1	Input signal channel 4
2		+Analog In 1	
	10	-Analog In 2	Input signal channel 5
3		+Analog In 2	
	11	-Analog In 3	Input signal channel 6
4		+Analog In 3	
	12	-Analog In 4	Input signal channel 7
5		+Analog In 4	
	13	GND	-
6		GND	-
	14	Sensor Monitor 1	-
7		Sensor Monitor 2	-
	15	Sensor Monitor 3	-
8		Analog Out 1	Output signal channel 4



* For further information on channel identifiers, see „Axes, Channels, Functional Elements” (p. 22).

INFORMATION

When using an analog input of the E-727, both the corresponding +Analog In and –Analog In line must be wired.

➢ Connect a differential signal (+analog, -analog, GND) as follows:

Customer Device	E-727.xxxA, .xxxAx
+Analog Out	+Analog In
-Analog Out	-Analog In
GND	GND

➢ Connect a single-ended signal as follows (recommended):

Customer Device	E-727.xxxA, .xxxAx
+Analog Out	+Analog In
GND	-Analog In*
GND	GND

* If it is not possible to connect –Analog In to GND on the customer side, -Analog In should be connected to GND on the E-727 side.

➢ In either case, use a shielded cable.

INFORMATION

To achieve the highest possible resolution and eliminate potential interference that affects the cable used:

➢ Filter the analog output signal of the E-727 in a suitable way, e.g., before you convert it to a digital format. Recommended: low-pass filter with max. 100 kHz cut-off frequency (characteristics: single pole, 6 dB/octave)

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

- Input impedance: 150 kohm
- Max. input voltage (single ended): ±14 V
- Resolution ADC: 18 bit
- E-727.3CDA, .3CDAx: 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-727.3RDA, .3RDAx, .3SDA, .3SDAx: 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 a piezoresistive or strain gauge sensor (input via pins 6, 24, 25 of the socket for the piezo stage(s) (p. 226)
    - 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 DAC: 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. 225 or p. 226) 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. 225 or p. 226).
  - 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 amplifier. 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. 234).

E-727.IO3x Analog Input Cable

The E-727.IO3x analog input cable splits the input lines of the ***Analog I/O*** socket (p. 228) up into separate wires.

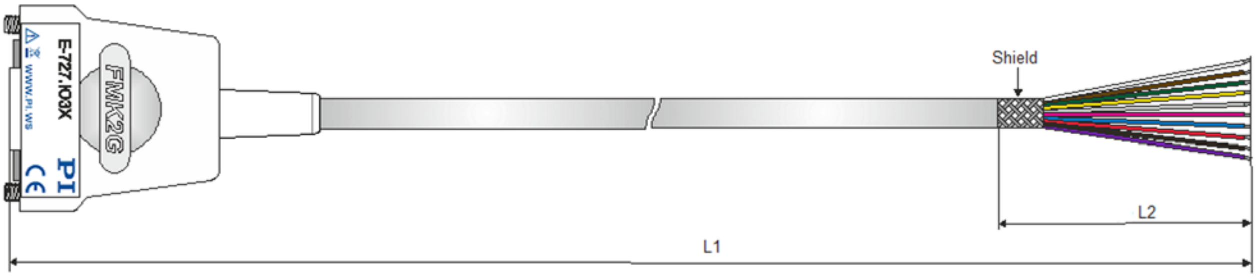


Figure 44: E-727.IO3x cable, D-Sub 15 (m) to open end

L1 = 1 m ±10 cm
L2 = 6 cm ±1 cm

D-Sub 15 (m)	**Signal**	Wire pair	Color
1	GND	Pair 1	brown
9	-Analog In 1	Pair 2	green
2	+Analog In 1	Pair 2	yellow
10	-Analog In 2	Pair 3	grey
3	+Analog In 2	Pair 3	pink
11	-Analog In 3	Pair 4	blue
4	+Analog In 3	Pair 4	red
12	-Analog In 4	Pair 5	black
5	+Analog In 4	Pair 5	purple
13	GND	Pair 1	white
Hood	Shield	Cable shield	---

INFORMATION

When using an analog input of the E-727, both the corresponding +Analog In and –Analog In line must be wired.

➢ Observe the wiring information in “E-727.xxxA, E-727.xxxAx: Analog I/O“ (p. 228).

E-727.IO8 Adapter Cable for Analog I/O

The E-727.IO8 adapter cable is available as accessory (p. 19). The cable splits the input and output lines of the ***Analog I/O*** socket (p. 228) up into 8 BNC connectors.

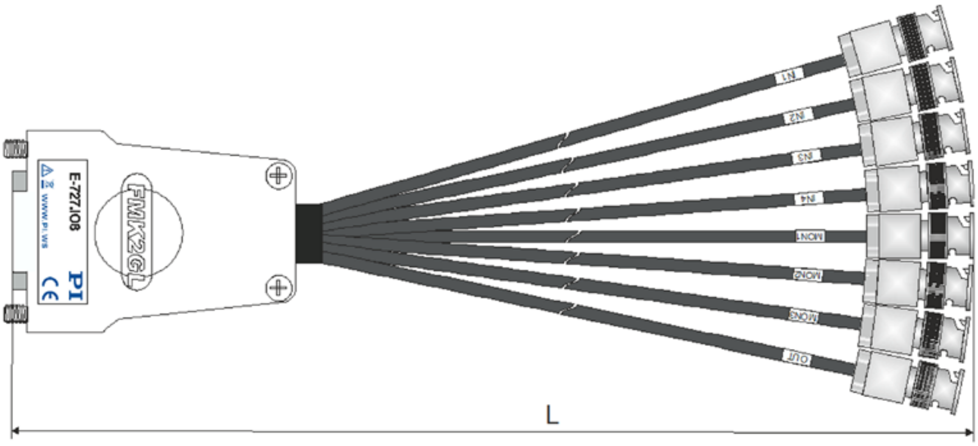


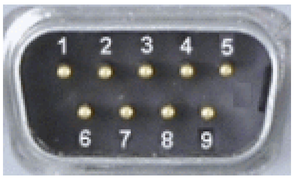
Figure 45: E-727.IO8 adapter cable, D-Sub 15 (m) to 8 x BNC, L = 0.4 m (tolerance +5 cm)

D-Sub 15 (m)	Signal	Cable labeling
1	GND	-
9	GND	IN1
2	+Analog In 1	
10	GND	IN2
3	+Analog In 2	
11	GND	IN3
4	+Analog In 3	
12	GND	IN4
5	+Analog In 4	
13	GND	-
6	GND	-
14	Sensor Monitor 1	Monitor1
7	Sensor Monitor 2	Monitor2
15	Sensor Monitor 3	Monitor3
8	Analog Out 1	OUT

RS-232

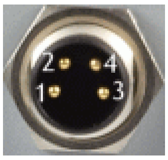
„RS-232“ – D-Sub 9 (m)

Pin	Function
1	nc
2	RXD receive data
3	TXD send data
4	nc
5	DGND ground
6	nc
7	RTS Hardware handshake, output
8	CTS Hardware handshake, input
9	nc



Power Supply 24 V

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



Pin	Function
1	GND (power)
2	GND (power)
3	Input: 24 V DC
4	Input: 24 V DC

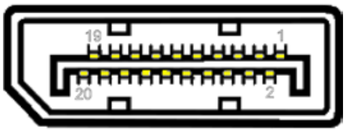
K050B0003 Adapter, Barrel Connector Side

Center: +24 V

Outer conductor: GND

SPI Interface

„SPI“ – DisplayPort (VESA), lockable



E-727 (SPI slave):

Pin	Signal Name	Direction
20	N.C.	
19	LDAT-	Input
18	LDAT+	Input
17	CS-	Input
16	GND	
15	CS+	Input
14	GND	
13	GND	
12	SCLK+	Input
11	GND	
10	SCLK-	Input
9	MOSI+	Input
8	GND	
7	MOSI-	Input
6	MISO+	Output
5	GND	
4	MISO-	Output
3	DCLK+	Output
2	GND	
1	DCLK-	Output

Host (SPI master):

Pin	Signal Name	Direction
20	N.C.	
19	LDAT-	Output
18	LDAT+	Output
17	CS-	Output
16	GND	
15	CS+	Output
14	GND	
13	GND	
12	DCLK-	Input
11	GND	
10	DCLK+	Input
9	MISO-	Input
8	GND	
7	MISO+	Input
6	MOSI-	Output
5	GND	
4	MOSI+	Output
3	SCLK-	Output
2	GND	
1	SCLK+	Output

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. 10) on our website.

Old Equipment Disposal

In accordance with EU law, electrical and electronic equipment may not be disposed of in EU member states via the municipal residual waste.

Dispose of your old equipment according to international, national, and local rules and regulations.

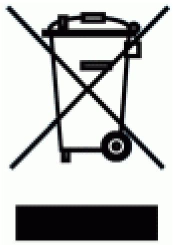
In order to fulfil its responsibility as the product manufacturer, Physik Instrumente (PI) GmbH & Co. KG undertakes environmentally correct disposal of all old PI equipment made available on the market after 13 August 2005 without charge.

Any old PI equipment can be sent free of charge to the following address:

Physik Instrumente (PI) GmbH & Co. KG

Auf der Roemerstr. 1

D-76228 Karlsruhe, Germany



EU Declaration of Conformity

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

Low Voltage Directive (LVD)

EMC Directive

RoHS Directive

The applied standards certifying the conformity are listed below.

Safety (LVD):	EN 61010-1
EMC:	EN 61326-1
RoHS:	EN 50581

Appendix: EtherCAT Object Dictionary

Object range	Index:Sub-index	Name	Access	Time of writing	Default Hex value (:Sub-index, Size)	Dec value	Data type	Comment
	1000	Device type	RO		0x00020192	131474	UNINT32	
	1001	Error register	RO		0x00	0	UINT8	
	1008	Manufacturer device name	RO				STRING	see p. 13
	1009	Manufacturer hardware version	RO		0		STRING	
	100A	Manufacturer software version	RO				STRING	
	1018:00	Idendity object	RO		4		UINT8	
	1018:01	Vendor ID	RO		0x0000076D	1901	UNINT32	
	1018:02	Product code	RO		0x00000001	1	UNINT32	
	1018:03	Revision number	RO		0	0	UNINT32	
	1018:04	Serial number	RO				UNINT32	
Mappings Rx	1600:00	RxPDO - CSP	RO		3			
	1600:01	Controlword	RO		0x6040:00, 16			
	1600:02	Modes of operation	RO		0x6060:00, 8			
	1600:03	Target position	RO		0x607A:00, 32			
	1601:00	RxPDO - PP	RO		5			
	1601:01	Controlword	RO		0x6040:00, 16			
	1601:02	Modes of operation	RO		0x6060:00, 8			
	1601:03	Target position	RO		0x607A:00, 32			
	1601:04	Profile velocity	RO		0x6081:00, 32			
	1601:05	Profile acceleration	RO		0x6083:00, 32			
	1610:00	RxPDO - CSP	RO		3			
	1610:01	Controlword	RO		0x6840:00, 16			
	1610:02	Modes of operation	RO		0x6860:00, 8			
	1610:03	Target position	RO		0x687A:00, 32			
	1611:00	RxPDO - PP	RO		5			
	1611:01	Controlword	RO		0x6840:00, 16			
	1611:02	Modes of operation	RO		0x6860:00, 8			
	1611:03	Target position	RO		0x687A:00, 32			
	1611:04	Profile velocity	RO		0x6881:00, 32			
	1611:05	Profile acceleration	RO		0x6883:00, 32			
	1620:00	RxPDO - CSP	RO		3			

Object range	Index:Sub-index	Name	Access	Time of writing	Default Hex value (:Sub-index, Size)	Dec value	Data type	Comment
	1620:01	Controlword	RO		0x7040:00, 16			
	1620:02	Modes of operation	RO		0x7060:00, 8			
	1620:03	Target position	RO		0x707A:00, 32			
	1621:00	RxPDO - PP	RO		5			
	1621:01	Controlword	RO		0x7040:00, 16			
	1621:02	Modes of operation	RO		0x7060:00, 8			
	1621:03	Target position	RO		0x707A:00, 32			
	1621:04	Profile velocity	RO		0x7081:00, 32			
	1621:05	Profile acceleration	RO		0x7083:00, 32			
Mappings Tx	1A00:00	TxPDO - CSP	RO		4			
	1A00:01	Statusword	RO		0x6041:00, 16			
	1A00:02	Modes of operation display	RO		0x6061:00, 8			
	1A00:03	Position actual value	RO		0x6064:00, 32			
	1A00:04	Following error actual value	RO		0x60F4:00, 32			
	1A01:00	TxPDO - PP	RO		3			
	1A01:01	Statusword	RO		0x6041:00, 16			
	1A01:02	Modes of operation display	RO		0x6061:00, 8			
	1A01:03	Position actual value	RO		0x6064:00, 32			
	1A10:00	TxPDO - CSP	RO		4			
	1A10:01	Statusword	RO		0x6841:00, 16			
	1A10:02	Modes of operation display	RO		0x6861:00, 8			
	1A10:03	Position actual value	RO		0x6864:00, 32			
	1A10:04	Following error actual value	RO		0x68F4:00, 32			
	1A11:00	TxPDO - PP	RO		3			
	1A11:01	Statusword	RO		0x6841:00, 16			
	1A11:02	Modes of operation display	RO		0x6861:00, 8			
	1A11:03	Position actual value	RO		0x6864:00, 32			
	1A20:00	TxPDO - CSP	RO		4			
	1A20:01	Statusword	RO		0x7041:00, 16			
	1A20:02	Modes of operation display	RO		0x7061:00, 8			
	1A20:03	Position actual value	RO		0x7064:00, 32			
	1A20:04	Following error actual value	RO		0x70F4:00, 32			
	1A21:00	TxPDO - PP	RO		3			

Object range	Index:Sub-index	Name	Access	Time of writing	Default Hex value (:Sub-index, Size)	Dec value	Data type	Comment
	1A21:01	Statusword	RO		0x7041:00, 16			
	1A21:02	Modes of operation display	RO		0x7061:00, 8			
	1A21:03	Position actual value	RO		0x7064:00, 32			
PDO Assign-ments	1C00:00	Sync Manager communication types	RO		4			
	1C00:01	SubIndex 001	RO		0x01	1		
	1C00:02	SubIndex 002	RO		0x02	1		
	1C00:03	SubIndex 003	RO		0x03	1		
	1C00:04	SubIndex 004	RO		0x04	1		
	1C10:00	Sync Manager 0 PDO assignment	RO		0			
	1C11:00	Sync Manager 1 PDO assignment	RO		0			
	1C12:00	Sync Manage r2 PDO assignment	RW	P->S	4			
	1C12:01	SubIndex 001	RW	P->S	0x1600	5632		
	1C12:02	SubIndex 002	RW	P->S	0x1610	5648		
	1C12:03	SubIndex 003	RW	P->S	0x1620	5664		
	1C12:04	SubIndex 004	RW	P->S	0	0		For future use
	1C13:00	Sync Manager3 PDO assignment	RW	P->S	4			
	1C13:01	SubIndex 001	RW	P->S	0x1A00	6656		
	1C13:02	SubIndex 002	RW	P->S	0x1A10	6672		
	1C13:03	SubIndex 003	RW	P->S	0x1A20	6688		
	1C13:04	SubIndex 004	RW	P->S	0	0		For future use
Axis 1	603F	Error code	RO		0x0000	0	UINT16	
	6040	Controlword	RW		0x0000	0	UINT16	
	6041	Statusword	RO		0x0000	0	UINT16	
	6060	Modes of operation	RW		0	0	INT8	
	6061	Modes of operation display	RO		0	0	INT8	
	6064	Position actual value	RO		0	0	INT32	
	607A	Target position	RW		0	0	INT32	
	607B:00	Position range limit	RO		2		UINT8	
	607B:01	Min position range limit	RW		min. axis limit		INT32	
	607B:02	Max position range limit	RW		max. axis limit		INT32	
	607D:00	Software position limit	RO		2		UINT8	
	607D:01	Min position limit	RW		min. axis limit		INT32	

Object range	Index:Sub-index	Name	Access	Time of writing	Default Hex value (:Sub-index, Size)	Dec value	Data type	Comment
	607D:02	Max position limit	RW		max. axis limit		INT32	
	6081	Profile velocity	RW		0	0	UINT32	
	6083	Profile acceleration	RW		0	0	UINT32	
	6098	Homing method	RW		0	0	INT8	
	60F4	Following error actual value	RO		0	0	INT32	
	60FF	Target velocity	RW		0	0	INT32	
	6502	Supported drive modes	RO		0x00000000	0	UINT32	
	67FF	Device profile number	RO		0x00000192	402	UINT32	
Axis 2	683F	Error code	RO		0x0000	0	UINT16	
	6840	Controlword	RW		0x0000	0	UINT16	
	6841	Statusword	RO		0x0000	0	UINT16	
	6860	Modes of operation	RW		0	0	INT8	
	6861	Modes of operation display	RO		0	0	INT8	
	6864	Position actual value	RO		0	0	INT32	
	687A	Target position	RW		0	0	INT32	
	687B:00	Position range limit	RO		2		UINT8	
	687B:01	Min position range limit	RW		min. axis limit		INT32	
	687B:02	Max position range limit	RW		max. axis limit		INT32	
	687D:00	Software position limit	RO		2		UINT8	
	687D:01	Min position limit	RW		min. axis limit		INT32	
	687D:02	Max position limit	RW		max. axis limit		INT32	
	6881	Profile velocity	RW		0	0	UINT32	
	6883	Profile acceleration	RW		0	0	UINT32	
	6898	Homing method	RW		0	0	INT8	
	68F4	Following error actual value	RO		0	0	INT32	
	68FF	Target velocity	RW		0	0	INT32	
	6D02	Supported drive modes	RO		0x00000000	0	UINT32	
	6FFF	Device profile number	RO		0x00000192	402	UINT32	
Axis 3	703F	Error code	RO		0x0000	0	UINT16	
	7040	Controlword	RW		0x0000	0	UINT16	
	7041	Statusword	RO		0x0000	0	UINT16	
	7060	Modes of operation	RW		0	0	INT8	
	7061	Modes of operation display	RO		0	0	INT8	

Object range	Index:Sub-index	Name	Access	Time of writing	Default Hex value (:Sub-index, Size)	Dec value	Data type	Comment
	7064	Position actual value	RO		0	0	INT32	
	707A	Target position	RW		0	0	INT32	
	707B:00	Position range limit	RO		2		UINT8	
	707B:01	Min position range limit	RW		min. axis limit		INT32	
	707B:02	Max position range limit	RW		max. axis limit		INT32	
	707D:00	Software position limit	RO		2		UINT8	
	707D:01	Min position limit	RW		min. axis limit		INT32	
	707D:02	Max position limit	RW		max. axis limit		INT32	
	7081	Profile velocity	RW		0	0	UINT32	
	7083	Profile acceleration	RW		0	0	UINT32	
	7098	Homing method	RW		0	0	INT8	
	70F4	Following error actual value	RO		0	0	INT32	
	70FF	Target velocity	RW		0	0	INT32	
	7502	Supported drive modes	RO		0x00000000	0	UINT32	
	77FF	Device profile number	RO		0x00000192	402	UINT32	