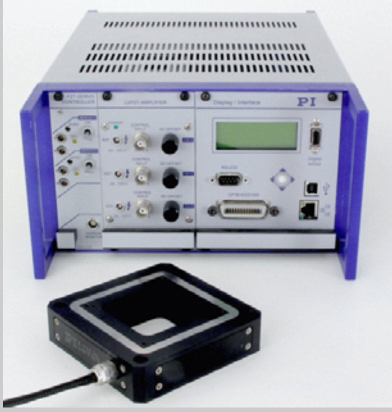
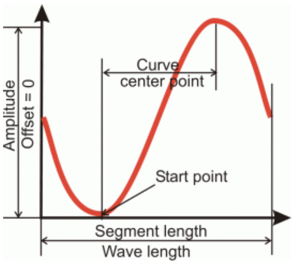
PZ214E User Manual
E-517 Digital Piezo Controller
Operation Module

Release: 1.0.0 Date: 08.12.2008

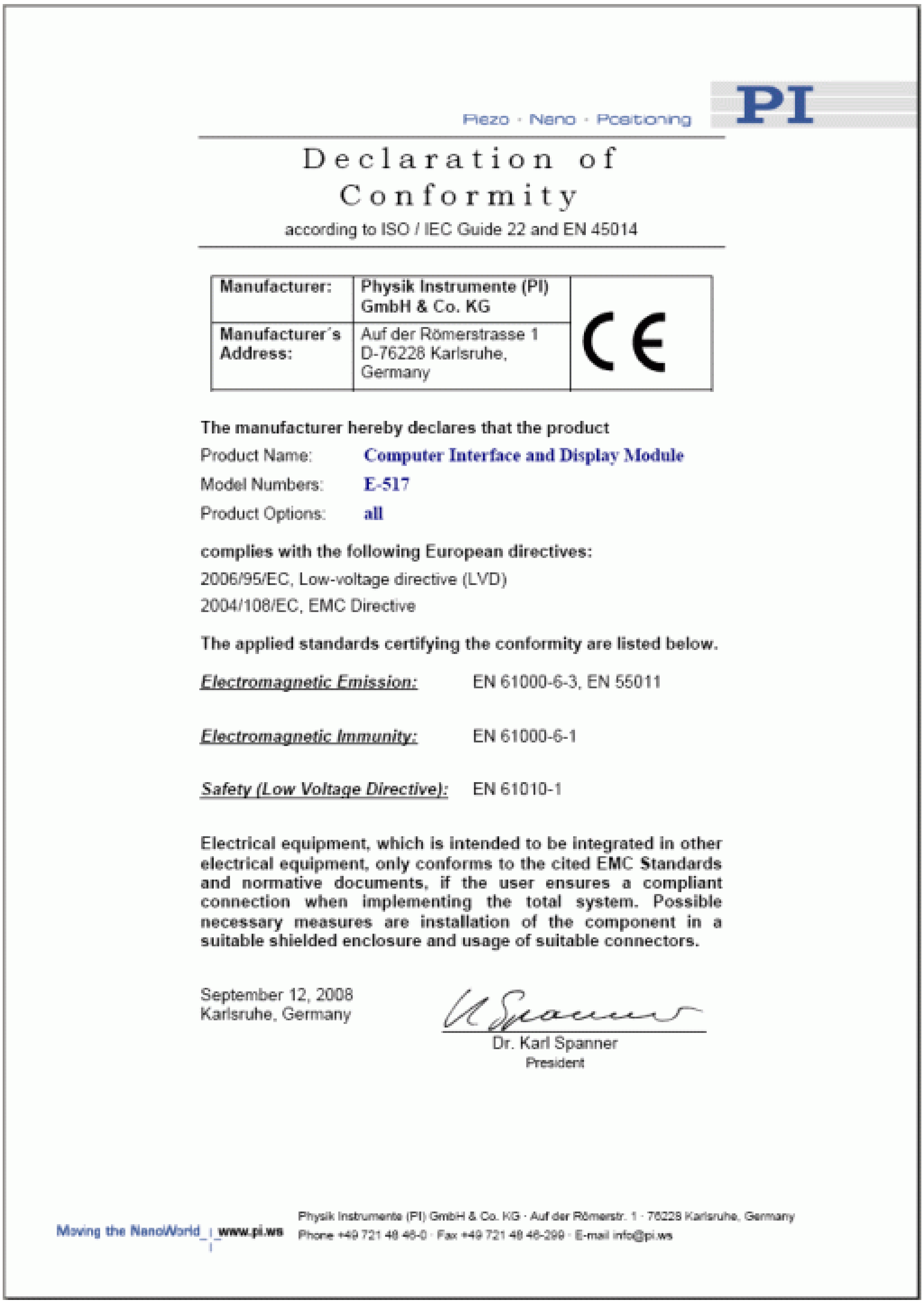


This document describes the following products:

- E-517.i3
  Interface / Display Module, TCP/IP, USB, RS-232,
  3 Channels
- E-517.i1
  Interface / Display Module, TCP/IP, USB, RS-232,
  1 Channel



© Physik Instrumente (PI) GmbH & Co. KG
Auf der Römerstr. 1 · 76228 Karlsruhe, Germany
Tel. +49 721 4846-0 · Fax: +49 721 4846-299
info@pi.ws · www.pi.ws



Piezo · Nano · Positioning PI

Declaration of Conformity

according to ISO / IEC Guide 22 and EN 45014

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

The manufacturer hereby declares that the product

Product Name:	**Computer Interface and Display Module**
Model Numbers:	**E-517**
Product Options:	**all**

complies with the following European directives:

2006/95/EC, Low-voltage directive (LVD)

2004/108/EC, EMC Directive

The applied standards certifying the conformity are listed below.

Electromagnetic Emission:	EN 61000-6-3, EN 55011
Electromagnetic Immunity:	EN 61000-6-1
Safety (Low Voltage Directive):	EN 61010-1

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

September 12, 2008
Karlsruhe, Germany

Dr. Karl Spanner
President

Moving the NanoWorld | www.pi.ws

Physik Instrumente (PI) GmbH & Co. KG · Auf der Römerstr. 1 · 76228 Karlsruhe, Germany
Phone +49 721 48 46-0 · Fax +49 721 48 46-299 · E-mail info@pi.ws

First printing 08.12.2008
Document Number PZ214E BRo, Release 1.0.0
E-517_User_PZ214E100.doc

Subject to change without notice. This manual is superseded by any new release. The newest release is available for download at www.pi.ws (http://www.pi.ws).

About This Document

Users of This Manual

This manual is designed to help the reader to operate the E-517 Interface / Display Module. It assumes that the reader has a fundamental understanding of basic servo systems, as well as motion control concepts and applicable safety procedures.
The manual describes the physical specifications and dimensions of the E-517 as well as the procedures and the commands which are required to put the associated motion system into operation.

Conventions

The notes and symbols used in this manual have the following meanings:

WARNING

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

DANGER

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

CAUTION

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

NOTE

Provides additional information or application hints.

Related Documents

The software tools and the mechanics which might be mentioned in this document are described in their own manuals. All documents are available as PDF files. Updated releases are available for download at www.pi.ws (http://www.pi.ws) or via email: contact your Physik Instrumente Sales Engineer or write info@pi.ws (mailto:info@pi.ws).

E-517 Quick Guide PZ214Equ
PIMikroMove SM148E
E-517 GCS LabVIEW PZ209E
E-517 PIGCS2 DLL PZ215E
GCSData SM146E

Contents

1 Introduction 4

1.1 Prescribed Use 5
1.2 Safety Precautions 6
1.3 Model Survey 8
1.4 Unpacking 8
1.5 Motion System Requirements 9
1.6 Software Description 10

2 First Steps 12

2.1 Installing the Software on the Host PC 12
2.2 Creating Backup File for Controller Parameters 12
2.3 Quick Start 14

3 Operation 20

3.1 Front Panel Elements 20
3.1.1 Front Panel Overview 20
3.1.2 Display Screens 21
3.1.3 Trackball Functions 22
3.2 Modes of Operation 23
3.2.1 Control Modes 23
3.2.2 Servo Mode (ON / OFF) 24
3.3 How to Command Axis Motion 26
3.3.1 Applicable Control Sources 26
3.3.2 Command Examples for ONLINE Mode 27
3.4 Power On / Reboot Sequence 30
3.5 How to Customize the System 30
3.5.1 Set System Parameters 31
3.5.2 Configure Axes and Channels 33
3.5.3 Calibration Settings 38
3.5.4 Select Command Set Version 44
3.6 Updates 45
3.6.1 Software Updates 45
3.6.2 Firmware Updates 46

4 System Description 48

4.1 Basic Elements 48
4.2 Accessible Items and Their Identifiers 50
4.3 Control Value Generation 53
4.3.1 Overview and Signal Path 53

4.3.2 Control Value Generation in ONLINE Mode 54
4.3.3 Drift Compensation 56
4.4 On Target Reading 57

5 Communication 58

5.1 Interfaces Available 58
5.2 Default and Current Settings 59
5.3 TCP/IP Connection 61
5.3.1 Network with DHCP Server 62
5.3.2 PC with Ethernet Card 63
5.4 RS-232 Serial Connection 67
5.5 USB Connection 68
5.6 GPIB (IEEE 488) Parallel Connection 69

6 Data Recording 71

6.1 How to Use the Data Recorder 71
6.2 Application Example: Frequency Response Measurements 72
6.3 Data-Recorder Related Commands and Parameters 75

7 External Triggering/Signaling 77

7.1 Using Trigger Input and Output 77
7.2 Configuring Trigger Output 77
7.2.1 Example—"Position Distance" Trigger Mode 79
7.2.2 Example—"On Target" Trigger Mode 80
7.2.3 Example—"MinMax Threshold" Trigger Mode 81
7.2.4 Example—"Generator Trigger" Mode 82
7.2.5 Example—Polarity Setting 83

8 Wave Generator 84

8.1 How to Work with the Wave Generator 84
8.1.1 Basic Data 84
8.1.2 Basic Operation 85
8.1.3 Additional Steps and Settings 86
8.1.4 Application Notes 89
8.2 Wave Generator Examples 92
8.2.1 Defining Waveforms 92
8.2.2 Modifying the Wave Generator Table Rate 96
8.2.3 Trigger Output Synchronized with Wave Generator 97
8.2.4 Wave Generator Started by Trigger Input 100
8.2.5 Scanning Examples with Wave Generator and Macros 104
8.3 Wave-Generator-Related Commands and Parameters 110

9 Working with Controller Macros 113

9.1 Defining Macros 113
9.2 Starting Macro Execution 115
9.3 Start-Up Macro 115

10 GCS Commands 117

10.1 Format 117
10.1.1 Notation 117
10.1.2 GCS Syntax 118
10.1.3 Limitations for GCS Commands 119
10.2 Command Survey 120
10.2.1 E-517 GCS Commands, Alphabetical Survey 120
10.2.2 E-517 GCS Commands, Survey by Function 124
10.3 Command Reference (alphabetical) 131
10.4 Error Codes 234

11 Controller Parameters 250

11.1 Parameter Handling 250
11.2 Parameter Overview 252

12 Troubleshooting 258

13 Customer Service 263

14 Old Equipment Disposal 264

15 Technical Data 265

15.1 Specifications 265
15.2 Pin Assignments 267
15.2.1 Digital In/Out Socket 267
15.2.2 RS-232 Socket 267
15.2.3 Main Connector 268

1 Introduction

- Low-Noise 24-bit D/A Converter
- Sample Rate 25 kHz
- TCP/IP, USB, IEEE 488 and RS-232 Interfaces
- 6-Digit Display for Voltage and Position
- 1- & 3-Channel Versions
- Wave Generator with Programmable Trigger-I/O
- Module for E-500 Piezo Controller Rack

The E -517 is a microprocessor controlled interface and display module for the E-500 piezo controller system. It is equipped with low-noise, 24-bit D/A converters and can be controlled through four digital interfaces: TCP/IP, USB, RS-232 and IEEE 488 (GPIB).
Alternatively, stand-alone operation is possible by uploading macro command sequences to the internal non-volatile memory. For manual control a trackball interface is provided. An LCD display indicates position or operating voltages of the individual channels.

Wave Generator

The integrated wave generator can output periodic motion profiles. In addition to sine and triangle waves, arbitrary, user-defined motion profiles can be created and stored.

Data Recorder

The flexibly configurable data recorder enables simultaneous recording and read-out of up to 3 input and output signals, such as for positions or control values. This makes possible the later association of events with saved data points.

Extensive Software Support

The controllers are delivered with Windows operating software. Comprehensive DLLs and LabVIEW drivers are available for automated control.

In this document, the E-517.i3 and .i1 models are also referred to as "E-517" or "interface/display module".

1.1 Prescribed Use

The E-517 Interface / Display Module is designed for installation in E-500/E-501, E-471 and E-481 modular analog piezo control electronics. It makes possible controlling the analog piezo control electronics via PC interface (TCP/IP, USB, RS-232, IEEE 488 (GPIB)), trackball, macro programming and wave generator output. The LCD display shows the current voltage and position values.

The E-517 may only be used for applications suitable according to the device specifications. Operation other than instructed in this User Manual may affect the safeguards provided.

The E-517 is a laboratory apparatus as defined by DIN EN 61010. It meets the following minimum specifications for safe operation (any more stringent specifications in the technical data table (p. 265) are, of course, also met):

- Indoor use only
- Altitude up to 2000 m
- Temperature range 5°C to 40°C
- Max. relative humidity 80% for temperatures up to 31°C, decreasing linearly to 50% relative humidity at 40°C
- Line voltage fluctuations not greater than ±10% of the line voltage
- Transient overvoltages as typical for public power supply
  Note: The nominal level of the transient overvoltage is the standing surge voltage according to the overvoltage category II (IEC 60364-4-443).
- Degree of pollution: 2

1.2 Safety Precautions

WARNING--READ INSTRUCTION

Install and operate the E-517 Amplifier / Controller only when you have read the operating instruction. Keep the instruction readily available close to the device in a safe place. When the instruction is lost or has become unusable, ask the manufacturer for a new copy. Add all information given by the manufacturer to the instruction, e.g. supplements or Technical Notes.

CAUTION

Your system will be fully calibrated before being shipped. If you have informed PI about your application, calibration is done with the corresponding setup, otherwise with a default setup. It is usually not necessary for you to do anything more than adjust the zero point(s) before operating the system.

Do not interchange piezo control electronics or piezo stages of a calibrated system. Respect the assignment of piezo stages to the electronics channels. The assignment is shown by the serial numbers on the device labels. With multi-axis stages also respect the channel / axis assignment given by the cable labeling.

Re-calibration should only be done by adequate trained personnel and after consultation with PI. Otherwise preset data will be lost.

CAUTION

Thermally stable systems have the best performance. For a thermally stable system, power on the E-517 at least one hour before you start working with it.

!

CAUTION

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

It is strongly recommended to save the parameter values of the E-517 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-517, use the *Device Parameter Configuration* window of PIMikroMove™. See "Create Backup File for Controller Parameters" (p. 12) for more information.

!

CAUTION

If the piezo stage starts oscillating (humming noise):

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

In open-loop operation, stop the motion immediately. Do not operate the piezo stage at its resonant frequency because the notch filter(s) may be deactivated in open-loop operation. You can measure the resonant frequency using PIMikroMove™; see "Data Recorder" (p. 71) for more information.

Otherwise the piezo stage could be irreparable damaged.

Re-calibration should only be done by adequate trained personnel and after consultation with PI. Otherwise preset data will be lost.

1.3 Model Survey

The following E-517 models are available:

E-517.i3	Interface / Display Module, 24 Bit D/A, TCP/IP, USB, RS-232, 3 Channels
E-517.i1	Interface / Display Module, 24 Bit D/A, TCP/IP, USB, RS -232, Single Channel

The E-517 comes normally installed in the chassis of the analog piezo control electronics, in the configuration ordered by the customer. The system is calibrated at the factory and ready for use.

E-516 COMPATIBILITY NOTE

Users who have written software for the E-516 interface / display module which is the predecessor of the E-517 can switch to the GCS syntax used with the E-516, see "Select Command Set Version" (p. 44). PI software and documentation for the E-516 GCS syntax version is available for download at www.pi.ws or can be obtained from your Physik Instrumente Sales Engineer.

1.4 Unpacking

The E-517 Interface / Display Module usually comes installed in an analog piezo control system. Unpack the system with care. Compare the contents against the items covered by the contract and against the packing list.

The E-517 includes the following components:

- E-517 Interface / Display Module as covered by the contract
- C-815.34 RS-232 cable for connecting E-517 and host PC (null-modem cable)
- C-815.563 special cross-over network cable, can be used to connect the E-517 to a network access point or to a PC with Ethernet socket
- 000011448 USB cable (USB-A/USB-B)
- MDR connector for Digital In/Out socket, consisting of:
  4347 MDR connector, 14-pol, RM 1.27
  4348 MDR cover 14-pol

- E-517.CD Distribution CD, containing E-517 host software (see "Software Description" (p. 10)) and manuals as PDF files

- E-517 Quick Guide in printed form (PZ214Equ)
  Note that the E-517 User Manual (E-517_User_PZ214Exxx.pdf) containing the complete documentation is provided as PDF file on the E-517 CD.

Inspect the contents for signs of damage. If parts are missing or you notice signs of damage, contact PI immediately.
Save all packing materials in case the product need be shipped again.

1.5 Motion System Requirements

To start working with the E-517 interface / display module, your motion system must also include the following components:

- The piezo control electronics in which the E-517 is integrated (e.g. E-500 / E-501 system, E-471 or E-481)

- The mechanics (piezo stage) with which the piezo control electronics was calibrated (compare serial numbers)

- For remote operation via the computer interface:

  A PC with Windows operating system (2000, XP, Vista) or Linux operating system (kernel 2.6, GTK 2.0, glibc 2.4).

  Communications interface to the PC and the appropriate cable. Possible interfaces on the PC are COM port, USB port, GPIB interface (IEEE 488), or an Ethernet card. The connection between E-517 and PC can furthermore be made via a free access point on a network to which the PC is connected.

  To command the E-517, at least a terminal emulator must be installed on the PC. For convenient operation without the need to learn any commands, it is recommended to install PIMikroMove™ from the E-517 CD. This graphical user interface is included in the "typical" installation which requires about 50 MB free disk space. Note that not all software components are available for Linux PCs. See "Software Description" (p. 10) for more information.

1.6 Software Description

Operating system details:

- "Windows" stands for 2000, XP and Vista
- "Linux" stands for kernel 2.6, GTK 2.0, glibc 2.4

The table below lists the software tools which are on the E-517 product CD with application recommendations.

For more information see the corresponding software manuals.

Software Tool	Supported Operating System	Short Description	Recommended for
GCS Library	Windows, Linux	Allows program access to the E-517 from languages like C++. The functions in the library are based on the PI General Command Set (GCS). Windows operating systems: PI_GCS2_DLL; Linux operating systems: libpi_pi_gcs2.so.x.x.x and libpi_pi_gcs2-x.x.x.a where x.x.x gives the version of the library	Recommended for customers who want to use a library for their applications. The dynamic version of the library is needed by the LabVIEW driver set and by PIMikroMove™.
LabVIEW drivers	Windows, Linux	LabVIEW is a software tool (available separately from National Instruments) for data acquisition and process control. The E-517 LabVIEW software consists of a collection of virtual instrument (VI) drivers for the E-517 controller. This driver set supports the PI General Command Set (GCS). Included are Vis for GCS commands and high-level Vis for various tasks.	Users who want to use LabVIEW for programming their applications based on the GCS. See the GCS LabVIEW manual of your controller for more information.

Software Tool	Supported Operating System	Short Description	Recommended for
PIMikroMove™	Windows	PIMikroMove™ permits you to start your motion system—host PC, controller and stage(s)—immediately without the need to write customized software. It offers motion-control displays and features that in many cases make it unnecessary to deal with ASCII-format commands. It also has a complete command input facility, which represents an easy way to experiment with various commands. PIMikroMove™ uses the GCS DLL described above to command the controller. Note that the program offers comprehensive online support.	Users who want to test the equipment before or instead of programming an application and who want to learn how to use the commands. For motor controllers, PIMikroMove™ offers an easy way to optimize the servo parameters.
PITerminal	Windows	PITerminal is a Windows GUI which can be used as a simple terminal with almost all PI controllers.	Users who want to send the commands of the PI General Command Set (GCS) directly.
Firmware Update Wizard	Windows	The Firmware Update Wizard guides you through the update of the firmware for the hardware modules of your E-517 system.	Users who want to update the firmware.

NOTE

Except for the PITerminal, the software on the E-517 CD can not be used when the E-517 is switched to the E-516 GCS syntax version (predecessor of the E-517). Do not switch to the E-516 GCS syntax version except when necessary for compatibility reasons! See "Select Command Set Version" (p. 44) for details.

2 First Steps

2.1 Installing the Software on the Host PC

Windows operating systems:

1. Insert the E-517 CD in your host PC.
2. If the Setup Wizard does not open automatically, start it from the root directory of the CD with the icon.
3. Follow the on-screen instructions and select the “typical” installation. Typical components are LabView drivers, GCS DLL, PIMikroMove™.

Linux operating systems:

1. Insert the E-517 CD in the host PC.
2. Open a terminal and go to the /linux directory on the E-517 CD.
3. Log in as superuser (root).
4. Start the install script with ./INSTALL
   Keep in mind the case sensitivity of Linux when typing the command.
5. Follow the on-screen instructions. You can choose the individual components to install.

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

For an overview over the host software provided see "Software Description" (p. 10).

2.2 Creating Backup File for Controller Parameters

It is strongly recommended to save the parameter values of the E-517 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-517, use the *Device Parameter Configuration* window of PIMikroMove™.

See "Installing the Software on the Host PC" (p. 12) for how to install the program.

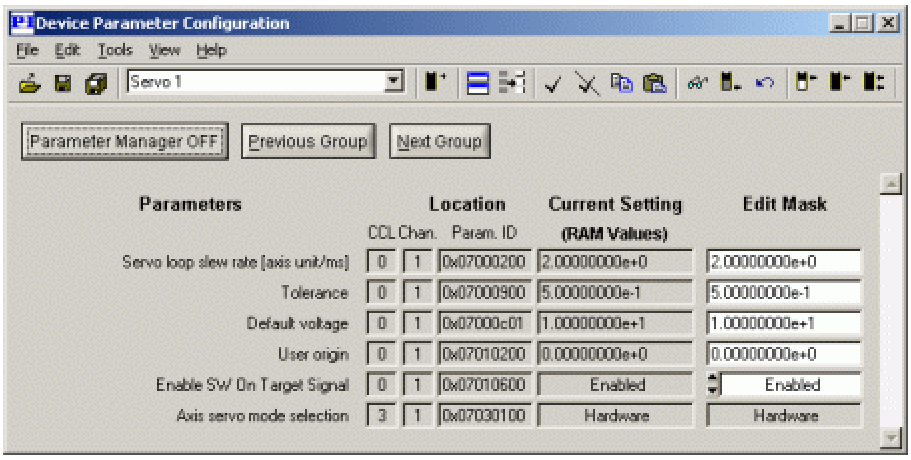
Proceed as follows to create a parameter file:

1 Power on the system. Find details in a dedicated section (p. 30).

2 Start PIMikroMove™ on the host PC and establish a connection to the E-517 as described in "Quick Start" (p. 14).

3 In the PIMikroMove™ main window, open the *Device Parameter Configuration* window using the *E-517* ⇒ *Parameter Configuration* menu sequence.

4 In the *Device Parameter Configuration* window, save the controller parameters to a file. Use the *Save* or *Save As* buttons in the top left-hand corner of the window, or use the *File* ⇒ *Save Edit Values* or *File* ⇒ *Save Edit Values As* menu sequences. In fact, the values from the *Edit Mask* column are saved.



Parameters	Location CCL	 Chan.	 Param. ID	Current Setting (RAM Values)	Edit Mask
Servo loop slew rate [axis unit/ms]	0	1	0x07000200	2.00000000e+0	2.00000000e+0
Tolerance	0	1	0x07000900	5.00000000e-1	5.00000000e-1
Default voltage	0	1	0x07000c01	1.00000000e+1	1.00000000e+1
User origin	0	1	0x07010200	0.00000000e+0	0.00000000e+0
Enable SW On Target Signal	0	1	0x07010600	Enabled	Enabled
Axis servo mode selection	3	1	0x07030100	Hardware	Hardware

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

1 Use the *File* ⇒ *Load and select* menu sequence. The loaded parameter values are written in the corresponding *Edit Mask* fields.

2 You can copy the loaded parameter values from the *Edit Mask* fields to the RAM (volatile memory) or non-volatile memory of your controller using the three *Write selected edit values...* buttons in the top right-hand corner of the window.

Note that depending on the controller, changing parameters may require a certain command control level (CCL). You can switch to command level 1 and change level-1 parameters (see *CCL* column), while command levels >1 are reserved for service personnel. In the dialog which pops up automatically when you try to copy the values of protected parameters from the *Edit Mask* column to the controller, enter the password "advanced" to switch to command level 1.

See the PIMikroMove™ Manual for further information.

2.3 Quick Start

The following instructions illustrate the first steps with the E-517 using PIMikroMove™ because that software offers a convenient user interface for operating the system. See the PIMikroMove™ Manual for the complete, more-detailed software description and "Installing the Software on the Host PC" (p. 12) for how to install the program. In the example below, an E-517.i3 3-channel model and an X-Y-Z stage are used.

CAUTION

Thermally stable systems have the best performance. For a thermally stable system, power on the E-517 at least one hour before you start working with it.

!

CAUTION

If the piezo stage starts oscillating (humming noise):

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

In open-loop operation, stop the motion immediately. Do not operate the piezo stage at its resonant frequency because the notch filter(s) may be deactivated in open-loop operation. You can measure the resonant frequency using PIMikroMove™; see "Data Recorder" (p. 71) for more information.

Otherwise the piezo stage could be irreparable damaged.

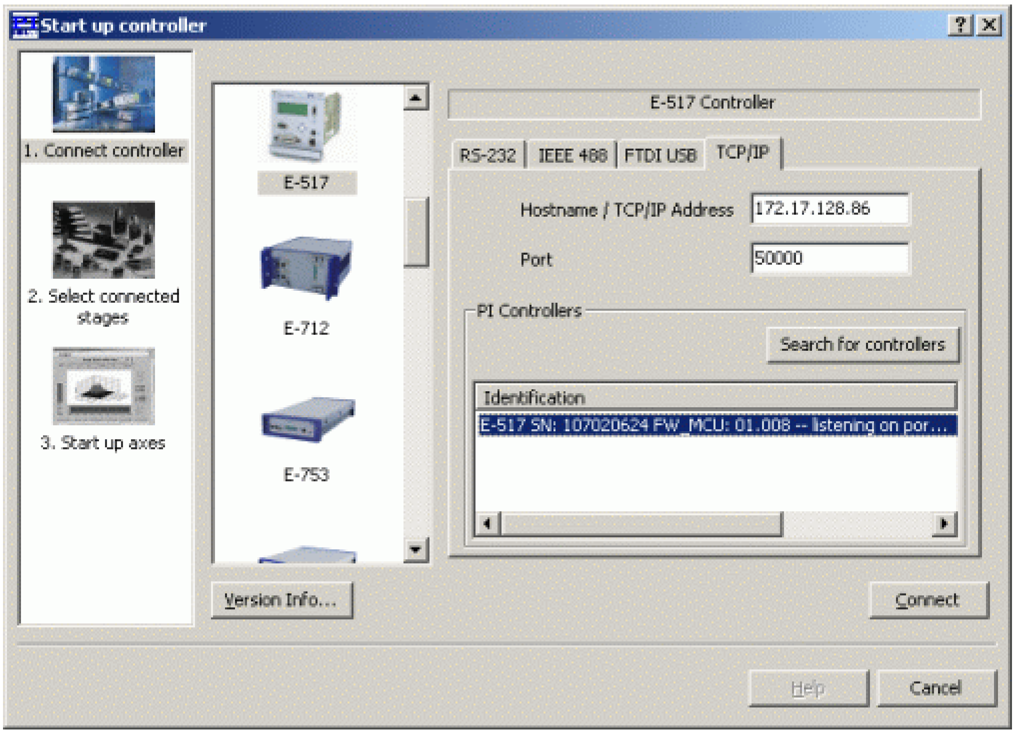
Re-calibration should only be done by adequate trained personnel and after consultation with PI. Otherwise preset data will be lost.

!

1 Interconnect your system. Find details in the User manual of the piezo control electronics in which the E-517 is integrated, and in "Communication" (p. 58) and "External Triggering/Signaling" (p. 77).

2 Power on the system. Find details in a dedicated section (p. 30).

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



Start up controller

E-517 Controller

Hostname / TCP/IP Address: 172.17.128.86

Port: 50000

Identification: E-517 SN: 107020624 FW_MCU: 01.008 -- listening on por...

4 Establish a connection to the E-517 from PIMikroMove™. This can be done via the TCP/IP, FTDI USB, RS-232 or IEEE 488 (GPIB) interface, using the corresponding tab card in the *Start up controller* window (tab card selection depends on the current hardware connection). See "Communication" (p. 58) for details.

Notes:

When using the USB interface for the first time, two FTDI USB drivers must be installed on the host PC. These drivers are provided on the E-517 CD in the \USB_Serial_Driver directory.

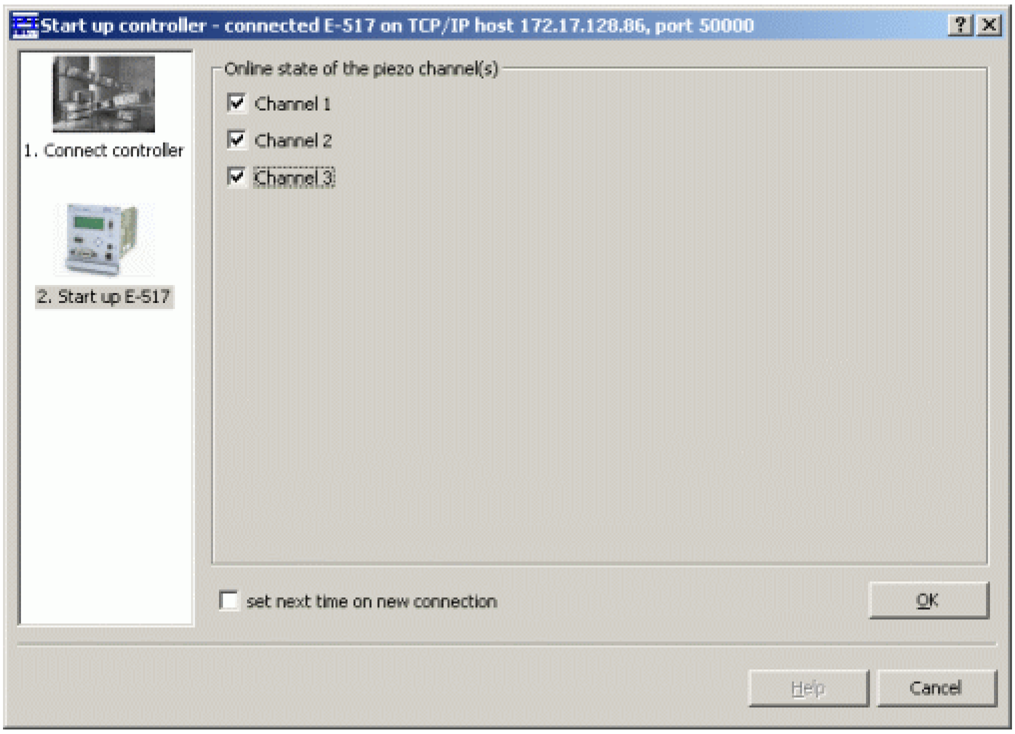
TCP/IP communication: If no DHCP server is available on the network or if a point-to-point connection between host PC and controller is being used, after power-on or reboot it might take a period of about 30 seconds before communication is possible. Use the *Search for controllers* button, select your controller in the resulting list and press *Connect*. If the controller is already connected to your or another host PC via TCP/IP, a second TCP/IP session cannot be established.

The *IEEE 488* tab card is only present in the host software if a

GPIB driver from National Instruments (NI) or an NI-compatible driver is installed on your PC.

5 Give the E-517 interface / display module control over the output voltage channels of the piezo control electronics by switching them to ONLINE mode. This can be done in the *Start up controller* window which should open automatically with the *Start up E-517* step, see figure below. A channel is in ONLINE mode when its check box is checked, otherwise it is in OFFLINE mode. When the mode selection is finished, press the *OK* button. The PIMikroMove™ main window will open.

You can reopen the window for ONLINE / OFFLINE selection at any time via the *E-517* ⇒ *Start up axes ...* menu sequence from the PIMikroMove™ main window.



Start up controller - connected E-517 on TCP/IP host 172.17.128.86, port 50000

Online state of the piezo channel(s)

☑ Channel 1

☑ Channel 2

☑ Channel 3

□ set next time on new connection

Notes:

In ONLINE mode the SERVO switches of all channels must be set to OFF on the piezo control electronics. This gives the E-517 complete control over the servo mode settings.

Channels which are in OFFLINE mode can not be commanded by the E-517 (and hence not by the host PC) but only via controls on the piezo control electronics (e.g. analog control input and / or DC-offset potentiometers).

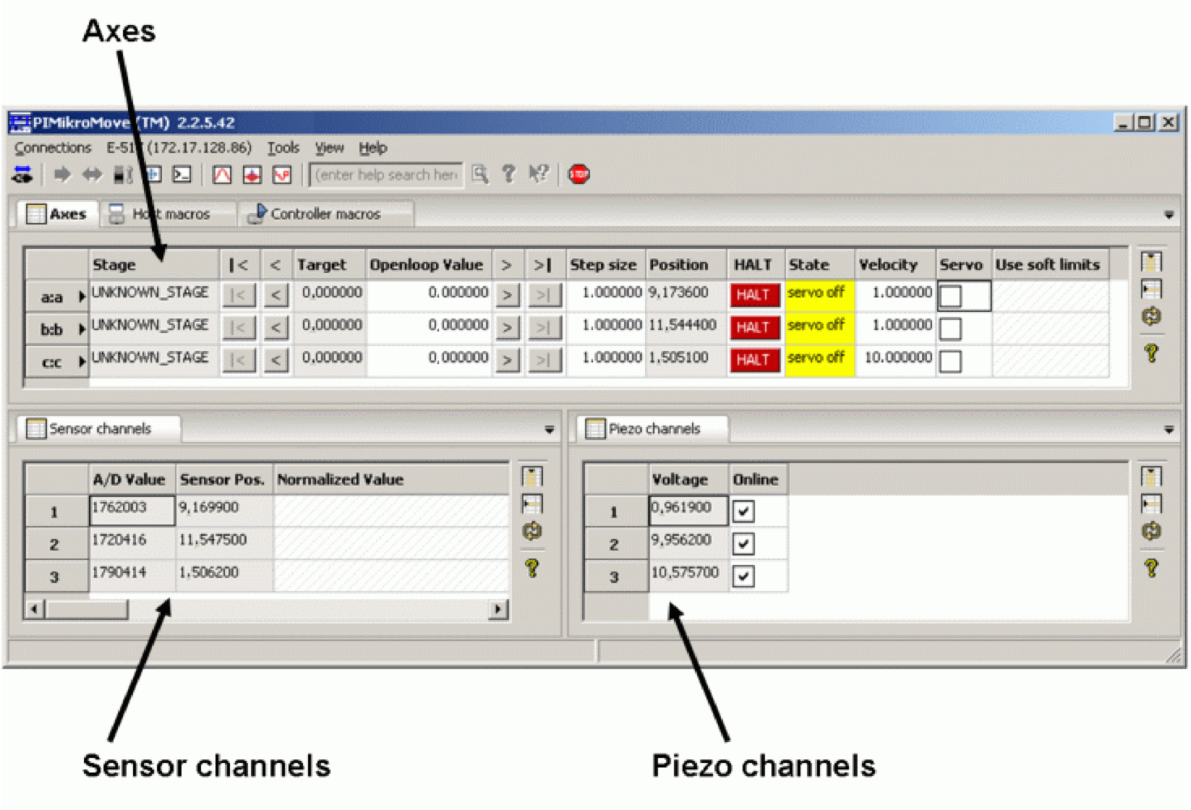
The current mode of the channels is also visible in the main screen of the E-517 display (top left corner; ‘I’ for ONLINE, ‘O’ for OFFLINE).

Find details in "Modes of Operation" (p. 23).

6 Optionally: Configure the PIMikroMove™ main window:

It is recommended to see the tab cards for axes, sensor and piezo 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, you can start axis motion (see step 7 below). The channel tab cards show the current sensor and output voltage values which are also visible on the main screen of the E-517 display.



	Stage	|<	<	Target	Openloop Value	>	>|	Step size	Position	HALT	State	Velocity	Servo	Use soft limits
a:a	UNKNOWN_STAGE	|<	<	0,000000	0.000000	>	>|	1.000000	9,173600	HALT	servo off	1.000000	□	
b:b	UNKNOWN_STAGE	|<	<	0,000000	0,000000	>	>|	1.000000	11,544400	HALT	servo off	1.000000	□	
c:c	UNKNOWN_STAGE	|<	<	0,000000	0,000000	>	>|	1.000000	1,505100	HALT	servo off	10.000000	□	

	A/D Value	Sensor Pos.	Normalized Value
1	1762003	9,169900	
2	1720416	11,547500	
3	1790414	1,506200	

	Voltage	Online
1	0,961900	☑
2	9,956200	☑
3	10,575700	☑

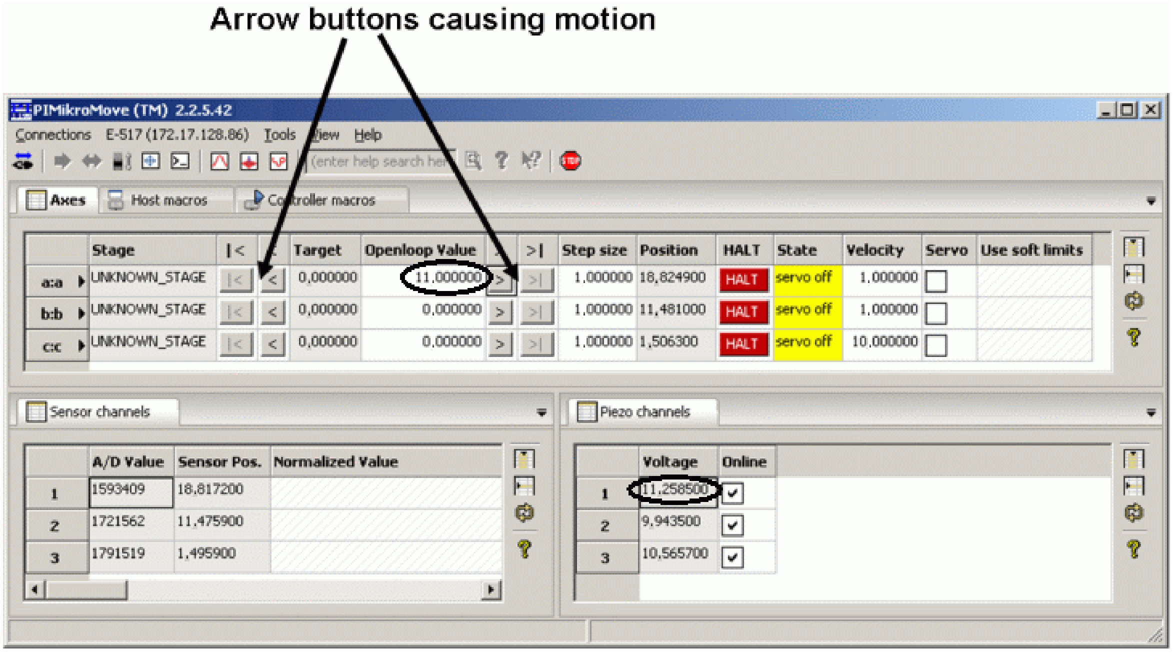
Note:

With the E-517, the axis-to-channel assignment is fixed so that the first motion axis ("A" by default) always is driven by the first piezo channel and measured by the first sensor channel, the second axis ("B" by default) belongs to the second piezo channel and to the second sensor channel, and the third axis ("C" by default) belongs to the third piezo channel and to the third sensor channel. Each line on the tab cards belongs to one axis or one channel of the system respectively.

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

The first moves should be made in open-loop operation (*Servo* boxes must be unchecked). With the E-517, open-loop commanding means to give the desired output voltage values for the axes. Command an output voltage of 0 V by entering 0 in the *Openloop Value* field of an axis and pressing Enter on your keyboard. Then enter a new value of about 10% of the axis voltage range and press Enter. Increment the *Openloop Value* this way by steps of a suitable size, up to the upper voltage limit of the axis (see datasheet or measurement protocol of the mechanics), and then reduce it in an analogous manner to the lower voltage range limit of the axis. In doing so, observe the position display for the axis (in the *Position* field) and the current output voltage for the corresponding piezo actuator in the stage (in the *Voltage* field of the *Piezo channels* tab card). The output voltage values should follow the commanded values. At 0 V output voltage, the current position value of the axis should be approximately 0 µm, but due to the calibration settings of the system, the axis position can differ from 0 by about 20% of the axis travel range.

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



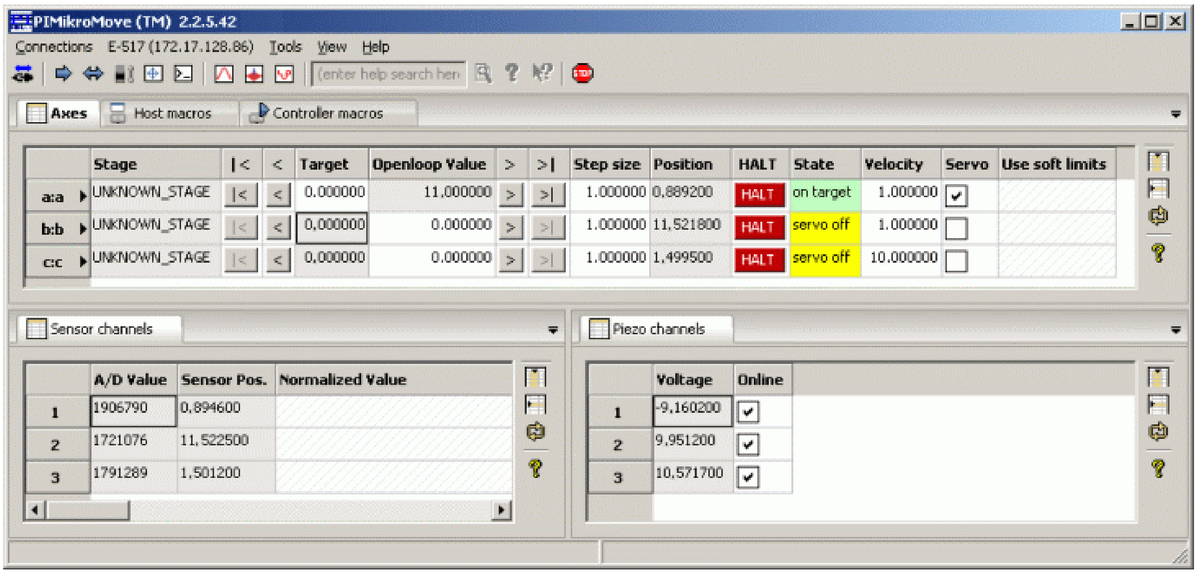
Arrow buttons causing motion

	Stage	|<	<	Target	Openloop Value	>	>|	Step size	Position	HALT	State	Velocity	Servo	Use soft limits
a:a	UNKNOWN_STAGE	|<	<	0,000000	11,000000	>	>|	1,000000	18,824900	HALT	servo off	1,000000	□	
b:b	UNKNOWN_STAGE	|<	<	0,000000	0,000000	>	>|	1,000000	11,481000	HALT	servo off	1,000000	□	
c:c	UNKNOWN_STAGE	|<	<	0,000000	0,000000	>	>|	1,000000	1,506300	HALT	servo off	10,000000	□	

	A/D Value	Sensor Pos.	Normalized Value
1	1593409	18,817200	
2	1721562	11,475900	
3	1791519	1,495900	

	Voltage	Online
1	11,258500	☑
2	9,943500	☑
3	10,565700	☑

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



	Stage	|<	<	Target	Openloop Value	>	>|	Step size	Position	HALT	State	Velocity	Servo	Use soft limits
a:a	UNKNOWN_STAGE	|<	<	0.000000	11.000000	>	>|	1.000000	0,889200	HALT	on target	1.000000	☑	
b:b	UNKNOWN_STAGE	|<	<	0,000000	0.000000	>	>|	1.000000	11,521800	HALT	servo off	1.000000	☐	
c:c	UNKNOWN_STAGE	|<	<	0,000000	0.000000	>	>|	1.000000	1,499500	HALT	servo off	10.000000	☐	

	A/D Value	Sensor Pos.	Normalized Value
1	1906790	0,894600	
2	1721076	11,522500	
3	1791289	1,501200	

	Voltage	Online
1	-9,160200	☑
2	9,951200	☑
3	10,571700	☑

3 Operation

3.1 Front Panel Elements

3.1.1 Front Panel Overview

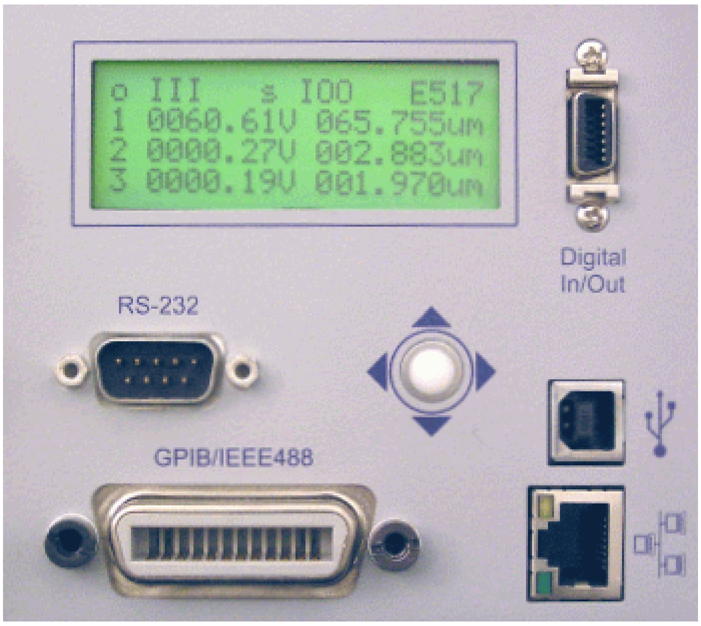
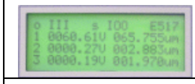
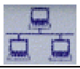


Figure 1: E-517.i3 interface / display module, 3 channels. Single-channel E-517.i1 models have identical front panels but show only one channel in the display.

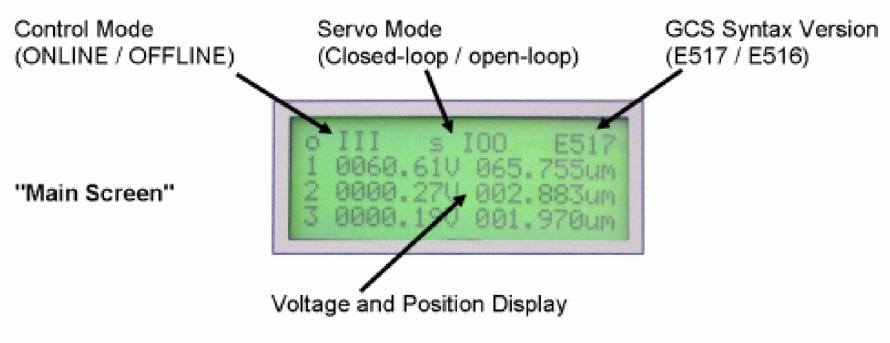
Name	Function
	LCD display. See "Display Screens" (p. 21) for details.
Digital In/Out	MDR14 socket for digital input and output lines; can be used to trigger external devices and to send start and synch. signals to the wave generator(s). See "Digital In/Out Socket" (p. 267) for the availability of the lines and for pinout.
RS-232	Serial connection to host PC. See "RS-232 Socket" (p. 267) for pinout.
	Trackball for display selection, device configuration and ONLINE motion. See "Trackball Functions" (p. 22) for details.
	Universal Serial Bus (USB-B socket) for connection to host PC. See "USB Connection" (p. 68) for more information.
GPIB/IEEE488	General Purpose Interface Bus (GPIB) for connection to host PC. See "GPIB Connection" (p. 69) for more information.
	Network connection over TCP/IP. See "TCP/IP Connection" for more information.

3.1.2 Display Screens

The individual display screens can be accessed by the trackball (p. 22) on the E-517 front panel. You can check the different screens by powering up the piezo control electronics with no actuator/stage and no host PC connected.

The following screens are available via the trackball:

- "Main screen", the first line shows the current control mode ("o") and servo mode ("s") for all channels (0 = OFFLINE / open-loop operation, I = ONLINE / closed-loop operation; see "Modes of Operation" (p. 23) for details) and the active syntax version of the PI General Command Set (E517 or E516, see "Select Command Set Version" (p. 44) for details). Lines 2 to 4 show the current voltage and position values (in V and µm) for all connected channels. The current voltage value can also be read with the VOL? command (p. 209) and the current position value with the TSP? command (p. 199).
  The main screen is active upon power-on and reboot.



```
o III    s I00   E517
1 0060.61V 065.755um
2 0000.27V 002.883um
3 0000.19V 001.970um
```

- DISPLAY SERVICE, where you can adjust the brightness and the contrast of the display. Ranges are 10-31 for brightness and 5-31 for contrast. The settings can be permanently saved as power-on defaults with WPA 100 (see WPA command (p. 229)) and are also available via controller parameters (brightness: ID 0x0E000D00; contrast: ID 0x0E000D01; see "Controller Parameters" (p. 250) for more information)

- COMMUNICATION, where you can select the RS-232 baud rate and the IEEE 488 address. The settings can be permanently saved as power-on defaults with WPA 100 (see WPA command (p. 229)). They are also available with IFC (p. 159), IFC? (p. 161) (temporary settings) and IFS (p. 163), IFS? (p. 164) (default settings) and via controller parameters (RS-232 baud rate: ID 0x11000400; IEEE 488 address: ID 0x11000900; see "Default and Current Settings" (p. 59) and "Controller Parameters" (p. 250) for more information)

- CHANNEL SETTING, where you can command motion (in ONLINE mode) or the check current position and voltage (in OFFLINE mode) for individual channels. See "Trackball Functions" (p. 22) for details

You can configure the position and voltage display via parameters (e.g. show / hide channels, set the number of digits). See "Activate/Deactivate Axes and Channels" (p. 33) for more information.

3.1.3 Trackball Functions

You can operate the trackball on the E-517 front panel by pressing it or rolling it left/right or up/down with your fingers. It gives access to the following functionality:

- Display screen selection:
  You can select the active display screen (p. 21) by rolling the trackball to the left or to the right. After power-on or reboot, always the "main screen" is active (all channels displayed)

- ONLINE/OFFLINE setting:
  When the "main screen" is active, you can switch between ONLINE and OFFLINE mode for all channels by pressing the trackball (see "Modes of Operation" (p. 23) for details)

- Skipping start-up macro:
  To skip the execution of the start-up macro, press the trackball once immediately after you have powered on or rebooted the piezo control electronics (while "Physik Instrumente" is still shown on the display). For details about start-up macro definition, see "Start-Up Macro" (p. 115)

- E-517 configuration:
  On the DISPLAY SERVICE and COMMUNICATION screens you can use the trackball to access and modify several settings. See "Display Screens" (p. 21) for details

- Commanding motion in ONLINE mode:
  The CHANNEL SETTING screen gives access to channel-specific screens. There you can enter target values (position or voltage, depending on the current servo mode) to move the corresponding axis. Rolling the trackball left/right selects the digit to be changed, rolling the trackball up/down selects the value for that digit. You can switch between the individual channels by pressing the trackball. While trackball settings and move commands can be used in parallel to command axis motion, they are not accepted when the wave generator is running for the axis. See "Control Value Generation" (p. 53) for more information.

In OFFLINE mode, you can check the current voltage and position values on the channel-specific screens.

3.2 Modes of Operation

Control mode: The piezo channels (amplifiers) of the control electronics can be operated in either ONLINE or OFFLINE control mode. The active mode determines the applicable control sources for the output voltage. See "Control Modes" (p. 23) and see "Display Screens" (p. 21) for more information.

Servo mode: The current servo mode determines if a motion axis is driven in open-loop (servo OFF) or closed-loop (servo ON) operation. In closed-loop operation a servo loop participates in the generation of the control value for the piezo channel. The servo loop thus maintains the current axis position, based on a given target position and the position feedback of the corresponding sensor channel. See "Servo Modes" (p. 24) and "Control Value Generation" (p. 53) for more information.

The individual control and servo modes can be combined arbitrarily. The current states are visible on the main screen of the E-517 display.

3.2.1 Control Modes

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

- OFFLINE mode:
  The output voltage depends on analog control input and DC offset applied to the channel (e.g. a voltage applied to the CONTROL INPUT socket; setting of the DC-OFFSET knob).
  Move commands (received via interface or from a running macro), wave generator output and trackball target settings are ignored and may provoke an error message.

- ONLINE mode:
  The E-517 controls the generation of the output voltage. Target values for the axis motion can be given by move commands (received via interface or from a running macro), wave generator output and trackball settings. Analog control input voltage and DC offset settings are ignored.

See "Control Value Generation" (p. 53) for details.

Switching between ONLINE and OFFLINE mode can be done by:

- Channel-specific ONL commands (p. 174) sent over the communications interface or received from a macro running on the E-517
- Pressing the trackball on the E-517 front panel when the main screen is displayed; switches the mode for all channels at once

The current control mode is visible for the individual channels on the main screen of the E-517 display, see "Display Screens" (p. 21). Using the ONL? command (p. 175), you can check the current control mode on a per-channel basis.

NOTES

Upon power-on or reboot, all piezo channels of the system are by default in OFFLINE mode. You can switch them to ONLINE mode using a start-up macro, see "Start-Up Macro" (p. 115) for details.

In OFFLINE mode, the E-517 accepts all commands just as in ONLINE mode. The only difference between the modes is the control source selection of the channel.

In ONLINE mode the SERVO switches of all channels must be set to OFF on the piezo control electronics.

3.2.2 Servo Mode (ON / OFF)

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

- Closed-loop operation:
  Any control input (analog, DC offset, E-517 input like move commands, trackball settings and wave generator output) is interpreted as target position. Based on this target position and on the position feedback of the corresponding sensor channel, a servo loop generates the control value for the piezo channel. The servo loop thus maintains the axis position. It is not located in the E-517 but in the analog part of the piezo control electronics (see the corresponding User Manuals of your system for more information). The servo loop applies a proportional-integral (P-I) servo-controller, a notch filter and a slew rate limiter. The slew rate can also be influenced by the E-517 (parameter ID 0x07000200).

- Open-loop operation:
  Any control input is interpreted as piezo voltage target. Open-loop operation omits the servo loop, and the control input directly controls the output voltage of the piezo channel. The slew rate can be influenced by the E-517 (parameter ID 0x07000200).

The servo mode can be set by:

- The SERVO ON/OFF toggle switches on the piezo control electronics
- Axis-specific SVO commands (p. 193) sent over the communications interface or received from a macro running on the E-517

The current servo mode is visible for the individual axes on the main screen of the E-517 display, see "Display Screens" (p. 21). Using the SVO? command (p. 194), you can check the current servo mode on a per-axis basis.

NOTE

In ONLINE mode the SERVO switches of all channels must be set to OFF on the piezo control electronics. This gives the E-517 complete control over the servo mode settings.

Using a start-up macro, you can set up the device to start with closed-loop operation.

The example below illustrates the interpretation of the control input as piezo voltage target or target position, depending on the current servo mode. The E-517 is in OFFLINE mode in this example, and the control input is given by the DC offset potentiometer. The DC offset potentiometer setting remains unchanged while switching from open-loop to closed-loop operation, and the voltages and positions are read out in the display.

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

3.3 How to Command Axis Motion

3.3.1 Applicable Control Sources

The applicable control sources for the axis motion depend on the current control mode of a piezo channel (OFFLINE or ONLINE mode, see "Control Modes" (p. 23) for details).

In OFFLINE mode, the axis motion is commanded by analog control input and DC offset applied to the channel. The DC offset plus the analog control voltage must be in the -2 V to +12 V range. The voltage gain is 10 or 100 (depending on the amplifier type). For detailed information regarding OFFLINE control see the User Manual of the piezo control electronics (OFFLINE control may be referred to as "analog operation" there).

In ONLINE mode, the axis motion can be commanded by multiple sources (see "Control Value Generation" (p. 53) for details):

- Move commands (received via interface or from a running macro): SVA (p. 191) and SVR (p. 195) in open-loop operation; MOV (p. 170), MVR (p. 172) and GOH (p. 154) in closed-loop operation; IMP (p. 165) and STE (p. 189) for both servo modes (see below for examples and "Servo Modes" (p. 24) for more information)
- Trackball settings on the CHANNEL SETTING screen (see "Trackball Functions" (p. 22) for more information)
- Wave generator output for periodic motion (see "Wave Generator" (p. 84) for more information and examples)

While move commands and trackball settings can be used in parallel to command axis motion, they are not accepted when the wave generator is running for the axis.

The macro feature allows defining command sequences and storing them permanently in non-volatile memory in the controller. Macros can run in OFFLINE and ONLINE mode, but move commands from macros are only accepted in ONLINE mode and only when the wave generator is not running. See "Working with Controller Macros" (p. 113) for more information.

The E-517 has a real-time data recorder. It is able to record several input and output signals from different data sources during the axis motion. Data recording is triggered automatically with four commands:

- STE (step response measurement),
- IMP (impulse response measurement),

- WGO (wave generator start) and
- 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. See "Data Recording" (p. 71) for more information.

It is possible to trigger external devices during the axis motion and to send start/stop signals to the wave generators with the digital I/O lines of the E-517. See "External Triggering / Signaling" (p. 77) for more information and examples and "Digital In/Out Socket" (p. 267) for the lines and pinout.

3.3.2 Command Examples for ONLINE Mode

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

Example 1:

This example moves axis A of the mechanics in open-loop operation. The output voltage is varied to see the influence of the voltage limits for the corresponding piezo channel: hardware limits are given by parameters 0x0B000007 and 0x0B000008 and can not be changed, "soft limits" can be set with VMA and VMI. In this example, the output voltage will be limited to the range of +10 to +90 volts.

Hardware: A piezo stage and an E-500 chassis with one E-505 amplifier module and E-517 (single-channel system).

Command String to Send	Response	Action Performed
`ONL 1 1`		Switch piezo channel 1 (identifiers 1) to ONLINE mode so that axis motion can be caused by move commands (received via interface or from a running macro), wave generator output and trackball settings
`SVO? A`	`A=0`	Check current servo mode for axis A. The axis is in open-loop operation (response = 0), i.e. there is no correction of drift or other effects.
`SVA A 80`		Move axis A by setting the corresponding output voltage to 80 volts.
`VOL? 1`	`1=+0079.9742`	Query the current output voltage of piezo channel 1 which drives axis A.
`SVA A 150`		Attempt to set the output voltage for axis A to 150 volts.
`VOL? 1`	`1=+0079.9725`	The last commanded voltage value lay beyond the allowed range and the SVA command therefore was ignored.

`SVA? A`	`A=+0080.0000`	Query the commanded output voltage. It resulted from the last valid SVA command.
`VMA? A`	`A=+0120.0000`	Get voltage output high limit for the axis.
`VMA A 90`		Set voltage output high limit for the axis to 90 volts.
`VMI A 10`		Set voltage output low limit for the axis to 10 volts.
`SVA A 85`		Set the output voltage for axis A to 85 volts.
`VOL? 1`	`1=+0085.4611`	Query the current output voltage of piezo channel 1.
`SVA A 100`		Set the output voltage for axis A to 100 volts.
`VOL? 1`	`1=+0085.4606`	Query the current output voltage of piezo channel 1.
`SVA? A`	`A=+0085.0000`	Query the commanded output voltage. It resulted from the last valid SVA command.
`ERR?`	`302 - (error 302 - "Voltage out of limits")`	Get error code of the last occurred error. The error code is cleared by the ERR? command
`ERR?`	`0 - (error 0 - "No error")`	No error occurred since the last ERR? query
`VMA A 100`		Set voltage output high limit for the axis to 100 volts.
`SVA A 100`		Set the output voltage for axis A to 100 volts.
`SVA? A`	`A=+0100.0000`	Query the commanded output voltage. It resulted from the last valid SVA command.
`VOL? 1`	`1=+0099.3135`	Query the current output voltage of piezo channel 1.
`POS? A`	`A=+0106.1647`	Query the current position axis A. The nominal travel range @ 0 to 100 V is 100 µm, with a tolerance of ±20%.
`SVR A -20`		Move axis A relative by subtracting 20 volts from the current output voltage.
`VOL? 1`	`1=+0080.4106`	Query the current output voltage of piezo channel 1.
`POS? A`	`A=+0093.9297`	Query the current position axis A.

Example 2:

This example moves three axes in closed-loop operation. Afterwards the current position is queried. It is discovered that axis B is blocked by an external hard stop. It will then be moved back to its zero position.
Hardware: E-500 system with 3 channels, each equipped with a sensor servo-module, piezo stage with 3 axes

Command String to Send	Response	Action Performed
`ONL 1 1 2 1 3 1`		Switch all piezo channels (identifiers 1, 2, 3) to ONLINE mode so that axis motion can be caused by move commands (received via interface or from a running macro), wave generator output and trackball settings
`SVO A 1 B 1 C 1`		Set servo-control on (closed-loop operation) for all axes; this also writes the current axis position to the target register, to avoid jumps of the mechanics.
`DCO A 1 B 1 C 1`		Activate drift compensation for the axes to avoid an unwanted change in displacement over time. See "Drift Compensation" (p. 56) for details.
`MOV A 30.5`		Move axis A to the given absolute position of 30.5 µm
`POS? A`	`A=+0030.4804`	Query the current position axis A.
`MOV B 80`		Move axis B to the given absolute position of 80 µm
`POS? B`	`B=+0056.8775`	Query the current position axis B.
`POS? B`	`B=+0056.8768`	Query the current position axis B. The commanded position has not been reached.
`MOV B 0 C 30.5`		Move axis B to the given absolute position of 0 µm and axis C to 30.5 µm.

`POS?`	`A=+0030.4806`
	`B=+0000.8773`
	`C=+0030.5080`
		Query the current positions of all axes.

`MVR A -2 B 3 C 5`		Move axes A, B and C relative by the given position values.

`POS?`	`A=+0028.4797`
	`B=+0003.8769`
	`C=+0035.5079`
		Query the current positions of all axes.

3.4 Power On / Reboot Sequence

CAUTION

!

Thermally stable systems have the best performance. For a thermally stable system, power on the E-517 at least one hour before you start working with it.

Power on the piezo control electronics as described in the corresponding User Manual(s).

On power-on or reboot (with the RBT command (p. 178)), the E-517 copies information from non-volatile memory to volatile memory. The control mode is reset to OFFLINE and the servo mode to open-loop operation.

You can define a start-up macro which runs on power-on or reboot, see "Working with Controller Macros" (p. 113) for more information. To skip the execution of the start-up macro, press the trackball on the E-517 once immediately after you have powered on or rebooted the piezo control electronics (while "Physik Instrumente" is still shown on the display).

NOTE

TCP/IP communication: If no DHCP server is available on the network or if a point-to-point connection between host PC and controller is being used, after power-on or reboot it might take a period of about 30 seconds before communication is possible.

3.5 How to Customize the System

NOTES

Values stored in non-volatile memory are power-on defaults, so that the system can be used in the desired way immediately. Note that PI records the data files of every E-517 controller calibrated at the factory for easy restoration of original settings should that ever be necessary.

3.5.1 Set System Parameters

!

CAUTION

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

It is strongly recommended to save the parameter values of the E-517 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-517, use the *Device Parameter Configuration* window of PIMikroMove™. See "Create Backup File for Controller Parameters" (p. 12) for more information.

To adapt the E-517 to your application, you can modify parameter values. The parameters available depend on the controller firmware. With HPA? (p. 157) you can obtain a list of all available parameters with information about each (e.g. short descriptions). The volatile and non-volatile memory parameter values can be read with the SPA? (p. 188) or SEP? (p. 184) commands, respectively. Note that many parameters are "protected" by higher command levels, as indicated in the "Command Level" column in the "Parameter Overview" table (p. 252). By going to command level 1 using the CCL command (p. 139), it is possible to change level-1 parameters. Parameters with level 2 or higher are reserved for service personnel.

Using the "general" modification commands SPA, RPA, SEP and WPA, all parameters for which the currently active command level has write permission can be changed in volatile memory (SPA (p. 185), RPA (p. 179)) or in non-volatile memory (SEP (p. 183), WPA (p. 229)). It is recommended that any modifications be first made with SPA, and when the controller runs well, saved using WPA. If you change the current interface settings with SPA, it may be necessary to close the current connection and re-open it with the new settings.

In addition to the "general" modification commands, there are commands which change certain specific parameters. All the commands listed below, except of IFS, change the parameter value only in volatile memory, and WPA must be used to save the value to non-volatile memory. IFS changes and saves the interface parameters directly in non-volatile memory only.

DFH (p. 146) ("User Origin", ID 0x07010200)
IFC (p. 159) (interface parameters: "RS-232 Baud Rate", ID 0x11000400, "GPIB Address", ID 0x11000900, "IP Address", ID 0x11000600, "IP Configuration", ID 0x11000800, "IP Mask", ID 0x11000700)
RTR (p. 180) ("Table Rate" for data recording, ID 0x16000000)
SAI (p. 181) ("Axis Name", i.e. the axis identifier, ID 0x07000600)
VEL (p. 205) ("Servo Loop Slew-Rate", ID 0x07000200)
VMA (p. 207) ("Output Voltage High Limit", ID 0x0C000001)
VMI (p. 208) ("Output Voltage Low Limit", ID 0x0C000000)

WAV (p. 209) (with CFG wave type; "Wave Offset", ID 0x1300010B, and "Wave Generator Table Rate", ID 0x13000109)
WGC (p. 220) ("Wave Generator Cycles", ID 0x13000003)
WOS (p. 228) ("Wave Offset", ID 0x1300010B)
WTR (p. 231) ("Wave Generator Table Rate", ID 0x13000109)

IFS (p. 163) (the same interface parameters as IFC, but in non-volatile memory)

The PIMikroMove™ host software gives access to parameter values in a more convenient way. Use its *Device Parameter Configuration* window to check/edit the individual parameters. See the PIMikroMove™ manual for more information.

See "Controller Parameters" (p. 250) for detailed information.

NOTES

With the E-517, the WPA command saves also the settings made by the following commands, although they are no parameters:

CSV (p. 141) ("Set GCS Syntax Version", selects E-517 or E-516 GCS syntax)
CTO (p. 142) ("Set Configuration Of Trigger Output")
DRC (p. 150) ("Set Data Recorder Configuration")
NLM (p. 173) ("Set Low Position Soft Limit", limits the axis travel range in closed-loop operation)
PLM (p. 177) ("Set High Position Soft Limit", limits the axis travel range in closed-loop operation)
VCO (p. 204) ("Set Velocity Control Mode (On/Off)", activates/deactivates velocity limitation)

The settings of the E-517 are also valid if the device is switched to the E-516 GCS syntax version. Not all parameters of the E-517 are present as E-516 parameters, and the IDs of the available parameters differ from the E-517 parameter IDs. See the E-516 documentation for available parameters.

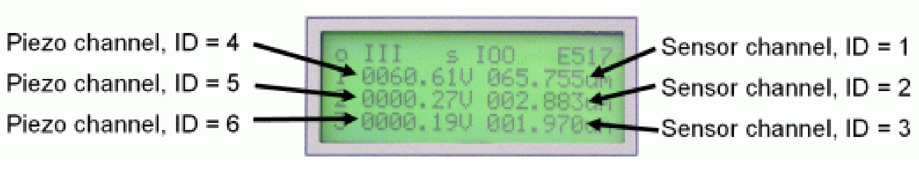
3.5.2 Configure Axes and Channels

The E-517 comes preconfigured, i.e. the number of axes and channels available on the display and in the firmware corresponds to the hardware configuration of your system, as well as the display appearance. You should change the E-517 configuration only in case of hardware changes, e.g. if you install additional sensor and/or amplifier channels in the system. Configuration is done via controller parameters, see "Set System Parameters" (p. 31) for detailed information regarding parameter handling. Switch to command level 1 (CCL command (p. 139)) before you change parameter values with SPA (p. 185) or SEP (p. 183).

NOTES

With the display configuration parameters described here, sensor channels have the identifiers 1 to 3, and piezo channels have the identifiers 4 to 6. Channels 1 and 4 belong to the first motion axis ("A" by default), 2 and 5 to the second axis ("B" by default) and 3 and 6 to the third axis ("C" by default). Keep in mind this assignment when configuring the E-517.

In the display, the numerical values for sensor channels give always positions, and for piezo channels always voltage values. Make sure to set a position unit (e.g. µm) for channels 1 to 3, and set the unit to "V" for channels 4 to 6.



```
o III    s IOO   E517
1 0060.61V 065.755um
2 0000.27V 002.883um
3 0000.19V 001.970um
```

Changing the unit display via the corresponding parameter value has no influence on the numerical value itself. The numerical value can be adjusted by parameters 0x02000300 and 0x02000200, see "Calibration Settings" (p. 38) for details.

With E-517.i1 models, only one axis, one sensor and one piezo channel are physically available, even though you should manage it to enable more in firmware and display.

With E-500 modular systems, the channels in the chassis are counted from right to left (front view). The actual occupancy of the slots has no influence on the channel and axis counting. E.g. the amplifier and sensor / servo modules in the leftmost slots are always identified as the third piezo channel (ID 6) and the third sensor channel (ID 3), and the corresponding axis is always the third axis ("C" by default), whether the other slots are occupied by modules or not.

You can configure the following E-517 settings:

Setting	Effect On	Parameter ID	Possible Values	Notes
Axes/channels accessible in the E-517 firmware	All commands related to axes and channels; direct influence on the response to SAI? (p. 181) (current axis identifiers), TSC? (p. 199) (number of sensor channels), TPC? (p. 198) (number of piezo channels)	Sensor Enable, ID 0x02000000	0 = Disabled 1 = Enabled	This parameter applies to the sensor channels (input signal channels 1 to 3). If it is set to "disabled" for a sensor channel, the corresponding axis and piezo channel are disabled too.
LCD display content	Channel hidden or shown in the display; number of decimal places and unit display for a channel	Display Format, ID 0x04000E01	-1 = channel is not displayed 0 = no decimal places (e.g. 100.) 1 = one decimal place (e.g. 100.9) 2 = 2 decimal places (e.g. 100.99) 3 = 3 decimal places (e.g. 100.999) 4 = 4 decimal places (e.g. 100.9999)	The parameter value -1 hides the channel from the display. The maximum number of decimal places is 4.

		Display Unit, ID 0x04000E00	V = volt
			um = µm
			mm = mm
			ur = µrad
			mr = mrad
			@@ = no unit is displayed
				The parameter value @@ hides the unit from the display.
				The maximum number of characters for the unit is 2.
				The unit is only displayed if the value of parameter 0x04000E01 is different from -1

Example 1: Hardware upgrade

Modular E-500 system with one E-505.00 single-channel amplifier module, one E-509.C1A single-channel sensor / servo module and one E-517.i3 is upgraded: two E-505.00 and two E-509.S1 (single-channel) modules are added to the slots for the second and third amplifier and sensor / servo channels. With the new hardware configuration, three axes, three piezo channels and three sensor channels are available. Two single-axis stages will be connected to the new channels, one with 0 to 50 µm and one with ±25 µm travel range. After the hardware installation, the E-517 must be configured as follows (with all stages connected to the channels!):

Command String to Send	Response	Action Performed
`CCL 1 advanced`		Switch to command level 1 to have write access to the configuration parameters
`SPA 2 0x02000000 1 3 0x02000000 1`		Enable sensor channels 2 and 3 and hence also the second and the third axis ("B" and "C" by default) and piezo channels 2 and 3

`SAI?`	`A`
	`B`
	`C`
		Get the axis identifiers of all axes which are accessible by commands. If enabling was successful, the response shows axes A, B and C.

`TSC?`	`3`	Get the number of sensor channels. If enabling was successful, the response is 3.
`TPC?`	`3`	Get the number of piezo channels. If enabling was successful, the response is 3.
`SPA 2 0x04000e01 3 2 0x04000e00 um`		Show the second sensor channel (ID is 2) in the display, with 3 decimal places and the unit µm
`SPA 5 0x04000e01 2 5 0x04000e00 V`		Show the second piezo channel (ID is 5) in the display, with 2 decimal places and the unit V
`SPA 3 0x04000e01 3 3 0x04000e00 um`		Show the third sensor channel (ID is 3) in the display, with 3 decimal places and the unit µm
`SPA 6 0x04000e01 2 6 0x04000e00 V`		Show the third piezo channel (ID is 6) in the display, with 2 decimal places and the unit V
`SPA 2 0x02000300 5 2 0x02000200 0 2 0x0B000003 10 2 0x0B00000A 0`		Adjust the Ksen, Osen, Kpzt and Opzt values (gain and offset for sensor electronics and piezo amplifier) for the new stage connected to the second sensor and piezo channel. See "Calibration

		Settings" (p. 38) for details.
`SPA 3 0x02000300 5 3 0x02000200 -25 3 0x0B000003 10 3 0x0B00000A 0`		Adjust the Ksen, Osen, Kpzt and Opzt values for the new stage connected to the third sensor and piezo channel.
`ATC 1 1 2 1 3 1`		Start autocalibration of the E-517 inputs and outputs for all channels (e.g. ADC gain/offset, DAC gain/offset). Note: The new hardware modules come calibrated. Nevertheless the ATC procedure is required to adapt several E-517 parameters for highest precision of the E-517. See "Calibration Settings" (p. 38) and ATC command description (p. 134) for details.

`ATS? 1 1 2 1 3 1`	`1 1=0`
	`2 1=0`
	`3 1=0`
		Check the autocalibration results. If autocalibration was successful, the response is 0 for the individual channels.

`TSP?`	`1=+0000.4806`
	`2=+0000.8773`
	`3=+0000.5080`
		Query the current position of all sensor channels. Compare it to the display on the E-517. The display should show position values suitable to the TSP? response.

`VOL?`	`1=-0002.4736`
	`2=-0000.0339`
	`3=+0000.8918`
		Query the current output voltage of all piezo channels. Compare it to the display on the E-517. The display should show voltage values suitable to the VOL? response.
		Note that all piezo-channel related commands and parameters use the identifiers 1 to 3, except of the display configuration parameters above which use identifiers 4 to 6 for the piezo channels.

`WPA 100`		Save the current E-517 configuration to make it the power-on default.

Example 2: Hardware is removed

The second amplifier channel is removed from a modular E-500 system with two single-channel amplifier modules, one E-509 two-channel sensor / servo module and one E-517.i3. With the new hardware configuration, one axis, one piezo channel and two sensor channels are available, but only the first sensor channel is still in use. After the hardware was removed, the E-517 must be configured as follows:

Command String to Send	Response	Action Performed
`CCL 1 advanced`		Switch to command level 1 to have write access to the configuration parameters
`SPA 2 0x02000000 0`		Disable sensor channel 2 and hence also the second axis ("B") and piezo channel 2 (sensor channel 3 is already disabled since no hardware was present in the original configuration)
`SAI?`	`A`	Get the axis identifiers of all axes which are accessible by commands. If disabling was successful, the response shows only axis A.
`TSC?`	`1`	Get the number of sensor channels. If disabling was successful, the response is 1.
`TPC?`	`1`	Get the number of piezo channels. If disabling was successful, the response is 1.
`SPA 2 0x04000e01 -1 2 0x04000e00 @@`		Hide the second sensor channel (ID 2) from the display, and remove the unit setting
`SPA 5 0x04000e01 -1 5 0x04000e00 @@`		Hide the second piezo channel (ID 5) from the display, and remove the unit setting
`WPA 100`		Save the current E-517 configuration to make it the power-on default.

3.5.3 Calibration Settings

The E-517 comes calibrated, i.e. the offset and the gain settings for the sensor and piezo electronics and for the internal A/D and D/A converters are optimized for highest precision of the present hardware (electronics and stage(s)).

Recalibrate the <Produktname only in the following cases:

- New stage added to the system, or stage replaced
- Changes of the piezo control electronics

NOTES

For successful calibration of the E-517 as described here, it is important that

- All stages are connected to the piezo control electronics
- The SERVO switches of all channels are set to OFF on the piezo control electronics
- The hardware is free from defects
- The calibration facilities physically present on the hardware (e.g. potentiometers) are properly adjusted (see the User Manual(s) of the piezo control electronics for calibration instructions). If you install new electronics channels which were ordered together with mechanics, these components should come calibrated so that you can immediately start with the calibration described here.

If the E-517 is switched to the E-516 GCS syntax version, the calibration settings described here are also valid, but the IDs of the corresponding parameters differ. See the list below for details.

New Stage Added to the System, or Stage Replaced

The settings to be adjusted are the offset and gain for sensor electronics and piezo amplifier. This is done using the Osen, Ksen, Opzt, Kpzt values which can be set directly via controller parameters (use SPA, SEP, WPA commands; see "Set System Parameters" (p. 31) for detailed information regarding parameter handling):

- Ksen (when sensor voltage changes 1 V, the position change of stage is Ksen (µm)): parameter ID is 0x02000300

- Osen (when sensor voltage is 0 V, the actual position of stage is Osen (µm)): parameter ID is 0x02000200

- Kpzt (when control voltage changes 1 V, the piezo voltage change is Kpzt (V)): parameter ID is 0x0B000003

- Opzt (when control voltage is 0 V, the piezo voltage is Opzt (V)): parameter ID is 0x0B00000A

The formulae for calculating the values for Ksen, Osen, Kpzt, Opzt are:

Ksen = $(P_{10}-P_0)$ / 10.0
Osen = P_0
Kpzt = $(V_{10}-V_0)$ / 10.0
Opzt = V_0
Where:
P_{10} is the actual stage position when sensor monitor voltage is 10 V
P_0 is the actual stage position when sensor monitor voltage is 0 V
V_{10} is the actual piezo voltage when the control voltage for the amplifier is 10 V in open-loop operation (servo OFF)
V_0 is the actual piezo voltage when the control voltage for the amplifier is 0 V in open-loop operation (servo OFF).

Example 1:

A stage has sensor monitor output of 0 to 10 V, the stage travel is to be 0 µm to ~50 µm, the piezo voltage is to be 0 and 100 V when the control voltage is 0 V and 10 V, respectively, in open–loop operation. The settings must then be:

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

The stage is connected to the first sensor channel and the first amplifier channel. So you have to send

```
SPA 1 0x02000300 5 1 0x02000200 0 1 0x0B000003 10 1 0x0B00000A 0
```

to change the parameters in volatile memory.

Example 2:

The stage travel is to be ±25 µm. The piezo voltage in open-loop operation is to be -0.5 and 100.5 V when the control voltage is 0 V and 10 V respectively. Then:

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

Opzt = -0.5

The stage is connected to the third sensor channel and the third amplifier channel. So you have to send

```
SPA 3 0x02000300 5 3 0x02000200 -25 3 0x0B000003 10.1 3 0x0B00000A -0.5
```

to change the parameters in volatile memory.

Example 3:

A stage has sensor monitor output of 0 to 10 V, the nominal stage extension is to be 0 µm to ~15 µm, the piezo voltage is to be 0 and 100 V. However, after being calibrated with another system, the real piezo extension is 0 µm to ~14.5 µm, and the piezo voltage is to be 0 and 98 V. Then

Ksen = (14.5 - 0) / 10.0 = 1.45
Osen = 0
Kpzt = (98 - 0) / 10.0 = 9.8
Opzt = 0
The stage is connected to the second sensor channel and the second amplifier channel. So you have to send

```
SPA 2 0x02000300 1.45 2 0x02000200 0 2 0x0B000003 9.8 2 0x0B00000A 0
```

to change the parameters in volatile memory.

Changes of the Piezo Control Electronics

Possible changes are, for example, the installation of additional sensor and/or amplifier channels in the system, the replacement of that channels and/or hardware adjustments, e.g. setting of internal potentiometers for the channels. If channels are added to the system, they are probably used with new stages. In this case first adjust the Osen, Ksen, Opzt and Kpzt parameters as described above before you adjust the settings described below with the ATC command.

The settings to be adjusted are the gain and offset for the E-517 A/D and D/A converters, for the sensor input to the P-I-controller and for the piezo monitor voltage output of the amplifier. These settings are implemented as protected controller parameters which can only be changed by an automated autocalibration procedure. This procedure can be started with the ATC command (p. 134). Switch to command level 1 (CCL command (p. 139)) before you apply ATC. It is recommended to use ATC option 1 for a complete calibration. You can ask with ATC? (p. 137) for the options and with ATS? (p. 138) for the success of the last autocalibration procedure. To save the autocalibration results to non-volatile memory, use WPA 100. See "Configure Axes and Channels" (p. 33) and the ATC command description (p. 134) for examples. If the ATC procedure fails, contact your Physik Instrumente Sales Engineer.

Calibration Details

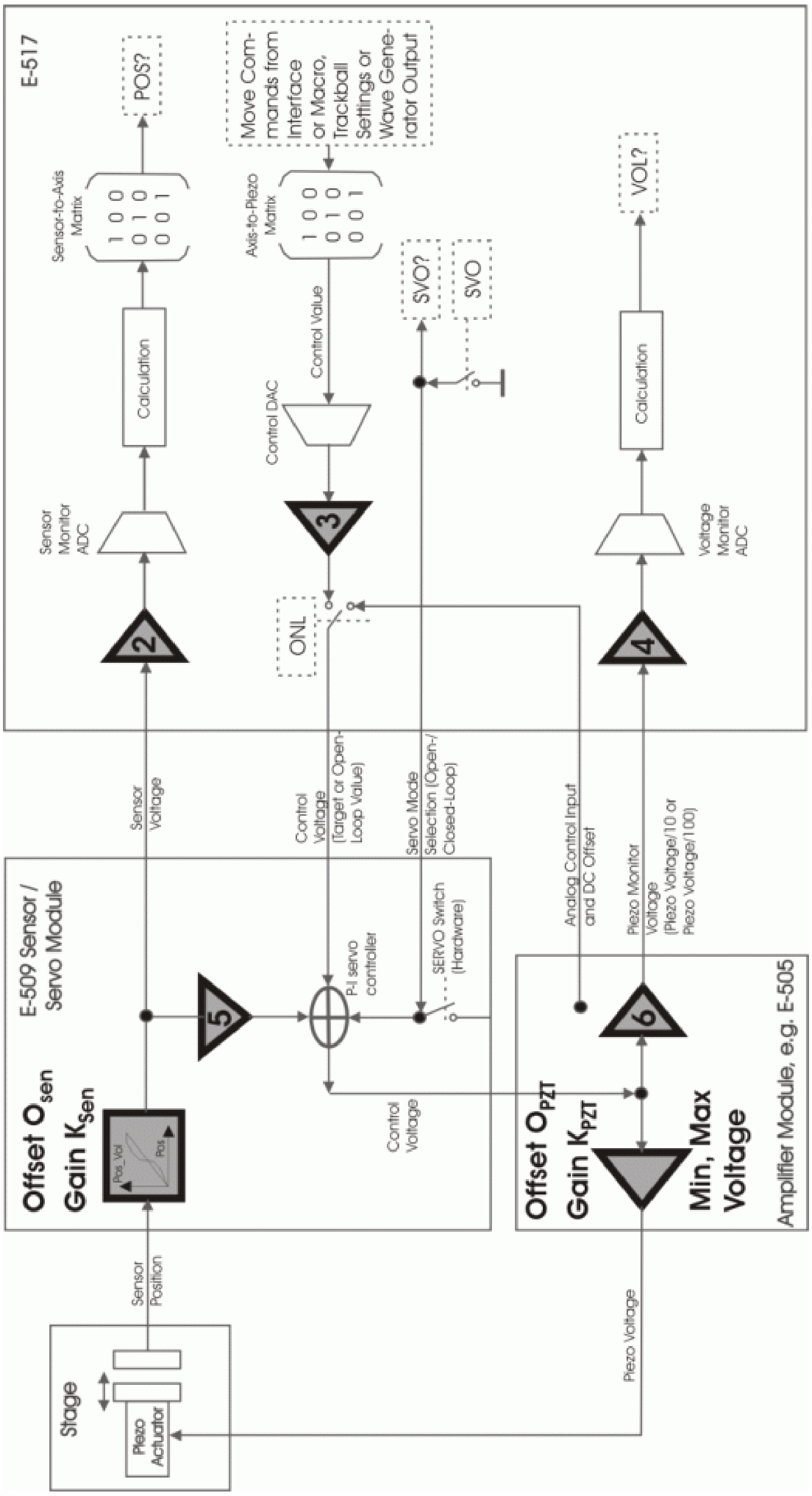


Figure 2: E-517 calibration settings (controller parameters and ATC options highlighted), shown for one channel / axis of the piezo control electronics; details stated below

The table below gives an overview over the IDs and definitions of the controller parameters used for calibration.

Description	Adjustable by	E-517 Parameter ID (hexa-decimal)	Command Level for write access (2 = protected)	Data Type	E-516 Parameter ID
Sensor correction 0 order (offset): Osen (when sensor voltage is 0 V, the actual position of stage is Osen (µm))	SPA, SEP	0x02000200	0	FLOAT	8
Sensor correction 1st order (gain): Ksen (when sensor voltage changes 1 V, the position change of stage is Ksen (µm))	SPA, SEP	0x02000300	0	FLOAT	7
ADC gain	ATC option 2 for sensor monitor ADC ATC option 4 for voltage monitor ADC	0x04000500 Items 1 to 3 for this parameter refer to monitor for sensor channels (position), items 4 to 6 refer to monitor for piezo channels (voltage)	2	FLOAT	3 with sensor monitor channels 1 with voltage monitor channels
ADC offset	ATC option 2 for sensor monitor ADC ATC option 4 for voltage monitor ADC	0x04000600 Items 1 to 3 for this parameter refer to monitor for sensor channels (position), items 4 to 6 refer to monitor for piezo channels (voltage)	2	FLOAT	4 with sensor monitor channels 2 with voltage monitor channels

Description	Adjustable by	E-517 Parameter ID (hexa-decimal)	Command Level for write access (2 = protected)	Data Type	E-516 Parameter ID
HW gain	ATC option 5 for sensor input to P-I-controller ATC option 6 for piezo monitor voltage output of the amplifier	0x04000700 Items 1 to 3 for this parameter refer to sensor channels, items 4 to 6 refer to piezo channels	2	FLOAT	13 with sensor channels 11 with piezo channels
HW offset	ATC option 5 for sensor input to P-I-controller ATC option 6 for piezo monitor voltage output of the amplifier	0x04000800 Items 1 to 3 for this parameter refer to sensor channels, items 4 to 6 refer to piezo channels	2	FLOAT	14 with sensor channels 12 with piezo channels
DAC offset	ATC option 3, affects the DACs which output the control voltages for the piezo amplifier channels	0x0A000010	2	FLOAT	6
DAC gain	ATC option 3, affects the DACs which output the control voltages for the piezo amplifier channels	0x0A000020	2	FLOAT	5
Gain Kpzt (when control voltage changes 1 V, the piezo voltage change is Kpzt (V))	SPA, SEP	0x0B000003	0	FLOAT	9
Offset Opzt (when control voltage is 0 V, the piezo voltage is Opzt (V))	SPA, SEP	0x0B00000A	0	FLOAT	10

NOTES

Although the item IDs for the parameters affected by ATC range from 1 to 6, the channel IDs for ATC always range from 1 to 3.

ATC option 1 affects all settings which can be adjusted separately by options 2 to 6.

The protected parameters "Min Voltage" (ID 0x0B000007) and "Max Voltage" (ID 0x0B000008) which limit the output voltage for the piezo channels are adapted automatically if Opzt and Kpzt are changed and/or ATC option 6 is used.

3.5.4 Select Command Set Version

Users who have written software for the E-516 interface / display module which is the predecessor of the E-517 can switch to the GCS syntax used with the E-516. Except for the PITerminal, the software on the E-517 CD can not be used when the E-517 is switched to the E-516 GCS syntax version. Do not switch to the E-516 GCS syntax version except when necessary for compatibility reasons!

PI software and documentation for the E-516 GCS syntax version is available for download at www.pi.ws or can be obtained from your Physik Instrumente Sales Engineer. The E-516 GCS syntax version is included in the E-517 firmware so that no extra firmware update is required to make it available.

Proceed as follows to select the GCS syntax version:

1 Send
CCL 1 advanced
to switch to command level 1

2 Send
CSV 1
to switch to E-516 GCS syntax version or
CSV 2
to switch to E-517 GCS syntax version (default setting)

3 Save the selection by sending
WPA 100
to make it available after the next power-on or reboot

The current active GCS syntax version is shown in the display (main screen, rightmost corner; "E517" or "E516") and can be queried with the CSV? command (p. 141).

If switched to E-516 GCS syntax version, the functions available on the E-517 are the same as with the original E-516 (e.g. data recorder and trigger output independent of the wave generator are not available). See the E-516 documentation for the available functions and commands.

The communications interfaces (Ethernet, USB, RS-232, IEEE 488) work independent of the selected GCS syntax version.

The settings of the E-517 are also valid if the device is switched to the E-516 GCS syntax version. Not all parameters of the E-517 are present as E-516 parameters, and the IDs of the available parameters differ from the E-517 parameter IDs. See the E-516 documentation for available parameters.

3.6 Updates

3.6.1 Software Updates

Updated releases of software and manuals are available for download at www.pi.ws. While the manuals are freely accessible, you need a password for the software download. This password is provided on the E-517 CD in the E-517 Releasenews PDF file in the \Manuals directory.
To download the latest software (complete CD mirror) from the PI Website, proceed as follows:

1. On the www.pi.ws front page, click on *Download/Support* in the *Service* section on the left

2. On the *Download/Support* page, click on *Manuals and Software*

3. On the *PI Download Server* page, enter the Username and the Password which are provided in the E-517 Releasenews xxxxx.pdf on the E-517 CD and click on *Login*

4. Click on *Download* in the navigation bar across the top

5. Click on the *E Piezo Drivers & Nanopositioning controllers* category

6. Click on *E-517*

7. Click on *Software* (if you click on *Documents* you will get the latest manuals)

8. Click the download button below the latest CD-Mirror (includes the manual versions that were with the release)

3.6.2 Firmware Updates

The overall firmware revision of your E-517 system can be identified in the answer of the *IDN? command (p. 134). For the revisions of the individual firmware unit ask with the VER? command.

In the E-517, there are the following individual firmware units which can be updated separately:

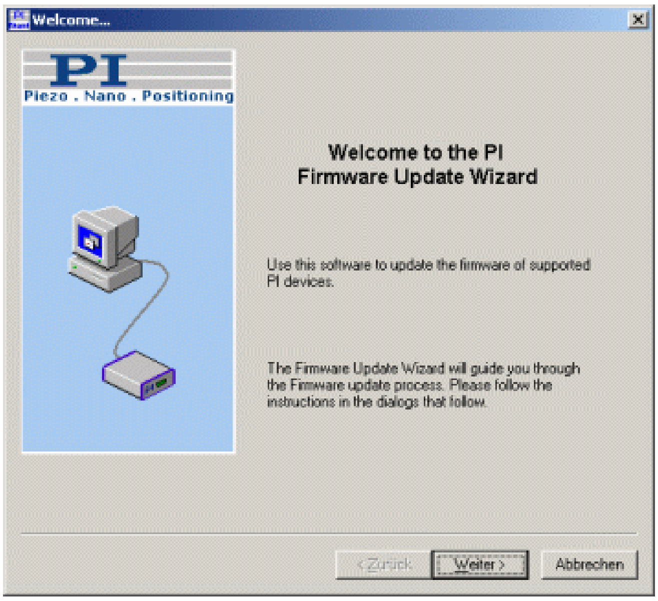
Firmware Unit	Filename
DSP	E-517_DSP_Firmware_HW_*xxxxx*_FW_*xxxxx*.hex
FPGA	E-517_FPGA_Firmware_HW_*xxxxx*_FW_*xxxxx*.hex
MCU	MCU_Firmware_HW_*xxxxx*_FW_*xxxxx*.hex

All firmware updates can be made by running the Firmware Update Wizard on the host computer, using the same communication interfaces as for normal operation. The Firmware Update Wizard is available on the E-517 CD and can be installed as follows:

1 Insert the E-517 CD in your host PC.

2 If the Setup Wizard does not open automatically, start it from the root directory of the CD with the icon.

3 Follow the on-screen instructions, select "custom" installation and then select the Firmware Update Wizard.



Welcome...

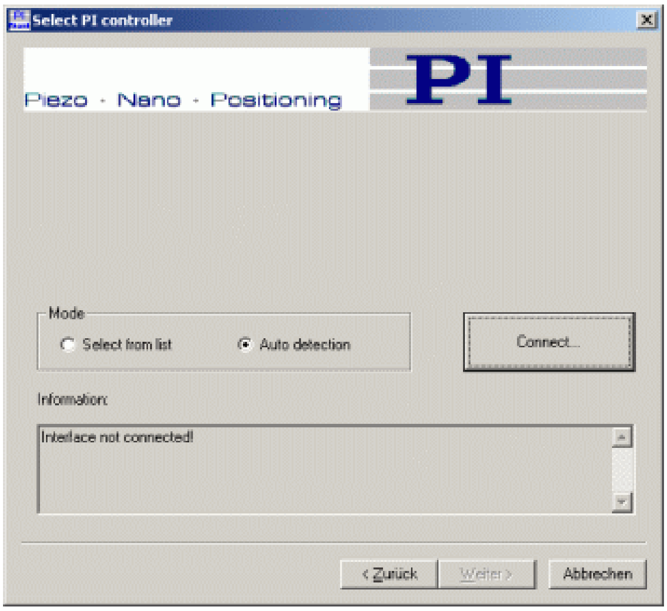
Welcome to the PI Firmware Update Wizard

Use this software to update the firmware of supported PI devices.

The Firmware Update Wizard will guide you through the Firmware update process. Please follow the instructions in the dialogs that follow.

Figure 3: The Firmware Update Wizard Start Screen

In the *Select PI controller* window, select *Auto detection* and press the *Connect...* button. The Firmware Update Wizard guides you through the firmware update of your E-517 system.



Information:

Interface not connected!

Figure 4: Select controller to be connected to Firmware Update Wizard

4 System Description

4.1 Basic Elements

For successful operation of the E-517, you should familiarize yourself with the following features of the device.

Logical Axes:
The E-517 controls logical axes.

With the E-517, the axis-to-channel assignment is fixed so that the first motion axis ("A" by default) always is driven by the first piezo channel and measured by the first sensor channel, the second axis ("B" by default) belongs to the second piezo channel and to the second sensor channel, and the third axis ("C" by default) belongs to the third piezo channel and to the third sensor channel. The assignment is given by two read-only matrices which are implemented via controller parameters: sensor-to-axis matrix (parameter IDs 0x07000500, 0x07000501, 0x07000502) and axis-to-piezo matrix (parameter IDs 0x09000000, 0x09000001, 0x09000002).

Input and Output Signals:
Input signal channels are the sensor channels and output signal channels the piezo channels of the system. Furthermore, the E-517 provides digital in- and output lines for triggering tasks. See "Accessible Items and Their Identifiers" (p. 50) for details.

Communication Interfaces:
The E-517 can be controlled from a host computer (not included) with ASCII commands sent via:

- TCP/IP
- RS-232 serial connection
- USB connection (FTDI; USB 1.1, compatible with USB 2.0)
- GPIB (IEEE 488) connection

All interfaces can be active simultaneously. The commands from the interfaces are queued in the order the completed command lines are received.
See "Communication" (p. 58) for more information.

E-517 Firmware:
The firmware comprises the ASCII command set and the controller parameters and also includes some special features. For version information and updates see "Firmware Update" (p. 46).

- ASCII Commands:

  The E-517 understands the PI General Command Set (GCS; version 2.0).
  The PI General Command Set (GCS) is supported by a wide range of PI systems. This command set is well-suited for positioning tasks with one or more axes. The command set itself is independent of the specific hardware (controller or attached stages).

  Commands are used, for example, to set operating modes, to initiate motion of the mechanics and to query system and motion values. See "GCS Commands" (p. 117) for more information.

- Controller Parameters:

  The key features of the E-517 are mirrored in parameters. Some of the parameters are protected so that their factory settings can not be changed, other parameters can be modified by the user to adapt the system to the individual application. See "Controller Parameters" (p. 250) for more information.

- Command Levels:

  "Command levels" determine the availability of commands and the write access to the controller parameters. Changing the current active command level may require a password and can be done with the CCL command (p. 139).

- Special Features:

  Wave generator: Each axis can be controlled by a "wave generator" which outputs user-specified patterns, so-called "waveforms". This feature is especially important in dynamic applications which require periodic, synchronous motion of the axes. See "Wave Generator" (p. 84) for more information.

  Data recorder: The E-517 comprises a real-time data recorder. It is able to record several input and output signals (e.g. current position, control voltage) from different data sources (e.g. axes or output signal channels). See "Data Recording" (p. 71) for more information.

  Macros: The E-517 can store macros. The macro feature allows defining command sequences and storing them permanently in non-volatile memory in the device. It is possible to define a macro that will be executed automatically every time the E-517 is started, facilitating stand-alone operation without a host computer. See "Working with Controller Macros" (p. 113) for more information.

Software on Host PC

Usually, a host computer is used to operate or at least configure the E-517. Therefore an ample array of software tools for installation on the host computer comes with the E-517. For a complete list of all software on the E-517 CD, see "Software Description" (p. 10).

E-516 COMPATIBILITY NOTE

Users who have written software for the E-516 interface / display module which is the predecessor of the E-517 can switch to the GCS syntax used with the E-516, see "Select Command Set Version" (p. 44). PI software and documentation for the E-516 GCS syntax version is available for download at www.pi.ws or can be obtained from your Physik Instrumente Sales Engineer.

4.2 Accessible Items and Their Identifiers

The identifiers listed below are used to address the appropriate items with the commands of the PI General Command Set (GCS 2.0) which is supported by the firmware of the E-517:

- Logical axes: up to three axes, the default identifiers are A, B, C.
  A logical axis is an axis of a linear, orthogonal coordinate system and represents a basic direction of motion in the E-517 firmware. All motion of the mechanics is commanded for logical axes.
  The axis identifiers can be changed using the SAI command (p. 181) which sets the "Axis Name" parameter, ID 0x07000600. An axis identifier can consist of up to 8 characters; valid characters are
  123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ_
  (ask with the TVI? command (p. 200)). You can ask with SAI? (p. 182) for the current valid axis identifiers.

- Input signal channels: up to three channels, the identifiers are 1 to 3 (can not be changed)
  In the E-517 firmware, the input signal channels represent the sensor channels of the piezo control electronics. The current number of sensor channels can be queried with the TSC? command (p. 199) (reads the Number Of Sensor Channels parameter, ID 0x0E000B03).

- Output signal channels: up to three channels, the identifiers are 1 to 3 (can not be changed)

In the E-517 firmware, the output signal channels represent the piezo amplifier channels of the piezo control electronics. The current number of piezo channels can be queried with the TPC? command (p. 198) (reads the Number Of Piezo Channels parameter, ID 0x0E000B04).

- Digital output lines: up to three lines, the identifiers are 1 to 3 (can not be changed)
  1 to 3 identify the DIO_O1 to DIO_O3 lines on the Digital In/Out Socket (p. 267).
  The maximum number of digital output lines is given by the Number Of Trigger Outputs parameter, ID 0x0E000B05. The number of currently usable digital output lines can be queried with the TIO? command (p. 196).
  See "External Triggering / Signaling" (p. 77) for more information.

- Digital input lines: up to three lines, the identifiers are 1 to 3 (can not be changed)
  1 to 3 identify the DIO_I1 to DIO_I3 lines on the Digital In/Out Socket (p. 267). The number of currently usable digital input lines can be queried with the TIO? command (p. 196).
  See "External Triggering / Signaling" (p. 77) for more information.

- Wave generators: three wave generators, the identifiers are 1 to 3 (can not be changed)
  The number of wave generators is the same as the number of logical axes, and each wave generator is dedicated to one axis. See "Wave Generator" (p. 84) for more information.

- Wave tables (memory tables for waveform data): three tables with 8192 points per table, the identifiers are 1 to 3 (can not be changed)
  The number of wave tables is the same as the number of logical axes, and each wave table is dedicated to one axis (and hence to one wave generator). The maximum number of wave tables is given by the Number of Waves Tables parameter, ID 0x1300010A. See "Wave Generator" (p. 84) for more information.

- Data recorder tables (memory tables for recorded data): three tables with 8192 points per table, the identifiers are 1 to 3 (can not be changed)
  The number of tables is given by the Max Number Of Data Recorder Channels parameter, ID 0x16000100.
  See "Data Recording" (p. 71) for more information.

- Whole system: the E-517 system as a whole, the identifier is 1

- Hardware components: two components, identifiers are 1 and 2: 1 = main board, 2 = display board of the E-517.

NOTES

There is a fixed one-to-one assignment of axes to sensor and piezo channels, to wave tables and wave generators and to the digital input and output lines.

Axes and channels can be activated / deactivated to reflect the current hardware configuration of the piezo control electronics (see "Configure Axes and Channels" (p. 33) for details). If axes are deactivated, only the digital input and output lines of the active axes and the corresponding wave generators and wave tables can be used.

To save the currently valid E-517 configuration to non-volatile memory (e.g. axis identifiers, active axes/channels), where it becomes the power-on default, you must use WPA (p. 229).

4.3 Control Value Generation

4.3.1 Overview and Signal Path

The E-517 interface and display module can be combined in several configurations in a piezo control electronics system. The signal path diagram below shows a usual configuration.

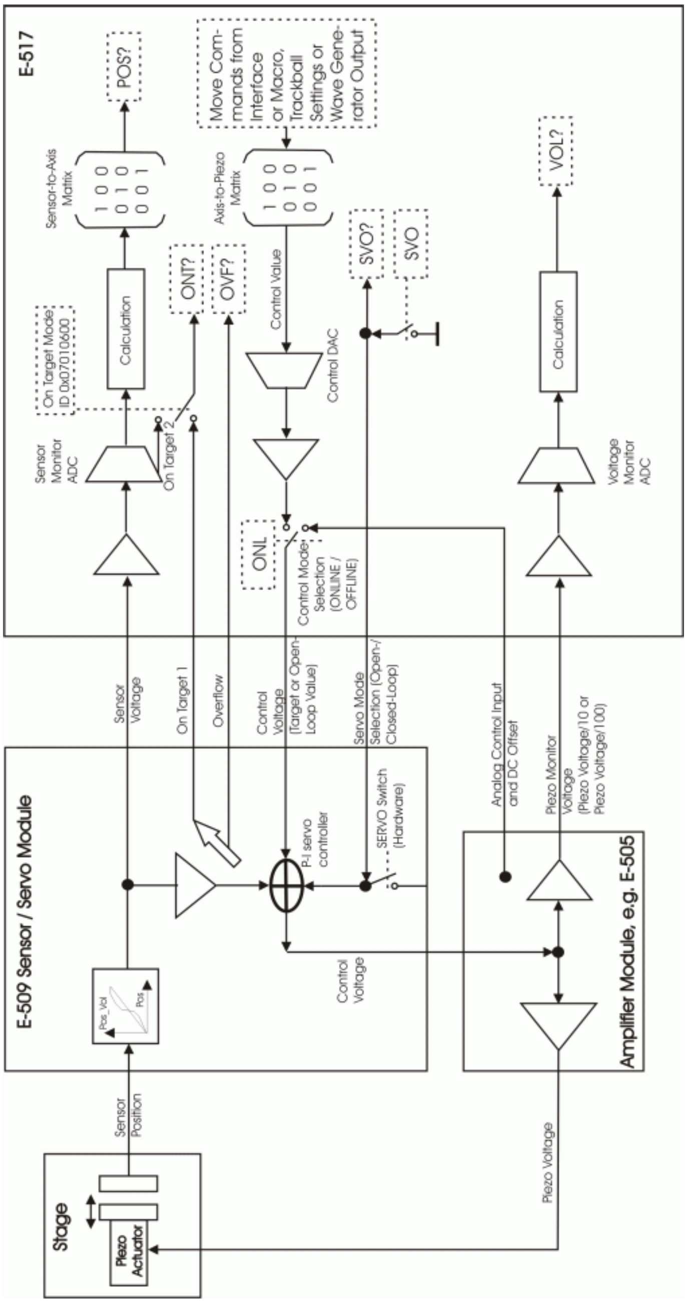

E-517
POS?
Move Commands from Interface or Macro, Trackball Settings or Wave Generator Output
Sensor-to-Axis Matrix
1 0 0
0 1 0
0 0 1
Axis-to-Piezo Matrix
1 0 0
0 1 0
0 0 1
VOL?
ONT?
OVF?
Control Value
SVO?
SVO
On Target Mode ID 0x07010600
Calculation
Control DAC
Calculation
Sensor Monitor ADC
On Target 2
Voltage Monitor ADC
ONL
Control Mode Selection (ONLINE / OFFLINE)
Sensor Voltage
On Target 1
Overflow
Control Voltage (Target or Open-Loop Value)
Servo Mode Selection (Open-/ Closed-Loop)
Analog Control Input and DC Offset
Piezo Monitor Voltage (Piezo Voltage/10 or Piezo Voltage/100)
E-509 Sensor / Servo Module
P-I servo controller
SERVO Switch (Hardware)
Control Voltage
Amplifier Module, e.g. E-505
Pos_Vol
Pos
Sensor Position
Stage
Piezo Actuator
Piezo Voltage

Figure 5: Signal path (only one channel shown) in a system with an E-509 sensor/servo module, a piezo amplifier module and an E-517 interface and display module

4.3.2 Control Value Generation in ONLINE Mode

In ONLINE mode, the E-517 controls the generation of the output voltage. The axis motion can be commanded by multiple sources:

- Move commands (received via interface or from a running macro): SVA (p. 191) and SVR (p. 195) in open-loop operation; MOV (p. 170), MVR (p. 172) and GOH (p. 154) in closed-loop operation; IMP (p. 165) and STE (p. 189) for both servo modes
- Trackball settings on the CHANNEL SETTING screen (see "Trackball Functions" (p. 22) for more information)
- Wave generator output for periodic motion (see "Wave Generator" (p. 84) for more information and examples)

While move commands and trackball settings can be used in parallel to command axis motion, they are not accepted when the wave generator is running for the axis.

The macro feature allows defining command sequences and storing them permanently in non-volatile memory in the controller. Move commands from macros are only accepted in ONLINE mode and only when the wave generator is not running. See "Working with Controller Macros" (p. 113) for more information.

In ONLINE mode the SERVO switches of all channels must be set to OFF on the piezo control electronics. Otherwise the SVO command (p. 193) has no complete control over the servo mode settings. When the servo is switched on for an axis, the target position is set to the current position, and when it is switched off, the last valid control value remains active. This means that when servo is switched on or off during motion caused by move commands, the axis stops. Servo-control can not be switched on or off while the wave generator is running for the axis.

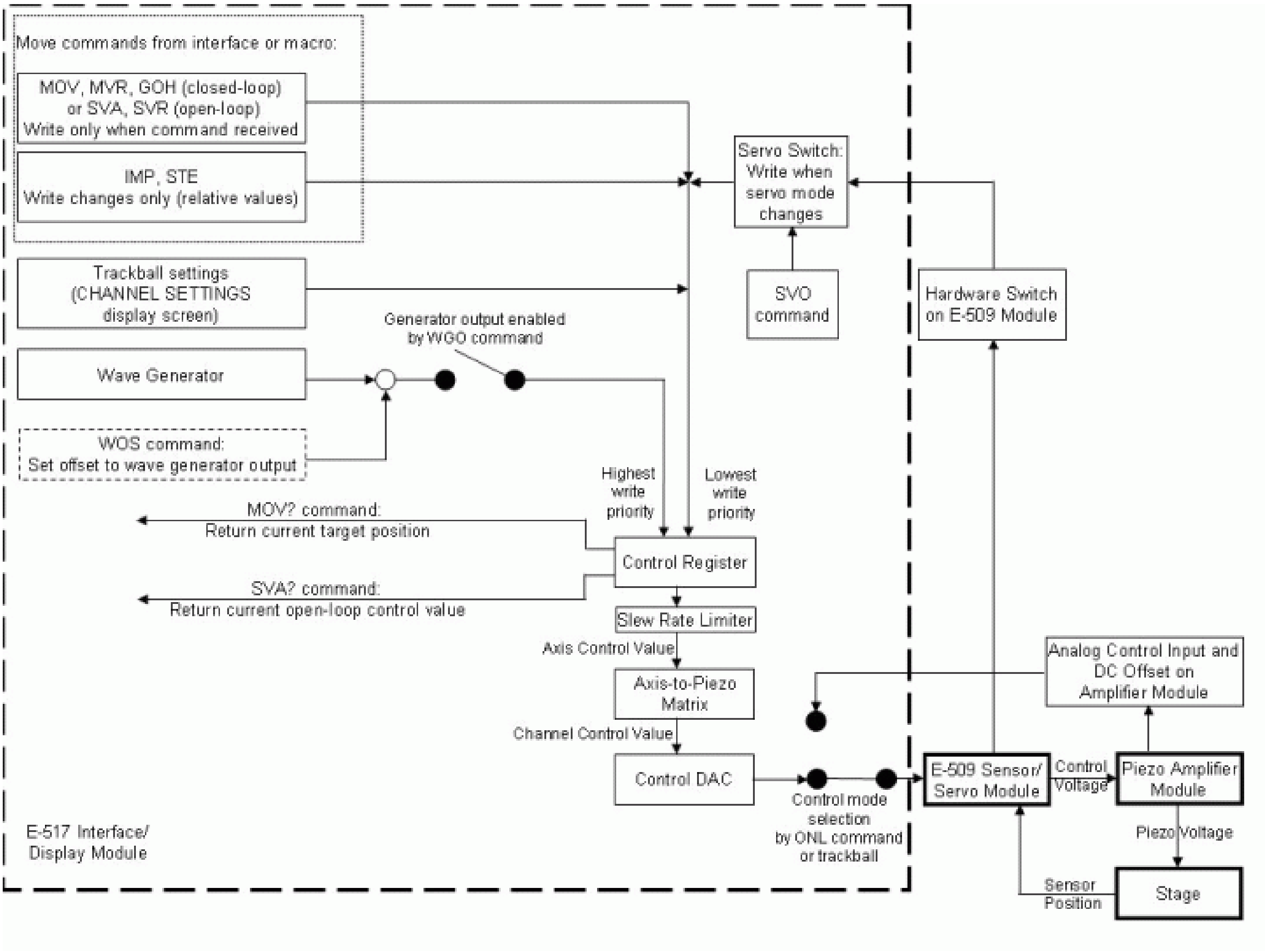

Move commands from interface or macro:
MOV, MVR, GOH (closed-loop) or SVA, SVR (open-loop) Write only when command received
IMP, STE Write changes only (relative values)
Trackball settings (CHANNEL SETTINGS display screen)
Wave Generator
WOS command: Set offset to wave generator output
Generator output enabled by WGO command
Servo Switch: Write when servo mode changes
SVO command
Hardware Switch on E-509 Module
Highest write priority
Lowest write priority
MOV? command: Return current target position
SVA? command: Return current open-loop control value
Control Register
Slew Rate Limiter
Axis Control Value
Axis-to-Piezo Matrix
Channel Control Value
Control DAC
Control mode selection by ONL command or trackball
Analog Control Input and DC Offset on Amplifier Module
E-509 Sensor/ Servo Module
Control Voltage
Piezo Amplifier Module
Piezo Voltage
Stage
Sensor Position
E-517 Interface/ Display Module

Figure 6: Control sources for an axis in ONLINE mode

4.3.3 Drift Compensation

Drift compensation is recommended in static operation to avoid an unwanted change in displacement over time. It can be activated with the DCO command (p. 145).

The drift compensation acts as an additional control-loop in ONLINE mode. The E-517 DAC output (control voltage) is automatically adjusted according to the following criterion:

- Open-loop operation: The current piezo voltage of the channel (shown on the display) must match the last commanded open-loop control value.
- Closed-loop operation: The current sensor position of the channel (shown on the display) must match the last commanded target position.

In dynamic operation DCO should be OFF. As long as you use the wave generator, the drift compensation is deactivated automatically. After the wave generation has finished, the primary DCO setting becomes effective again. Depending on the desired behavior after wave generator operation, you should decide if you want to activate or deactivate DCO before you start the wave generator:

- If the last position after wave generator stop is important, DCO should be off.
- If the last position after wave generator stop does not matter, you can keep DCO on. The position of the piezo actuator then may show a stepwise change.

4.4 On Target Reading

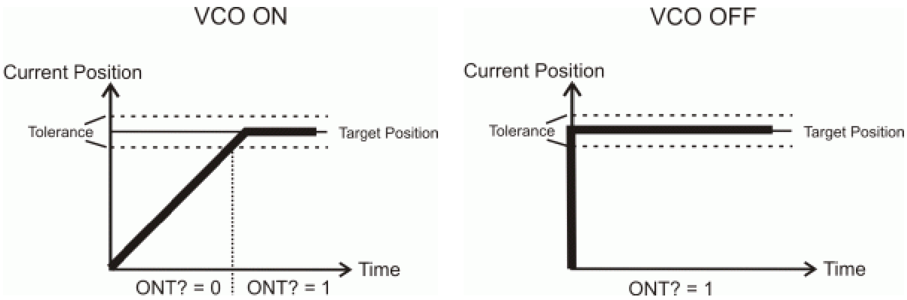
You can read the on-target status of the individual axes with the ONT? command (p. 176). To provide a meaningful value even if the system hardware (stage, controller electronics) does not support an on-target status signal, software-emulated on-target reading is enabled by default with parameter 0x07010600 (1 = enabled).

The on-target status can only be identified in closed-loop operation (servo ON) and is influenced by the Tolerance parameter (ID 0x07000900): the on-target status is true if
|Current Position - Target Position| ≤ Tolerance
The default tolerance value is 0.01 µm.

Software-emulated on-target reading requires the following:

- Successful hardware operation

- The velocity control mode for the axis should be set to ON (with VCO command (p. 204)) to have the control voltage increasing continuously during a certain (small) time period. Otherwise the axis would immediately reach the target position, and ONT? monitoring would make no sense. See figures below for examples.



With the "On Target" trigger mode set by the CTO command (p. 142), the on-target status of the axis can be written to the corresponding trigger line. See "Example-"On Target" Trigger Mode" (p. 80) for more information.

5 Communication

5.1 Interfaces Available

The E-517 can be controlled from a host computer (not included) with ASCII commands sent via:

- TCP/IP
- RS-232 serial connection
- USB connection (FTDI; USB 1.1, compatible with USB 2.0)
- GPIB (IEEE 488) connection

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

NOTES

TCP/IP communication: If no DHCP server is available on the network or if a point-to-point connection between host PC and controller is being used, after power-on or reboot it might take a period of about 30 seconds before communication is possible.

It is not possible to connect multiple TCP/IP command streams to the E-517 via TCP/IP—only one TCP/IP connection is possible at any one time.

When using the USB interface for the first time, two FTDI USB drivers must be installed on the host PC. These drivers are provided on the E-517 CD in the \USB_Serial_Driver directory.

With TCP/IP and USB connections, communication can not be maintained after the E-517 is power-cycled or rebooted. The connection must then be closed and reopened.

The *IEEE 488* tab card is only present in the host software if a GPIB driver from National Instruments (NI) or an NI-compatible driver is installed on your PC.

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

With PIMikroMove™, it is possible to connect to multiple controllers in one program instance. With PITerminal, you need a new program instance for each E-517 you want to connect to.

5.2 Default and Current Settings

The default communication parameters are stored on the E-517. You can read the default settings using the IFS? command (p. 164) and change them with IFS (p. 163). Changes become active with the next power-on or reboot, when the default values are loaded into the controllers volatile memory (RAM).

The current active interface settings can be temporarily changed with IFC (p. 159). The new settings become active immediately and the host PC interface configuration may need to be changed to maintain communication. When the controller is powered down, the settings made with IFC are lost if they were not saved with WPA (p. 229).

To read all current active communication parameters use the IFC? command (p. 161).

NOTE

The IFS, IFS?, IFC and IFC? commands affect the controller side only.

It is also possible to change the default settings with SEP (p. 183) and to read them with the SEP? command (p. 184). If you use RPA (p. 179) to activate the changed settings, it may be necessary to close the current connection and re-open it with the new settings. The appropriate parameter IDs are given below.

The factory defaults of the communication settings stored in the controller are as follows (response to IFS?):

- RSBAUD: gives the baud rate to be used for RS-232 communication,
  default is 115200
  also accessible as Uart Baudrate parameter, ID 0x11000400

- GPADR: gives the device address to be used for GPIB (IEEE 488) communication,
  default is 4
  also accessible as GPIB Address parameter, ID 0x11000900

- IPADR: the first four portions specify the default IP address for TCP/IP communication, the last portion specifies the default port to be used,
  default is 192.168.168.10:50000
  also accessible as IP Address parameter, ID 0x11000600
  Note: While the IP address can be changed, the port must always be 50000!

- IPSTART: defines the startup behavior for configuration of the IP address for TCP/IP communication,
  default is 1 = "use DHCP to obtain IP address"
  also accessible as IP Configuration parameter, ID 0x11000800

- IPMASK: gives the IP mask to be used for TCP/IP communication,
  default is 255.255.255.0
  also accessible as IP Mask parameter, ID 0x11000700

- MACADR: is the unique address of the network hardware in the E-517, read-only, example:
  0-d0-c9-a7-1f-86
  also accessible as MAC Address parameter, ID 0x11000B00

NOTES

A TCP/IP connection will fail if no DCHP server is present but the startup behavior for IP address configuration is set to "use DHCP to obtain IP address, if this fails, use IPADR" (IPSTART = 1). The E-517 will use the address given by IPADR only if IPSTART = 0.

If the IP address for the current TCP/IP connection was obtained from a DHCP server, this address is not reflected in the response to the IFC? command.

The currently active RS-232 baud rate and GPIB (IEEE 488) address can also be checked/set on the E-517 front panel in the COMMUNICATION display screen (p. 21). This screen can be accessed by the trackball (p. 22). The front panel settings correspond with the communication parameter settings described above.

5.3 TCP/IP Connection

The TCP/IP connection is available on the E-517 front panel of the controller, via the RJ 45 socket with the network icon. The access differs depending on the network availability, which can be as follows:

- Network with DHCP server
- PC equipped with an Ethernet connection or network without DHCP server

For the default IP address, IP mask and startup behaviour settings of the TCP/IP interface see "Default and Current Settings" (p. 59).

!

CAUTION

With TCP/IP and USB connections, communication can not be maintained after the E-517 is power-cycled or rebooted. The connection must then be closed and reopened.

NOTES

Make sure that your network administrator has not set the network to forbid unknown devices like the E-517 from logging on.

Only one TCP/IP connection is allowed at any one time: Presently, only one port (50000) is available at the E-517 so that only one application at a time can use the TCP/IP connection. The default port setting can not be changed.

5.3.1 Network with DHCP Server

NOTES

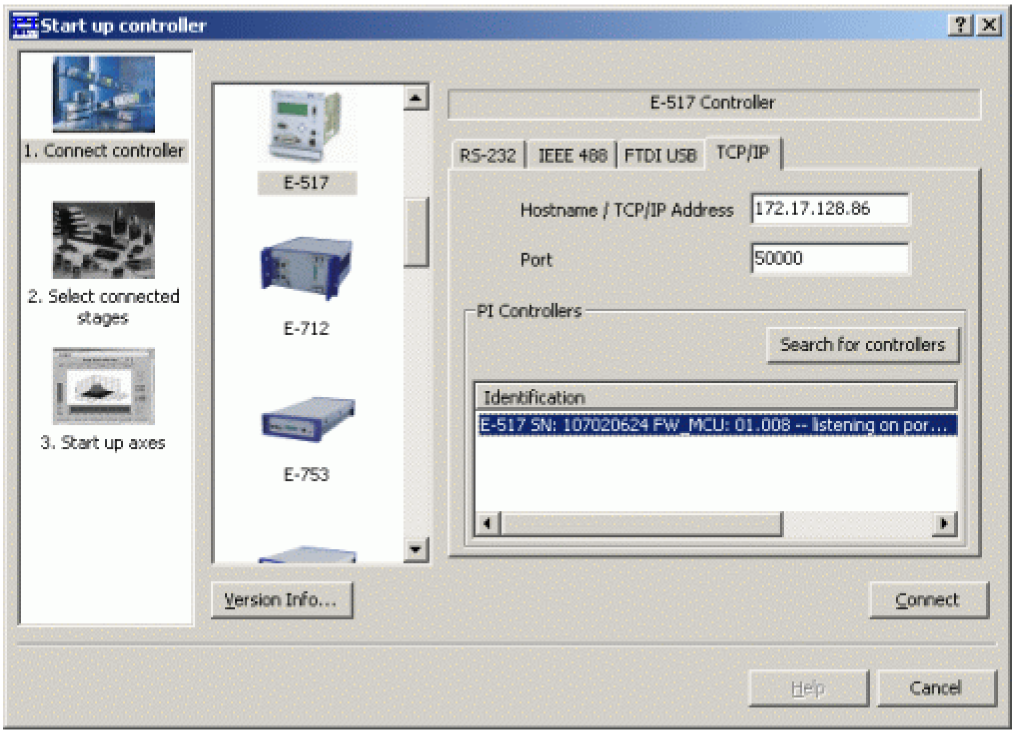
For successful connection, the E-517 startup behaviour for IP address configuration must be set to "Use DHCP to obtain IP address" (IPSTART=1; default selection).

You can check the IPSTART settings with the IFS? command (power-on default setting) and the IFC? command (current setting).

If the IPSTART setting is "Use IP address defined with IPADR" (IPSTART=0), send
IFS 100 IPSTART 1
The change becomes active with the next power-on or reboot.

First, make sure that the proper startup behaviour is set (see Note above). Then, if a network with DHCP server is available, connect the controller to a network access point and power cycle it (newer switches accept both cross-over and straight-through network cables). The controller will automatically obtain an IP address over DHCP.

In the host software (e.g. PIMikroMove™, PITerminal or LabView drivers), you can use the "Search for controllers" functionality in the connection dialog to see all available E-517 controllers with their IP address and port settings. In PIMikroMove™ you have, for example, to press the *Search for controllers* button in the *Start up controller* window (see figure below). In the resulting list, click on the controller to which you want to connect. Check that the IP address and port number were correctly transferred to the *Hostname* and *Port* fields. Then press the *Connect* button to establish the connection.



Hostname / TCP/IP Address: 172.17.128.86

Port: 50000

E-517 SN: 107020624 FW_MCU: 01.008 -- listening on por...

Figure 7: The connection dialog in PIMikroMove™

NOTE

In the controller selection list, you can also identify the controllers which already have a TCP/IP connection open. If you try to connect to such a controller, an error message will be generated as no multiple TCP/IP connections are possible.

5.3.2 PC with Ethernet Card

The IP address and IP mask settings of PC and E-517 must be compatible with each other in the following cases, because otherwise no connection can be established:

- The E-517 is directly connected to an Ethernet connection of the PC.
- E-517 and 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 network).

You can configure either the PC or the E-517 settings to be compatible. If you have a network with multiple E-517s, the settings of the individual devices must be changed to have unique IP addresses for all devices in the network. See below for how to proceed.

NOTES

For successful connection, the E-517 startup behaviour for IP address configuration must be set to "Use IP address defined with IPADR" (IPSTART=0).

You can check the IPSTART settings with the IFS? command (power-on default setting) and the IFC? command (current setting).

If the IPSTART setting is "Use DHCP to obtain IP address" (IPSTART=1; default), send
IFS 100 IPSTART 0
The change becomes active with the next power-on or reboot.

After power-on / reboot of the E-517 you have to wait for a period of about 30 seconds before communication is possible.

If you want to change the IP address and IP mask settings of the PC:

Configure the connection on the PC according to the IP address and IP mask settings of the E-517 (see "Default and Current Settings" (p. 59) for the default E-517 settings). Note that the following steps may vary in some details depending on the version of your Windows operating system:

1 Follow the menu sequence *Start* ⇒ *Settings* ⇒ *Control Panel* ⇒ *Network and Dial-up Connections* ⇒ *Local Area Connection*.

2 In the *Local Area Network (LAN)* status window, click *Properties*.

3 In the *Properties* window of the LAN connection, click on *Internet Protocol (TCP/IP)*.

4 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-517:

Set the first three portions of *IP address* identical to those of the IP address of the E-517, 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-517 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-517 is

255.255.255.0).

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

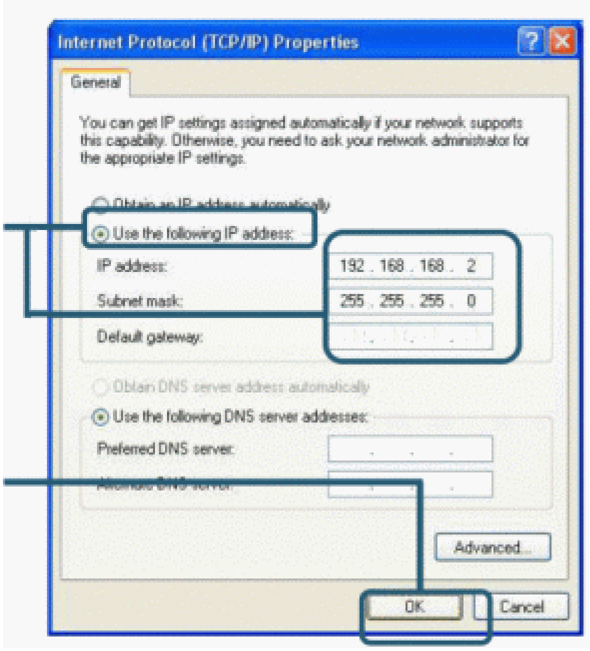
5 Connect the E-517 to the Ethernet card in the PC using the included, special, cross-over cable ("point-to-point" connection). If a hub is used to allow for connection of several devices to the PC, connect the E-517 to the hub using the straight-through network cable.

6 Power on the E-517.

7 Use the "Search for controllers" functionality, described in "Network with DHCP Server" (p. 62), to establish the connection between PC and the E-517.

NOTE

If the connection fails, change the last portion of the *IP address* setting on the PC and try again to connect via "Search for controllers".



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

Press *OK*

Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 168 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Figure 8: Internet Protocol (TCP/IP) Properties window, the settings shown are only examples, maybe they does not match that of your controller

If you want to change the IP address and IP mask settings of the E-517:

1 Establish a serial connection between PC and E-517 as described in "RS-232 Serial Connection" (p. 67).

2 Use the IFS command (p. 163) in the command entry facility of the program to adapt the IP address and IP mask settings of the E-517 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 (subnet 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*
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-517 to the Ethernet card in the PC using the included, special, cross-over cable ("point-to-point" connection). If you connect the E-517 to a free access point (e.g. to a hub) on a network to which the PC is connected, use the straight-through network cable.

5 Power-cycle the E-517.

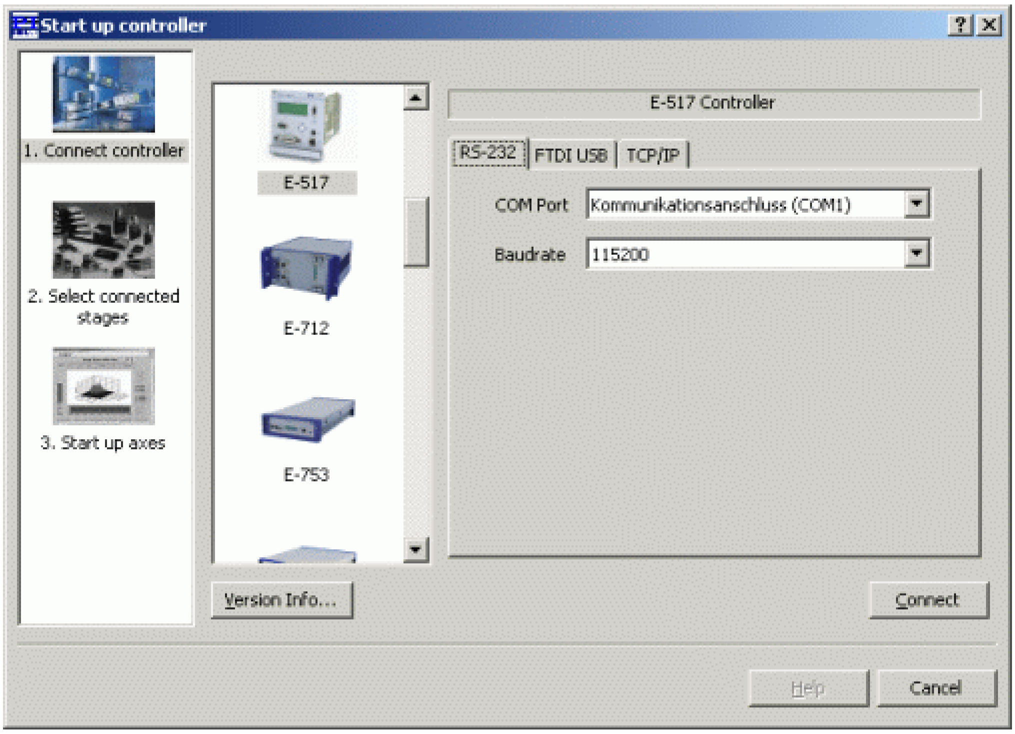
6 Use the "Search for controllers" functionality, described in "Network with DHCP Server" (p. 62), to establish the connection between PC and the E-517.

5.4 RS-232 Serial Connection

The serial communications port is accessed via the sub-D 9m "RS-232" connector (p. 267) on the front panel of the E-517. Use the included null-modem cable to connect the E-517 to the host PC; if the PC has only one COM port, it is probably COM 1.

The serial port on the E-517 is preset to the following parameters:
115,200 baud, 8 bits, no parity, RTS/CTS

In the connection dialog of the host software (e.g. PIMikroMove™, PITerminal or LabView drivers), you make the settings on the host PC side. Select the correct PC COM port and make sure that the baud rate (and other settings) in the dialog match those of the E-517. Otherwise no communication can be established.



COM Port: Kommunikationsanschluss (COM1)

Baudrate: 115200

Figure 9: RS-232 configuration of the host PC side in PIMikroMove™

If you have established a connection and want to change the currently active baud rate, proceed as follows:

1. Use the IFC command (p. 159) in the command entry facility of the program, e.g. by sending IFC RSBAUD 57600.

2. Close the connection.

3 Open the connection again with the baud rate you just set with IFC (in the example 57600).

The currently active RS-232 baud rate can also be checked/set on the E-517 front panel in the COMMUNICATION display screen (p. 21). This screen can be accessed by the trackball (p. 22). The front panel setting corresponds with the baud rate setting available over the PC interface.

NOTE

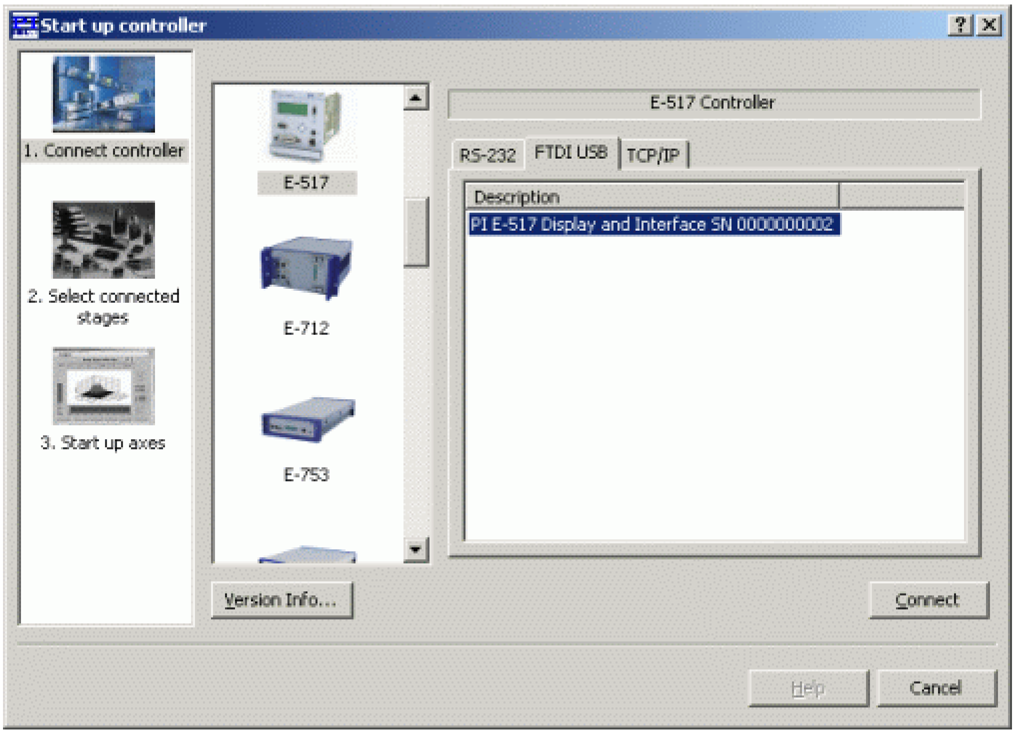
It is recommended that the host PC has a "genuine" hardware RS-232 interface. If the host PC uses a USB-to-serial adapter instead, data loss could occur during communication, especially when transferring large amounts of data.

5.5 USB Connection

The USB interface is available on the front panel of the E-517 via the type B USB socket. Use the included USB cable (USB-A/USB-B) to connect the E-517 to the host PC.

The first time you connect over the USB interface, be sure you are logged on the PC as a user having administrator rights. After the E-517 is powered on, a message will appear saying that new hardware has been detected. Follow the on-screen instructions and insert the E-517 CD. The required FTDI hardware drivers are found in the \USB_Serial_Driver directory.

In the host software (e.g. PIMikroMove™, PITerminal or LabView drivers), you see all E-517s which are connected to the USB sockets of the host PC. In the *Start up controller* window of PIMikroMove™, for example, the present devices are listed on the *FTDI USB* tab card (see figure below). Click on the E-517 to which you want to connect. Then press the *Connect* button to establish the connection.



PI E-517 Display and Interface SN 0000000002

Figure 10: The connection dialog in PIMikroMove™

NOTE

With TCP/IP and USB connections, communication can not be maintained after the E-517 is power-cycled or rebooted. The connection must then be closed and reopened.

5.6 GPIB (IEEE 488) Parallel Connection

The parallel communications bus is accessed via the "GPIB/IEEE488" connector on the front panel of the E-517. Use a suitable cable to connect the E-517 to the host PC. The host PC must be equipped with GPIB hardware, and the corresponding driver must be installed (GPIB driver from National Instruments (NI) or NI-compatible driver).

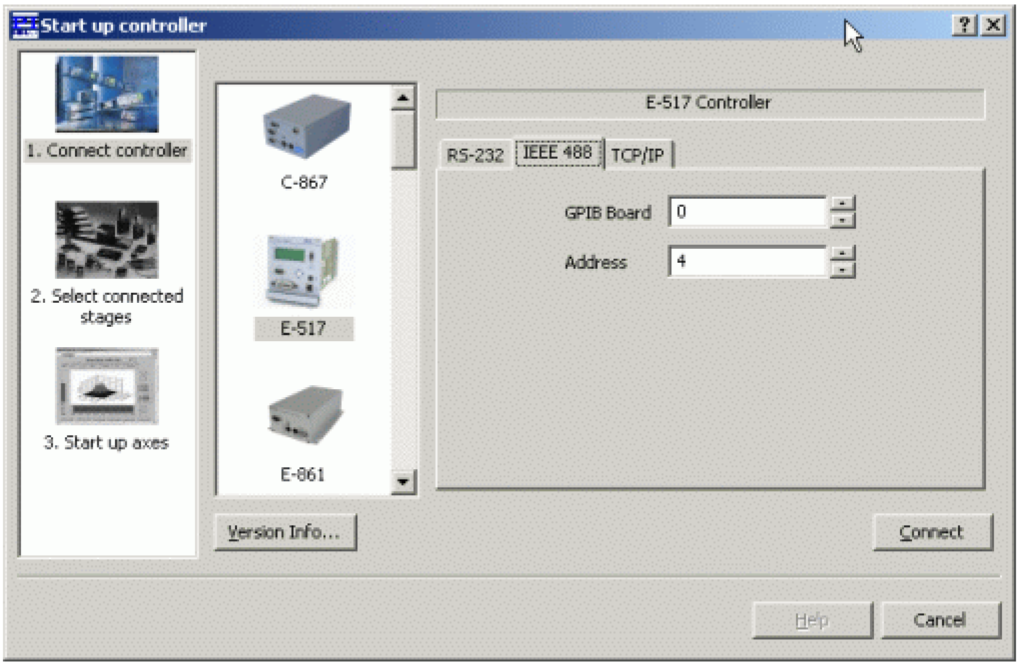
The GPIB device address of the E-517 is preset to 4. Make sure that all devices which share the same GPIB bus have unique address settings. Possible addresses are 1 to 31.

In the connection dialog of the host software (e.g. PIMikroMove™, PITerminal or LabView drivers), select the correct GPIB board in the PC

(*GPIB Board*) and enter the address of the E-517 (*Address*). Otherwise no communication can be established.

NOTE

The *IEEE 488* tab card is only present in the host software if a GPIB driver from National Instruments (NI) or an NI-compatible driver is installed on your PC.



GPIB Board 0

Address 4

Figure 11: GPIB configuration in PIMikroMove™

If you have established a connection and want to change the currently active address of the E-517, proceed as follows:

1 Use the IFC command (p. 159) in the command entry facility of the program, e.g. by sending IFC GPADR 7.

2 Close the connection.

3 Open the connection again with the address you just set with IFC (in the example 7).

The currently active address can also be checked/set on the E-517 front panel in the COMMUNICATION display screen (p. 21). This screen can be accessed by the trackball (p. 22). The front panel setting corresponds with the address setting available over the PC interface.

6 Data Recording

6.1 How to Use the Data Recorder

The E-517 includes a real-time data recorder. It is able to record several input and output signals (e.g. current position, control voltage) from different data sources (e.g. axes or output channels). 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. The E-517 provides 3 data recorder tables (can be read with TNR? (p. 198)) with 8192 points per table.

Start Recording

Data recording is triggered automatically with four commands:

- STE (p. 189) (step response measurement),
- IMP (p. 165) (impulse response measurement),
- WGO (p. 221) (wave generator start) and
- WGR (p. 227) (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.

Read Data

The last recorded data can be read with the DRR? command (p. 152). The data is reported in GCS array format. For details regarding GCS array see the separate manual (SM146E), which is provided on the E-517 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.

Configure Recording

For general information regarding the data recording you can send HDR? (p. 155), which lists available record options, and gives information about additional parameters and commands concerned with data recording.

The data recorder configuration, i.e. the assignment of data sources and record options to the recorder tables, can be changed with DRC (p. 150), and the current configuration can be read with DRC? (p. 152). Upon delivery, the data recorder is preset to record the current positions of the axes.

The data recorder sampling period can be read with the RTR? command (p. 181). 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 (p. 180) or change the parameter value directly.

The current data recorder configuration made with DRC and RTR is saved with the WPA command, in addition to the current parameter values of the E-517 and other settings (see the WPA description (p. 229) for details). When the controller is powered down, the contents of the data recorder tables and all settings which were not saved with WPA are lost.

6.2 Application Example: Frequency Response Measurements

Mechanical resonances of the system may exaggerate the response to certain frequencies in the control signal. To determine the resonant frequencies of the mechanics, use the E-517 data recorder to observe the system response to an impulse in open-loop operation.

NOTE

Very stiff mechanics, e.g. S-325, S-316, P-8xx or P-2xx, have resonant frequencies of several kilohertz. With such mechanics, the frequency response measurement described below will not deliver meaningful results since the sensor bandwidth of the system is limited by the hardware settings on the E-509 analog sensor/servo module. Meaningful results can only be attained with mechanics whose resonant frequency is ≤ 1 kHz.

Proceed as follows for the axis whose resonant frequency is to be measured. In this example, the resonant frequency of axis A is to be measured:

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

2 Start PIMikroMove™ on a host PC connected to the E-517 (see the PIMikroMove™ software manual on the included CD for details).

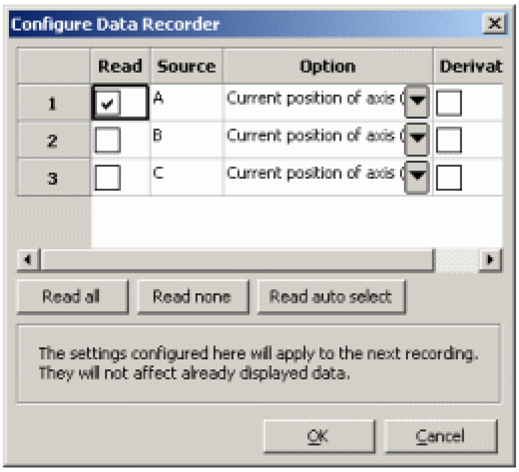
3 In the *Command Entry* window of PIMikroMove™, deactivate the digital filtering of the sensor signal for the sensor channel which belongs to the measured axis. This is recommended to achieve more reliable results. In this example, you have to do this for the

first channel:
Send: SPA? 1 0x05000000
to query the current filter type setting, response should be 2 (= FIR filter; default setting of the Digital Filter Type parameter)
Send: SPA 1 0x05000000 0
to deactivate the digital filter.
Close the *Command Entry* window.

4 Open the *Data Recorder* window via the *E-517* ⇒ *Show/Hide data recorder ...* menu entry.

5 Configure recording:

Press the *Configure...* button in the down right corner of the *Data Recorder* window. In the *Configure Data Recorder* dialog which opens, check the box in the *Read* column and select "Current position of axis" in the *Option* column for the axis to be measured. In the example shown in the figure below, the current position of axis A will be recorded and read afterwards.



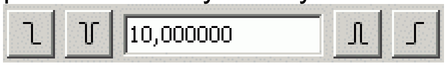
	Read	Source	Option	Derivat
1	☑	A	Current position of axis (	□
2	□	B	Current position of axis (	□
3	□	C	Current position of axis (	□

The settings configured here will apply to the next recording. They will not affect already displayed data.

Click *OK* to close the dialog.

6 Configure the measurement in the bottom line of the *Data Recorder* window:

Select the axis to be measured in the leftmost field (A in this example).
Deselect the *Servo* checkbox since the measurement must be done in open-loop operation.
Enter the amplitude of the impulse in V in the field between the start buttons for step and impulse response measurements and press Enter on your keyboard:

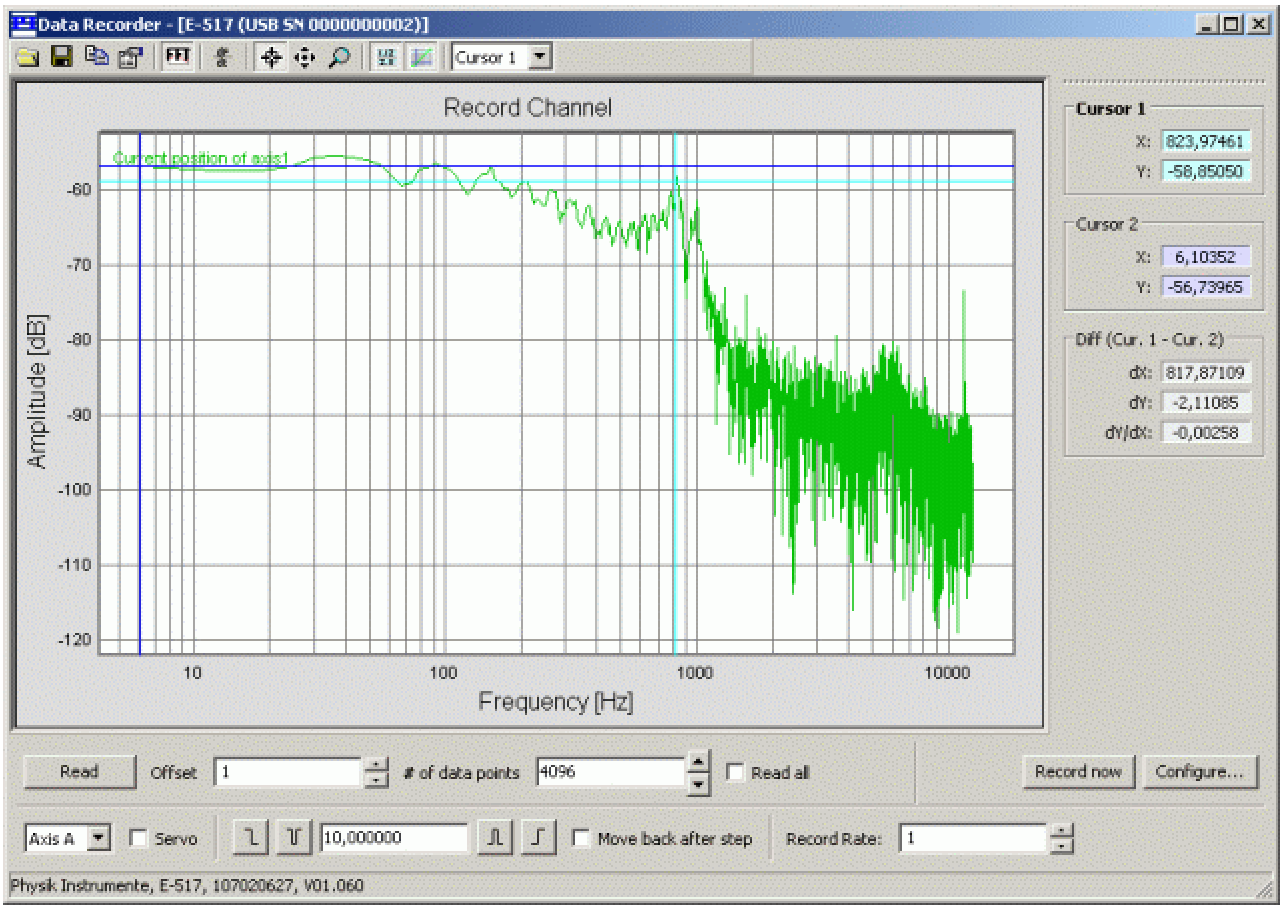

10,000000

The Record Rate value should be 1.

7 Start the impulse response measurement by clicking on the button right beside the amplitude field. An impulse move is commanded (with the IMP (p. 165) command) and the recorded position is displayed in the graphics pane of the *Data Recorder* window.

8 To analyze the recorded data and display it as a Bode frequency response diagram, click on the *FFT* button in the icon bar above the graphics display.

9 Optionally, you can configure the display properties and activate cursors using the buttons in the icon bar above the graphics display (see the PIMikroMove™ manual for details).
This helps to identify the resonance peak on the Bode plot. For example, you can place a cursor on the peak and read out the cursor value which is displayed beside the graph (see figure below; the resonant frequency is about 824 Hz).



Data Recorder - [E-517 (USB SN 0000000002)]

Cursor 1
X: 823,97461
Y: -58,85050

Cursor 2
X: 6,10352
Y: -56,73965

Diff (Cur. 1 - Cur. 2)
dX: 817,87109
dY: -2,11085
dY/dX: -0,00258

Offset: 1
# of data points: 4096
□ Read all

Axis A
□ Servo
10,000000
□ Move back after step
Record Rate: 1

Physik Instrumente, E-517, 107020627, V01.060

10 When the measurement is finished, open the *Command Entry* window again and set the Digital Filter Type parameter back to its initial value (see step 3).

NOTE

If the resonant frequency should not be damped well enough (the measured resonance peak is higher than the highest amplitude value for lower frequencies), then you should adjust the notch filter (center frequency, damping) of the analog piezo control electronics as described in the User manual of the E-802 submodule.

6.3 Data-Recorder Related Commands and Parameters

Command	Description	Notes
DRC (p. 150)	Set Data Recorder Configuration	Assigns data sources and record options to data recorder tables. Settings can be saved with WPA.
DRC? (p. 152)	Get Data Recorder Configuration	Reads current data recorder settings
DRR? (p. 152)	Get Recorded Data Values	Reading can take some time, depending on the number of points.
HDR? (p. 155)	Get All Data Recorder Options	Lists available record options, gives information about additional parameters concerned with data recording
IMP (p. 165)	Start Impulse and Response Measurement	Triggers recording
RTR (p. 180)	Set Record Table Rate	Changes the data recorder table rate in volatile memory (Data Recorder Table Rate parameter, ID 0x16000000)
RTR? (p. 181)	Get Record Table Rate	Reads the current setting of the data recorder table rate (Data Recorder Table Rate parameter, ID 0x16000000)
STE (p. 189)	Start Step and Response Measurement	Triggers recording
TNR? (p. 198)	Get Number of Record Tables	Reads the number of available data recorder tables (Maximum Number of Channels parameter, ID 0x16000100)
WGO (p. 221)	Set Wave Generator Start/Stop Mode	Triggers recording
WGR (p. 227)	Start Recording Synchronous to Wave Generator	Triggers recording

See "How to use the Data Recorder" (p. 71) for more information. For detailed command descriptions see "Command Reference" (p. 131). For the identifiers of the items which can be addressed with the commands see "Accessible Items and Their Identifiers" (p. 50).

Parameter ID	Command Level	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x16000000	0	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
0x16000201	3	Data Recorder Table	3	INT	Maximum record points of table i

See "Controller Parameters" (p. 250) for more information regarding the controller parameters and their handling.

7 External Triggering/Signaling

7.1 Using Trigger Input and Output

It is possible to trigger external devices and to send start/stop signals to the wave generators with the digital I/O lines of the E-517. See "Digital In/Out Socket" (p. 267) for the lines and pinout. With the E-517, there is a fixed one-to-one assignment of axes to digital output and input lines. This means that:

- With E-517.i1 models only the DIO_O1 and DIO_I1 lines are available

- If axes / channels are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the output and input lines of the active axes are available

You can ask for the available lines with the TIO? command (p. 196).

You can program the DIO_O1 to DIO_O3 output lines (TTL, active high) to trigger other devices. Programming can be done using the CTO command (p. 142). If the trigger output is to be synchronized with the wave generator output, you can use CTO in combination with

- TWS (p. 201) which defines trigger action for certain waveform points

- Certain WGO (p. 221) options which start the wave generator and the corresponding trigger output

See "Configuring Trigger Output" (p. 77) and "Trigger Output Synchronized with Wave Generator" (p. 97) for examples.

The DIO_I1 to DIO_I3 input lines (TTL, active high) can be used in conjunction with the WGO command (p. 221) to start and synchronize the wave generator output. See "Wave Generator Started by Trigger Input" (p. 100) for an example.

7.2 Configuring Trigger Output

This section describes how to program the digital output lines of the E-517 using the CTO command (p. 142) to trigger other devices.

The current trigger output configuration set with CTO can be saved with the WPA command (p. 229) (send WPA 100).

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 CTO trigger modes are supported by the E-517:

- 0 = Position Distance; a trigger pulse is written whenever the axis has covered a given distance. See "Example—"Position Distance" Trigger Mode"
- 2 = OnTarget; the on-target status of the selected axis is written to the selected trigger output line. See "Example—"On Target" Trigger Mode"
- 3 = MinMaxThreshold; values for MinThreshold and MaxThreshold must be defined. When the axis position of the selected axis is inside the band specified by the MinThreshold and MaxThreshold values, the selected trigger output line is set high, otherwise it is set low. See "Example—"MinMax Threshold" Trigger Mode"
- 4 = Generator Trigger; enables trigger output synchronized with the wave generator output. The trigger line action at certain waveform points can be defined with TWS (p. 201), and using the WGO command (p. 221) the wave generator can be started with certain trigger output options. See "Example—"Generator Trigger" Mode", "Trigger Output Synchronized with Wave Generator via CTO and TWS" (p. 97) and "Trigger Output Synchronized With Wave Generator via CTO, WGO and WAV" for examples.

To select the mode, set <CTOPam> = 3 and <Value> to the code of the mode; default selection is Generator Trigger (4).

Furthermore, it is possible to select the signal polarity for the digital output line (active high / active low). See "Example—Polarity Setting" (p. 83).

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

NOTE

The width of a trigger pulse is 30 µs by default, except with the MinMaxThreshold trigger mode where the pulse width depends on the threshold settings. You can change the default pulse width using the Pulse Width parameter, ID 0x0E000900. Possible values are in the range of 10 to 150 µs.

7.2.1 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 width of each trigger pulse is 30 µs by default (can be changed using the Pulse Width parameter, ID 0x0E000900).

The unit of <TriggerStep> is µm or µrad.

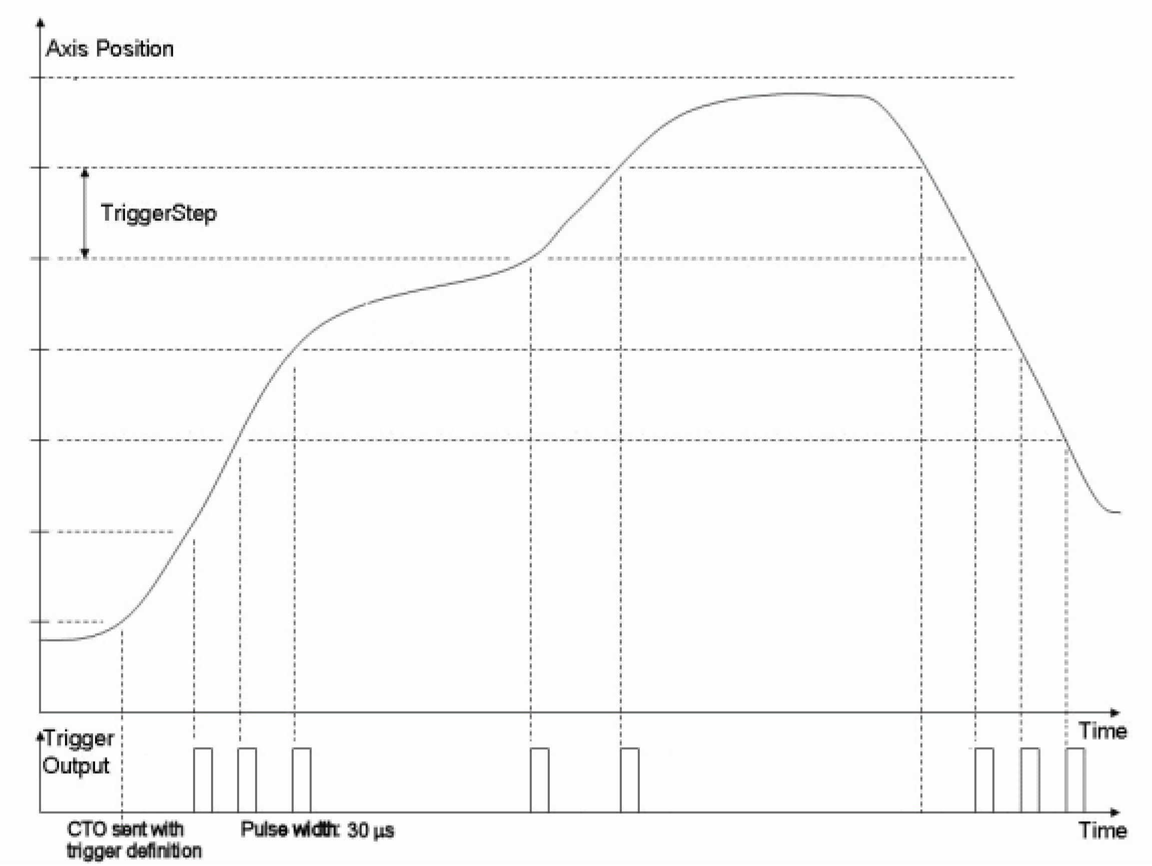


Figure 12: "Position Distance" Trigger Mode

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

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

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	Trigger mode selection	Step size setting
CTO	<TrigOutID> 3 0	<TrigOutID> 1 *Stepsize*

Example: A pulse is to be generated whenever the first axis ("A" by default) has covered a distance of 0.1 µm. Since the assignment of axes to digital output lines is fixed (A belongs to DIO_I1), the first digital output line (<TrigOutID> = 1) must be configured. Send:

```
CTO 1 3 0 1 1 0.1
```

7.2.2 Example—"On Target" Trigger Mode

With the "On Target" trigger mode, the on-target status of the axis is written to the corresponding trigger line. The on-target status can only be identified in closed-loop operation (servo ON) and is influenced by the Tolerance parameter (ID 0x07000900): the on-target status is true if
|Current Position - Target Position| ≤ Tolerance

For the digital output line <TrigOutID> which belongs to the axis whose on-target state is to be output, set the trigger mode (<CTOPam> = 3) as follows:

Command mnemonic	Trigger mode selection
CTO	<TrigOutID> 3 2

Example: The On-Target status flag of the first axis ("A" by default) is to be written. Hence you have to set the "On Target" trigger mode for the digital output line DIO_O1 (ID 1). The following command sequence could be used:

```
ONL 1 1
SVO A 1
MOV A 0.0
CTO 1 3 2
MOV A 1.0
```

NOTE

If software-emulated on-target reading is enabled with parameter 0x07010600, the on-target status can also be read with the ONT? command (p. 176). See "On Target Reading" (p. 57) for more information.

7.2.3 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.

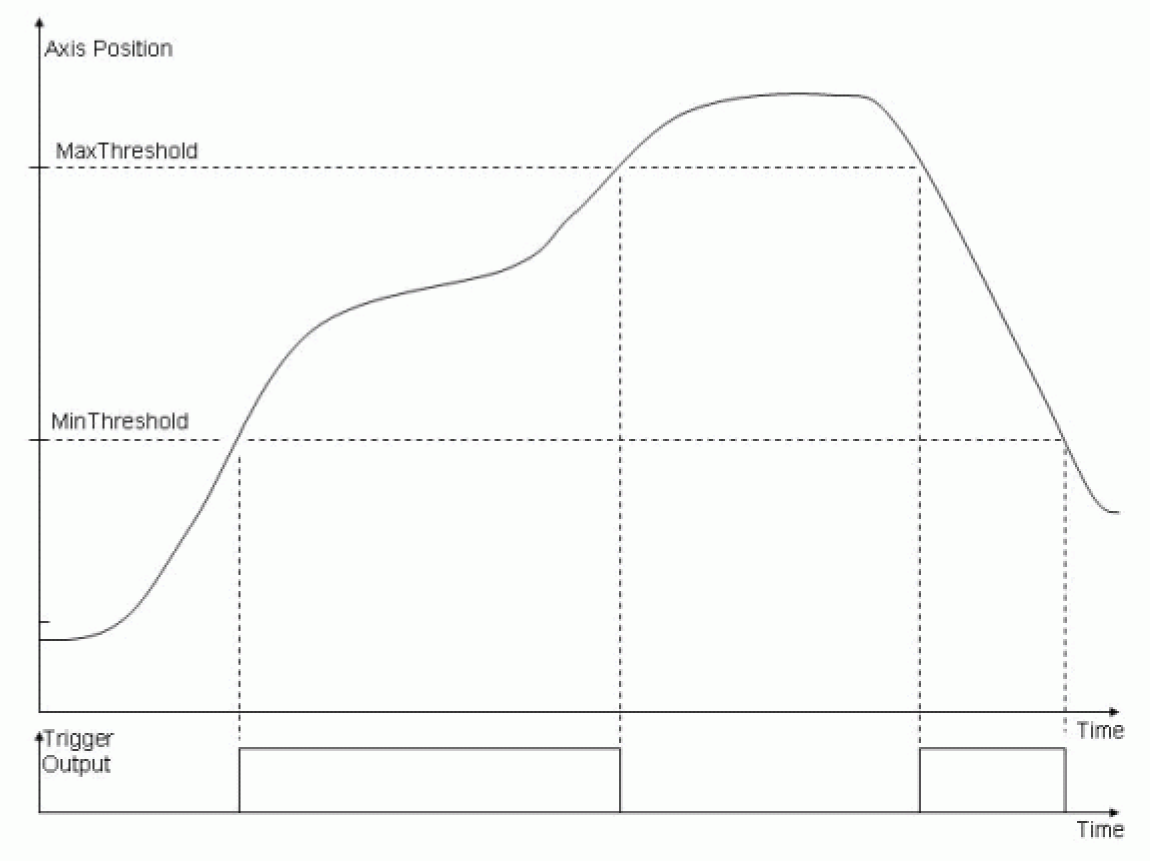


Figure 13: "MinMax Threshold" Trigger Mode

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

- 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 three "portions", each starting with the <TrigOutID> declaration):

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

Example: Whenever the axis position of the first axis ("A" by default) is higher than 0.3 µm and lower than 0.6 µm, the corresponding digital output line is to be set high. Since the assignment of axes to digital output lines is fixed (A belongs to DIO_I1), the first digital output line (<TrigOutID> = 1) must be configured. Send:

CTO 1 3 3 1 5 0.3 1 6 0.6

7.2.4 Example—"Generator Trigger" Mode

With the "Generator Trigger" mode, the trigger output will be synchronized with the wave generator output. CTO must be used in combination with TWS (p. 201) and / or WGO (p. 221). The number of wave generator output cycles during which trigger pulses are to be output can be set using the Number Of Trigger Cycles parameter, ID 0x18000100. The width of each trigger pulse is 30 µs by default (can be changed using the Pulse Width parameter, ID 0x0E000900).

To enable the "Generator Trigger" mode, the following CTO 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

See "Trigger Output Synchronized with Wave Generator" (p. 97) for a detailed example.

7.2.5 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	Polarity selection
CTO	<TrigOutID> 7 *pol.code*

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

CTO 1 7 0

8 Wave Generator

Each axis can be controlled by a "wave generator" which outputs user-specified patterns, so-called "waveforms". This feature is especially important in dynamic applications which require periodic, synchronous motion of the axes. The waveforms to be output are stored in "wave tables" in the controllers volatile memory—one waveform per wave table. Waveforms can be created based on predefined "curve" shapes. This can be sine, ramp, polynomial, tangent and single scan line curves. Additionally you can freely define curve shapes. Programmable trigger inputs and outputs facilitate synchronization of external events.

In "How to Work with the Wave Generator" (p. 84) and "Wave Generator Examples" (p. 92) you will learn how to use the wave generator, and "Wave-Generator-Related Commands and Parameters" gives an overview.

During the wave generator output, data is recorded in "record tables" on the controller. See "Data Recording" (p. 71) for more information.

8.1 How to Work with the Wave Generator

The following subsections describe the wave generator handling in detail. See also "Wave Generator Examples" (p. 92).

8.1.1 Basic Data

The E-517 provides three wave generators and three wave tables with 8192 points per table. There is a fixed one-to-one assignment of axes to wave generators and wave tables. If axes are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the wave generators and wave tables belonging to the active axes can be used. With E-517.i1 models only wave generator 1 and wave table 1 can be used.

To ask for the number of wave generators, use the TWG? command (p. 200). The maximum number of wave tables is given by the Number of Waves Tables parameter, ID 0x1300010A. The maximum number of points per table can be queried with the WMS? command (p. 227) or via the Max Wave Points of table i parameter, ID 0x13000201. Use the SPA? command (p. 188) to ask for the parameter values.

8.1.2 Basic Operation

1 Define the waveform segment-by-segment using the WAV command (p. 209). The waveform will be written to the selected wave table.

2 Start the wave generator output and hence the motion of the axis using the WGO command (p. 221). You can choose several start options (e.g. start/stop by external trigger; see the description of the WGO command for more information).
When starting the wave generator, data recording is started automatically.

3 Stop the wave generator output with WGO or #24 (p. 133) or STP (p. 190).

The simple example below shows how to start periodic motion of the first axis ("A" by default) using the command entry facilities of PIMikroMove™ or PITerminal:

Command String to Send	Action Performed
`WAV 1 X SIN_P 2000 20 10 2000 0 1000`	Define a inverted cosine waveform for Wave Table 1 (belongs to the first axis); see WAV description for details
`WGO 1 1`	Start output of Wave Generator 1 (belongs to the first axis) immediately and synchronized by servo cycles
`WGO 1 0`	Stop output of Wave Generator 1

NOTE

PIMikroMove™ provides the *PI Wave Generator Tool* with which you can create waveforms and start the wave generator output without the need to learn the corresponding commands. See the PIMikroMove™ manual for more information.

8.1.3 Additional Steps and Settings

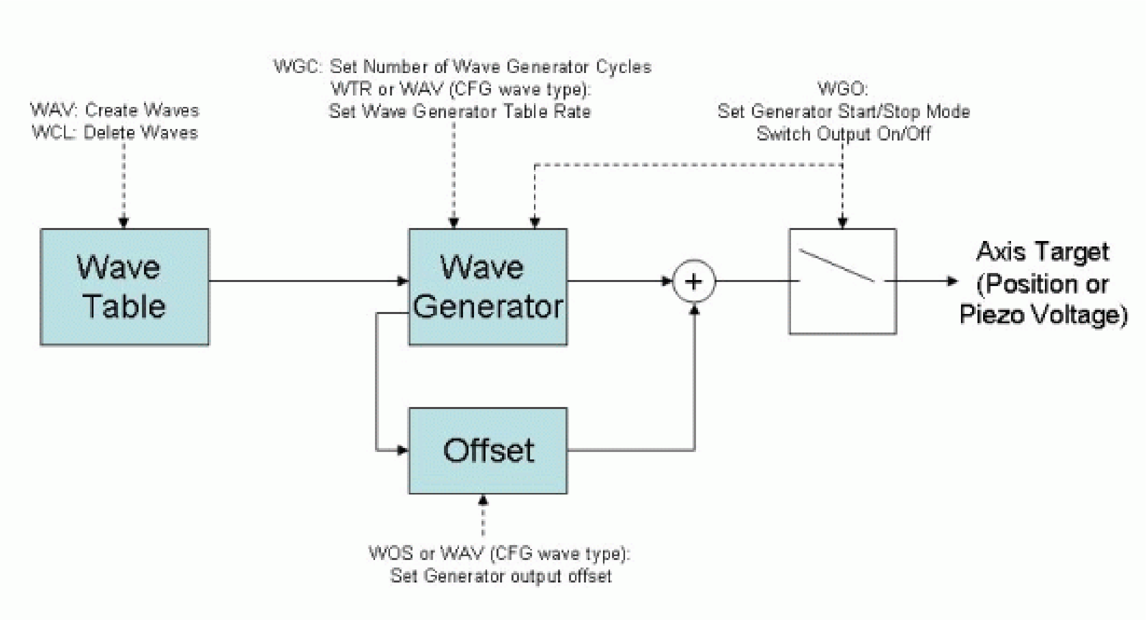

WAV: Create Waves
WCL: Delete Waves
WGC: Set Number of Wave Generator Cycles
WTR or WAV (CFG wave type):
Set Wave Generator Table Rate
WGO:
Set Generator Start/Stop Mode
Switch Output On/Off
Wave Table
Wave Generator
+
Axis Target (Position or Piezo Voltage)
Offset
WOS or WAV (CFG wave type):
Set Generator output offset

Figure 14: Block diagram of one wave generator

You can ask with WAV? (p. 219) for the current waveform lengths and delete the content of selected wave tables with the WCL command (p. 219).

The CFG wave type is a special type with which you can configure additional parameters for a waveform. The settings should be made subsequent to the real waveform definition and will be applied to the waveform when the wave generator output starts. Some CFG settings can be saved to non-volatile memory as power-up defaults—if not, they are valid until a new WAV command is sent or the wave table content is cleared or the E-517 is powered down or rebooted. See the WAV command description (p. 209) for more information.

After you send the waveform definition to the wave table (with WAV (p. 209)), 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 (p. 154). Note that the response to GWD? does not contain any offset set with WOS (p. 228) to the wave generator output.

You can add an offset to the output of a wave generator using the WOS command (p. 228). 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" (i.e. WGO bit 8 is set), the E-517 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 (p. 185) and

SEP (p. 183) or using the CFG wave type (see WAV command (p. 209)). To save the current offset value to non-volatile memory use WPA (p. 229). Keep in mind that this value may also result from the last wave generator run with the "start at the endpoint option" and can therefore be very large. Deleting wave table content with WCL (p. 219) has no effect on the WOS settings.

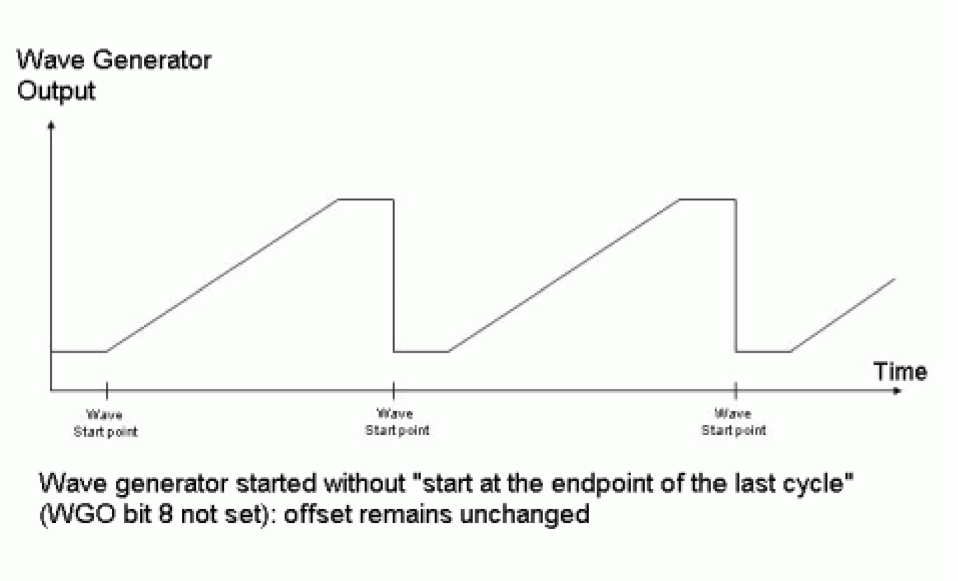


Figure 15: Constant offset

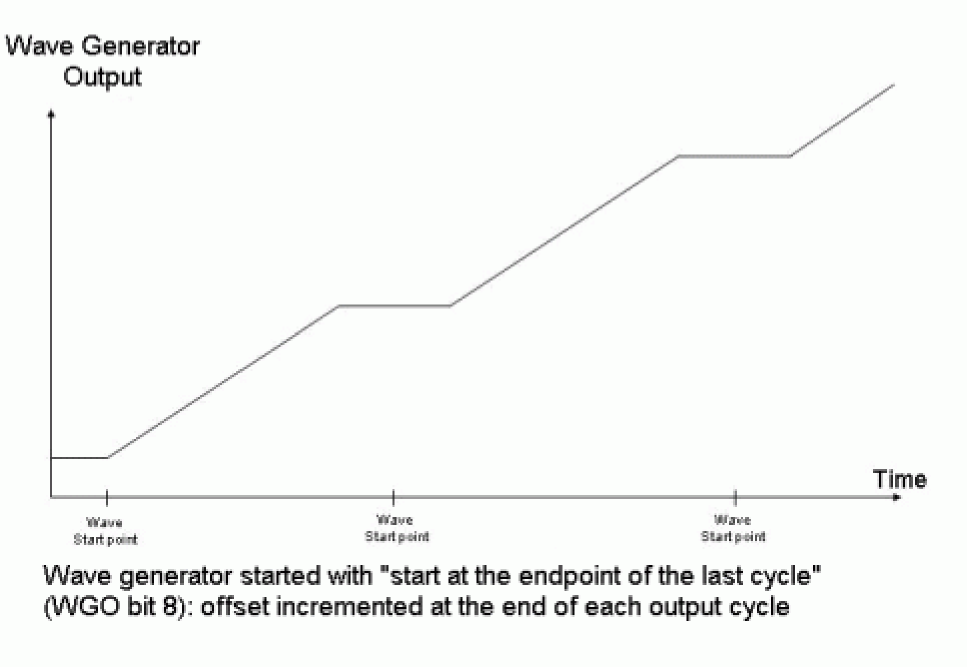


Figure 16: Offset incremented during output

For triggering purposes, the wave generator output can be coupled with the digital output lines DIO_O1 to DIO_O3 of the E-517 (see "Digital In/Out

Socket" (p. 267)). Enable the "Generator Trigger" mode for selected output lines with the CTO command (p. 142) (note that the CTO settings are saved with WPA). Furthermore, TWS (p. 201) and / or WGO (p. 221) must be used to configure the trigger output.
When using the TWS command: First use TWC (p. 200) to set the signal state of the output lines to "low" for all waveform points ("low" is also the power-on default). Then define the trigger line actions with TWS by setting the desired signal states (high or low) of selected output lines for selected waveform points. The number of arguments following the command mnemonic is limited to 32, i.e. you can define at most 10 trigger points per TWS command line. When the wave generator output is started with WGO afterwards, the trigger pulses will be output as configured.
In addition, the WGO command offers several start options for trigger output (WGO bits 3, 4 and 5). If you start the wave generator with those start options, the corresponding pulses are output in addition to any output pulses defined with TWS.
The number of wave generator output cycles during which trigger pulses are to be output can be set using the Number Of Trigger Cycles parameter, ID 0x18000100. The width of each trigger pulse is 30 µs by default (can be changed using the Pulse Width parameter, ID 0x0E000900).

The #9 single-character command (p. 133) 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. With WGO? you can ask for the last-commanded wave generator start options (WGO settings (p. 221)).

You can limit the duration of the wave generator output by setting the number of output cycles with WGC (p. 220). The waveform itself remains unchanged.

Using the WTR command, you can lengthen the individual output cycles of the waveform. As long as the wave generator output is synchronized by servo-cycles and not paused by an external signal (see WGO (p. 221) for details), 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 in seconds is given by parameter 0x0E000200
WTR (wave table rate) value gives the number of servo cycles the output of a waveform point takes, default is 1
Number of Points is the length of the waveform (i.e. the length of the wave table)

If the wave generator is started with WGO bit 2 (triggered and synchronized by external signal), the wave table rate gives the number of HIGH pulses the output of a point takes.

WTR sets the value of the Wave Generator Table Rate parameter, ID 0x13000109, in volatile memory. You can set the wave table rate also using the CFG wave type (see WAV command (p. 209)) or by directly changing the parameter with SPA (p. 185) or SEP (p. 183). Save the value with WPA (p. 229) to non-volatile memory, where it becomes the power-on default. The value of the parameter in volatile memory can be read with the WTR? command (p. 231). The E-517 provides no interpolation so that the interpolation type parameter of the WTR command must always be
0 = no interpolation

With WGR (p. 227) you can restart data recording while the wave generator is running. The recorded data can be read with the DRR? command (p. 152). See "Data Recording" (p. 71) for more information.
If more than one wave generator is running, recording starts at the waveform start point which occurs first.

8.1.4 Application Notes

The high voltage output of the piezo control electronics may be deactivated automatically when an internal temperature sensor detects overheating. If this occurs, it is recommended to reduce the frequency and/or the amplitude and/or the duration of the wave generator output.

The E-517 can not produce waveforms with arbitrary frequency. It can change the output voltage only once every 40 µs (i.e. the maximum update rate for the wave generator output is 25 kHz). Therefore the period of the waveform must be a multiple of 40 µs.
For example: If you want a 70 Hz sine wave, the closest possible value is 70.028 Hz and the resulting waveform will be output with 70.028 Hz and not exactly 70 Hz.

All wave generators can run simultaneously. The wave generators are clocked individually so that multiple wave generators only run synchronously when started with the same WGO command.

As long as a wave generator is running, it is not possible to change (WAV (p. 209)) or to delete (WCL (p. 219)) the connected wave table (i.e. the waveform). The wave generator table rate (WTR (p. 231)), the number of output cycles (WGC (p. 220)), the wave offset (WOS (p. 228)) and the output trigger settings (TWS (p. 201)) can be modified while a wave generator is running.

When the E-517 is in OFFLINE mode, the wave generator output can not be started. If the wave generator output was started in ONLINE mode, it remains active when switching to OFFLINE mode, but it is not used as control value. This means that the wave generator output will not cause axis motion until switching back to ONLINE mode.
Motion commands like MOV (p. 170) or SVA (p. 191) (from command line or from a running macro) are not allowed when the wave generator output

is active.
See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Velocity control (see VCO command (p. 204)) is not effective when the wave generator is running.

As long as you use the wave generator, the drift compensation is deactivated automatically (see "Drift Compensation" (p. 56) for more information).

A wave generator outputs absolute values. In closed-loop operation (servo ON), the output is interpreted as target positions. In open-loop operation (servo OFF), the output corresponds to piezo voltage values (amplifier output in V). Servo-control can not be switched on or off while the wave generator is running for the axis.

When a wave generator finishes by running through a specified number of cycles completely, the final position will be the last point of the waveform. If the option "start at the endpoint of the last cycle" was selected, the final position is the sum of the endpoint of the last output cycle and any offset defined with WAV (p. 209) 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. Exceptions are possible using several start mode bits of the WGO command:
WGO bit 1: The wave generator runs as long as the signal on the corresponding digital input line is HIGH and is paused as long as the signal is LOW.
WGO bit 2: The wave generator outputs one waveform point each time the signal on the corresponding digital input line is HIGH and is paused as long as the signal is LOW (with wave table rate = 1). If the wave table rate is larger than 1, a point is output for the corresponding number of HIGH pulses. See also WTR (p. 231).
WGO bit 14: Wave generator output starts with the point at which the wave generator was last stopped (as if the wave output had been paused). Note that bit 14 simply specifies a start option and must always be combined with one of the start modes specified by bit 0 (0x1 or 1), bit 1 (0x2 or 2) and bit 2 (0x4 or 4).
See the WGO command (p. 221) for more information.
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.

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

- Wave table content (waveform defined with WAV (p. 209))
- Output trigger settings (TWS (p. 201))

Nevertheless, you can keep waveform(s) and trigger settings using the macro feature. The macro feature allows defining command sequences and storing them permanently in non-volatile memory in the controller. This way you can define waveforms and output trigger settings and save them permanently to the E-517. See "Working with Controller Macros" (p. 113) and "Scanning Examples with Wave Generator and Macros" (p. 104) for more information.
The following settings can be saved with WPA (p. 229) to non-volatile memory, where they become the power-on defaults:

- Wave offset (set with WOS (p. 228) or WAV with CFG wave type, or set automatically when running the wave generator with bit 8 ("start at the endpoint"))
- Wave generator table rate (set with WTR (p. 231) or WAV with CFG wave type)
- Number of cycles for wave generator output (set with WGC (p. 220))
- "Generator Trigger" mode selection made with CTO (p. 142)
- Data recorder configuration made with DRC (p. 150)

The different software interfaces provided for the E-517 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 (p. 209) and WGO (p. 221). If using the wave generator with the GCS DLL, PIMikroMove™ or LabView, read the descriptions in the associated software manual first.

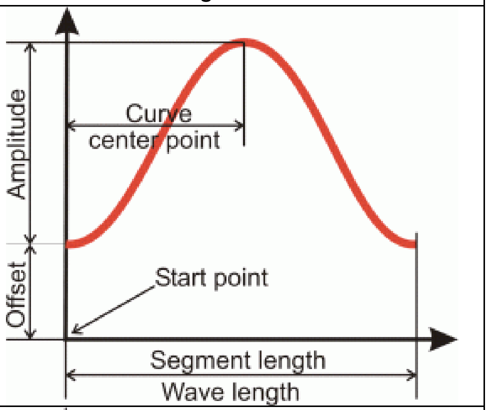
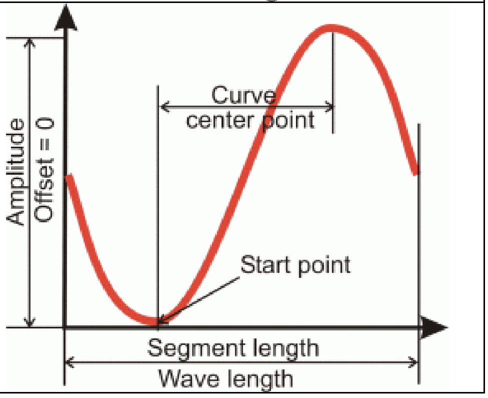
8.2 Wave Generator Examples

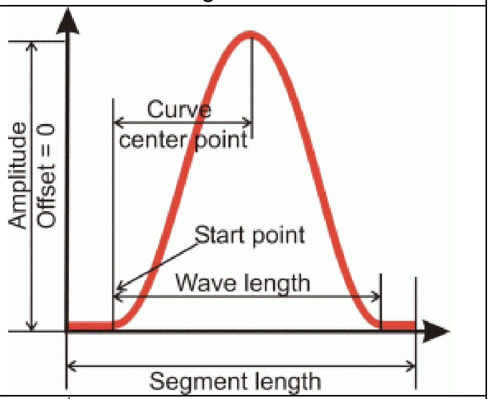
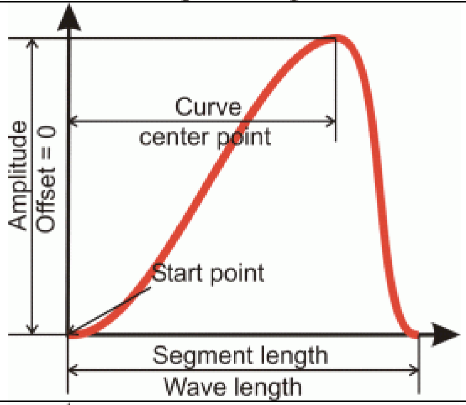
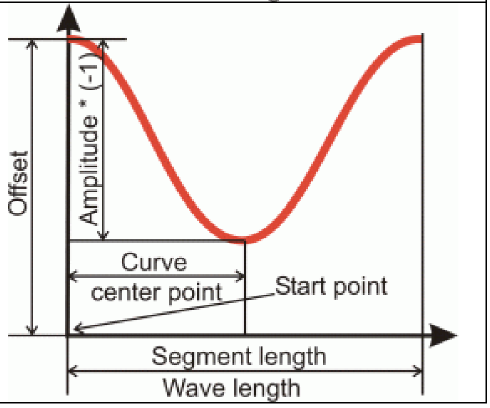
The following examples can be reproduced using the command entry facilities of PIMikroMove™ or PI Terminal. Note that it might be necessary to adapt them to your hardware configuration.

8.2.1 Defining Waveforms

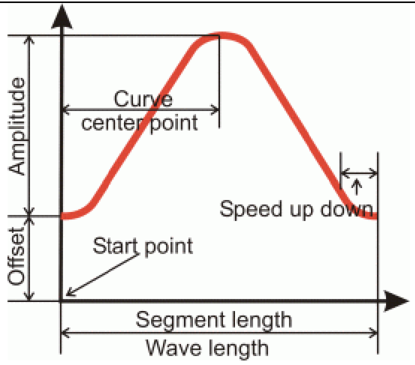
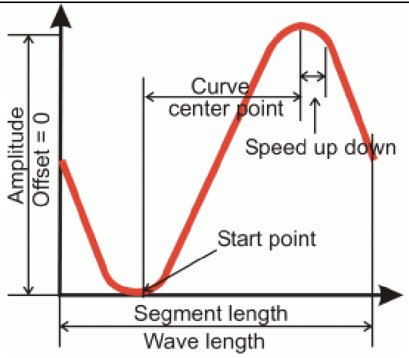
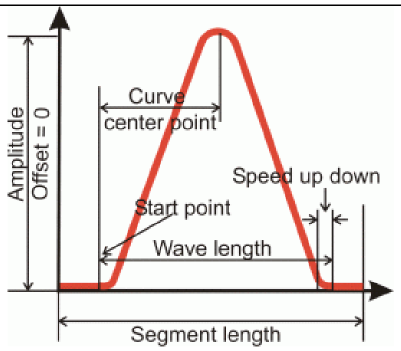
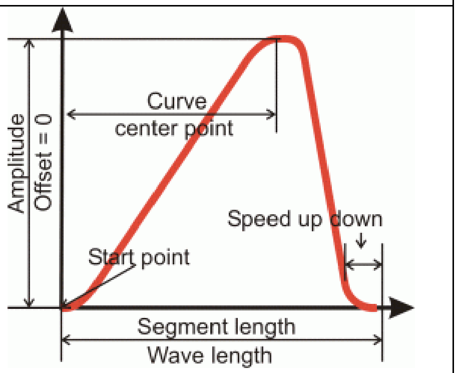
Examples for how to define waveform segments for the wave tables, based on predefined curve shapes (each WAV command defines a waveform segment which either replaces or is appended to the waveform in the specified wave table):

Inverted Cosine Curves

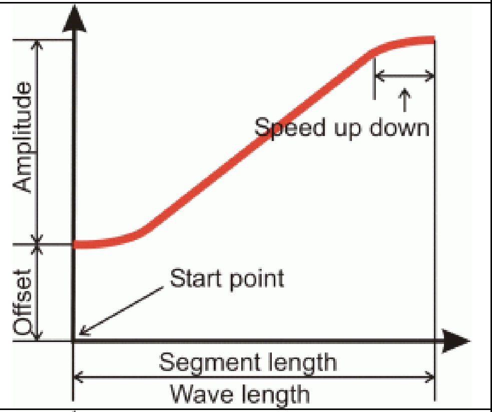
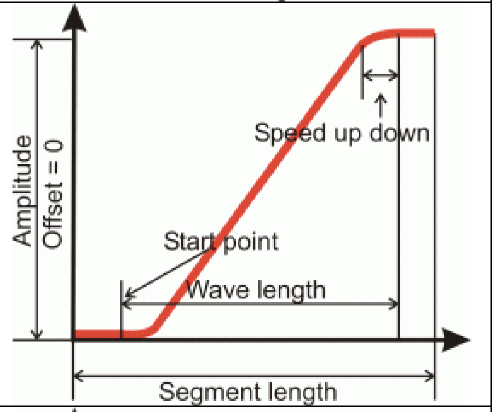
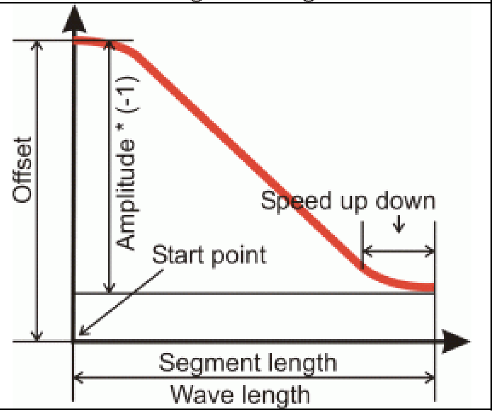
WAV command	Comments	Waveform Segment
WAV 2 X SIN_P 2000 20 10 2000 0 1000 <WaveTableID> = 2 <AppendWave> = X <WaveType> = SIN_P <SegLength> = 2000 <Amp> = 20 <Offset> = 10 <WaveLength> = 2000 <StartPoint> = 0 <CurveCenterPoint> = 1000	The previous contents of the wave table are overwritten by the new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

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> = 110 <CurveCenterPoint> = 900	The defined segment will be appended to the existing wave table contents, symmetric curve	 Amplitude Offset = 0 Curve center point Start point Wave length Segment length
WAV 3 X SIN_P 4000 20 0 4000 0 3100 <WaveTableID> = 2 <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	 Amplitude Offset = 0 Curve center point Start point Segment length Wave length
WAV 1 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	 Offset Amplitude * (-1) Curve center point Start point Segment length Wave length

Ramp Curves

WAV command	Comments	Waveform Segment
WAV 4 X RAMP 2000 20 10 2000 0 100 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	 Amplitude Offset Curve center point Speed up down Start point Segment length Wave length
WAV 4 X RAMP 2000 35 0 2000 499 100 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	 Amplitude Offset = 0 Curve center point Speed up down Start point Segment length Wave length
WAV 5 X RAMP 2000 15 0 1800 120 50 900 <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	 Amplitude Offset = 0 Curve center point Speed up down Start point Wave length Segment length
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	 Amplitude Offset = 0 Curve center point Speed up down Start point Segment length Wave length

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!)	 Amplitude Offset Speed up down Start point Segment length Wave length
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	 Amplitude Offset = 0 Speed up down Start point Wave length Segment length
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-amplitud e curve	 Offset Amplitude * (-1) Speed up down Start point Segment length Wave length

8.2.2 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 to be clocked by servo cycles in this example, see WGO (p. 221) for details). The E-517 has a servo update time of 40 µs.
`WTR?`	Ask for the current wave table rate values, 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) 0.000040 s * 1 * 2000 = 0.08 s
`WTR 2 3 0`	Set the wave table rate for the second wave generator to 3, tripling the duration of one wave generator output cycle (each wave table point will now "occupy" 3 servo cycles). Note that the E-517 does not support any interpolation to smooth the output. Duration of one output cycle will now be: 0.000040 s * 3 * 2000 = 0.24 s

8.2.3 Trigger Output Synchronized with Wave Generator

Using the digital output lines DIO_O1 to DIO_O3 of the E-517, it is possible to trigger external devices. See "Digital I/O Socket" (p. 267) for the availability of the lines (pinout) and "Configuring Trigger Output" (p. 77) for trigger applications without wave generator usage. There is a one-to-one assignment of the digital output lines to the wave generators, i.e. all trigger output synchronized with a certain wave generator is to be done via the digital output line dedicated to that wave generator.

In the example below, trigger pulses are to be generated synchronized with the output of the second wave generator, i.e. the second digital output line DIO_O2 must be configured:

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 which belongs to the second wave generator. 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 2 500 1 2 1500 1 2 1900 1 2 2000 1`	Set trigger actions for the digital output line DIO_O2 (identifier is 2): 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). You can define at most 10 trigger points per command line.
`CTO 2 3 4`	The digital output line DIO_O2 is set to "Generator Trigger" mode.
`WGO 2 1`	Start output of the second Wave Generator immediately (synchronized by servo cycle). Now the trigger output action on the DIO_O2 line will take place as specified with TWS.
`WGO 2 0`	Stop output of the second Wave Generator and hence also the trigger output.

The trigger output is now to be limited to a certain number of wave generator output cycles, and the pulse width is to be changed. In addition to the trigger points defined with TWS, pulses are also to be output every time the axis reaches the amplitude limit.

Command String to Send	Action Performed
`SPA 2 0x18000100 4`	The trigger cycles are to be output on the DIO_O2 line only for the first four wave generator output cycles.
`SPA 2 0x0E000900 50`	The pulse width of the trigger output on the DIO_O2 line is set to 50 µs.
`WPA 100`	The trigger output configuration is saved as power-on default (except of the TWS settings).
`WGO 2 33`	Start output of the second Wave Generator immediately (synchronized by servo cycle) with the "trigger output when amplitude limit is reached" start option (WGO bits 1 and 5 are set: 1+32=33) Now the trigger output action on the DIO_O2 line will take place as specified with TWS and WGO.

WGO details:

The WGO command (p. 221) offers the following start options for digital output ("Generator Trigger" mode must be enabled with CTO):

- bit 3 (0x8 or 8):
  synchronized trigger pulse is output on the corresponding digital output line when the wave generator outputs a new data point

- bit 4 (0x10 or 16):
  synchronized trigger pulse is output on the corresponding digital output line when the axis finishes each period (A_{SL} in the figure below, is the curve amplitude set with WAV during the waveform definition)

- bit 5 (0x20 or 32):
  synchronized trigger pulse is output on the corresponding digital output line when the axis reaches the amplitude limit (A_{SF} in the figure below, is the field limit set with the CFG wave type of the WAV command, parameter <L>)

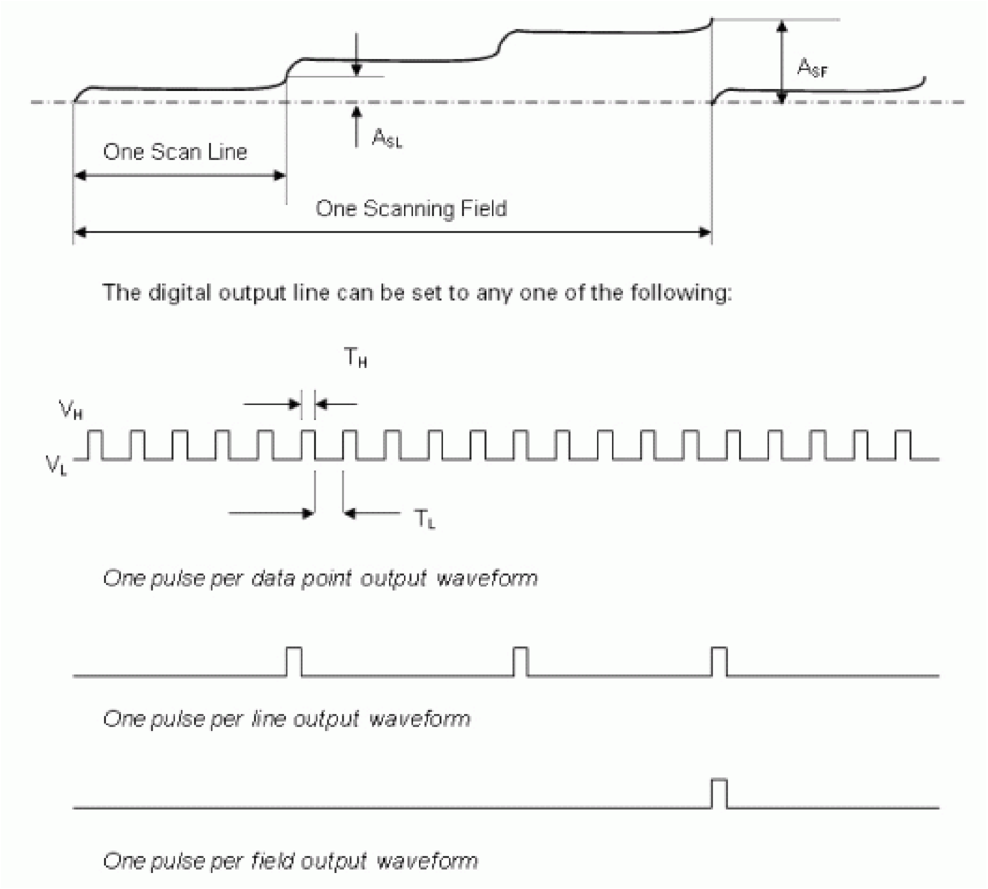

A_SF
A_SL
One Scan Line
One Scanning Field
The digital output line can be set to any one of the following:
T_H
V_H
V_L
T_L
One pulse per data point output waveform
One pulse per line output waveform
One pulse per field output waveform

Figure 17: Digital trigger output synchronized with wave generator output; top down: wave generator output, trigger output caused by bit 3, 4 and 5

PARAMETERS	MAX	TYPE	MIN
VH, voltage of logic high	-	5.0 V	-
VL, voltage of logic low	-	0 V	-
TH, Time of pulse high	150 µs	-	10 µs
TL, Time of pulse low	-	-	-

8.2.4 Wave Generator Started by Trigger Input

Using the digital input lines of the E-517, it is possible to apply start/stop signals for the wave generator output. See the pinout description of the Digital In/Out socket (p. 267) for the availability of the lines and "Using Trigger Input and Output" for an overview. There is a one-to-one assignment of the digital input lines to the wave generators, i.e. all trigger actions for one wave generator are to be done via the digital input line dedicated to that wave generator.

The input trigger configuration is made via the WGO command (p. 221) which offers the following start modes and options for use of digital input signals:

- bit 1 (0x2 or 2): start mode
  Wave generator output is triggered by external signal and synchronized by servo cycles (internal clock). The wave generator runs as long as the signal is HIGH and is paused as long as the signal is LOW.

- bit 2 (0x4 or 4), start mode
  Wave generator output is triggered and synchronized by an external signal. The wave generator outputs one waveform point each time the signal is HIGH and is paused as long as the signal is LOW (with wave table rate = 1). If the wave table rate is larger than 1, a point is output for the corresponding number of HIGH pulses. See also WTR (p. 231).

- bit 12 (0x1000 or 4096), start option
  The wave generator is started the first time the external signal is HIGH and runs continuously even if the signal becomes LOW.

- bit 13 (0x2000 or 8192), start option
  The wave generator is stopped when the external signal is HIGH. Further trigger pulses have no effect.

Bit 12 and 13 specify start options which are not effective on their own. To start the wave generator, you have to combine them with bit 0 ("start wave generator output immediately" start mode). Do not combine bit 12 and 13 with the start modes of bit 1 or 2.

If the wave generator output is started with bit 1 or bit 2 set while 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 is completed, irrespective of any further trigger pulses.

In the following example, the first wave generator is to be started by the first trigger pulse and stopped by the second trigger pulse.

Command String to Send	Action Performed
`WAV 1 X SIN_P 2000 20 10 2000 0 1000`	Define a sine waveform for Wave Table 1, the segment length and hence the number of points in the wave table is 2000
`WGO 1 12289`	Start and stop output of Wave Generator 1 triggered by external signal. To provide the external signal, the digital input line DIO_I1 must be used. Details: bit 12 and 13 are to be set on, contributing values of 4096 and 8192 (decimal) to the <StartMode> argument of the WGO command. Because bits 12 and 13 are start options which do not actually start the wave generator output, bit 0 as "start mode" must be set in addition, contributing 1. The resulting (decimal) <StartMode> value is 12289. With the <StartMode> value given in hexadecimal format, the command would be WGO 1 0x3001

WGO bit 1 details:

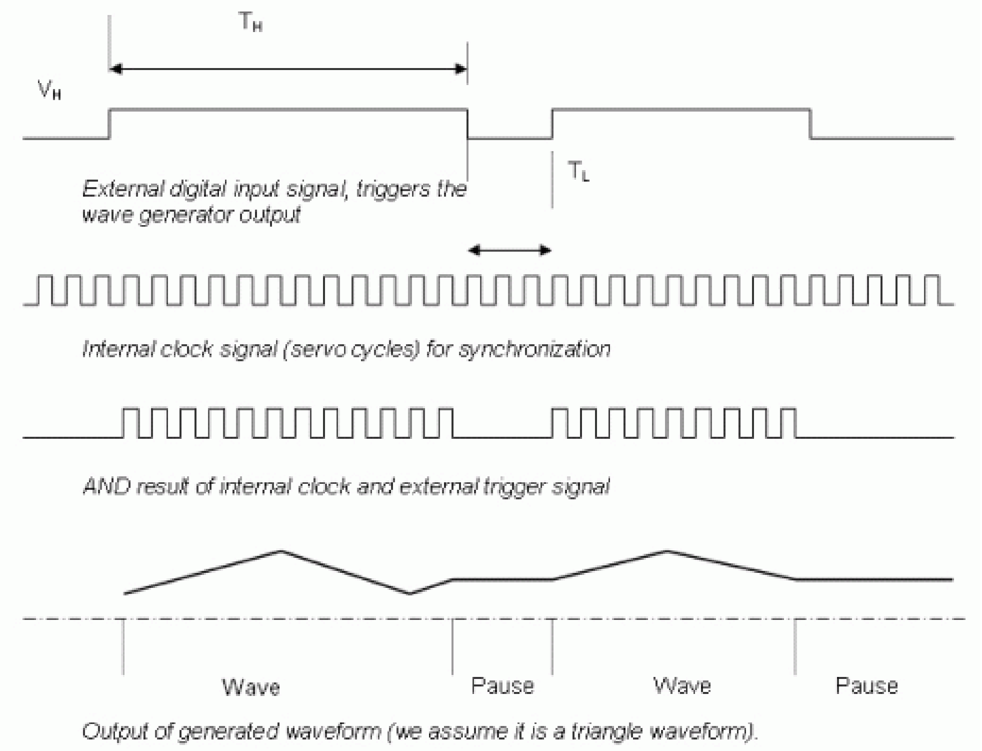

T_H
V_H
External digital input signal, triggers the wave generator output
T_L
Internal clock signal (servo cycles) for synchronization
AND result of internal clock and external trigger signal
Wave
Pause
Wave
Pause
Output of generated waveform (we assume it is a triangle waveform).

Figure 18: Wave generator started by external signal and synchronized by servo cycles

PARAMETERS	MAX	MIN
VH, Voltage of logic high	5.5 V	2.4 V
VL, Voltage of logic low	0.5 V	-0.5 V
TH, Time of pulse high	-	40 µs
TL, Time of pulse low	-	-

WGO bit 2 details:

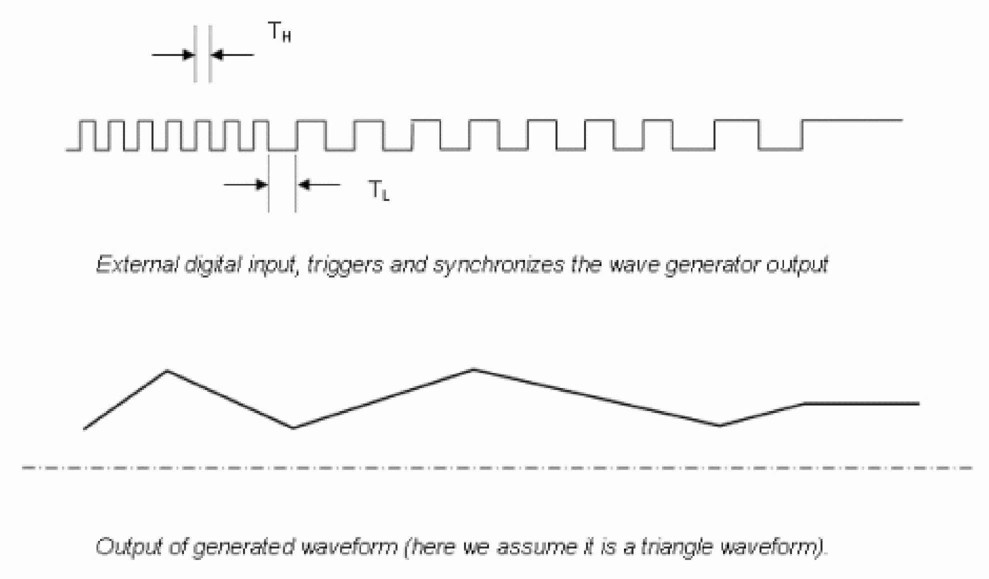

T_H
T_L
External digital input, triggers and synchronizes the wave generator output
Output of generated waveform (here we assume it is a triangle waveform).

Figure 19: Wave generator output triggered and synchronized by external signal

PARAMETERS	MAX	MIN
VH, Voltage of logic high	5.5 V	2.4 V
VL, Voltage of logic low	0.5 V	-0.5 V
TH, Time of pulse high	-	10 µs
TL, Time of pulse low	-	10 µs

The frequency of the digital input signal must not exceed 25 kHz.

8.2.5 Scanning Examples with Wave Generator and Macros

Using the macro feature you can save the command sequences required for scan motions permanently to the E-517. See "Working with Controller Macros" (p. 113) for details regarding macros.

NOTE

PIMikroMove™ provides the *PI Wave Generator Tool* with which you can create waveforms and start the wave generator output without the need to learn the corresponding commands. In the *Log* window of PIMikroMove™, you can not only monitor the commands which are sent to the controller when you use the controls of the *PI Wave Generator Tool* but also save them to a text file. Furthermore, PIMikroMove™ offers a comfortable macro editor on the *Controller macros* tab card.

Using these features, you can easily create macros like those listed below. See the PIMikroMove™ manual for more information.

Unidirectional XY-SCAN

In this example a simple XY-scan is to be performed.

An XY-scan involves the motion shown below. Assume that axis A is parallel to the X-direction and axis B is parallel to the Y-direction. The forward X-motion consists of multiple small steps. A trigger signal is output after each X-step (numbered dots in the figure below).

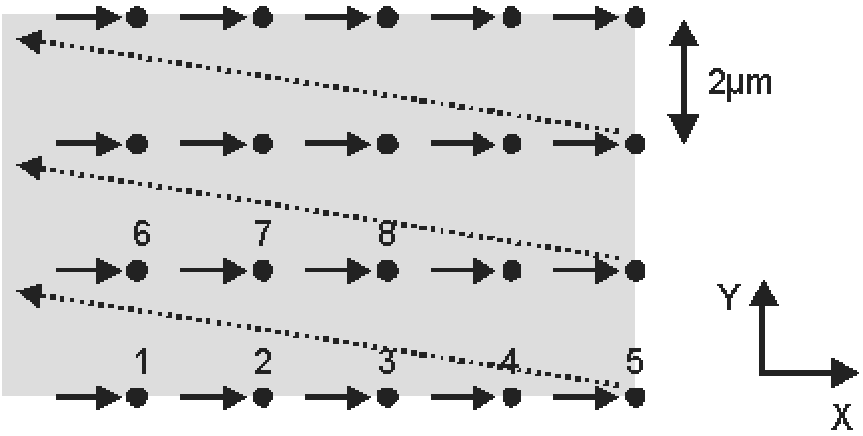


Figure 20: Unidirectional XY-scan

The correlation between the two motions is shown in the graphs below, which plot position against time for axis A and B respectively.

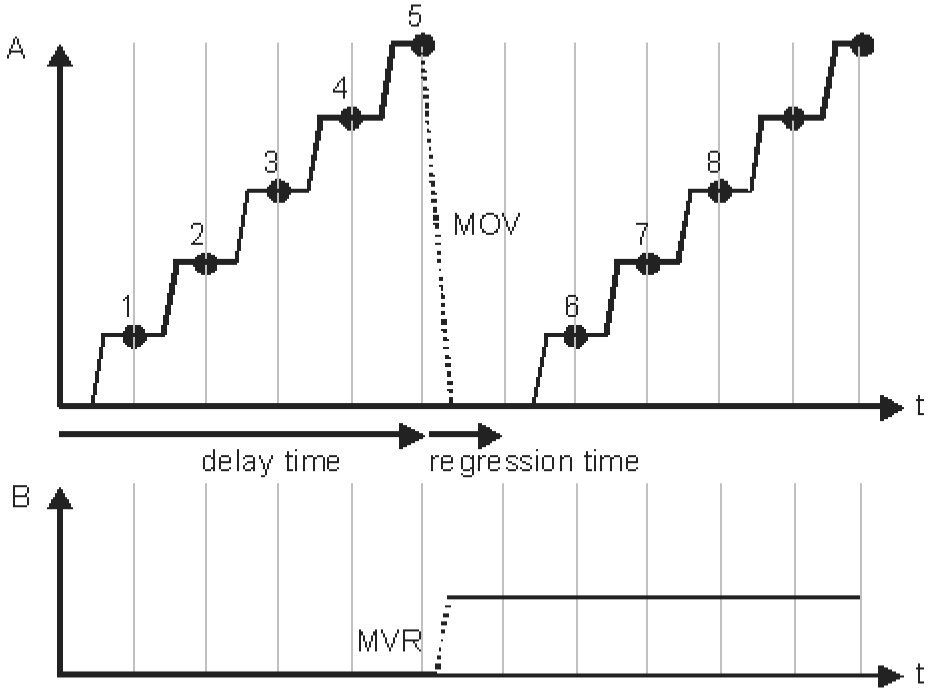


Figure 21: Position over time for axis A and B

The regression is much longer than the small X-steps, so extra time is required (short arrow under MOV command in the upper graph).

In the X-direction we want to return to the starting point (left edge), whereas in Y-direction we continue making equal steps (here: 2 µm) away from the bottom edge and so use a relative motion command (MVR B 2).

This XY-scan can be carried out with a command sequence like the one listed below. Write this command sequence to a macro named "unidir":

```
MAC BEG unidir
WGO 1 0
WAV 1 X POL 3846 3846 0 2 0
WAV 1 X POL 0 3846 0 0 0
WAV 1 X CFG 7692 1 0 5 2 10
WGO 1 17
DEL 10000
MVR B 2
DEL 100
MAC END
```

NOTE

The "Field Limit" parameter set with the CFG wave type in the WAV command defines the scan area border. This value has large-scale consequences because the controller automatically returns to the bottom line when reaching the upper border.

In the example above, the field limit is 10 ((WAV 1 X CFG 7692 1 0 5 2 10).

The macro listed above starts the wave generator output with the option to output a trigger signal after each axis A step (WGO bit 4 is set). A delay of 10 seconds (delay time) is required before a relative step in Y-direction is done. The wave generator automatically returns to its point of origin. Another delay of 100 milliseconds allows the system to settle down (regression time).

We need another macro which initializes the E-517 and then calls the scanline macro. Create the following macro named "scan":

```
MAC BEG scan
ONL 1 1 2 1
SVO A 1 B 1
VEL A 100 B 100
MOV A 0 B 0
WGC 1 5
MAC NSTART UNIDIR 5
MAC END
```

Run the macro with:

```
MAC START SCAN
```

Watch the motion on the controller display. After 5 steps of 2 µm, axis A returns to 0.00 µm whereas the position of axis B is increased by 2 µm. If you decide to change the starting point of the scan area (lower left corner) you have to adjust the MOV A 0 B 0 command in the SCAN macro as well as the offset parameter in the wave definitions.

Bidirectional XY-Scan

If measurements can be done in back-and-forth motion the scan can be performed more rapidly because there is just a small Y-step between the X-scanlines. But as can be seen in the graph below the backward motion (falling edge) differs from the forward motion (rising edge). This is why a macro is required that toggles between the two waveforms.

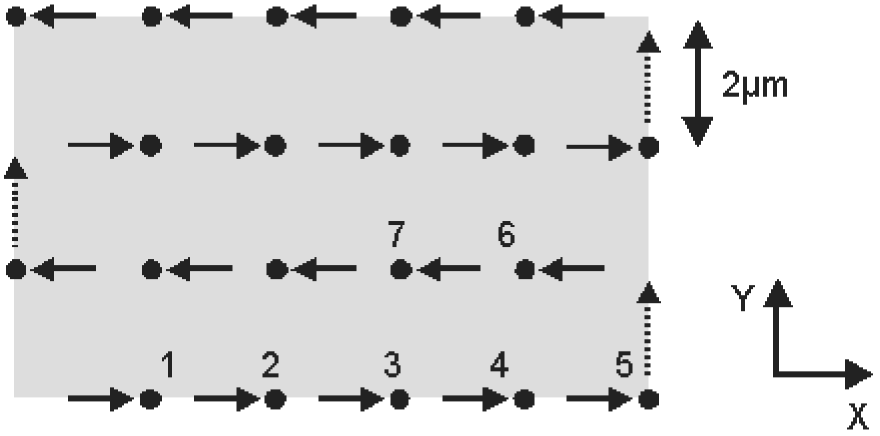


Figure 22: Bidirectional XY-scan

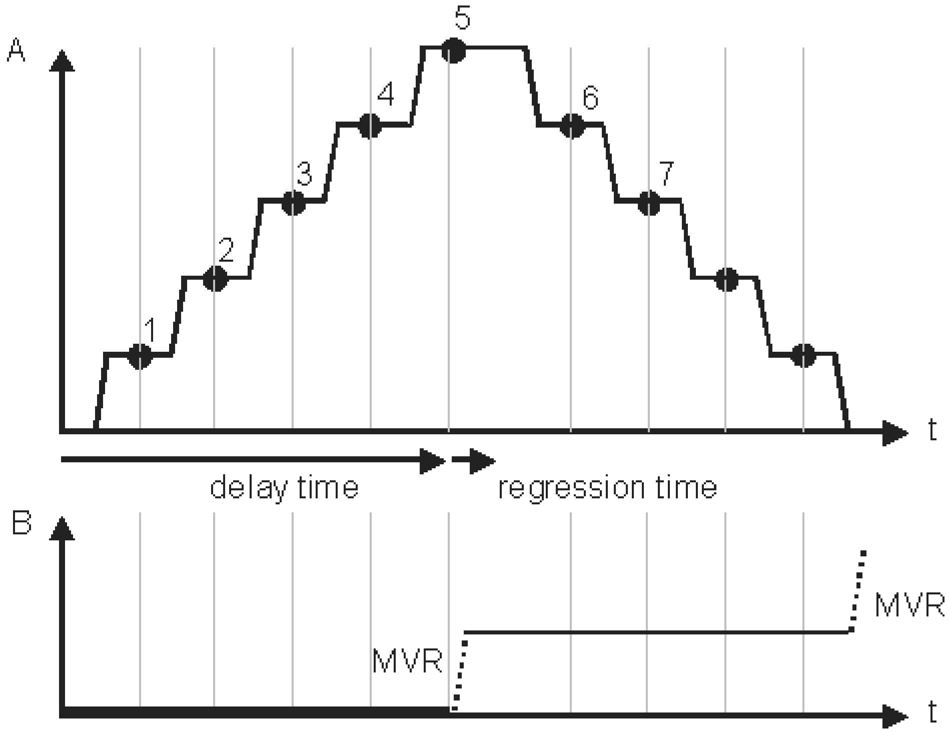


Figure 23: Position over time for axis A and B

This XY-scan can be carried out with a command sequence like the one listed below. Write this command sequence to a macro named "bidir":

```
MAC BEG bidir
WGO 1 0
WAV 1 X POL 3846 3846 0 2 0
WAV 1 X POL 0 3846 0 0 0
WAV 1 CFG 7692 1 0 5 2 10
WGO 1 17
DEL 10000
MVR B 2
```

Forward scan

```
WGO 1 0
WAV 1 X POL 3846 3846 0 8 0
WAV 1 X POL 0 3846 0 10 0
WAV 1 X CFG 7692 1 0 5 -2 10
WGO 1 17
DEL 10000
MVR B 2
```

Backward scan

```
MAC END
```

You can abstain from regression time because the step in Y-direction is of the same distance as the up/down scan line, and the trigger signal does not come until the end of it.

Modify the last line of the SCAN macro

```
MAC NSTART BIDIR 5
```

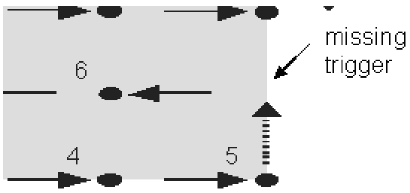
and run the macro with

```
MAC START SCAN
```

You should recognize the bidirectional motion on the E-517 display. After 5 axis A steps of 2 µm, axis B position is increased by 2 µm and axis A then steps down to the starting position, where axis B position is increased again. If you decide to change the starting point of the scan area (lower left corner) you have to adjust the MOV A 0 B 0 command in the SCAN macro as well as the offset parameter in the wave definitions.

Improved Bidirectional XY-Scan

With WGO bit 4 set as start option (WGO 1 17), the E-517 always outputs a trigger signal at the end of a step, line or area. In the previous bidirectional XY-scan (macro "bidir"), the end point of a scan line has the same X-position as the start point of the succeeding scan line so there is always one trigger point missing.



To achieve a scan area with no missing trigger points the axis can move onward one step-width before scanning the next line. This is done by increasing the scanline field height and adjusting its offset.

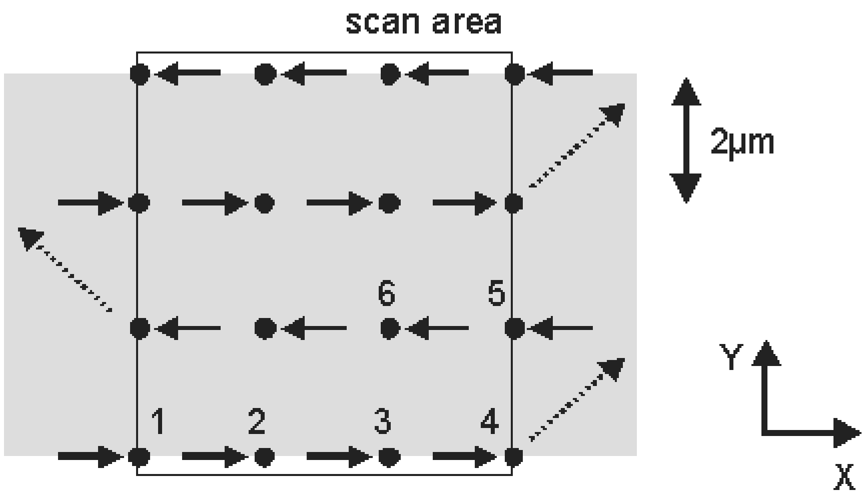


Figure 24: Improved bidirectional XY-scan

This improved bidirectional XY-scan can be carried out with a command sequence like the one listed below. The extra step in the scan area requires increased delay times. Write this command sequence to a macro named "bidir2":

```
MAC BEG bidir2
WGO 1 0
WAV 1 X POL 3846 3846 0 0 0
WAV 1 X POL 0 3846 0 -2 0
WAV 1 X CFG 7692 1 0 5 2 12
WGO 1 17
DEL 12000
MVR B 2
```

Forward scan

```
WGO 1 0
WAV 1 X POL 3846 3846 0 0 0
WAV 1 X POL 0 3846 0 -2 0
WAV 1 X CFG 7692 1 0 5 2 12
WGO 1 17
DEL 12000
MVR B 2
```

Backward scan

```
MAC END
```

Modify the last line of the SCAN macro

```
MAC NSTART BIDIR2 5
```

and run the macro with

```
MAC START SCAN
```

You should recognize the enlarged motion area on the E-517 display. The first trigger signal is at 0 µm and at 10 µm respectively.

8.3 Wave-Generator-Related Commands and Parameters

Command	Description	Notes
CTO (p. 142)	Set Configuration Of Trigger Output	Activates the Generator Trigger output mode which is required for the triggerline actions set with TWS and/or WGO; settings can be saved with WPA (p. 229)
DRR? (p. 152)	Get Recorded Data Values	Reads the last recorded data. Data recording is triggered by the WGO and WGR commands (among others).
GWD? (p. 154)	Get Wave Table Data	Should be used to check the waveform before the wave generator output is started.
TWC (p. 200)	Clear All Wave Related Triggers	Clears only the TWS settings, but not the CTO settings.
TWG? (p. 200)	Get Number Of Wave Generators	Number of wave generators = number of axes = number of wave tables
TWS (p. 201)	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. You can define at most 10 trigger points per command line.
TWS? (p. 203)	Get TriggerLine Action At Waveform Point	Reads the current TWS settings
WAV (p. 209)	Set Waveform Definition	A waveform must be defined before the wave generator output can be started. With the CFG wave type you can configure additional parameters for the waveform (amongst others, Wave Offset parameter, ID 0x1300010b, and Wave Generator Table Rate parameter, ID 0x13000109, in volatile memory)
WAV? (p. 219)	Get Waveform Definition	Reads the number of waveform points currently written to the wave table
WCL (p. 219)	Clear Wave Table Data	Clears the wave table content, but not the WOS settings.
WGC (p. 220)	Set Number Of Wave Generator Cycles	If WGC is not used, the wave generator must be stopped with WGO, #24 (p. 133) or STP (p. 190); WGC sets the value of the Wave Generator Cycles parameter, ID 0x13000003, in volatile memory).
WGC? (p. 220)	Get Number Of Wave Generator Cycles	Gets the value of the Wave Generator Cycles parameter, ID 0x13000003, from volatile memory.

Command	Description	Notes
WGO (p. 221)	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? (p. 226)	Get Wave Generator Start/Stop Mode	Gets the last commanded start options, but not the activation status (use #9 instead)
WGR (p. 227)	Starts Recording in Sync with Wave Generator	Restarts data recording as long as a wave generator is running.
WMS? (p. 227)	Get Maximum Number Of Wave Table Points	Gets the value of the Wave table length of table i parameter, ID 0x13000201.
WOS (p. 228)	Set Wave Generator Output Offset	Sets the value of the Wave Offset parameter, ID 0x1300010b, in volatile memory. Can also be set with the CFG wave type of the WAV command.
WOS? (p. 229)	Get Wave Generator Output Offset	Gets the value of the Wave Offset parameter, ID 0x1300010b, from volatile memory.
WTR (p. 231)	Set Wave Generator Table Rate	Sets the value of the Wave Generator Table Rate parameter, ID 0x13000109, in volatile memory. Can also be set with the CFG wave type of the WAV command. The interpolation type set with WTR must be 0.
WTR? (p. 233)	Get Wave Generator Table Rate	Gets the value of the Wave Generator Table Rate parameter (ID 0x13000109) from volatile memory.
#9 (p. 133)	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. 84) for more information. For detailed command descriptions see "Command Reference" (p. 131). For the identifiers of the items which can be addressed with the commands see "Accessible Items and Their Identifiers" (p. 50).

Parameter ID	Command Level	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x13000003	0	Wave Generator	3	INT	Wave generator cycles
0x13000004	3	System	1	INT	Max Wave Points
0x13000109	0	Wave Generator	3	INT	Wave Generator Table Rate
0x1300010A	3	System	1	INT	Number of Wave tables

Parameter ID	Command Level	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description
0x1300010B	0	Wave Generator	3	FLOAT	Wave Offset
0x13000201	3	Wave Table	3	INT	Max Wave Points of table i

See "Controller Parameters" (p. 250) for more information regarding the controller parameters and their handling.

9 Working with Controller Macros

The macro feature allows defining command sequences and storing them permanently in non-volatile memory in the controller. Each defined macro can be called up by its own user-defined name. In addition, it is possible to define a macro that will be executed automatically every time the E-517 is started, making possible stand-alone operation without a host computer. See the subsections below and the MAC command (p. 167) description for more details and examples.

For further examples see also "Scanning Examples with Wave Generator and Macros" (p. 104).

NOTES

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

Furthermore, PIMikroMove™ offers the "Host macro" feature which makes it possible to save macros on the host PC. With the Host macro feature you can also program conditions and loops in macros which is not directly supported by the E-517. See the PIMikroMove™ manual for more information regarding Host macros.

9.1 Defining Macros

To define a macro command sequence, first activate macro recording mode with the command MAC BEG <macroname> where <macroname> is a user-settable name with a maximum of 8 characters. While in macro recording mode, commands are not executed but stored in macro storage. Recording mode is exited by the MAC END command. The maximum number of macros to be stored on the E-517 is 60.

A macro is immediately written to non-volatile memory. No additional storage procedure is required. Note that saving a macro to non-volatile memory typically takes about 1 s.

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

During macro recording no macro execution is allowed.

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

Macros can be recorded when the wave generator is running.

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

Example 1: This simple macro initializes the E-517 and performs a move forward and backward.

`MAC BEG macro1`	Start recording macro "macro1"
`ONL 1 1 2 1 3 1`	Switch all piezo channels to ONLINE mode
`SVO A 1 B 1 C 1`	Enable servo control mode for all axes.

`VCO A 1 B 1 C 1`
`DCO A 1 B 1 C 1`
	Enable velocity control mode and drift compensation mode. The E-517 now keeps track of proper motion.

`VEL A 100 B 100 C 100`	Set velocity to 100 µm/s.
`MOV A 10 B 10 C 10`	Move all axes to position 10 µm. This moves take 0.1 s.
`DEL 1000`	Delay of approx. 1000 ms.
`MOV A 0 B 0 C 0`	Move all axes to its home.
`DEL 1000`	This delay only matters if the macro is run several times in a loop.
`MAC END`	Exit the macro recording mode. The macro now is complete.

Example 2: Macro1 now is splitted in two separate macros (macro2 and macro3) because it in not necessary to initialize the E-517 every time motion is to be performed. When you start macro3 this implicates that macro2 is executed 10 times.
Write macro2:

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

Write macro3:

```
MAC BEG macro3
ONL 1 1 2 1 3 1
SVO A 1 B 1 C 1
VCO A 1 B 1 C 1
DCO A 1 B 1 C 1
VEL A 100 B 100 C 100
MAC NSTART macro2 10
MAC END
```

9.2 Starting Macro Execution

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

To run a macro multiple times, call it with MAC NSTART <macroname> n where n gives the number of times the macro is to be run.

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

If the E-517 is in OFFLINE mode, move commands from a running macro are not allowed (they provoke an error message). See "Control Modes" (p. 23) for details.

A macro can be started when the wave generator is running, but move commands from the macro are not executed and provoke an error message.

Commands sent when a macro is running will be executed not until the macro is finished or stopped. The only exceptions are #5, #6, #7, #8, #9 and #24 which are executed immediately.

When a macro is running, trackball settings can be used in parallel to command axis motion. See "Trackball Functions" (p. 22) for details.

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

Macro execution can be stopped from the command line with #24 (p. 133). The STP command (p. 190) stops macro execution only when it is part of the macro.

A running macro may not be deleted.

You can query with #8 (p. 132) if a macro is currently running on the controller.

9.3 Start-Up Macro

With MAC DEF <macroname> it is possible to set the specified macro as start-up macro. This macro will be automatically executed with the next power-on or reboot of the controller.

Example:

```
MAC BEG init
```

```
ONL 1 1 2 1 3 1
SVO A 1 B 1 C 1
VCO A 1 B 1 C 1
DCO A 1 B 1 C 1
VEL A 20 B 20 C 20
MAC END
```

This macro switches the E-517 to ONLINE mode and closed-loop operation, enables velocity control mode and drift compensation mode and sets the velocity to 20 µm/s.

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

```
MAC DEF?
```

To undo the current start-up macro selection, send

```
MAC DEF
```

i.e. omit <macroname>.

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

NOTE

To skip the execution of the start-up macro, press the trackball once immediately after you have powered on or rebooted the piezo control electronics (while "Physik Instrumente" is still shown on the display). For details about the trackball handling, see "Trackball Functions" (p. 22).

10 GCS Commands

The PI General Command Set (GCS) is supported by a wide range of PI systems. This command set is well-suited for positioning tasks with one or more axes. The command set itself is independent of the specific hardware (controller or attached stages).

Commands are used to set operating modes, initiate axis motion and to query system and motion values. Because of the variety of functions and parameters, a sequence of commands must often be transferred in order to achieve a desired system action.

You can type commands, for example, in the *Command Entry* window of PIMikroMove™, or in the PITerminal.

NOTE

All information in this section refers solely to the E-517 GCS syntax version. For information regarding the E-516 GCS syntax version see the E-516 documentation which is available for download at www.pi.ws.

10.1 Format

10.1.1 Notation

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

<...> Angle brackets indicate an argument of a command, can be an item identifier (p. 50) or a command-specific parameter

[…] Square brackets indicate an optional entry

{…} Braces 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

SP Space (ASCII char #32)

10.1.2 GCS Syntax

Except as listed below, a GCS command consists of 3 characters, e.g. CMD. The corresponding query command has a "?" appended, e.g. CMD?. Command mnemonic:

CMD ::= character1 character2 character3 [?]

Exceptions:

- Special commands, e.g. fast polling commands, consist only of one character. The 24th ASCII character e.g. is called #24. Note that these commands are not followed by a termination character (but the responses to them are).
- *IDN? (for GPIB compatibility).

The command mnemonic is not case-sensitive.

General:

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

That means the command mnemonic and all arguments (e.g. axis IDs, channel IDs, parameters, etc.) must be separated from each other by one space.

Example:
Send: MOVSP1SP10.0LF
to move Axis 1 to position 10.0 (the unit depends on the controller, can be µm or mm, for example)

More than one command mnemonic per line is not allowed. Several groups of arguments following a command mnemonic are allowed, e.g.
MOVSP1SP17.3SP2SP2.05LF
if there were 2 axes. The command line ends with the termination character (LF).

If part of a command line can not 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. For example,
RPALF
resets all parameters in volatile memory.

The <AxisID> argument is used for the logical axes of the controller. Depending on the controller, an axis could be identified with up to 16 characters—all alphanumeric characters and the underscore are allowed. See "Accessible Items and Their Identifiers" (p. 50) for the identifiers supported by the E-517.

Definitions for query commands (report commands):

CMD?[{{SP}<argument>}]LF

When all arguments are optional and are omitted, all possible values are reported. For example,
POS?
queries the position of all axes.

Reply syntax:

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

Multi-line reply syntax:
{[<argument>[{SP<argument>}]"="]<value>SP LF}
[<argument>[{SP<argument>}]"="]<value>LF   for the last line!

The command
CMD?SP<arg3>SP <arg1>SP <arg2>LF
replies in the same order:
<arg3>"="<value3>SP LF
<arg1>"="value1SP LF
<arg2>"="value2 LF

Example:
Send: TSP? 2 1
Report: 2=-1158.4405SP LF
          1=+0000.0000LF

10.1.3 Limitations for GCS Commands

More than one command mnemonic per line is not allowed.

The number of characters per line is limited to 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 (p. 153) afterwards because the number of arguments is 33.

The number of characters to use for axis identifiers is limited to 8 (SAI command (p. 181) and "Axis Name" parameter, ID 0x07000600).

10.2 Command Survey

10.2.1 E-517 GCS Commands, Alphabetical Survey

#5 (p. 131)	Request Motion Status
#6 (p. 131)	Query If Position Has Changed Since Last POS? Command
#7 (p. 132)	Request Controller Ready Status
#8 (p. 132)	Query If Macro Is Running
#9 (p. 133)	Get Wave Generator Status
#24 (p. 133)	Stop All Motion
*IDN? (p. 134)	Get Device Identification
ATC (p. 134)	Start Auto Calibration
ATC? (p. 137)	Get Auto Calibration Settings
ATS? (p. 138)	Get Auto Calibration Status
CCL (p. 139)	Set Command Level
CCL? (p. 140)	Get Command Level
CSV (p. 141)	Set GCS Syntax Version
CSV? (p. 141)	Get Current Syntax Version
CTO (p. 142)	Set Configuration Of Trigger Output
CTO? (p. 145)	Get Configuration Of Trigger Output
DCO (p. 145)	Set Drift Compensation Mode
DCO? (p. 146)	Get Drift Compensation Mode
DEL (p. 146)	Delay The Command Interpreter
DFH (p. 146)	Define Current Position As Axis Home Position

DFH? (p. 149)	Get Home Position Definition
DIO? (p. 149)	Get Digital Input Lines
DRC (p. 150)	Set Data Recorder Configuration
DRC? (p. 152)	Get Data Recorder Configuration
DRR? (p. 152)	Get Recorded Data Values
ERR? (p. 153)	Get Error Number
GOH (p. 154)	Go To Home Position
GWD? (p. 154)	Get Wave Table Data
HDR? (p. 155)	Get All Data Recorder Options
HLP? (p. 156)	Get List of Available Commands
HLT (p. 156)	Halt Motion Smoothly
HPA? (p. 157)	Get List of Available Parameters
IFC (p. 159)	Set Interface Parameters Temporary
IFC? (p. 161)	Get Current Interface Parameters
IFS (p. 163)	Set Interface Parameters As Default Values
IFS? (p. 164)	Get Interface Parameters As Default Values
IMP (p. 165)	Start Impulse And Response - Measurement
IMP? (p. 166)	Get IMP Settings
MAC (p. 167)	Call Macro Function
MAC? (p. 170)	List Macros
MOV (p. 170)	Set Target Position
MOV? (p. 171)	Get Target Position
MVR (p. 172)	Set Target Relative To Current Position
NLM (p. 173)	Set Low Position Soft Limit
NLM? (p. 174)	Get Low Position Soft Limit

ONL (p. 174)	Set Control Mode
ONL? (p. 175)	Get Control Mode
ONT? (p. 176)	Get On Target State
OVF? (p. 177)	Get Overflow State
PLM (p. 177)	Set High Position Soft Limit
PLM? (p. 178)	Get High Position Soft Limit
POS? (p. 178)	Get Real Position
RBT (p. 178)	Reboot System
RPA (p. 179)	Reset Volatile Memory Parameters
RTR (p. 180)	Set Record Table Rate
RTR? (p. 181)	Get Record Table Rate
SAI (p. 181)	Set Current Axis Identifiers
SAI? (p. 182)	Get List Of Current Axis Identifiers
SEP (p. 183)	Set Nonvolatile Memory Parameters
SEP? (p. 184)	Get Nonvolatile Memory Parameters
SPA (p. 185)	Set Temporary Memory Parameters
SPA? (p. 188)	Get Temporary Memory Parameters
SSN? (p. 189)	Get Device Serial Number
STE (p. 189)	Start Step And Response - Measurement
STE? (p. 190)	Get STE Settings
STP (p. 190)	Stop All Motion
SVA (p. 191)	Set Open-Loop Axis Value
SVA? (p. 193)	Get Open-Loop Axis Value
SVO (p. 193)	Set Servo State (Open-Loop / Closed-Loop Operation)

SVO? (p. 194)	Get Servo State (Open-Loop / Closed-Loop Operation)
SVR (p. 195)	Set Relative Open-Loop Axis Value
TAD? (p. 196)	Get ADC Value Of Input Signal
TIO? (p. 196)	Tell Digital I/O Lines
TMN? (p. 197)	Get Minimum Commandable Position
TMX? (p. 197)	Get Maximum Commandable Position
TNR? (p. 198)	Get Number Of Record Tables
TPC? (p. 198)	Get Number Of Output Signal Channels
TSC? (p. 199)	Get Number Of Input Signal Channels
TSP? (p. 199)	Get Input Signal Position Value
TVI? (p. 200)	Tell Valid Character Set For Axis Identifiers
TWC (p. 200)	Clear All Wave Related Triggers
TWG? (p. 200)	Get Number Of Wave Generators
TWS (p. 201)	Set TriggerLine Action To Waveform Point
TWS? (p. 203)	Get TriggerLine Action At Waveform Point
VCO (p. 204)	Set Velocity Control Mode
VCO? (p. 204)	Get Velocity Control Mode
VEL (p. 205)	Set Velocity
VEL? (p. 206)	Get Velocity
VER? (p. 206)	Get Versions Of Firmware And Drivers
VMA (p. 207)	Set Voltage Output High Limit
VMA? (p. 207)	Get Voltage Output High Limit
VMI (p. 208)	Set Voltage Output Low Limit
VMI? (p. 208)	Get Voltage Output Low Limit

VOL? (p. 209)	Get Voltage Of Output Signal Channel
WAV (p. 209)	Set Waveform Definition
WAV? (p. 219)	Get Waveform Definition
WCL (p. 219)	Clear Wave Table Data
WGC (p. 220)	Set Number Of Wave Generator Cycles
WGC? (p. 220)	Get Number Of Wave Generator Cycles
WGO (p. 221)	Set Wave Generator Start/Stop Mode
WGO? (p. 226)	Get Wave Generator Start/Stop Mode
WGR (p. 227)	Starts Recording In Sync With Wave Generator
WMS? (p. 227)	Get Maximum Number Of Wave Table Points
WOS (p. 228)	Set Wave Generator Output Offset
WOS? (p. 229)	Get Wave Generator Output Offset
WPA (p. 229)	Save Parameters To Nonvolatile Memory
WTR (p. 231)	Set Wave Generator Table Rate
WTR? (p. 233)	Get Wave Generator Table Rate

10.2.2 E-517 GCS Commands, Survey by Function

The E-517 commands are listed below according to the following function groups:

- Motion Commands
- Macros
- Wave Generator
- Data Recorder
- Interface
- Trigger / Digital I/O Lines
- System Setup / Calibration

- Status / Signal Values
- Device Information
- Miscellaneous

Note that some commands are assigned to multiple groups.

Motion Commands

#24 (p. 133)	Stop All Motion
DRR? (p. 152)	Get Recorded Data Values
GOH (p. 154)	Go To Home Position
HLT (p. 156)	Halt Motion Smoothly
IMP (p. 165)	Start Impulse And Response - Measurement
IMP? (p. 166)	Get IMP Settings
MOV (p. 170)	Set Target Position
MOV? (p. 171)	Get Target Position
MVR (p. 172)	Set Target Relative To Current Position
STE (p. 189)	Start Step And Response - Measurement
STE? (p. 190)	Get STE Settings
STP (p. 190)	Stop All Motion
SVA (p. 191)	Set Open-Loop Axis Value
SVA? (p. 193)	Get Open-Loop Axis Value
SVR (p. 195)	Set Relative Open-Loop Axis Value

Macros

#8 (p. 132)	Query If Macro Is Running	
#24 (p. 133)	Stop All Motion	Stops macro execution
MAC (p. 167)	Call Macro Function	

MAC? (p. 170)	List Macros	
STP (p. 190)	Stop All Motion	Stops macro execution when used inside the macro

Wave Generator

#9 (p. 133)	Get Wave Generator Status	
#24 (p. 133)	Stop All Motion	Stops wave generator output
CTO (p. 142)	Set Configuration Of Trigger Output	
DRR? (p. 152)	Get Recorded Data Values	
GWD? (p. 154)	Get Wave Table Data	
STP (p. 190)	Stop All Motion	Stops wave generator output
TWC (p. 200)	Clear All Wave Related Triggers	
TWG? (p. 200)	Get Number Of Wave Generators	
TWS (p. 201)	Set TriggerLine Action To Waveform Point	
TWS? (p. 203)	Get TriggerLine Action At Waveform Point	
WAV (p. 209)	Set Waveform Definition	
WAV? (p. 219)	Get Waveform Definition	
WCL (p. 219)	Clear Wave Table Data	
WGC (p. 220)	Set Number Of Wave Generator Cycles	
WGC? (p. 220)	Get Number Of Wave Generator Cycles	
WGO (p. 221)	Set Wave Generator Start/Stop Mode	Starts wave generator output
WGO? (p. 226)	Get Wave Generator Start/Stop Mode	
WGR (p. 227)	Starts Recording In Sync With Wave Generator	
WMS? (p. 227)	Get Maximum Number Of Wave Table Points	
WOS (p. 228)	Set Wave Generator Output Offset	
WOS? (p. 229)	Get Wave Generator Output Offset	

WTR (p. 231)	Set Wave Generator Table Rate
WTR? (p. 233)	Get Wave Generator Table Rate

Data Recorder

DRC (p. 150)	Set Data Recorder Configuration
DRC? (p. 152)	Get Data Recorder Configuration
DRR? (p. 152)	Get Recorded Data Values
HDR? (p. 155)	Get All Data Recorder Options
IMP (p. 165)	Start Impulse And Response – Measurement Starts recording
RTR (p. 180)	Set Record Table Rate
RTR? (p. 181)	Get Record Table Rate
STE (p. 189)	Start Step And Response – Measurement Starts recording
TNR? (p. 198)	Get Number Of Record Tables
WGO (p. 221)	Set Wave Generator Start/Stop Mode Starts recording
WGR (p. 227)	Starts Recording In Sync With Wave Generator

Interface

IFC (p. 159)	Set Interface Parameters Temporary	
IFC? (p. 161)	Get Current Interface Parameters	
IFS (p. 163)	Set Interface Parameters As Default Values	
IFS? (p. 164)	Get Interface Parameters As Default Values	
ONL (p. 174)	Set Control Mode	ONLINE or OFFLINE operation
ONL? (p. 175)	Get Control Mode	

Trigger / Digital I/O Lines

CTO (p. 142)	Set Configuration Of Trigger Output

CTO? (p. 145)	Get Configuration Of Trigger Output
DIO? (p. 149)	Get Digital Input Lines
TIO? (p. 196)	Tell Digital I/O Lines
TWC (p. 200)	Clear All Wave Related Triggers
TWS (p. 201)	Set TriggerLine Action To Waveform Point
TWS? (p. 203)	Get TriggerLine Action At Waveform Point
WGO (p. 221)	Set Wave Generator Start/Stop Mode Selects trigger input and output options for the wave generator

System Setup / Calibration

ATC (p. 134)	Start Auto Calibration
ATC? (p. 137)	Get Auto Calibration Settings
ATS? (p. 138)	Get Auto Calibration Status
CCL (p. 139)	Set Command Level
CCL? (p. 140)	Get Command Level
CSV (p. 141)	Set GCS Syntax Version
CSV? (p. 141)	Get Current Syntax Version
DCO (p. 145)	Set Drift Compensation Mode
DCO? (p. 146)	Get Drift Compensation Mode
DFH (p. 146)	Define Current Position As Axis Home Position
DFH? (p. 149)	Get Home Position Definition
NLM (p. 173)	Set Low Position Soft Limit
NLM? (p. 174)	Get Low Position Soft Limit
PLM (p. 177)	Set High Position Soft Limit
PLM? (p. 178)	Get High Position Soft Limit
RPA (p. 179)	Reset Volatile Memory Parameters

SAI (p. 181)	Set Current Axis Identifiers
SAI? (p. 182)	Get List Of Current Axis Identifiers
SEP (p. 183)	Set Nonvolatile Memory Parameters
SEP? (p. 184)	Get Nonvolatile Memory Parameters
SPA (p. 185)	Set Temporary Memory Parameters
SPA? (p. 188)	Get Temporary Memory Parameters
SVO (p. 193)	Set Servo State (Open-Loop / Closed-Loop Operation)
SVO? (p. 194)	Get Servo State (Open-Loop / Closed-Loop Operation)
VCO (p. 204)	Set Velocity Control Mode
VCO? (p. 204)	Get Velocity Control Mode
VEL (p. 205)	Set Velocity
VEL? (p. 206)	Get Velocity
VMA (p. 207)	Set Voltage Output High Limit
VMA? (p. 207)	Get Voltage Output High Limit
VMI (p. 208)	Set Voltage Output Low Limit
VMI? (p. 208)	Get Voltage Output Low Limit
WPA (p. 229)	Save Parameters To Nonvolatile Memory

Status / Signal Values

#5 (p. 131)	Request Motion Status
#6 (p. 131)	Query If Position Has Changed Since Last POS? Command
#7 (p. 132)	Request Controller Ready Status
#8 (p. 132)	Query If Macro Is Running
#9 (p. 133)	Get Wave Generator Status

ERR? (p. 153)	Get Error Number
ONT? (p. 176)	Get On Target State
OVF? (p. 177)	Get Overflow State
POS? (p. 178)	Get Real Position
TAD? (p. 196)	Get ADC Value Of Input Signal
TSP? (p. 199)	Get Input Signal Position Value
VOL? (p. 209)	Get Voltage Of Output Signal Channel

Device Information

*IDN? (p. 134)	Get Device Identification
HLP? (p. 156)	Get List of Available Commands
HPA? (p. 157)	Get List of Available Parameters
SSN? (p. 189)	Get Device Serial Number
TIO? (p. 196)	Tell Digital I/O Lines
TMN? (p. 197)	Get Minimum Commandable Position
TMX? (p. 197)	Get Maximum Commandable Position
TNR? (p. 198)	Get Number Of Record Tables
TPC? (p. 198)	Get Number Of Output Signal Channels
TSC? (p. 199)	Get Number Of Input Signal Channels
TVI? (p. 200)	Tell Valid Character Set For Axis Identifiers
TWG? (p. 200)	Get Number Of Wave Generators
VER? (p. 206)	Get Versions Of Firmware And Drivers
WMS? (p. 227)	Get Maximum Number Of Wave Table Points

Miscellaneous

DEL (p. 146)	Delay The Command Interpreter
RBT (p. 178)	Reboot System

10.3 Command Reference (alphabetical)

#5 (Request Motion Status)

Description: Requests motion status of the axes.

Format: #5 (single ASCII character number 5)

Arguments: none

Response: The answer <uint> is bit-mapped and returned as the hexadecimal sum of the following codes:

1=first axis is moving
2=second axis is moving
4=third axis is moving
...

Examples: 0 indicates motion of all axes complete
3 indicates that the first and the second axis are moving

Notes: The #5 response is influenced by the current control and servo mode as follows:

- OFFLINE mode: response is always 0
- Open-loop operation: response is always 0
- Closed-loop operation, ONLINE mode and axis on target: response is 0
- Closed-loop operation, ONLINE mode and axis not on target: response is the non-zero code indicating that the axis is moving

#6 (Query If Position Has Changed Since Last POS?)

Description: Requests if the positions of the axes have changed since the last POS? command.

Format: #6 (single ASCII character number 6)

Arguments: none

Response: The answer <uint> is bit-mapped and returned as the hexadecimal sum of the following codes:

1=position of first axis has changed
2=position of second axis has changed
4=position of third axis has changed
...

Examples: 0 indicates position of all axes has not changed
3 indicates that the positions of the first and the second axis have changed

Notes: #6 can be used in open-loop and closed-loop operation.

The query considers only motion caused by control sources (e.g. move commands), but ignores position changes caused by amplifier noise.

#7 (Request Controller Ready Status)

Description: Asks controller for ready status (tests if controller is ready to perform a new command).

Note: Use #5 (p. 131) instead of #7 to verify if motion has finished.

Format: #7 (single ASCII character number 7)

Arguments: none

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

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

Troubleshooting: The response characters may appear differently in non-Western character sets or other operating systems. They may be indistinguishable on the controller screen.

#8 (Query If Macro Is Running)

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

Format: #8 (single ASCII character number 8)

Arguments: none

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

#9 (Get Wave Generator Status)

Description: Requests the status of the wave generator(s).

The #9 single-character command (p. 133) 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. With WGO? you can ask for the last-commanded wave generator start options (WGO settings (p. 221)).

Format: #9 (single ASCII character number 9)

Arguments: none

Response: The answer <uint> is bit-mapped and returned as the hexadecimal sum of the following codes:
1 = Wave Generator 1 is running,
2 = Wave Generator 2 is running,
4 = Wave Generator 3 is running, etc.

Examples: 0 indicates that no wave generator is running
5 indicates that wave generators 1 and 3 are running

#24 (Stop All Motion)

Description: Stops all motion abruptly. For details see the notes below.

Sets error code to 10.

This command is identical in function to STP (p. 190), but only one character must be send via the interface. Therefore #24 can also be used while the controller is performing time-consuming tasks.

Format: #24 (ASCII character 24)

Arguments: none

Response: none

Notes: #24 stops motion of all axes caused by move commands (MOV (p. 170), MVR (p. 172), GOH (p. 154), SVA (p. 191), SVR (p. 195)). Furthermore, it stops macros (MAC (p. 167)) and wave generator output (WGO (p. 221)).

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

HLT (p. 156) in contrast to #24 stops motion smoothly.

*IDN? (Get Device Identification)

Description: Reports the device identity number.

Format: *IDN?

Arguments: none

Response: One-line string terminated by line feed with controller name, serial number and firmware version

Notes: For E-517, *IDN? replies something like:

Physik Instrumente, E-517, 107020627, V01.000

ATC (Start Auto Calibration)

Description: Automatic calibration. Adjusts the gain and offset for the E-517 A/D and D/A converters, for the sensor input to the P-I-controller and for the piezo monitor voltage output of the amplifier.

To make the ATC command available, switch to command level 1 (CCL command (p. 139)).

The settings to be calibrated are implemented as protected controller parameters which can only be changed by the auto calibration procedure. You can query for the parameter values with the SPA? and SEP? commands; see the <Option> list below for the parameter IDs and the item IDs to use. To save the calibrated values of those parameters to non-volatile memory, where they become the power-on defaults, use WPA (p. 229) when the ATC procedure is finished successfully.

The auto calibration procedure has the highest priority, i.e. it will overwrite the control values given by all other sources. The auto calibration procedure automatically switches between the servo modes (open-loop / closed-loop operation) and the control modes (ONLINE / OFFLINE) of the E-517. When finished, the mode settings which were active at the start of the procedure are restored. For that reason, the SERVO switches of all channels must be set to off on the piezo control electronics.

The success of the auto calibration procedure can be queried with the ATS? command (p. 138). With ATC? (p. 137) you can ask for the options used with the latest auto calibration procedure.

The auto calibration procedure can take several seconds. During this time, the controller is busy and only very limited able to execute or answer commands. Ask with #7 if the procedure is finished.

See "Calibration Settings" (p. 38) for more information.

Format: ATC {<ChannelID> <Option>}

Arguments <ChannelID> is one channel of the piezo control electronics, can be a sensor channel (for <Option> = 2 and 5) or a piezo channel (for <Option> = 3, 4, 6) or both (for <Option> = 1);
possible IDs are 1, 2, 3

<Option> selects the settings to be calibrated. The IDs of the parameters affected by the individual options are listed below. In addition, the item IDs are given which are to be used when asking for the parameter values with SPA? or SEP? commands.
Possible options are:
1 = Complete calibration; includes all settings affected by options 2 to 6
2 = Sensor monitor ADC (gain, ID 0x04000500, and offset, ID 0x04000600; parameter item IDs 1 to 3)
3 = DAC for control voltage (gain, ID 0x0A000020, and offset, ID 0x0A000010; parameter item IDs 1 to 3)
4 = Voltage monitor ADC (gain, ID 0x04000500, and offset, ID 0x04000600; parameter item IDs 4 to 6)
5 = Sensor input to P-I controller (E-509 module;

gain, ID 0x04000700, and offset, ID 0x04000800; parameter item IDs 1 to 3)
6 = Piezo monitor voltage output of amplifier (e.g. E-503, E-505 or E-507; gain, ID 0x04000700, and offset, ID 0x04000800; parameter item IDs 4 to 6)

Response: none

Troubleshooting: ATC will fail if

- the stage is not connected
- the hardware is defective
- the calibration facilities physically present on the hardware (e.g. potentiometers) are not adjusted properly
- the SERVO switch of the channel is set to ON on the piezo control electronics
- An invalid channel identifier is used, e.g. ATZ 4 1

Notes: CAUTION: The ATC procedure will move the axis, and the motion may cover the whole travel range. Make sure that it is safe for the stage to move.

The protected parameters "Min Voltage" (ID 0x0B000007) and "Max Voltage" (ID 0x0B000008) which limit the output voltage for the piezo channels are adapted automatically if ATC with options 1 or 6 is used.

Example: New hardware modules are added to the piezo control electronics (see "Configure Axes and Channels" (p. 33) for an example). Now the auto calibration procedure is required to adjust all gain and offset parameters properly.

Send:	CCL 1 advanced
Send:	ATC 1 1 2 1 3 1
Note:	A complete auto calibration procedure ist started for all channels.
Send:	ATS?
Receive:	1 1=0 2 1=1 3 1=0

Note:	The ATC procedure was successful for channels 1 and 3 but not for channel 2. Check the hardware of channel 2 for errors and repeat the auto calibration for channel 2, if the hardware is free from defects.
Send:	ATC 2 1
Send:	ATS? 2 1
Receive:	2 1=0
Note:	ATC now was successful for channel 2 (if the ATC procedure should continue to fail , contact your Physik Instrumente Sales Engineer)
Send:	WPA 100
Note:	Save the results of the auto calibration procedure to non-volatile memory.

ATC? (Get Auto Calibration Settings)

Description: Get the options used for the latest auto calibration procedure (started with ATC).

To make the ATC? command available, switch to command level 1 (CCL command (p. 139)).

The success of the auto calibration procedure can be queried with the ATS? command (p. 138).

See "Calibration Settings" (p. 38) for more information.

Format: ATC? [{<ChannelID>}]

Arguments <ChannelID> is one channel of the piezo control electronics, can be 1, 2, 3; see ATC for details

Response: {<ChannelID>"="<Option> LF}

For <Option> see ATC (p. 134)

ATS? (Get Auto Calibration Status)

Description: Query the results of the latest auto calibration procedure (started with ATC).

To make the ATS? command available, switch to command level 1 (CCL command (p. 139)).

The options used for the latest auto calibration procedure can be queried with the ATC? command (p. 137).

See "Calibration Settings" (p. 38) for more information.

Format: ATS? [{<ChannelID> <Option>}]

Arguments <ChannelID> is one channel of the piezo control electronics, can be 1, 2, 3; see ATC for details

<Option> gives the ATC option to be queried, see ATC (p. 134) for details

Response: {<ChannelID> <Option>"="<Status> LF}

where

<Status> gives the results of the latest auto calibration procedure. If 0, the ATC procedure was successful. Values >0 indicate option specific error codes; multiple non-zero error codes for the same channel and option will be listed one after another, see example below for details.

Option	Possible Status values
1	0 = no error occurred (this answer is only possible if ATC has been called with option 1 before) 1 = ATC procedure failed
2	0 = no error occurred 1 = Sensor monitor ADC error
3	0 = no error occurred 1 = DAC error
4	0 = no error occurred 1 = Voltage monitor ADC error
5	0 = no error 1 = Error with sensor input to P-I-controller

6 0 = no error
1 = Amplify range error for piezo monitor voltage
2 = Amplify error for piezo amplifier

Example 1:
Send: ATS? 2 1
Receive: 2 1=0
Note: No error at all for channel 2, this response is only possible if ATC <channel> 1 was sent before (i.e. complete calibration performed). Otherwise only the individual options 2 to 6 can return 0.

Example 2:
Send: ATS? 3 6
Receive: 3 6=2
Note: Amplify error channel 3

Example 3:
Send: ATS? 3 6
Receive: 3 6=21
Note: Amplify range and amplify error channel 3

CCL (Set Command Level)

Description: Changes the active "command level" and determines thus the availability of commands and of write access to system parameters.

Format: CCL <Level> [<PSWD>]

Arguments: <Level> is one command level of the controller

<PSWD> is the password required for changing to the appropriate command level

The following command levels and passwords are valid:

Level = 0 is the default setting, all commands provided for "normal" users are available, read access to all parameters, no password required.

Level = 1 adds additional commands and write access to level-1 parameters (commands and parameters from level 0 are included). The required password is "advanced".

Level > 1 is provided for PI service personnel only. Users can not change to a level > 1. Contact your Physik Instrumente Sales Engineer or write info@pi.ws if there seem to be problems with level 2 or higher parameters.

Response: none

Troubleshooting: Invalid password

Notes: HLP? (p. 156) lists all commands available in the current command level.

HPA? (p. 157) lists the parameters including the information about which command level allows write access to them. For more information about parameter handling see "Controller Parameters" (p. 250).

After controller power-on or reboot, the active command level is always Level 0.

CCL? (Get Command Level)

Description: Get the active "command level".

Format: CCL?

Arguments: none

Response: <Level> is the currently active command level; uint.

Notes: <Level> should be 0 or 1.

<Level> = 0 is the default setting, all commands provided for "normal" users are available, as is read access to all parameters

<Level> = 1 provides additional commands and write access to level-1 parameters (commands and parameters from Level 0 are included)

CSV (Set Syntax Version)

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

Format: CSV <Version>

Arguments: <Version> is the GCS syntax version to be used, can be 1.0 (for GCS 1.0) or 2.0 (for GCS 2.0)

Response: none

Troubleshooting: Current command level too low, check command level with CCL? (p. 140) and change with CCL (p. 139).

Notes: To make the CSV command available, switch to command level 1 (CCL command (p. 139)).

In this document, <Version> = 1.0 is referred to as "E-516 GCS syntax version", and <Version> = 2.0 as "E-517 GCS syntax version".

The current selection of the syntax version is saved with the WPA command (send WPA 100), in addition to the current parameter values and other settings. See the WPA description (p. 229) for details.

The current active GCS syntax version is shown in the display (main screen, rightmost corner; "E517" or "E516") and can be queried with the CSV? command (p. 141).

See "Select Command Set Version" (p. 44) for more information.

CSV? (Get Current Syntax Version)

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

Format: CSV?

Arguments: none

Response: The current GCS syntax version, can be 1.0 (for GCS 1.0) or 2.0 (for GCS 2.0)

Notes: In this document, <Version> = 1.0 is referred to as "E-516 GCS syntax version", and <Version> = 2.0 as "E-517 GCS syntax version".

See "Select Command Set Version" (p. 44) for more information.

CTO (Set Configuration of Trigger Output)

Description: Configures the trigger output conditions for the given digital output line.

The trigger output conditions will become active immediately.

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

Arguments: <TrigOutID> is one digital output line of the controller, see below for details

<CTOPam> is the CTO parameter ID in decimal format, see below for the available IDs

<Value> is the value to which the CTO parameter is set, see below

Response: None

Note: The current trigger output configuration is saved with the WPA command, in addition to the current parameter values and other settings. See the WPA description (p. 229) for details.

The width of a trigger pulse is 30 µs by default, except with the MinMaxThreshold trigger mode where the pulse width depends on the threshold settings. You can change the default pulse width using the Pulse Width parameter, ID 0x0E000900. Possible values are in the range of 10 to 150 µs.

Available output lines and trigger conditions: <TrigOutID> corresponds to the output lines DIO_O1 to DIO_O3, IDs = 1 to 3; see "Digital In/Out Socket" (p. 267). The assignment of these lines to the axes of the E-517 is fixed (DIO_O1 belongs to the first axis (A by default), DIO_O2 to the second axis (B by default) and DIO_O3 to the third axis (C by default). This means that with E-517.i1 models only DIO_O1 can be used. If axes / channels are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the output lines of the active axes can be used.

<CTOPam> parameter IDs available for E-517:
1 = TriggerStep
2 = Axis
3 = TriggerMode
4 = Trigger Delay (always 0)
5 = MinThreshold
6 = MaxThreshold
7 = Polarity

<Value> available for the appropriate <CTOPam> ID:

for TriggerStep: step size in physical units (default value is 0.1)

for Axis: the axis to connect to the trigger output line. The assignment is fixed (see above).

for TriggerMode (default value is 4):

0 = PositionDistance; with this TriggerMode, a trigger pulse is written whenever the axis has covered the TriggerStep distance (<CTOPam> ID 1).

2 = OnTarget; with this TriggerMode, the on-target status of the selected axis is written to the selected trigger output line (this status can also be read with the ONT? command)

3 = MinMaxThreshold; with this TriggerMode, values for MinThreshold and MaxThreshold (<CTOPam> IDs 5 and 6) must be defined. When the axis position of the selected axis is inside the band specified by the MinThreshold and MaxThreshold values, the selected trigger output line is set high, otherwise it is set low (with the default polarity setting).

4 = Generator Trigger; with this TriggerMode, the trigger line action requires use of TWS (p. 201) and/or WGO (p. 221). The number of wave generator output cycles during which trigger pulses are to be output can be set using the Number Of Trigger Cycles parameter, ID 0x18000100.

for MinThreshold/MaxThreshold: position value in physical units; used for the MinMaxThreshold TriggerMode; both values must be set to form a band (default values are 0.0 (Min) and 1.0 (Max))

for Polarity (default value is 1): sets the signal polarity for the trigger line
0 = Active Low
1 = Active High

For application examples see "Configuring Trigger Output" (p. 77) and the lines below.

Example 1:

A pulse on the digital output line DIO_O1 (ID 1) is to be generated whenever axis A has covered a distance of 0.05 µm. The following parameters must be set:

TrigOutID = 1
TriggerMode = 0
TriggerStep = 0.05

Send:	CTO 1 3 0 1 1 0.05
Send:	WPA 100
Note:	The trigger output configuration is saved as power-on default.

Example 2:

On the digital output line DIO_O2 (ID 2), pulses are to be generated at certain waveform points during the wave generator output, i.e. the trigger outputs are to be controlled by the wave generator. To do this, the trigger line is programmed using the TWS and TWC commands, and the corresponding trigger mode is set by CTO. Optionally, you can restrict the trigger output to a certain number of wave generator output cycles.

Send:	TWC
Note:	Clears all trigger settings for the wave generator by switching the signal state for all points to "low". It is recommended to do this before new trigger actions are defined.
Send:	TWS 2 1 1 2 2 0 2 3 0
Note:	Sets trigger action for output line DIO_O2 (identifier is 2), at waveform point 1 it is set high, points 2 and 3 are set low.
Send:	SPA 2 0x18000100 4
Note:	The trigger cycles configured with TWS are to be output on the DIO_O2 line for the first four wave generator output cycles.
Send:	CTO 2 3 4
Note:	The TriggerMode for output line DIO_O2 (ID 2) is set to "Generator

Trigger".

Send: WPA 100

Note: The trigger output configuration is saved as power-on default (except of the TWS settings).

Now the second wave generator can be started with WGO (a waveform must have been defined before), and the trigger action on the DIO_O2 line will take place as specified (the assignment of axes to wave generators and of digital output lines to axes is fixed). See also "Trigger Output Synchronized with Wave Generator" (p. 97).

CTO? (Get Configuration of Trigger Output)

Description: Replies with the values set for specified trigger output lines and parameters

Format: CTO? [{<TrigOutID> <CTOPam>}]

Arguments: <TrigOutID>: is one digital output line of the controller; see CTO

<CTOPam>: parameter ID; see CTO

If all arguments are omitted, the values for all parameters are given for all output lines.

Response: {<TrigOutID> <CTOPam>"="<Value> LF}

For <Value> see CTO.

DCO (Set Drift Compensation Mode)

Description: Sets drift compensation mode for given axes (on or off).

Drift compensation is applied to avoid unwanted changes in displacement over time and is therefore recommended for static operation. For a detailed description see "Drift Compensation" (p. 56).

Drift compensation is automatically deactivated as long as the wave generator is activated.

Format: DCO {<AxisID> <DCOState>}

Arguments: <AxisID> is one axis of the controller

<DCOState> can have the following values:
0 = drift compensation off
1 = drift compensation on

Response: none

Troubleshooting: Illegal axis identifier

DCO? (Get Drift Compensation Mode)

Description: Gets drift compensation mode of given axes.

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

Format: DCO? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<DCOState> LF}

where

<DCOState> is the current drift compensation mode of the axis:
0 = drift compensation off
1 = drift compensation on

Troubleshooting: Illegal axis identifier

DEL (Delay The Command Interpreter)

Description: Delays <uint> milliseconds.

Format: DEL <uint>

Arguments: <uint> is the delay value in milliseconds.

Response: none

DFH (Define Home Position)

Description: Defines the current position of given axes as the axis home position (by setting the position value to 0.00).

Format: DFH [{<AxisID>}]

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

Response: none

Troubleshooting: Illegal axis identifier

Notes: DFH sets the User Origin parameter, ID 0x07010200, to the current sensor position which was valid at the time the DFH command was processed (is used as offset for the calculation of the new axis position). Furthermore DFH adapts the limits of the travel range to the new home position (parameters 0x07000000 and 0x07000001). The parameters are set in volatile memory (RAM) only. To save the currently valid values to non-volatile memory, where they become the power-on defaults, you must use WPA (p. 229). Changes not saved with WPA will be lost when the controller is powered down.

The position shown in the E-517 display is not affected by DFH because it is that of the sensor channel (can also be queried with the TSP? command (p. 199)). DFH affects only the axis position which can be queried with the POS? command (p. 178).

The home position is reset to default (value from non-volatile memory) by the calibration procedures described in "Calibration Settings" (p. 38).

Example:

```
Send:      SVO A 1
Send:      MOV A 9.87
Send:      POS? A
Receive:   A=+0009.8712
Send?      TSP? 1
Receive:   1=+0009.8721
           DFH? A
           A=+0000.0000
Note:      Axis A is moved to an absolute position
           of 9.87 µm. Then the axis position and
           the position of the sensor channel are
           queried. Both are identical because the
           current home position definition is 0
           (DFH? query).
Send:      SPA? A 0x07010200 A 0x07000000 A
           0x07000001
Receive:   A 0X07010200=0.00000000e+0
           A 0X07000000=0.00000000e+0
           A 0X07000001=1.00000000e+2
```

Note:	Query the current values of the User Origin and range limit parameters.
Send:	NLM? A
Receive:	A=+0000.0000
Send:	PLM? A
Receive:	A=+0100.0000
Note:	Query the low and high position soft limits of the axis.
Send:	DFH A
Send:	POS? A
Receive:	A=+0000.0008
Send:	TSP? 1
Receive:	1=+0009.8720
Note:	Query the current axis position and the position of the sensor channel again. Now the axis position is 0 because it was defined to be the new home position (based on internal calculations using the User Origin parameter). The sensor position is still the same as before (also shown in the E-517 display).
Send:	SPA? A 0x07010200 A 0x07000000 A 0x07000001
Receive:	A 0X07010200=9.87220000e+0 A 0X07000000=-9.87220320e+0 A 0X07000001=9.01277760e+1
Note:	Query the current values of the User Origin and range limit parameters again. They were changed by the DFH command.
Send:	DFH? A
Receive:	A=+0009.8722
Note:	Query the new home position definition.
Send:	NLM? A
Receive:	A=-0009.8722
Send:	PLM? A
Receive:	A=+0090.1277
Note:	The low and high position soft limits of the axis were also adapted to the new home position definition.
Send:	WPA 100
Note:	This command saves the new home position setting and the adapted range settings (i.e. the new parameter values and the new values received by NLM? and PLM?) as power-on defaults.

DFH? (Get Home Position Definition)

Description: Returns the sensor position on which the current home position definition of given axes is based.

User Origin parameter, ID 0x07010200

Format: DFH? [{<AxisID>}]

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

Response: {<AxisID>"="<SensorPosition> LF}

where

<SensorPosition> is the sensor position which was valid at the time the last DFH command was processed (is used as offset for the calculation of the current axis position)

Troubleshooting: Illegal axis identifier

Notes: The sensor position on which the current home position definition of given axes is based is saved as User Origin parameter, ID 0x07010200. You can change this parameter with DFH (p. 146) or using SPA / SEP.

See DFH for an example.

DIO? (Get Digital Input Lines)

Description: Lists the states of the specified digital input lines. Can be used to query externally generated signals.

Use TIO? (p. 196) to get the number of installed digital I/O lines.

Format: DIO? [{<DIOID>}]

Arguments: <DIOID> is the identifier of the digital input line, see below for details

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

where

<InputOn> gives the state of the digital input line, see below for details

Notes: Using the DIO? command, you can directly read the DIO_I1 to DIO_I3 lines which are located on the

"Digital In/Out Socket" (p. 267). These lines can be used to start and synchronize the wave generators, see WGO (p. 221) command for details.

With the E-517, there is a fixed one-to-one assignment of axes to the wave generators and the digital input lines. This means that

- the digital input line will only start and synchronize the corresponding wave generator but no other wave generators.
- with E-517.i1 models only DIO_I1 can be used.
- if axes / channels are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the input lines of the active axes and the corresponding wave generators are available.

The <DIOID> identifiers to use for the lines are 1 to 3.

If <InputOn>=0, the digital input is LOW/OFF, if <InputOn>=1, the digital input is HIGH/ON.

DRC (Set Data Recorder Configuration)

Description: Set data recorder configuration: determines the data source and the kind of data (RecordOption) used for the given data recorder table.

Format: DRC <RecTableID> <Source> <RecOption>

Arguments: <RecTableID>: is one data recorder table of the controller, see below

<Source>: is the data source, for example an axis, output signal channel or input signal channel of the controller. The required source depends on the selected record option.

<RecOption>: is the kind of data to be recorded (record option).

See below for a list of the available record options and the corresponding data sources.

Response: none

Notes: The current data recorder configuration is saved with the WPA command, in addition to the current parameter values and other settings. See the WPA description (p. 229) for details.

The E-517 provides 3 data recorder tables (can be read with TNR? (p. 198)) with 8192 points per table (Maximum record points of table i, ID 0x16000201).

With HDR? (p. 155) you will obtain a list of available record options and information about additional parameters concerned with data recording.

For detailed information see "Data Recording" (p. 71).

Record options for the appropriate data sources:

<Source>	<RecOption>
Axis	1 = Target Position of axis (i.e. target value in closed-loop operation), corresponds to the MOV? response 2 = Current Position of axis, corresponds to the POS? response 3 = Position Error of axis 15 = Control Output of axis (before the axis-to-piezo matrix transformation)
Output Signal Channel	7 = Control Voltage of piezo channel (after the axis-to-piezo matrix transformation)

See "Control Value Generation" (p. 53) for more information.

Example: Send DRC 3 A 2
to record the current position of the first axis ("A" by default) in record table 3.

DRC? (get Data Recorder Configuration)

Description: Returns settings made with DRC (p. 150).

Format: DRC? [{<RecTableID>}]

Arguments: <RecTableID>: is one data recorder table of the controller; if omitted settings for all tables are given.

Response: The current DRC settings:

{<RecTableID>"="<Source> <RecOption> LF}

where

<Source>: is the data source, for example an axis or an output signal channel of the controller. The source type depends on the record option.

<RecOption>: is the kind of data to be recorded

See DRC for a list of the available record options and the corresponding data sources.

DRR? (Get Recorded Data Values)

Description: Reading of the last recorded Data Set.

Reading 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.

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

Arguments: <StartPoint>: is the start point in the data recorder table, starts with index 1

<NumberOfPoints>: is the number of points to be read per table

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

Response: The recorded data in GCS array format, see the separate manual for GCS array, SM 146E, and the example below

Notes: If <RecTableID> is omitted, the data from all available tables will be read.

With HDR? (p. 155) you will obtain a list of available record options and trigger options and information about additional parameters concerned with data recording.

For detailed information see "Data Recording" (p. 71).

Example:

```
drr? 1 10
# TYPE = 1
# SEPARATOR = 9
# DIM = 3
# SAMPLE_TIME = 40E-6
# NDATA = 10
# NAME0 = Target position of axis1
# NAME1 = Current position of axis1
# NAME2 = Voltage of piezo channel1
# END_HEADER
+0001.6215     +0001.5383     -0000.3192
+0001.6215     +0001.5383     -0000.3192
+0001.6215     +0001.5383     -0000.3193
+0001.6215     +0001.5382     -0000.3192
+0001.6215     +0001.5382     -0000.3192
+0001.6215     +0001.5381     -0000.3192
+0001.6215     +0001.5380     -0000.3192
+0001.6215     +0001.5380     -0000.3192
+0001.6215     +0001.5380     -0000.3192
+0001.6215     +0001.5379     -0000.3192
```

ERR? (Get Error Number)

Description: Get error code <int> of the last occurred error and reset the error to 0.

Only the last error is buffered. Therefore you should call ERR? after each command.

The error codes and their descriptions are fully listed in "Error Codes" (p. 234).

Format: ERR?

Arguments: none

Response: The error code of the last occurred error (int).

Troubleshooting: Communication breakdown

GOH (Go To Home Position)

Description: Move given axes to home position.

GOH [{<AxisID>}]
is the same as
MOV {<AxisID> 0}

Servo must be enabled for the commanded axis prior to using this command (closed-loop operation).

This command can be interrupted by #24 (p. 133), STP (p. 190) and HLT (p. 156).

Format: GOH [{<AxisID>}]

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

Response: none

Troubleshooting: Illegal axis identifier

Notes: The definition of the home position can be changed with DFH (p. 146). The travel range limits will then be adapted to the new home position.

Motion commands like GOH are not allowed when the E-517 is in OFFLINE mode or when the wave generator output is active. When a macro is running on the E-517, GOH will be executed not until the macro is finished or stopped. See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

GWD? (Get Wave Table Data)

Description: Query waveform shape for given wave table.

The response to GWD? does not contain any offset to the wave generator output set with WOS (p. 228).

Format: GWD? [<StartPoint> [<NumberOfPoints> [{<WaveTableID>}]]]

Arguments: <StartPoint> is the start point in the wave table, starts with index 1

<NumberOfPoints> is the number of points to be read per table

<WaveTableID> is one wave table of the controller

Response: The wave table contents (waveform) in GCS array format (see the separate manual for the GCS array, SM 146E, and the example below)

Example:

```
gwd? 1 10
# TYPE = 1
# SEPARATOR = 9
# DIM = 3
# SAMPLE_TIME = 40E-6
# NDATA = 10
# NAME0 = Wave table1
# NAME1 = Wave table2
# NAME2 = Wave table3
# END_HEADER
+0000.0005     +0000.0001     +0000.0019
+0000.0000     +0000.0000     +0000.0000
+0000.0005     +0000.0001     +0000.0019
+0000.0019     +0000.0005     +0000.0079
+0000.0044     +0000.0011     +0000.0177
+0000.0079     +0000.0019     +0000.0315
+0000.0123     +0000.0030     +0000.0493
+0000.0177     +0000.0044     +0000.0710
+0000.0241     +0000.0060     +0000.0966
+0000.0315     +0000.0079     +0000.1262
```

HDR? (Get All Data Recorder Options)

Description: List a help string which contains all information available about data recording (record options and trigger options, information about additional parameters and commands concerned with data recording).

Format: HDR?

Arguments: none

Response #RecordOptions
{<RecordOption>"="<DescriptionString>[of <Channel>]}

#TriggerOptions
[{<TriggerOption>"="<DescriptionString>}]

#Parameters to be set with SPA
[{<ParameterID>“=“<DescriptionString>}]

#Additional information
[{<Command description>"("<Command>")"}]

end of help

Example:

```
hdr?
#RecordOptions
1=Target position of axis
2=Current position of axis
3=Position error of axis
7=Voltage of piezo channel
15=Control output of axis
#TriggerOptions
0=Default
#Parameters to be set with SPA
0x16000000=Data Recorder Table Rate
end of help
```

Note: TriggerOptions = 0 (default) means that recording is triggered by the IMP (p. 165), STE (p. 189), WGO (p. 221) and WGR (p. 227) commands

HLP? (Get List Of Available Commands)

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

Format: HLP?

Arguments: none

Response: List of commands available

Troubleshooting: Communication breakdown

Notes: The HLP? response contains the commands provided by the current command level. See CCL (p. 139) for more information.

HLT (Halt Motion Smoothly)

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

Error code 10 is set.

#24 (p. 133) and STP (p. 190) in contrast abort current motion as fast as possible for the controller without taking care of systems inertia or oscillations.

Format: HLT [{<AxisID>}]

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

Response: none

Troubleshooting: Illegal axis identifier

Notes: HLT stops motion of all axes caused by move commands (MOV (p. 170), MVR (p. 172), GOH (p. 154), SVA (p. 191), SVR (p. 195)).

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

HPA? (Get List Of Available Parameters)

Description: Responds with a help string which contains all available parameters with short descriptions. See "Controller Parameters" (p. 250) for further details.

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

SPA (p. 185) affects the parameter settings in volatile memory (RAM).

WPA (p. 229) copies parameter settings from RAM to non-volatile memory.

SEP (p. 183) writes parameter settings directly into non-volatile memory (without changing RAM settings).

RPA (p. 179) resets RAM to the values from non-volatile memory.

Format: HPA?

Arguments: none

Response

{<PamID>"="<string> LF}

where

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

<string> is a string which describes the corresponding parameter.

The string has following format:

```
<CmdLevel>TAB<MaxItem>TAB<DataType>TAB<FunctionGroup
Description>TAB<ParameterDescription>
[{TAB<PossibleValue>"="<ValueDescription>}]
```

where

<CmdLevel> is the command level which allows write access to the parameter value

<MaxItem> is the maximum number of items of the same type which are affected by the parameter (the meaning of "item" depends on the parameter, can be axis, sensor channel, piezo channel, wave generator, wave table, data recorder table, digital I/O line, hardware component or the whole system)

<DataType> is the data type of the parameter value, can be INT, FLOAT or CHAR

<FunctionGroupDescription> is the name of the function group to which the parameter belongs (parameters are grouped according to their purpose to clarify their interrelation)

<ParameterDescription> is the parameter name

<PossibleValue> is one value from the allowed data range

<ValueDescription> is the meaning of the corresponding value

IFC (Set Interface Parameters Temporarily)

Description: Interface configuration.

After IFC is sent, the new settings become active and the host PC interface configuration may need to be changed to maintain communication (close the current connection and re-open it with the new settings, see "Communication" (p. 58) for an example).

Interface settings made with IFC are lost when the controller is powered down. To save settings to non-volatile memory and thus make them the power-on defaults, use IFS (p. 163) instead.

Alternatively, you can change the interface settings with SPA (p. 185) or SEP (p. 183) and save the current value with WPA (p. 229) to non-volatile memory (provided that the current command level provides write access to the parameter, see CCL (p. 139)). For the appropriate parameter IDs see below.

Format: IFC {<InterfacePam> <PamValue>}

Arguments: <InterfacePam> is the interface parameter to be changed, see below

<PamValue> gives the value of the interface parameter, see below

Response: None

Notes: CAUTION: A TCP/IP connection will fail if no DCHP server is present but the startup behavior for IP address configuration is set to "use DHCP to obtain IP address, if this fails, use IPADR" (IPSTART = 1). The E-517 will use the address given by IPADR only if IPSTART = 0.

The current active RS-232 baud rate and GPIB (IEEE 488) address can also be set on the E-517 front panel in the COMMUNICATION display screen (p. 21). This screen can be accessed by the trackball (p. 22). The front panel settings interdepend with the interface parameter settings described below.

Possible interface parameters:

The following interface parameters can be set:

RSBAUD

<PamValue> gives the baud rate to be used for RS-232 communication, default is 115200; is also accessible as parameter ID 0x11000400, Uart Baudrate

GPADR

<PamValue> gives the device address to be used for GPIB (IEEE 488) communication, default is 4; is also accessible as parameter ID 0x11000900, GPIB Address

IPADR

The first four portions of <PamValue> specify the default IP address for TCP/IP communication, the last portion specifies the default port to be used, default is 192:168.168.10:50000; is also accessible as parameter ID 0x11000600, IP Address
Note: While the IP address can be changed, the port must always be 50000!

IPSTART

<PamValue> defines the startup behavior for configuration of the IP address for TCP/IP communication,
0 = use IP address defined with IPADR
1 = use DHCP to obtain IP address, if this fails, use IPADR (default);
is also accessible as parameter ID 0x11000800, IP Configuration

IPMASK

<PamValue> gives the IP mask to be used for TCP/IP communication, in the form uint.uint.uint.uint, default is 255.255.255.0; is also accessible as parameter ID 0x11000700, IP Mask

IFC? (Get Current Interface Parameters)

Description: Get the interface configuration parameter values from volatile memory.

The values from volatile memory can also be queried with SPA? (p. 188), for the corresponding parameter IDs see below.

Format: IFC? [{<InterfacePam>}]

Arguments: <InterfacePam> is the interface parameter to be queried, can be RSBAUD, GPADR, IPADR, IPSTART, IPMASK and MACADR (the availability of the parameters depends on the interfaces present on the controller)

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

where

<PamValue> gives the value of the interface parameter from volatile memory

For <InterfacePam> = RSBAUD, <PamValue>
gives the current baud rate of the RS-232 communication;
is also accessible as parameter ID 0x11000400, Uart Baudrate

For >InterfacePam> = GPADR, <PamValue> gives
the current device address for GPIB (IEEE 488) communication;
is also accessible as parameter ID 0x11000900, GPIB Address

For <InterfacePam> = IPADR, the first four portions
of <PamValue> give the IP address used for TCP/IP communication, the last portion gives the port;
is also accessible as parameter ID 0x11000600, IP Address

For <InterfacePam> = IPSTART, <PamValue>
gives the current startup behavior setting for configuration of the IP address for TCP/IP communication,
0 = use IP address defined with IPADR
1 = use DHCP to obtain IP address, if this fails, use IPADR;
is also accessible as parameter ID

0x11000800, IP Configuration

For <InterfacePam> = IPMASK, <PamValue> gives the current IP mask setting to be used for TCP/IP communication, in the form uint.uint.uint.uint;
is also accessible as parameter ID 0x11000700, IP Mask

For <InterfacePam> = MACADR, <PamValue> gives the fixed, unique address of the network hardware in the E-517;
is also accessible as parameter ID 0x11000B00, MAC Address

Notes:

CAUTION: A TCP/IP connection will fail if no DCHP server is present but the startup behavior for IP address configuration is set to "use DHCP to obtain IP address, if this fails, use IPADR" (IPSTART = 1). The E-517 will use the address given by IPADR only if IPSTART = 0.
If the IP address for the current TCP/IP connection was obtained from a DHCP server, this address is not reflected in the response to the IFC? command.

The current active RS-232 baud rate and GPIB (IEEE 488) address can also be checked on the E-517 front panel in the COMMUNICATION display screen (p. 21). This screen can be accessed by the trackball (p. 22). The front panel settings interdepend with the interface parameter settings described above.

IFS (Set Interface Parameters As Default Values)

Description: Interface parameter store.

The power-on default parameters for the interface are changed in non-volatile memory, but the current active parameters are not. Settings made with IFS become active with the next power-on or reboot.

To change the interface parameters immediately (but temporarily) use IFC (p. 159).

It is also possible to change the default settings in non-volatile memory with SEP (p. 183) and to read them with the SEP? (p. 184) command (provided that the current command level provides write access to the parameter, see CCL (p. 139)). For the appropriate parameter IDs see below. If you use RPA (p. 179) to activate the changed settings, it may be necessary to close the current connection and re-open it with the new settings.

Warning: The number of write cycles of non-volatile memory is limited. Write default values only when necessary.

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

Arguments: <Pswd> is the password for writing to non-volatile memory, default is "100"

<InterfacePam> is the interface parameter to be changed, see below

<PamValue> gives the value of the interface parameter, see below

Response: None

Note: CAUTION: A TCP/IP connection will fail if no DCHP server is present but the startup behavior for IP address configuration is set to "use DHCP to obtain IP address, if this fails, use IPADR" (IPSTART = 1). The E-517 will use the address given by IPADR only if IPSTART = 0.

Possible interface parameters: The following interface parameters can be set:

RSBAUD
<PamValue> gives the baud rate to be used

for RS-232 communication, default is 115200;
is also accessible as parameter ID
0x11000400, Uart Baudrate

GPADR

<PamValue> gives the device address to be used for GPIB (IEEE 488) communication, default is 4;
is also accessible as parameter ID
0x11000900, GPIB Address

IPADR

The first four portions of <PamValue> specify the default IP address for TCP/IP communication, the last portion specifies the default port to be used, default is 192:168.168.10:50000;
is also accessible as parameter ID
0x11000600, IP Address
Note: While the IP address can be changed, the port must always be 50000!

IPSTART

<PamValue> defines the startup behavior for configuration of the IP address for TCP/IP communication,
0 = use IP address defined with IPADR
1 = use DHCP to obtain IP address, if this fails, use IPADR (default);
is also accessible as parameter ID
0x11000800, IP Configuration

IPMASK

<PamValue> gives the IP mask to be used for TCP/IP communication, in the form uint.uint.uint.uint, default is 255.255.255.0;
is also accessible as parameter ID
0x11000700, IP Mask

IFS? (Get Interface Parameters As Default Values)

Description: Get the interface configuration parameter values stored in non-volatile memory (i.e. the current power-on default)

Format: IFS? [{<InterfacePam>}]

Arguments: <InterfacePam> is the interface parameter to be queried, can be RSBAUD, GPADR, IPADR, IPSTART, IPMASK or MACADR (the availability of the parameters depends on the interfaces present on the controller)

See IFS (p. 163) for details.

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

where

<PamValue> is the value of the interface parameter in non-volatile memory, see IFS for possible values

For <InterfacePam> = MACADR, <PamValue> gives the unique address of the network hardware in the E-517 (is also accessible as parameter ID 0x11000B00, MAC Address)

IMP (Start Impulse And Response Measurement)

Description: Starts performing an impulse and recording the impulse response for the given axis.

The data recorder configuration, i.e. the assignment of data sources and record options to the recorder tables, can be set with DRC (p. 150).

The recorded data can be read with the DRR? command (p. 152).

Format: IMP <AxisID> <Amplitude>

Arguments <AxisID> is one axis of the controller

<Amplitude> is the height of the impulse. See below for details.

Response: none

Troubleshooting: The control value resulting from the specified impulse height is out of limits:

Open-loop operation: the amplitude limitation results from the voltage limit parameters (IDs 0x0B000007, 0x0B000008, 0x0C000000 and 0x0C000001)
Closed-loop operation: use TMN? (p. 197) and TMX? (p. 197) to ask for the current valid travel range limits.

Motion commands like IMP are not allowed when the E-517 is in OFFLINE mode or when the wave generator output is active. When a macro is running on the E-517, IMP will be executed not until the macro is finished or stopped. See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Notes: An "impulse" consists of a relative move of the specified amplitude followed by an equal relative move in the opposite direction. Depending on the current servo mode, the impulse is performed relative to the current position (servo ON) or to the current piezo voltage (servo OFF).

In closed-loop operation (servo ON), the given amplitude is interpreted as relative position value. In open-loop operation (servo OFF), the amplitude corresponds to a relative piezo voltage value.

IMP? (Get IMP Settings)

Description: Get last sent IMP settings for the given axis.

Format: IMP? [{<AxisID>}]

Arguments <AxisID> is one axis of the controller

Response: {<AxisID>"="<Amplitude> LF}

where

<Amplitude> is the height of the last commanded impulse. See IMP (p. 165) for details.

MAC (Call Macro Function)

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

Format: MAC <keyword> {<parameter>}

in particular:

MAC BEG <macroname>
MAC DEF <macroname>
MAC DEF?
MAC DEL <macroname>
MAC END
MAC FREE?
MAC NSTART <macroname> <uint>
MAC START <macroname>

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

MAC BEG <macroname>

Start recording a macro to be named *macroname* on the controller; may not be used in a macro; the commands that follow become the macro, so if successful, the error code cannot be queried. End the recording with MAC END.

MAC END

Stop macro recording (cannot become part of a macro)

MAC FREE?

Ask for the free memory space for macro recording
Response: <uint> is the free memory in number of characters

MAC DEF <macroname>

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

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

MAC DEL <macroname>
Deletes specified macro
With <macroname> = *.* all macros are deleted.

MAC NSTART <macroname> <uint>
Repeat the specified macro <uint> times. Another execution is started when the last one is finished.

MAC START <macroname>
Starts one execution of specified macro.

Response: none

Troubleshooting: Macro recording is active (keywords BEG, DEL) or inactive (END)

Macro contains a disallowed MAC command

Notes: Macro recording:

- The maximum number of characters for the macro name is 8, the maximum number of macros to be stored on the E-517 is 60.
- A macro is immediately written to non-volatile memory. No additional storage procedure is required.
- During macro recording no macro execution is allowed.
- A macro can be overwritten by a macro with the same name.
- A macro can start another macro without any nesting level limitation. A macro can call itself to form an infinite loop.
- Macros can be recorded when the wave generator is running.

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

Macro execution:

- If the E-517 is in OFFLINE mode, move commands from a running macro are not allowed (they provoke an error message). See "Control Modes" (p. 23) for details.
- A macro can be started when the wave generator is running, but move commands from the macro are not executed and provoke an error message.
- Commands sent when a macro is running will be executed not until the macro is finished or stopped. The only exceptions are #5, #6, #7, #8, #9 and #24 which are executed immediately.
- When a macro is running, trackball settings can be used in parallel to command axis motion. See "Trackball Functions" (p. 22) for details.
- Macro execution can be stopped with #24 (p. 133) and STP (p. 190).
- Simultaneous execution of multiple macros is not possible. Only one macro can be executed at a time.
- A running macro may not be deleted.
- You can query with #8 (p. 132) if a macro is currently running on the controller.

See "Working with Controller Macros" (p. 113) for examples.

MAC? (List Macros)

Description: List macros or content of a given macro.

Format: MAC? [<macroname>]

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

Response: <string>
if <macroname> was given, <string> is the content of this macro;
if <macroname> was omitted, <string> is a list with the names of all stored macros

Troubleshooting: Macro <macroname> not found

MOV (Set Target Position)

Description: Set new absolute target position for given axis.

Servo must be enabled for the commanded axis prior to using this command (closed-loop operation).

Format: MOV {<AxisID> <Position>}

Arguments <AxisID> is one axis of the controller

<Position> is the new absolute target position in physical units.

Response: none

Troubleshooting: Target position out of limits. Use TMN? (p. 197) and TMX? (p. 197) to ask for the current valid travel range limits.

Illegal axis identifier

Servo is Off for one of the axes specified.

Motion commands like MOV are not allowed when the E-517 is in OFFLINE mode or when the wave generator output is active. When a macro is running on the E-517, MOV will be executed not until the macro is finished or stopped. See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Notes: During a move, a new move command resets the target to a new value and the old one may never be reached.

The MOV command can be interrupted by #24 (p. 133), STP (p. 190) and HLT (p. 156).

Example 1:

Send: MOV 1 10
Note: Axis 1 moves to 10 (target position in µm)

Example 2:

Send: MOV 1 243
Send: ERR?
Receive: 7
Note: The axis does not move. The error code "7" in the reply to the ERR? command (p. 153) indicates that the target position given in the move command is out of limits.

Example 3:

Send: MOV 1 10 2 100 3 4000
Send: ERR?
Receive: 7
Note: The axes do not move. The error code "7" in the reply to the ERR? command (p. 153) indicates that at least one of the target positions given in the move command is out of limits.

MOV? (Get Target Position)

Description: Returns last valid commanded target position.

Format: MOV? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<float> LF}

where

<float> is the last commanded target position in physical units

Troubleshooting: Illegal axis identifier

Notes: MOV? gives the target position settings made when the E-517 is in ONLINE mode (see "Control Modes" (p. 23) for details).

In ONLINE mode, the target position can be changed by move commands (received via interface or from a running macro; MOV (p. 170), MVR (p. 172), GOH (p. 154), IMP (p. 165), STE (p. 189)), by trackball settings (p. 22) and by the wave generator (p. 84). When the control mode is switched from OFFLINE to ONLINE, the target position is set to the current position. See "Control Value Generation" (p. 53) for details.

MOV? gets the commanded positions. Use POS? (p. 178) to get the current positions.

MVR (Set Target Relative To Current Position)

Description: Move given axes relative to the last commanded target position.

The new target position is calculated by adding the given value <Distance> to the last commanded target value.

Servo must be enabled for the commanded axis prior to using this command (closed-loop operation).

Format: MVR {<AxisID> <Distance>}

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

<Distance> gives the distance to move; the sum of the distance and the last commanded target position is set as new target position (in physical units).

Response: none

Troubleshooting: Target position out of limits. Use TMN? (p. 197) and TMX? (p. 197) to ask for the current valid travel range limits, and MOV? (p. 171) for the current target.

Illegal axis identifier

Servo is Off for one of the axes specified.

Motion commands like MVR are not allowed when the E-517 is in OFFLINE mode or when the wave generator output is active. When a macro is running on the E-517, MVR will be executed not until the macro is finished or stopped. See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Notes: The MVR command can be interrupted by #24 (p. 133), STP (p. 190) and HLT (p. 156).

Example:

```
Send:     MOV 1 0.5
Note:     This is an absolute move.
Send:     POS? 1
Receive:  1=0.500000
Send:     MOV? 1
Receive:  1=0.500000
Send:     MVR 1 2
Note:     This is a relative move.
Send:     POS? 1
Receive:  1=2.500000
Send:     MVR 1 2000
Note:     New target position of axis 1 would
          exceed motion range. Command is
          ignored, i.e. the target position remains
          unchanged, and the axis does not move.
Send:     MOV? 1
Receive:  1=2.500000
Send:     POS? 1
Receive:  1=2.500000
```

NLM (Set Low Position Soft Limit)

Description: Limits the low end of the axis travel range in closed-loop operation ("soft limit").

Format: NLM {<AxisID> <LowLimit>}

Arguments: <AxisID> is one axis of the controller

<LowLimit> is the limit position for the low end of the travel range, in physical units.

Response: None

Notes: The position value set with NLM must be equal to or larger than the range limit given by parameter 0x07000000, and smaller than the range limit given by parameter 0x07000001 or by PLM (p. 177). Use TMN? (p. 197) and TMX? (p. 197) to ask for the current valid travel range limits.

The current NLM settings are saved with the WPA command, in addition to the current parameter values and other settings. See the WPA description (p. 229) for details.

The definition of the home position can be changed with DFH (p. 146). The travel range limits will then be adapted to the new home position.

NLM? (Get Low Position Soft Limit)

Description: Get the position "soft limit" which determines the low end of the axis travel range in closed-loop operation.

Format: NLM? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<LowLimit> LF}

where

<LowLimit> is the limit position for the low end of the travel range, in physical units.

ONL (Set Control Mode)

Description: Sets control mode for given piezo channel (ONLINE or OFFLINE mode).

Format: ONL {<OutputSignalID> <ControlMode>}

Arguments: <OutputSignalID> is one piezo channel of the controller

<ControlMode> can have the following values:
0 = OFFLINE mode, the output voltage depends on analog control input and DC offset applied to the channel
1 = ONLINE mode, the E-517 controls the generation of the output voltage
In ONLINE mode the SERVO switches of all channels must be set to OFF on the piezo control electronics.

Response: none

Troubleshooting: Illegal channel identifier

Notes: The current control mode determines the applicable control sources for the output voltage and hence for the axis motion. See "Control Modes" (p. 23) for more information.

The current control mode is visible for the individual channels on the main screen of the E-517 display, see "Display Screens" (p. 21). Using the ONL? command (p. 175), you can check the current control mode on a per-channel basis.

Pressing the trackball on the E-517 front panel when the main screen is displayed switches the mode for all channels at once.

ONL? (Get Control Mode)

Description: Gets current control mode for given piezo channel.

Format: ONL? [{<OutputSignalID>}]

Arguments: <OutputSignalID> is one piezo channel of the controller

Response: {<OutputSignalID>"="<ControlMode> LF}

where

<ControlMode> is the current mode which can have the following values:
0 = OFFLINE mode, the output voltage depends on analog control input and DC offset applied to the channel
1 = ONLINE mode, the E-517 controls the generation of the output voltage

Notes: The current control mode is also visible for the individual channels on the main screen of the E-517 display, see "Display Screens" (p. 21).

In ONLINE mode the SERVO switches of all channels must be set to OFF on the piezo control electronics.

ONT? (Get On Target State)

Description: Get on-target status of given axis.

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

Format: ONT? [{<AxisID>}]

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

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

where

<uint> = "1" when the specified axis is on-target, "0" otherwise.

Troubleshooting: Illegal axis identifier

Notes: The on-target status can only be identified in closed-loop operation (servo ON).

Software-emulated on-target reading must be enabled by the Enable SW On Target Signal parameter, ID 0x07010600 (1 = enabled; default setting).

The on-target status is influenced by the Tolerance parameter (ID 0x07000900): the on-target status is true if
|Current Position - Target Position| ≤ Tolerance

See "On Target Reading" (p. 57) for more information.

OVF? (Get Overflow State)

Description: Get overflow status of given axis.

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

Overflow means that the control variables are out of range (can only happen if controller is in closed-loop operation).

Format: OVF? [{<AxisID>}]

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

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

where

<uint> = "0" (axis is not in overflow) or "1" (axis is in overflow)

Troubleshooting: Illegal axis identifier

PLM (Set High Position Soft Limit)

Description: Limits the high end of the axis travel range in closed-loop operation ("soft limit").

Format: PLM {<AxisID> <HighLimit>}

Arguments: <AxisID> is one axis of the controller

<HighLimit> is the limit position for the high end of the travel range, in physical units.

Response: None

Notes: The position value set with PLM must be equal to or smaller than the range limit given by parameter 0x07000001, and larger than the range limit given by parameter 0x07000000 or by NLM (p. 173). Use TMN? (p. 197) and TMX? (p. 197) to ask for the current valid travel range limits.

The current PLM settings are saved with the WPA command, in addition to the current parameter values and other settings. See the WPA description (p. 229) for details.

The definition of the home position can be changed with DFH (p. 146). The travel range limits will then be adapted to the new home position.

PLM? (Get High Position Soft Limit)

Description: Get the position "soft limit" which determines the high end of the axis travel range in closed-loop operation.

Format: PLM? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<HighLimit> LF}

where

<HighLimit> is the limit position for the high end of the travel range, in physical units.

POS? (Get Real Position)

Description: Returns the current axis position.

If all arguments are omitted, gets current position of all axes.

Format: POS? [{<AxisID>}]

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

Response: {<AxisID>"="<float> LF}

where

<float> is the current axis position in physical units

Troubleshooting: Illegal axis identifier

Note: To request the current position of input signal channels (sensors) in physical units, use the TSP? (p. 199) command instead.

RBT (Reboot System)

Description: Reboot system. Controller behaves just like after power-on.

Format: RBT

Arguments: none

Response: none

Notes: With TCP/IP and USB connections, communication can not be maintained after the E-517 is power-cycled or rebooted. The connection must then be closed and reopened.

RPA (Reset Volatile Memory Parameters)

Description: Resets the given parameter of the given item. The value from non-volatile memory is written into volatile memory.

Related commands:

With HPA? (p. 157) you can obtain a list of the available parameters. SPA (p. 185) affects the parameter settings in volatile memory, WPA (p. 229) writes parameter settings from volatile to non-volatile memory, and SEP (p. 183) writes parameter settings directly into non-volatile memory (without changing the settings in volatile memory).

See SPA for an example.

Format: RPA [{<ItemID> <PamID>}]

Arguments: <ItemID> is the item for which a parameter is to be reset. See below for details.

<PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.

Response: none

Troubleshooting: Illegal item identifier, wrong parameter ID

Notes: This procedure can take a few seconds.

If you use RPA to activate changed interface parameter settings, it may be necessary to close the current connection and re-open it with the new settings.

Available item IDs and parameter IDs:

<ItemID> can be an axis identifier, a sensor channel, a piezo channel, a wave generator, a digital output line or the whole system; the item type depends on the parameter, see "Parameter Overview" (p. 252) for the item type concerned. See "Accessible Items and Their Identifiers" (p. 50) for the identifiers of the items.

Valid parameter IDs are given in "Parameter Overview" (p. 252).

RTR (Set Record Table Rate)

Description:

Sets the record table rate, i.e. the number of servo-loop cycles to be used in data recording operations. Settings larger than 1 make it possible to cover longer time periods.

Format:

RTR <RecordTableRate>

Arguments:

<RecordTableRate> is the table rate to be used for recording operations (unit: number of servo-loop cycles), must be an integer value larger than zero

Response:

None

Notes:

RTR affects the Data Recorder Table Rate parameter, ID 0x16000000.

The duration of the recording can be calculated as follows:

Rec. Duration = Servo Update Time * RTR value * Number of Points

where

Servo Update Time is given in seconds by parameter 0x0E000200

Number of Points is the length of the data recorder table

For more information see "Data Recording" (p. 71).

The record table rate set with RTR is saved in volatile memory (RAM) only. To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA (p. 229). Changes not saved with WPA will be lost when the controller is powered down. To have write access to the parameter, it might be necessary to switch to a higher command level using CCL (p. 139).

RTR? (Get Record Table Rate)

Description: Gets the current record table rate, i.e. the number of servo-loop cycles used in data recording operations.

Format: RTR?

Arguments: None

Response: <RecordTableRate> is the table rate used for recording operations (unit: number of servo-loop cycles)

Notes: Gets the Data Recorder Table Rate parameter value in volatile memory (ID 0x16000000).

For more information see "Data Recording" (p. 71).

SAI (Set Current Axis Identifiers)

Description: Sets the axis identifiers for the given axes.

After it was set with SAI, the new axis identifier must be used as <AxisID> in all axis-related commands.

Format: SAI {<AxisID> <NewIdentifier>}

Arguments: <AxisID> is one axis of the controller

<NewIdentifier> is the new identifier to use for the axis, see below for details

Response: none

Notes: An axis could be identified with up to 8 characters. Use TVI? (p. 200) to ask for valid characters.

SAI affects the Axis Name parameter, ID 0x07000600, in volatile memory (RAM). To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA (p. 229). Changes not saved with WPA will be lost when the E-517 is powered down.

SAI? (Get List Of Current Axis Identifiers)

Description: Gets the axis identifiers.

See also "Accessible Items and Their Identifiers" (p. 50).

Format: SAI? [ALL]

Arguments: [ALL] is optional and provided for compatibility with controllers which allow for axis deactivation. [ALL] then ensures that the answer also includes the axes which are "deactivated".

Response: {<AxisID> LF}

<AxisID> is one axis of the controller.

SEP (Set Non-Volatile Memory Parameters)

Description: Set a parameter of a given item to a different value in non-volatile memory, where it becomes the new power-on default.

After parameters were set with SEP, you can use RPA (p. 179) to activate them (write them to volatile memory) without controller reboot.

Caution: This command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware!

Related commands:

HPA? (p. 157) returns a list of the available parameters.

SPA (p. 185) writes parameter settings into volatile memory (without changing the settings in non-volatile memory).

WPA (p. 229) writes parameter settings from volatile to non-volatile memory.

See SPA for an example.

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

Arguments <Pswd> is the password for writing to non-volatile memory, default is "100"

<ItemID> is the item for which a parameter is to be changed in non-volatile memory. See below for details.

<PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.

<PamValue> is the value to which the given parameter of the given item is set

Response: none

Troubleshooting: Illegal item identifier, wrong parameter ID, invalid password, command level too low for write access

Notes: To have write access to the parameter(s), it might be necessary to switch to a higher command level using CCL (p. 139).

Warning: The number of write cycles of non-volatile memory is limited. Write default values only when necessary.

Available item IDs and parameter IDs: <ItemID> can be an axis identifier, a sensor channel, a piezo channel, a wave generator, a digital output line or the whole system; the item type depends on the parameter, see "Parameter Overview" (p. 252) for the item type concerned. See "Accessible Items and Their Identifiers" (p. 50) for the identifiers of the items.

Valid parameter IDs are given in "Parameter Overview" (p. 252).

SEP? (Get Non-Volatile Memory Parameters)

Description: Get the value of a parameter of a given item from non-volatile memory.

With HPA? (p. 157) you can obtain a list of the available parameters and their IDs.

Format: SEP? [{<ItemID> <PamID>}]

Arguments: <ItemID> is the item for which a parameter value from non-volatile memory is to be queried. See below for details.

<PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.

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

where

<PamValue> is the value of the given parameter for the given item

Troubleshooting: Illegal item identifier, wrong parameter ID

Available item IDs and parameter IDs: <ItemID> can be an axis identifier, a sensor channel, a piezo channel, a wave generator, a wave table, a data recorder table, a digital output line, a hardware component or the whole system; the item type depends on the parameter, see "Parameter Overview" (p. 252) for the item type concerned. See "Accessible Items and Their Identifiers" (p. 50) for the identifiers of the items.

Valid parameter IDs are given in "Parameter Overview" (p. 252).

SPA (Set Volatile Memory Parameters)

Description: Set a parameter of a given item to a value in volatile memory (RAM). Parameter changes will be lost when the controller is powered down or rebooted or when the parameters are restored with RPA (p. 179).

Caution: This command is for setting hardware-specific parameters. Wrong values may lead to improper operation or damage of your hardware!

Related commands:

HPA? (p. 157) returns a list of the available parameters.

SEP (p. 183) writes parameter settings directly into non-volatile memory (without changing the settings in volatile memory).

WPA (p. 229) writes parameter settings from volatile to non-volatile memory.

RPA resets volatile memory to the value in non-volatile memory.

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

Arguments

<ItemID> is the item for which a parameter is to be changed in volatile memory. See below for details.

<PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.

<PamValue> is the value to which the given parameter of the given item is set

Response:

none

Troubleshooting:

Illegal item identifier, wrong parameter ID,value out of range, command level too low for write access

Notes:

If you change the current settings of the communication interface, it may be necessary to close the current connection and re-open it with the new settings.

To have write access to the parameter(s), it might be necessary to switch to a higher command level using CCL (p. 139).

Available item IDs and parameter IDs:

<ItemID> can be an axis identifier, a sensor channel, a piezo channel, a wave generator, a digital output line or the whole system; the item type depends on the parameter, see "Parameter Overview" (p. 252) for the item type concerned. See "Accessible Items and Their Identifiers" (p. 50) for the identifiers of the items.

Valid parameter IDs are given in "Parameter Overview" (p. 252).

Example 1:

Send:	SPA 1 0x16000000 8
Note:	Set the Data Recorder Table Rate for the controller to 8, parameter ID written in hexadecimal format
Send:	SPA 1 369098752 2
Note:	Sets the Data Recorder Table Rate for the controller to 2, parameter ID written in decimal format

Example 2: The position of the second sensor channel (ID is 2) is to be displayed on the E-517 front panel with 4 decimal places. For that purpose, the Display Format parameter, ID 0x04000E01, must be set to 4 for channel 2.

Send: CCL 1 advanced
Note: Switch to command level 1 because this level is required for write access to the Display Format parameter.
Send: SPA 2 0x04000E01 4
Note: The setting should become immediately effective on the display. It is made in volatile memory only.

If everything is okay and you want to use this system configuration after the next power-on, save the parameter settings from volatile to non-volatile memory.

Send: WPA 100
Note: When WPA is used without specifying any parameters, all currently valid parameter values from volatile memory and several other settings are saved (see WPA for details).
Send: SEP? 2 0x04000E01
Receive: 2 0x04000E01=4
Note: Check the parameter settings in non-volatile memory.

Example 3: The task performed in example 2 can also be done in the following way, provided you are sure that the new system configuration will work:

Send: CCL 1 advanced
Note: Switch to command level 1 because this level is required for write access to the Display Format parameter.
Send: SEP 100 2 0x04000E01 4
Note: The setting is made in non-volatile memory and hence is the new power-on default, but is not yet active (the display does not change).

To use the new settings immediately, you now have to load them to volatile memory (otherwise they would become active after the next power-on or reboot of the controller).

Send: RPA

Note: The new configuration is now active.
Send: SPA? 2 0x04000E01
Receive: 2 0x04000E01=4
Note: Check the parameter settings in volatile memory. The display should have changed now.

SPA? (Get Volatile Memory Parameters)

Description: Get the value of a parameter of a given item from volatile memory (RAM).

With HPA? (p. 157) you can obtain a list of the available parameters and their IDs.

Format: SPA? [{<ItemID> <PamID>}]

Arguments: <ItemID> is the item for which a parameter is to be queried in volatile memory. See below for details.

<PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.

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

where

<PamValue> is the value of the given parameter for the given item

Troubleshooting: Illegal item identifier, wrong parameter ID

Available item IDs and parameter IDs: <ItemID> can be an axis identifier, a sensor channel, a piezo channel, a wave generator, a wave table, a data recorder table, a digital output line, a hardware component or the whole system; the item type depends on the parameter, see "Parameter Overview" (p. 252) for the item type concerned. See "Accessible Items and Their Identifiers" (p. 50) for the identifiers of the items.

Valid parameter IDs are given in "Parameter Overview" (p. 252).

SSN? (Get Device Serial Number)

Description: Get the serial number of the E-517.

Format: SSN?

Arguments: none

Response <SerialNumber> is the serial number of the device; the answer gives the value of the Device Serial Number parameter, ID 0x0D000000.

STE (Start Step And Response Measurement)

Description: Starts performing a step and recording the step response for the given axis.

The data recorder configuration, i.e. the assignment of data sources and record options to the recorder tables, can be set with DRC (p. 150).

The recorded data can be read with the DRR? (p. 152) command.

Format: STE <AxisID> <Amplitude>

Arguments <AxisID> is one axis of the controller

<Amplitude> is the height of the step. See below for details.

Response: none

Troubleshooting: The control value resulting from the specified step height is out of limits:

Open-loop operation: the amplitude limitation results from the voltage limit parameters (IDs 0x0B000007, 0x0B000008, 0x0C000000 and 0x0C000001)
Closed-loop operation: use TMN? (p. 197) and TMX? (p. 197) to ask for the current valid travel range limits.

Motion commands like STE are not allowed when the E-517 is in OFFLINE mode or when the wave generator output is active. When a macro is running on the E-517, STE will be executed not until the macro is finished or stopped. See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Notes: A "step" consists of a relative move of the specified amplitude. Depending on the current servo mode, the step is performed relative to the current position (servo ON) or to the current piezo voltage (servo OFF).

In closed-loop operation (servo ON), the given amplitude is interpreted as relative position value. In open-loop operation (servo OFF), the amplitude corresponds to a relative piezo voltage value.

STE? (Get STE Settings)

Description: Get last sent STE settings for the given axis.

Format: STE? [{<AxisID>}]

Arguments <AxisID> is one axis of the controller

Response: {<AxisID>"="<Amplitude> LF}

where

<Amplitude> is the height of the last commanded step. See STE (p. 189) for details.

STP (Stop All Motion)

Description: Stops all motion abruptly. For details see the notes below.

Sets error code to 10.

This command is identical in function to #24 (p. 133) which should be preferred when the controller is performing time-consuming tasks.

Format: STP

Arguments: none

Response: none

Troubleshooting: Communication breakdown

Notes: STP stops motion of all axes caused by move commands (MOV (p. 170), MVR (p. 172), GOH (p. 154), SVA (p. 191), SVR (p. 195)) and wave generator output (WGO (p. 221)). If STP is part of a macro, it stops macro execution when accessed by the command interpreter (MAC (p. 167))

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

HLT (p. 156) in contrast to STP stops motion smoothly.

SVA (Set Open-Loop Axis Value)

Description: Set absolute open-loop control value to move the axis.

Servo must be switched off (open-loop operation) when using this command.

Format: SVA {<AxisID> <Amplitude>}

Arguments <AxisID> is one axis of the controller

<Amplitude> is the new absolute open-loop control value. See below for details.

Response: none

Troubleshooting: The control value specified by the given amplitude is out of limits. The limitation results from the voltage limit parameters (IDs 0x0B000007, 0x0B000008, 0x0C000000 and 0x0C000001) of the piezo channels which would be involved in the axis motion.

Illegal axis identifier

Servo is On for one of the specified axes

Motion commands like SVA are not allowed when the E-517 is in OFFLINE mode or when the wave generator output is active. When a macro is running on the E-517, SVA will be executed not until the macro is finished or stopped. See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Notes: The amplitude is to be given as a piezo voltage value in V.

The SVA command can be interrupted by #24 (p. 133), STP (p. 191) and HLT (p. 156).

Example 1:

Send: SVA A 10
Note: The piezo voltage for axis A (and hence for piezo channel 1) is set to 10 V. The axis moves accordingly with no position control.

Example 2:

Send: SVA A 300
Send: ERR?
Receive: 302
Note: The axis does not move. The error code "302" reported by the ERR? command (p. 153) indicates that the piezo voltage value set by SVA is out of limits.

Example 3:

Send: SVA A 300 B 60 C 100
Send: ERR?
Receive: 302
Note: The axes do not move. The error code "302" reported by the ERR? command (p. 153) indicates that at least one of the piezo voltage values set by SVA is out of limits.

SVA? (Get Open-Loop Axis Value)

Description: Returns last valid open-loop control value of given axis.

Format: SVA? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<float> LF}

where

<float> is the last commanded open-loop control value. See below for details.

Troubleshooting: Illegal axis identifier

Notes: SVA? gives the open-loop control value settings made when the E-517 is in ONLINE mode (see "Control Modes" (p. 23) for details).

In ONLINE mode, the open-loop control value can be changed by move commands (received via interface or from a running macro; SVA (p. 191), SVR (p. 195), IMP (p. 165), STE (p. 189)), by trackball settings (p. 22) and by the wave generator (p. 84). When the control mode is switched from OFFLINE to ONLINE, the open-loop control value is set to the value of the Default Voltage parameter, ID 0x07000C01. See "Control Value Generation" (p. 53) for details.

The open-loop control value returned by SVA? is to be interpreted as piezo voltage value in V.

SVA? gets the commanded open-loop values for the axes. Use VOL? (p. 209) to get the current voltage output of the piezo channels.

SVO (Set Servo State)

Description: Sets servo-control state for given axes (open-loop or closed-loop operation).

Format: SVO {<AxisID> <ServoState>}

Arguments: <AxisID> is one axis of the controller

<ServoState> can have the following values:
0 = servo off (open-loop operation)
1 = servo on (closed-loop operation)

Response: none

Troubleshooting: Illegal axis identifier

Notes: CAUTION: In ONLINE mode the SERVO switches of all channels must be set to OFF on the piezo control electronics. Otherwise the SVO command has no complete control over the servo mode settings.

When the servo is switched on for an axis, the target position is set to the current position, and when it is switched off, the last valid control value remains active. See "Control Value Generation" (p. 53) for more information.

The current servo state affects the applicable move commands:
servo-control off: use SVA (p. 191) and SVR (p. 195)
servo-control on: use MOV (p. 170), MVR (p. 172) and GOH (p. 154)

When servo is switched on or off during motion caused by move commands, the axis stops.

Servo-control can not be switched on or off while the wave generator is running for the axis.

Using a start-up macro, you can set up the device to start with closed-loop operation. See "Start-Up Macro" (p. 115) for details.

SVO? (Get Servo State)

Description: Gets servo-control state of given axes.

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

Format: SVO? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<ServoState> LF}

where

<ServoState> is the current servo state of the axis:
0 = servo off (open-loop operation)
1 = servo on (closed-loop operation)

Troubleshooting: Illegal axis identifier

SVR (Set Relative Open-Loop Axis Value)

Description: Set open-loop control value relative to the current open-loop control value to move the axis.

The new open-loop control value is calculated by adding the given value <Difference> to the last commanded open-loop control value.

Servo must be off when using this command (open-loop operation).

Format: SVR {<AxisID> <Difference>}

Arguments <AxisID> is one axis of the controller

<Difference> is the value which is added to the current open-loop control value. See below for details.

Response: none

Troubleshooting: The specified control value is out of limits. The limitation results from the voltage limit parameters (IDs 0x0B000007, 0x0B000008, 0x0C000000 and 0x0C000001) of the piezo channels which would be involved in the axis motion.

Illegal axis identifier

Servo is On for one of the specified axes

Motion commands like SVR are not allowed when the E-517 is in OFFLINE mode or when the wave generator output is active. When a macro is running on the E-517, SVR will be executed not until the macro is finished or stopped. See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Notes: The <Difference> value is to be given as a piezo voltage value in V.

The SVR command can be interrupted by #24 (p. 133), STP (p. 191) and HLT (p. 156).

TAD? (Get ADC Value Of Input Signal)

Description: Get the current value from the specified input signal channel's A/D converter. Using this command it is possible to check for sensor overflow.

Format: TAD? [{<InputSignalID>}]

Arguments: <InputSignalID> is one input signal channel of the controller

Response: {<InputSignalID>"="<uint> LF}

where

<uint> is the current A/D value, dimensionless

Note: The input signal channels to be queried with TAD? are the sensor channels of the piezo control electronics, IDs = 1 to 3 (actually available IDs depend on the response to the TSC? command (p. 199)).

The TAD? response represents the digitized signal value without digital filtering.

TIO? (Tell Digital I/O Lines)

Description: Tell number of installed digital I/O lines

Format: TIO?

Arguments: none

Response: I=<uint1>
O=<uint2>

where

<uint1> is the number of digital input lines.
<uint2> is the number of digital output lines.

Notes: The digital output lines reported by TIO? are DIO_O1 to DIO_O3. They can be set with CTO (p. 142), TWS (p. 201) and WGO (p. 221).

The digital input lines reported by TIO? are DIO_I1 to DIO_I3. The can be read with the DIO? command (p. 149).

With E-517.i1 models only DIO_O1 and DIO_I1 are available. If axes / channels are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the output and input lines of the active axes are available.

All the lines are located on the Digital In/Out Socket (p. 267) of the E-517.

TMN? (Get Minimum Commandable Position)

Description: Get the minimum commandable position in physical units.

Format: TMN? [{ <AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response {<AxisID>"="<float> LF}

where

<float> is the minimum commandable position in physical units

Note: The minimum commandable position is defined by the Range Limit min parameter, ID 0x07000000, or by the low position soft limit (NLM, NLM?), whichever is higher.

When a new home position is defined with DFH (p. 146), the minimum commandable position is automatically adapted to the appropriate new value.

TMX? (Get Maximum Commandable Position)

Description: Get the maximum commandable position in physical units.

Format: TMX? [{ <AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response {<AxisID>"="<float> LF}

where

<float> is the maximum commandable position in physical units

Description: The maximum commandable position is defined by the Range Limit max parameter, ID 0x07000001, or by the high position soft limit (PLM, PLM?), whichever is lower.

When a new home position is defined with DFH (p. 146), the maximum commandable position is automatically adapted to the appropriate new value.

TNR? (Get Number of Record Tables)

Description: Get the number of data recorder tables currently available on the controller.

Format: TNR?

Arguments: none

Response <uint> is the number of data recorder tables which are currently available

Notes: The answer gives the value of the Max Number of Data Recorder Channels parameter, ID 0x16000100.

For more information see "Data Recording" (p. 71).

TPC? (Get Number of Output Signal Channels)

Description: Get the number of output signal channels available on the controller.

Format: TPC?

Arguments: none

Response <uint> is the number of piezo channels which are available; the answer gives the value of the Number Of Piezo Channels parameter, ID 0x0E000B04. See "Accessible Items and Their Identifiers" (p. 50) for more information.

Note: Using the Sensor Enable parameter, ID 0x02000000, you can change the E-517 configuration in case of hardware changes, e.g. if you install additional sensor and/or amplifier channels in the system. If this parameter is changed, the Number Of Piezo Channels parameter is adapted automatically. E.g. if parameter 0x02000000 is set to "disabled" for a sensor channel, the corresponding piezo channel is disabled too and no longer included in the TPC? response. See "Configure Axes and Channels" (p. 33) for details.

TSC? (Get Number of Input Signal Channels)

Description: Get the number of input signal channels available on the controller.

Format: TSC?

Arguments: none

Response <uint> is the number of sensor channels which are available; the answer gives the value of the Number Of Sensor Channels parameter, ID 0x0E000B03. See "Accessible Items and Their Identifiers" (p. 50) for more information.

Note: Using the Sensor Enable parameter, ID 0x02000000, you can change the E-517 configuration in case of hardware changes, e.g. if you install additional sensor and/or amplifier channels in the system. If this parameter is changed, the Number Of Sensor Channels parameter is adapted automatically. E.g. if parameter 0x02000000 is set to "disabled" for a sensor channel, this sensor channel is no longer included in the TSC? response. See "Configure Axes and Channels" (p. 33) for details.

TSP? (Get Input Signal Position Value)

Description: Requests the current position of the selected input signal channel in physical units (µm).

Format: TSP? [{<InputSignalID>}]

Arguments: <InputSignalID> is one input signal channel of the controller

Response: {<InputSignalID>"="<float> LF}

where

<float> is the current position of the input signal channel, in physical units

Notes: The input signal channels to be queried with TSP? are the sensor channels of the piezo control electronics, IDs = 1 to 3 (actually available IDs depend on the response to the TSC? command (p. 199)).

TVI? (Tell Valid Character Set For Axis Identifiers)

Description: Gets a string with characters which can be used for axis identifiers.

Use SAI (p. 181) to change the axis identifiers and SAI? (p. 182) to ask for the current valid axis identifiers.

Format: TVI?

Arguments: none

Response: <string> is a list of characters

With the E-517, the string consists of
123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
_

TWC (Clear All Wave Related Triggers)

Description: Clears all output trigger settings for the wave generators (the settings made with TWS (p. 201)) by switching the signal state for all points to "low".

For a detailed description see "Wave Generator" (p. 84) and "Configuring Trigger Output" (p. 77).

Format: TWC

Arguments: none

Response: none

TWG? (Get Number of Wave Generators)

Description: Get the number of wave generators available on the controller.

Format: TWG?

Arguments: none

Response <uint> is the number of wave generators which are available

TWS (Set Trigger Line Action To Waveform Point)

Description: Associates output trigger line and trigger line action (signal state high or low) with waveform point.

The power-on default state of all points is low. Afterwards, the signal state of the trigger output line can be switched to "low" for all points using the TWC command (p. 200). It is recommended to use TWC before trigger actions are set with TWS.

Generator trigger mode must be activated for the selected trigger output line with the CTO command (p. 142).

See also "Wave Generator" (p. 84) and "Configuring Trigger Output" (p. 77).

Format: TWS {<TrigOutID> <PointNumber> <Switch>}

Arguments: <TrigOutID> is one digital output line of the controller, see below for details

<PointNumber> is one point in the waveform, starts with index 1, see below for the timing calculation

<Switch> is the signal state of the digital output line:
0 = low, 1 = high

Response: None

Notes: <TrigOutID> corresponds to the output lines DIO_O1 to DIO_O3, IDs = 1 to 3; see "Digital In/Out Socket" (p. 267).

With the E-517, there is a fixed one-to-one assignment of axes to the wave generators and the digital output lines. This means that

- the TWS settings for a digital output line will only be applied if the corresponding wave generator is started, but not if other wave generators run.

- with E-517.i1 models only DIO_O1 can be used.

- if axes / channels are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the output lines of the active axes and the corresponding wave generators can be used.

The number of arguments following the command mnemonic is limited to 32, i.e. you can define at most 10 trigger points per command line.

As long as the wave generator output is synchronized by servo-cycles, you can calculate the time for the trigger point in the waveform as follows:
time = generator cycle time * PointNumber
with
generator cycle time = Servo Update Time * WTR value
where
Servo Update Time in seconds is given by parameter 0x0E000200
WTR value is the wave table rate, i.e. the number of servo cycles the output of a waveform point takes, is given by parameter 0x13000109, default is 1

If you start the wave generator with the WGO start options given by bit 3, 4 or 5, the corresponding pulses are output in addition to the output pulses defined with TWS.

The number of wave generator output cycles during which trigger pulses are to be output can be set using the Number Of Trigger Cycles parameter, ID 0x18000100. The width of a trigger pulse is 30 µs by default. You can change the default pulse width using the Pulse Width parameter, ID 0x0E000900. Possible values are in the range of 10 to 150 µs.

Example: Send: TWS 2 1 1 2 2 0 2 3 0
Note: Sets trigger actions for the output line DIO_O2 (identifier 2), at waveform point 1 it is set high, points 2 and 3 are set low.

TWS? (Get Trigger Line Action At Waveform Point)

Description: Reading of the trigger line settings made with TWS (p. 201) for the waveform points.

To query the waveform shape, use the GWD? command (p. 154).

See also "Wave Generator" (p. 84) and "Configuring Trigger Output" (p. 77).

Format: TWS? [<StartPoint> [<NumberOfPoints> [{<TrigOutID>}]]]

Arguments: <StartPoint> is the start point in the waveform, starts with index 1

<NumberOfPoints> is the number of points to be read per digital output line

<TrigOutID> is one digital output line of the controller

Response: The trigger settings (signal states) in GCS array format, see the separate manual for GCS array, SM 146E, and the example below.

Example: The trigger settings for the output lines DIO_O2 (identifier 2) and DIO_O3 (identifier 3) are queried for the waveform points 1 to 20. The response gives the signal states of the digital output lines at the individual waveform points:
0 = low, 1 = high

```
tws? 1 20 2 3
# TYPE = 1
# SEPARATOR = 9
# DIM = 2
# SAMPLE_TIME = 40E-6
# NDATA = 20
# END_HEADER
0       0
0       0
0       0
0       0
0       1
0       0
0       0
0       0
0       0
0       0
0       0
0       0
```

```
0       0
0       0
1       0
0       0
0       0
0       0
0       0
0       1
```

VCO (Set Velocity Control Mode)

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

Format: VCO {<AxisID> <VelCtrlState>}

Arguments: <AxisID> is one axis of the controller

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

Response: none

Troubleshooting: Illegal axis identifier

Notes: When Velocity Control Mode is ON, the axis is driven with the velocity specified with VEL (p. 205). This is valid in ONLINE mode in open-loop and closed-loop operation, but the VCO settings are not effective when the wave generator is running.

Velocity Control Mode should be ON if you use the software-emulated on-target reading (ONT? command (p. 176)).

The current VCO settings are saved with the WPA command, in addition to the current parameter values and other settings. See the WPA description (p. 229) for details.

VCO? (Get Velocity Control Mode)

Description: Gets Velocity Control Mode of given axes.

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

Format: VCO? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

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

where

<VelCtrlState> is the current Velocity Control Mode of the axis:
0 = Velocity Control Mode OFF
1 = Velocity Control Mode ON

Troubleshooting: Illegal axis identifier

VEL (Set Closed-Loop Velocity)

Description: Set velocity of given axes.

VEL can be changed while the axis is moving.

Format: VEL {<AxisID> <Velocity>}

Arguments: <AxisID> is one axis of the controller

<Velocity> is the velocity value in physical units/s.

Response: none

Troubleshooting: Illegal axis identifiers, axis is under joystick control (via host PC)

Notes: The VEL setting only takes effect when Velocity Control Mode is ON for the given axis (see VCO command (p. 204)). The axis can be in closed-loop operation (servo on) or in open-loop operation (servo off). In open-loop operation, the velocity unit is V/s.

The velocity value must be ≥ 0.

VEL concerns the value of the Servo Loop Slew-Rate parameter, ID 0x07000200.

The velocity set with VEL is saved in volatile memory (RAM) only. To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA (p. 229). Changes not saved with WPA will be lost when the controller is powered down.

VEL? (Get Closed-Loop Velocity)

Description: Get the current velocity value.

If all arguments are omitted, gets current value of all axes.

Format: VEL? [{<AxisID>}]

Arguments: <AxisID> is one axis of the controller

Response: {<AxisID>"="<float> LF}

where

<float> is the current active velocity value in physical units / s.

Note: VEL? queries the current velocity value given by the Servo Loop Slew-Rate parameter, ID 0x07000200, in volatile memory. In open-loop operation, the unit of the velocity value is V/s.

VER? (Get Versions Of Firmware And Drivers)

Description: Get the versions of the E-517 firmware and the underlying drivers and libraries.

Format: VER?

Arguments: none

Response {<string1>":" <string2> [<string3>]LF}

where

<string1> is the name of the component
<string2> is the version information of the component <string1>
<string3> is an optional note

Notes: For E-517, VER? replies something like:

```
PI PI_GCS2_DLL.dll: V1.3.0.35
FW_DSP: 01.052
FW_FPGA: 01.060
FW_MCU: 01.008
```

VMA (Set Voltage Output High Limit)

Description: Gives the high limit of the piezo voltage ("soft limit").

Format: VMA {<OutputSignalID> <HighLimit>}

Arguments: <OutputSignalID> is one piezo channel of the piezo control electronics

<HighLimit> is the high limit of the output voltage, in V.

Response: None

Notes: VMA concerns the value of the Output Voltage High Limit parameter, ID 0x0C000001.

The voltage value set with VMA must be equal to or smaller than the voltage limit given by parameter 0x0B000008.

The voltage limit set with VMA is saved in volatile memory (RAM) only. To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA (p. 229). Changes not saved with WPA will be lost when the E-517 is powered down.

VMA? (Get Voltage Output High Limit)

Description: Get the high limit of the piezo voltage ("soft limit").

Format: VMA? [{<OutputSignalID>}]

Arguments: <OutputSignalID> is one piezo channel of the piezo control electronics

Response: {<OutputSignalID>"="<HighLimit> LF}

where

<HighLimit> is the high limit of the output voltage, in V.

Notes: VMA? queries the voltage limit given by the Output Voltage High Limit parameter, ID 0x0C000001, in volatile memory.

VMI (Set Voltage Output Low Limit)

Description: Gives the low limit of the piezo voltage ("soft limit").

Format: VMI {<OutputSignalID> <LowLimit>}

Arguments: <OutputSignalID> is one piezo channel of the piezo control electronics

<LowLimit> is the low limit of the output voltage, in V.

Response: None

Notes: VMI concerns the value of the Output Voltage Low Limit parameter, ID 0x0C000000.

The voltage value set with VMI must be equal to or larger than the voltage limit given by parameter 0x0B000007.

The voltage limit set with VMI is saved in volatile memory (RAM) only. To save the currently valid value to non-volatile memory, where it becomes the power-on default, you must use WPA (p. 229). Changes not saved with WPA will be lost when the E-517 is powered down.

VMI? (Get Voltage Output Low Limit)

Description: Get the low limit of the piezo voltage ("soft limit").

Format: VMI? [{<OutputSignalID>}]

Arguments: <OutputSignalID> is one piezo channel of the piezo control electronics

Response: {<OutputSignalID>"="<LowLimit> LF}

where

<LowLimit> is the low limit of the output voltage, in V.

Notes: VMI? queries the voltage limit given by the Output Voltage Low Limit parameter, ID 0x0C000000, in volatile memory.

VOL? (Get Voltage Of Output Signal Channel)

Description: Read the current voltage value of the given output signal channel.

Format: VOL? [{<OutputSignalID>}]

Arguments: <OutputSignalID> is one output signal channel of the controller

Response: {<OutputSignalID>"="<float> LF}

where

<float> is the current voltage value in V

Note: The output signal channels to be queried with VOL? are the piezo channels of the piezo control electronics, IDs = 1 to 3 (actually available IDs depend on the response to the TPC? command (p. 198)).

WAV (Set Waveform Definition)

Description: Define waveform of given type for given wave table.

To allow for flexible definition, a waveform (wave table contents) can be built up by adding "segments". Each segment is defined with a separate WAV command. To add a segment, the <AppendWave> argument (see below) is used to concatenate the new segment to the existing wave table contents.

A segment can be based on predefined "curve" shapes (see the <WaveType> argument below).

The CFG wave type is a special type with which you can configure additional parameters for a waveform. The settings should be made subsequent to the real waveform definition and will be applied to the waveform when the wave generator output starts. Some CFG settings can be saved to non-volatile memory as power-up defaults—if not, they are valid until a new WAV command is sent or the wave table content is cleared or the E-517 is powered down or rebooted.

Waveforms can not 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 waveform values are absolute values.

As long as the wave generator output is synchronized by servo-cycles and not paused by an external signal (see WGO (p. 221) for details), 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 in seconds is given by parameter 0x0E000200

WTR (wave table rate) value gives the number of servo cycles the output of a waveform point takes, default is 1

Number of Points is the length of the wave table (which is the sum of the lengths of all segments in this table)

See "How to work with the Wave Generator" (p. 84) for more information.

Format: WAV <WaveTableID> <AppendWave> <WaveType> <WaveTypeParameters>

Arguments: <WaveTableID> is the wave table identifier.

<AppendWave> This can be "X", "&" or "+":
"X" clears the wave table and starts writing with the first point in the table.
"&" appends the defined segment to the already existing wave table contents (i.e. concatenates a segment to lengthen the waveform).
"+" adds the content of the defined segment to the already existing wave table contents (i.e. the values of the defined points are added to the existing values of that points; if the defined segment is larger than the already existing wave table content, the difference will be concatenated to the wave table content).

<WaveType> The type of curve used to define the segment. This can be one of
"PNT" (user-defined curve)
"SIN_P"(inverted cosine curve)
"RAMP" (ramp curve)
"LIN" (single scan line curve)
"SIN" (sine curve)
"POL" (polynomial)
"TAN" (tangent curve)
"CFG" (special type which configures additionally parameters for a waveform)

<WaveTypeParameters> stands for the parameters of the curve and can be as follows:

For "PNT":

<SegStartPoint> <WaveLength> {<WavePoint>}

<SegStartPoint> The index of the segment starting point in the wave table. Must be 1.

<WaveLength> The length of the user-defined curve in points. The segment length, i.e. the number of points written to the wave table, is identical to the <WaveLength> value.

<WavePoint> The value of one single point.

For "SIN_P":

<SegLength> <Amp> <Offset> <WaveLength>
<StartPoint> <CurveCenterPoint>

<SegLength>: The length of the wave table segment in points. Only the number of points given by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the sine curve.

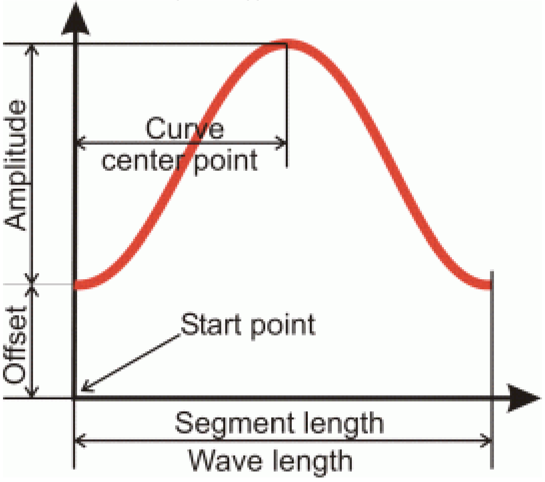
<Offset>: The offset of the sine curve.

<WaveLength>: The length of the sine curve in points (cycle duration).

<StartPoint>: The index of the starting point of the sine curve in the segment. Gives the phase shift. Lowest possible value is 0.

<CurveCenterPoint>: The index of the center point of the sine curve. Determines if the curve is symmetrical or not. Lowest possible value is 0.

Example (for more examples see "Defining Waveforms" (p. 92)):



For "RAMP":

<SegLength> <Amp> <Offset> <WaveLength>
<StartPoint> <SpeedUpDown> <CurveCenterPoint>

<SegLength>: The length of the wave table segment in points. Only the number of points given by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the ramp curve.

<Offset>: The offset of the ramp curve.

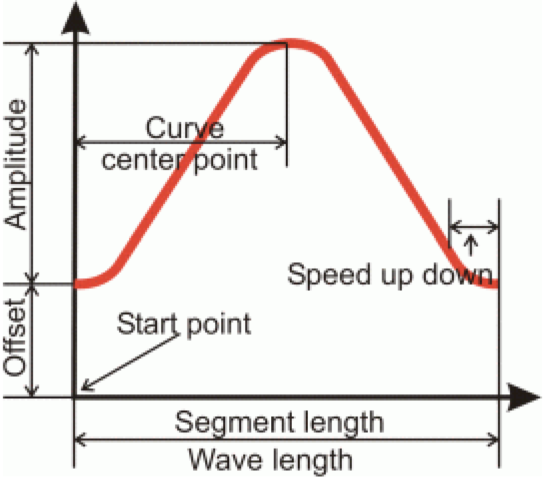
<WaveLength>: The length of the ramp curve in points (cycle duration).

<StartPoint>: The index of the starting point of the ramp curve in the segment. Gives the phase shift. Lowest possible value is 0.

<SpeedUpDown>: The number of points for speed-up and slow-down.

<CurveCenterPoint>: The index of the center point of the ramp curve. Determines if the curve is symmetrical or not. Lowest possible value is 0.

Example (for more examples see "Defining Waveforms" (p. 92)):



For "LIN":

<SegLength> <Amp> <Offset> <WaveLength>
<StartPoint> <SpeedUpDown>

<SegLength>: The length of the wave table segment in points. Only the number of points given by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <WaveLength> value, the missing points in the segment are filled with the endpoint value of the curve.

<Amp>: The amplitude of the scan line.

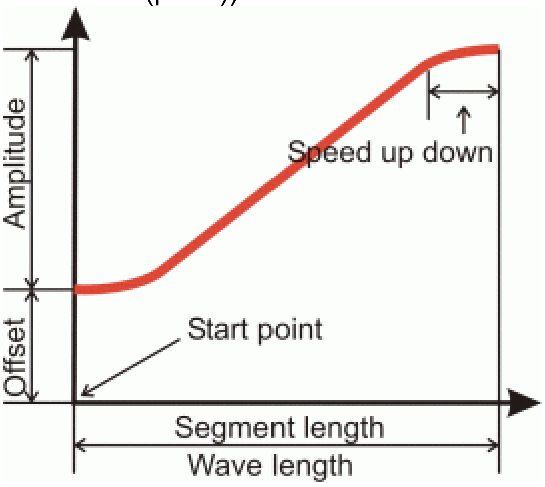
<Offset>: The offset of the scan line.

<WaveLength>: The length of the single scan line curve in points.

<StartPoint>: The index of the starting point of the scan line in the segment. Lowest possible value is 0.

<SpeedUpDown>: The number of points for speed-up and slow-down.

Example (for more examples see "Waveform Definition" (p. 92)):



For "SIN":

with

$$SIN = A \cdot \sin\left(\frac{2\pi}{N_p}(x - x_0) + \Phi\right) + B$$

the wave-type-dependent parameters are:

<SegStartPoint> < SegLength> <A> <N$_p$> <x$_0$> <ϕ>
<B>

<SegStartPoint> The index of the segment starting point in the wave table. Lowest possible value is 1. Writing to the wave table starts with this point. In the wave table, all points with an index smaller than <SegStartPoint> remain unchanged. This means that former written content for that points will not be deleted and is shown by GWD? and output by the wave generator.

<SegLength> The length of the wave table segment in points. Only the number of points given by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <N$_p$> value, the missing points in the segment are filled with recurrences of the sine curve.

<A> The amplitude of the sine curve.

<N$_p$> The wave length, i.e. the length of the sine curve in points (cycle duration).

<x$_0$> The index of the starting point of the sine curve in the segment. Gives the phase shift in points. Lowest possible value is 0.

<ϕ> The phase shift in degrees. Lowest possible value is 0.

<B> The offset of the sine curve.

For "POL":

with

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

the wave-type-dependent parameters are:

<SegStartPoint> <WaveLength> x_0 A_0 [{A_n}]

with n ≤ 5

<SegStartPoint> The index of the segment starting point in the wave table. Lowest possible value is 1. Writing to the wave table starts with this point. In the wave table, all points with an index smaller than <SegStartPoint> remain unchanged. This means that former written content for those points will not be deleted and is shown by GWD? and output by the wave generator.

<WaveLength> The wave length, i.e. the length of the polynomial curve in points (cycle duration). It gives the x value of the equation shown above.

The segment length, i.e. the number of points written to the wave table, results from the <SegStartPoint> and <WaveLength> values as follows:
Segment Length = SegStartPoint + WaveLength -1

For "TAN":

with

$$TAN = A \cdot \tan\left(\frac{2\pi}{N_p}\left(x - x_0\right) + \Phi\right) + B$$

the wave-type-dependent parameters are:

<SegStartPoint> < SegLength> <A> <N_p> <x_0> <ϕ> <B>

<SegStartPoint> The index of the segment starting point in the wave table. Lowest possible value is 1.Writing to the wave table starts with this point. In the wave table, all points with an index smaller than <SegStartPoint> remain unchanged. This means that former written content for those points will not be deleted and is shown by GWD? and output by the wave

generator.

<SegLength> The length of the wave table segment in points. Only the number of points given by <SegLength> will be written to the wave table. If the <SegLength> value is larger than the <N_p> value, the missing points in the segment are filled with recurrences of the tangent curve.

<A> The amplitude of the tangent curve.

<N_p> The wave length, i.e. the length of the tangent curve in points (cycle duration).

<x_0> The index of the starting point of the tangent curve in the segment. Gives the phase shift in points. Lowest possible value is 0.

<ϕ> The phase shift in degrees. Lowest possible value is 0.

<B> The offset of the tangent curve.

For CFG the wave-type-dependent parameters are:

<n> <m> <p> <k> <s> <L>

The settings for parameters <k> and <s> can be saved to non-volatile memory as power-up defaults and are also available via separate commands (see below). The settings for parameters <n>, <m>, <p> and <L> are valid until a new WAV command is sent or the wave table content is cleared or the controller is powered down or rebooted. If less than six parameters are specified, the values are assigned in order left to right and the unspecified parameters retain their previous values.

<n> integer, the length of the periodic waveform, i.e., how many points are to be included in one period of the waveform. Must be between 1 and 8192.

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

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

<k> integer, the number of interrupts (samples) before incrementing the current-point pointer. Must be equal to or greater than 1. If not specified 1 is used. The value can also be set / queried with the WTR / WTR? commands and is also available as "Wave generator table rate" parameter, ID 0x13000109 (use WPA to save it to non-volatile memory).

<s> double, amplitude shift after each complete period. In scanning applications this parameter is the distance between lines. If not specified 0 is used. The value can also be set / queried with the WOS / WOS? commands and is also available as "Curve Offset" parameter, ID 0x1300010B (use WPA to save it to non-volatile memory). Note that if the wave generator is started with the option "start at the endpoint of the last cycle" (bit 8), the E-517 at the end of each output cycle equates the offset value with the current generator output.

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

The period of the resulting output wave results from the following equation:

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

where t_i is the Servo Update Time in seconds is given by parameter 0x0E000200

The initial phase angle of the output waveform is:

$$\phi_s = \frac{p}{n} \times 360°$$

Response: None

Troubleshooting: Invalid wave table identifier

The total number of points for the waveform (which may consist of several segments) exceeds the available number of memory points.

Notes: Make sure not to exceed the output power limits of the hardware when defining the waveform. The amplifier output power of the piezo control electronics is proportional to the amplitude and the frequency of the control signal (e.g. wave generator output). See the data sheet or the User Manual of the piezo amplifiers used for specifications.

WAV? (Get Waveform Definition)

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

See "How to work with the Wave Generator (p. 84)" for more information.

Format: WAV? [{<WaveTableID> <WaveParameterID>}]

Arguments: <WaveTableID> is the wave table identifier.

<WaveParameterID> is the wave parameter ID, 1 = current wave table length in number of points; more parameters may be defined in the future

Response: {<WaveTableID> <WaveParameterID>"="<float> LF}

where

<float> depends on the <WaveParameterID>; gives the current number of waveform points in the wave table for <WaveParameterID> = 1

Troubleshooting: Invalid wave table identifier

WCL (Clear Wave Table Data)

Description: Clears the content of the given wave table.

As long as a wave generator is running, it is not possible to clear the connected wave table.

For a detailed description see "Wave Generator" (p. 84).

Format: WCL {<WaveTableID>}

Arguments: <WaveTableID> is the wave table identifier.

Response: none

WGC (Set Number Of Wave Generator Cycles)

Description: Sets the number of output cycles for the given wave generator (the output itself is started with WGO (p. 221)).

For a detailed description see "Wave Generator" (p. 84).

Format: WGC {<WaveGenID> <Cycles>}

Arguments: <WaveGenID> is the wave generator identifier

<Cycles> is the number of wave generator output cycles. If cycles = 0 then the waveform is output without period limitation until it is stopped by WGO or #24 (p. 133) or STP (p. 190).

Response: None

Notes: WGC sets the value of the Wave Generator Cycles parameter, ID 0x13000003, in volatile memory. You can set the wave generator cycles also by directly changing the parameter with SPA or SEP. Save the value with WPA to non-volatile memory, where it becomes the power-on default. The value of the parameter in volatile memory can be read with the WGC? command.

If the digital input line is used to trigger the wave generator output (see WGO for details), the count of output cycles continues with each generator restart. The generator will be stopped when the number of cycles given by WGC are completed, irrespective of any further trigger pulses.

WGC? (Get Number Of Wave Generator Cycles)

Description: Gets the number of output cycles set for the given wave generator.

For a detailed description see "Wave Generator" (p. 84).

Format: WGC? [{<WaveGenID>}]

Arguments: <WaveGenID> is the wave generator identifier

Response: {<WaveGenID>"="<Cycles> LF}

where

<Cycles> is the number of wave generator output cycles set with WGC (p. 220).

Notes: The number of wave generator cycles read by WGC? is the Wave Generator Cycles parameter value in volatile memory (ID 0x13000003).

If <WaveGenID> is omitted, all wave generators are queried.

WGO (Set Wave Generator Start/Stop Mode)

Description: Start and stop the specified wave generator in the given mode. In addition, one data recording cycle is started.

The number of output cycles can be limited by WGC (p. 220).

Using the WTR command (p. 231) or the CFG wave type (see WAV command (p. 209)), you can lengthen the individual output cycles of the waveform.

Using the WOS command (p. 228) or the CFG wave type (see WAV command (p. 209)), you can add an offset to the output of a wave generator.

The data recorder configuration can be made with DRC (p. 150). Recording can be restarted with WGR (p. 227).

Using the CTO command (p. 142), the E-517 can be set up for trigger output synchronized with the wave generator output. If you then start the wave generator with the WGO start options given by bit 3, 4 or 5 (see below), the corresponding pulses are output, in addition to any output pulses defined with TWS (p. 201). Note that with CTO, you can also change the polarity of the trigger output ("active high" is default).

The number of wave generator output cycles during which trigger pulses are to be output can be set using the Number Of Trigger Cycles parameter, ID 0x18000100. The width of a trigger pulse is 30 µs by default. You can change the default pulse width using the Pulse Width parameter, ID 0x0E000900. Possible values are in the range of 10 to 150 µs.

Keep in mind that wave generator output will continue even if the terminal or the program from which it was started is quit.

The #9 single-character command (p. 133) 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. With WGO? you can ask for the last-commanded wave generator start options (WGO settings (p. 221)).

For more information see "Wave Generator" (p. 84) and "Configuring Trigger Output" (p. 77).

Format: WGO {<WaveGenID> <StartMode>}

Arguments: <WaveGenID> is the wave generator identifier

With the E-517, there is a fixed one-to-one assignment of wave generators to axes and wave tables. E.g. starting wave generator 1 affects the first axis ("A" by default), and the waveform is taken from wave table 1.
With E-517.i1 models only wave generator 1 can be used. If axes / channels are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the wave generators of the active axes can be used.

<StartMode> is the start mode for the specified wave generator.

In the WGO command, you supply the start mode in hex or decimal format. When no bits are set (<StartMode> = 0), there is no wave generator output for the associated axis.
Note that the following bits cannot start the wave generator output by themselves: bit 3 (0x8 or 8), bit 4 (0x10 or 16), bit 5 (0x20 or 32), bit 8 (0x100 or 256), bit 12 (0x1000 or 4096), bit 13 (0x2000 or 8192) and bit 14 (0x4000 or 16384). These bits simply specify certain start

options and must always be combined with one of the start modes specified by bit 0 (0x1 or 1), bit 1 (0x2 or 2) and bit 2 (0x4 or 4). Note that if you should combine bits 0, 1 and 2, the wave generator starts with the mode given by the least significant bit. Do not combine bit 12 and 13 with bit 1 or 2 but only with bit 0 to start the wave generator.
See the examples below.

The start mode values in detail:

0: wave generator output is stopped. You can also use #24 (p. 133) or STP (p. 190) to stop the wave generator output, but WGO? (p. 226) will then still report the last commanded start mode.

bit 0 = 0x1 (hex format) or 1 (decimal format):
start wave generator output immediately, synchronized by servo cycle

bit 1 = 0x2 (hex format) or 2 (decimal format):
start wave generator output triggered by external signal, synchronized by servo cycle. The external signal is provided by the corresponding digital input line (see "Digital In/Out Socket" (p. 267)). The wave generator runs as long as the signal is HIGH and is paused as long as the signal is LOW.

bit 2 = 0x4 (hex format) or 4 (decimal format):
start wave generator output, triggered and synchronized by external signal.
The external signal is provided by the corresponding digital input line (see "Digital In/Out Socket" (p. 267)). The wave generator outputs one waveform point each time the signal is HIGH and is paused as long as the signal is LOW (with wave table rate = 1). If the wave table rate is larger than 1, a point is output for the corresponding number of HIGH pulses. See also WTR (p. 231).

bit 3 = 0x8 (hex format) or 8 (decimal format):
synchronized trigger pulse is output on the corresponding digital output line (see "Digital In/Out Socket" (p. 267)) when the wave generator outputs a new data point; start option

bit 4 = 0x10 (hex format) or 16 (decimal format):
synchronized trigger pulse is output on the corresponding digital output line (see "Digital In/Out Socket" (p. 267)) when the axis finishes each period (end of scan line, see WAV CFG); start option

bit 5 = 0x20 (hex format) or 32 (decimal format):
synchronized trigger pulse is output on the corresponding digital output line (see "Digital In/Out Socket" (p. 267)) when the axis reaches the amplitude limit (scan field limit, see WAV CFG); start option

bit 8 = 0x100 (hex format) or 256 (decimal format):
wave generator started at the endpoint of the last cycle; start option.
The second and all subsequent output cycles each start at the endpoint of the preceding cycle which makes this start option appropriate to scanning applications. The final position is the sum of the endpoint of the last output cycle and any offset defined with WAV (p. 209) for the waveform.

bit 12 = 0x1000 (hex format) or 4096 (decimal format):
wave generator output is triggered by external signal; start option
The external signal is provided by the corresponding digital input line (see "Digital In/Out Socket" (p. 267)). The wave generator is started the first time the external signal is HIGH and runs continuously even if the signal becomes LOW.

bit 13 = 0x2000 (hex format) or 8192 (decimal format):
wave generator output is stopped by external signal; start option
The external signal is provided by the corresponding digital input line (see "Digital In/Out Socket" (p. 267)). The wave generator is stopped when the external signal is HIGH. Further trigger pulses have no effect.

bit 14 = 0x4000 (hex format) or 16384 (decimal format):
wave generator output starts with the point at

which the wave generator was last stopped (as if the wave output had been paused); start option

Response: None

Troubleshooting: Invalid wave generator identifier

When the E-517 is in OFFLINE mode, the wave generator output can not be started. If the wave generator output was started in ONLINE mode, it remains active when switching to OFFLINE mode, but it is not used as control value. This means that it will not cause axis motion until switching back to ONLINE mode.

Motion commands like MOV (p. 170) or SVA (p. 191) (from command line or from a running macro) are not allowed when the wave generator output is active.

See "Control Value Generation" (p. 53) and "Control Modes" (p. 23) for details.

Example 1: Wave generator 1 is to be started by the first trigger pulse and stopped by the second trigger pulse, i.e. bit 12 and 13 are to be set on, contributing values of 0x1000 (dec.: 4096) and 0x2000 (dec.: 8192) to <StartMode>. Because bits 12 and 13 do not actually start the wave generator output, bit 0 as "start mode" must be set in addition, contributing 0x1 (dec.: 1). The resulting <StartMode> value is 0x3001 (dec.: 12289).

Send the following WGO command, with the <StartMode> given in hex format:
WGO 1 0x3001
The same command with <StartMode> given in decimal format:
WGO 1 12289

Example 2: Wave generator 1 is to be started with the option "start at the endpoint of the last cycle" (bit 8, value 0x100; dec.: 256). The start mode is to be "triggered by external signal, synchronized by servo cycle" (bit 1, value 0x2; dec.: 2). Hence the resulting <StartMode> value is in hex format

0x100 + 0x2 = 0x102
or in dec format
256 + 2 = 258

Send
WGO 1 0x102
or
WGO 1 258

WGO? (Get Wave Generator Start/Stop Mode)

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

The #9 single-character command (p. 133) 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. With WGO? you can ask for the last-commanded wave generator start options (WGO settings (p. 221)).

Note that #24 (p. 133) or STP (p. 191) stop the wave generator output, but do not reset the start/stop mode settings so that WGO? will still report the start mode which was set by the last WGO command (p. 221).

For more information see "Wave Generator" (p. 84).

Format: WGO? [{<WaveGenID>}]

Arguments: <WaveGenID> is the wave generator identifier

Response: {<WaveGenID>"="<StartMode> LF}

where

<StartMode> is the last commanded start mode of the wave generator, in decimal format. The value may be the sum of several start options and one start mode. See the WGO command description for details.

WGR (Starts Recording In Sync With Wave Generator)

Description: Restarts recording when the wave generator is running (a first data recording cycle is started with the WGO command (p. 221) which starts the wave generator output).

The data recorder configuration can be made with DRC (p. 150). The recorded data can be read with the DRR? command (p. 152).

For more information see "Wave Generator" (p. 84) and "Data Recording" (p. 71).

Format: WGR

Arguments: None

Response: None

Note: WGR restarts recording with the next occurring waveform startpoint.

WMS? (Get Maximum Number of Wave Table Points)

Description: Get the maximum number of points for the waveform written to the specified wave table.

Format: WMS? [{<WaveTableID>}]

Arguments: <WaveTableID> is the wave table identifier

Response {<WaveTableID>"="<NumberOfPoints> LF}

where

<NumberOfPoints> is the number of points available for the wave table

WOS (Set Wave Generator Output Offset)

Description: Sets an offset to the output of a wave generator. The current wave generator output is then created by adding the offset value to the current wave value:

Generator Output = Offset + Current Wave Value

Do not confuse the output-offset value set with WOS with the offset settings specified during waveform creation with WAV (p. 209). While the WAV offset affects only one segment (i.e. only one waveform), the WOS offset is added to all waveforms which are output by the given wave generator.

WOS sets the value of the Wave Offset parameter, ID 0x1300010b, in volatile memory.

If the wave generator is started with the option "start at the endpoint of the last cycle", the E-517 at the end of each output cycle equates the volatile value of the Wave Offset parameter with the current generator output.

Deleting wave table content with WCL (p. 219) has no effect on the settings for the wave generator output offset.

For more information see "Wave Generator" (p. 84).

Format: WOS {<WaveGenID> <Offset>}

Arguments: <WaveGenID> is the wave generator identifier

<Offset> is the wave generator output offset, any float number. See below for details.

Response: None

Notes: You can set the offset also using the CFG wave type (see WAV command (p. 209)) or by directly changing the Wave Offset parameter, ID 0x1300010b, with SPA (p. 185) or SEP (p. 183). Save the value with WPA (p. 229) to non-volatile memory, where it becomes the power-on default.

In closed-loop operation (servo ON), the given offset is interpreted as position value. In open-loop operation (servo OFF), the offset corresponds to a piezo voltage value.

WOS? (Get Wave Generator Output Offset)

Description: Reads the current value of the offset which is added to the wave generator output.

For more information see also "Wave Generator" (p. 84).

Format: WOS? [{<WaveGenID>}]

Arguments: <WaveGenID> is the wave generator identifier

Response: {<WaveGenID>"="<Offset> LF}

where

<Offset> is the current wave generator output offset.
In closed-loop operation (servo ON), the offset is interpreted as position value. In open-loop operation (servo OFF), the offset corresponds to a piezo voltage value.

Notes: The offset read by WOS? is the Wave Offset parameter value in volatile memory (ID 0x1300010b). This value results either from settings made with WOS (p. 228), WAV (p. 209), SPA (p. 185) or SEP (p. 183), or from internal calculation during the wave generator output; see WOS for details.

WPA (Save Parameters To Non-Volatile Memory)

Description: Write the currently valid value of a parameter of a given item from volatile memory (RAM) to non-volatile memory. The values saved this way become the power-on defaults.

Caution: If current parameter values are incorrect, the system may malfunction. Be sure that you have the correct parameter settings before using the WPA command.

RAM settings not saved with WPA will be lost when the controller is powered down or rebooted or when RPA (p. 179) is used to restore the parameters.

With HPA? (p. 157) you can obtain a list of all available parameters.

Use SPA? (p. 185) to check the current parameter settings in volatile memory.

See SPA (p. 185) for an example.

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

Arguments <Pswd> is the password for writing to non-volatile memory. See below for details.

<ItemID> is the item for which parameters are to be saved from volatile to non-volatile memory. See below for details.

<PamID> is the parameter ID, can be written in hexadecimal or decimal format. See below for details.

Response: none

Troubleshooting: Illegal item identifier, wrong parameter ID, invalid password, command level too low for write access

Notes: Parameters can be changed in volatile memory with SPA (p. 185), DFH (p. 146), IFC (p. 159), RTR (p. 180), SAI (p. 181), VEL (p. 205), VMA (p. 207), VMI (p. 208), WAV (p. 209), WGC (p. 220), WOS (p. 228) and WTR (p. 231).

When WPA is used without specifying any arguments except of the password, all currently valid parameter values are saved. With the E-517, the WPA command saves also the settings made by the following commands, although they are no parameters:

CSV (p. 141) ("Set GCS Syntax Version", selects E-517 or E-516 GCS syntax)
CTO (p. 142) ("Set Configuration Of Trigger Output")
DRC (p. 150) ("Set Data Recorder Configuration")
NLM (p. 173) ("Set Low Position Soft Limit", limits the axis travel range in closed-loop operation)
PLM (p. 177) ("Set High Position Soft Limit", limits

the axis travel range in closed-loop operation)
VCO (p. 204) ("Set Velocity Control Mode (On/Off)", activates/deactivates velocity limitation)

To have write access to the parameter(s), it might be necessary to switch to a higher command level using CCL (p. 139).

Warning: The number of write cycles of non-volatile memory is limited. Write default values only when necessary.

CAUTION: Avoid powering down the E-517 during the WPA procedure.

Available passwords, item IDs and parameter IDs

The password for writing to non-volatile memory is "100".

<ItemID> can be an axis identifier, a sensor channel, a piezo channel, a wave generator, a digital output line or the whole system; the item type depends on the parameter, see "Parameter Overview" (p. 252) for the item type concerned. See "Accessible Items and Their Identifiers" (p. 50) for the identifiers of the items.

Valid parameter IDs are given in "Parameter Overview" (p. 252).

WTR (Set Wave Generator Table Rate)

Description: Set wave generator table rate and interpolation type.

Format: WTR {<WaveGenID> <WaveTableRate> <InterpolationType>}

Arguments: <WaveGenID> is the wave generator identifier. See below for details.

<WaveTableRate> is the table rate to be used for wave generator output (unit: number of servo-loop cycles), must be an integer value larger than zero

<InterpolationType> When a wave generator table rate higher than 1 is set, this option can be used to apply interpolation to the wave generator output between wave table points. For the available interpolation types see below.

Response: None

Notes: <WaveGenID> details: With E-517.i1 models only wave generator 1 can be used. If axes / channels are deactivated (see "Configure Axes and Channels" (p. 33) for details), only the wave generators of the active axes can be used.

Interpolation types available:
The E-517 provides no interpolation so that the interpolation type must always be
0 = no interpolation

Using the WTR command, you can lengthen the individual output cycles of the waveform. As long as the wave generator output is synchronized by servo-cycles and not paused by an external signal (see WGO (p. 221) for details), 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 in seconds is given by parameter 0x0E000200
WTR (wave table rate) value gives the number of servo cycles the output of a waveform point takes, default is 1
Number of Points is the length of the waveform (i.e. the length of the wave table)

If the wave generator is started with WGO bit 2 (triggered and synchronized by external signal), the wave table rate gives the number of HIGH pulses the output of a point takes.

WTR sets the value of the Wave Generator Table Rate parameter, ID 0x13000109, in volatile memory. You can set the wave table rate also using the CFG wave type (see WAV command (p. 209)) or by directly changing the parameter with SPA (p. 185) or SEP (p. 183). Save the value with WPA (p. 229) to non-volatile memory, where it becomes the power-on default. The value of the parameter in volatile memory can be read with the WTR? command (p. 231).

For more information see "Wave Generator" (p. 84). An application example can be found in "Modifying the Wave Generator Table Rate" (p. 96).

WTR? (Get Wave Generator Table Rate)

Description: Gets the current wave generator table rate. Gets also the interpolation type used with table rate values > 1.

For more information see "Wave Generator" (p. 84). An application example can be found in "Modifying the Wave Generator Table Rate" (p. 96).

Format: WTR? [{<WaveGenID>}]

Arguments: <WaveGenID> is the wave generator identifier

Response: {<WaveGenID>"="<WaveTableRate> <InterpolationType> LF}

where

<WaveTableRate> is the table rate used for wave generator output (unit: number of servo-loop cycles)

<InterpolationType> interpolation type applied to outputs between wave table points when a wave generator table rate higher than 1 is set. See below for available interpolation types.

Notes: If <WaveGenID> is omitted, all wave generators are queried.

The wave table rate gives the number of servo-loop cycles used by the wave generator to output one waveform point. If the wave generator is started with WGO bit 2 (triggered and synchronized by external signal), the wave table rate gives the number of HIGH pulses the output of a point takes.

The wave table rate read by WTR? is the Wave Generator Table Rate parameter value in volatile

	memory (ID 0x13000109).
Interpolation types:	The E-517 provides no interpolation so that the interpolation type is always 0 = no interpolation

10.4 Error Codes

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

Controller Errors

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

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

39	PI_CNTR_INVALID_CNTR_NUMBER	Controller number invalid
40	PI_CNTR_NO_JOYSTICK_CONNECTED	No joystick configured
41	PI_CNTR_INVALID_EGE_AXIS	Invalid axis for electronic gearing, axis can not be slave
42	PI_CNTR_SLAVE_POSITION_OUT_OF_RANGE	Position of slave axis is out of range
43	PI_CNTR_COMMAND_EGE_SLAVE	Slave axis cannot be commanded directly when electronic gearing is enabled
44	PI_CNTR_JOYSTICK_CALIBRATION_FAILED	Calibration of joystick failed
45	PI_CNTR_REFERENCING_FAILED	Referencing failed
46	PI_CNTR_OPM_MISSING	OPM (Optical Power Meter) missing
47	PI_CNTR_OPM_NOT_INITIALIZED	OPM (Optical Power Meter) not initialized or cannot be initialized
48	PI_CNTR_OPM_COM_ERROR	OPM (Optical Power Meter) Communication Error
49	PI_CNTR_MOVE_TO_LIMIT_SWITCH_FAILED	Move to limit switch failed
50	PI_CNTR_REF_WITH_REF_DISABLED	Attempt to reference axis with referencing disabled
51	PI_CNTR_AXIS_UNDER_JOYSTICK_CONTROL	Selected axis is controlled by joystick
52	PI_CNTR_COMMUNICATION_ERROR	Controller detected communication error
53	PI_CNTR_DYNAMIC_MOVE_IN_PROCESS	MOV! motion still in progress
54	PI_CNTR_UNKNOWN_PARAMETER	Unknown parameter
55	PI_CNTR_NO_REP_RECORDED	No commands were recorded with REP
56	PI_CNTR_INVALID_PASSWORD	Password invalid
57	PI_CNTR_INVALID_RECORDER_CHAN	Data Record Table does not exist
58	PI_CNTR_INVALID_RECORDER_SRC_OPT	Source does not exist; number too low or too high

59	PI_CNTR_INVALID_RECORDER_SRC_CHAN	Source Record Table number too low or too high
60	PI_CNTR_PARAM_PROTECTION	Protected Param: current Command Level (CCL) too low
61	PI_CNTR_AUTOZERO_RUNNING	Command execution not possible while Autozero is running
62	PI_CNTR_NO_LINEAR_AXIS	Autozero requires at least one linear axis
63	PI_CNTR_INIT_RUNNING	Initialization still in progress
64	PI_CNTR_READ_ONLY_PARAMETER	Parameter is read-only
65	PI_CNTR_PAM_NOT_FOUND	Parameter not found in non-volatile memory
66	PI_CNTR_VOL_OUT_OF_LIMITS	Voltage out of limits
67	PI_CNTR_WAVE_TOO_LARGE	Not enough memory available for requested wave curve
68	PI_CNTR_NOT_ENOUGH_DDL_MEMORY	Not enough memory available for DDL table; DDL can not be started
69	PI_CNTR_DDL_TIME_DELAY_TOO_LARGE	Time delay larger than DDL table; DDL can not be started
70	PI_CNTR_DIFFERENT_ARRAY_LENGTH	The requested arrays have different lengths; query them separately
71	PI_CNTR_GEN_SINGLE_MODE_RESTART	Attempt to restart the generator while it is running in single step mode
72	PI_CNTR_ANALOG_TARGET_ACTIVE	Motion commands and wave generator activation are not allowed when analog target is active
73	PI_CNTR_WAVE_GENERATOR_ACTIVE	Motion commands are not allowed when wave generator is active
74	PI_CNTR_AUTOZERO_DISABLED	No sensor channel or no piezo channel connected to selected axis (sensor and piezo matrix)
75	PI_CNTR_NO_WAVE_SELECTED	Generator started (WGO) without having selected a wave table (WSL).
76	PI_CNTR_IF_BUFFER_OVERRUN	Interface buffer did overrun and command couldn't be received correctly

77	PI_CNTR_NOT_ENOUGH_RECORDED_DATA	Data Record Table does not hold enough recorded data
78	PI_CNTR_TABLE_DEACTIVATED	Data Record Table is not configured for recording
79	PI_CNTR_OPENLOOP_VALUE_SET_WHEN_SERVO_ON	Open-loop commands (SVA, SVR) are not allowed when servo is on
80	PI_CNTR_RAM_ERROR	Hardware error affecting RAM
81	PI_CNTR_MACRO_UNKNOWN_COMMAND	Not macro command
82	PI_CNTR_MACRO_PC_ERROR	Macro counter out of range
83	PI_CNTR_JOYSTICK_ACTIVE	Joystick is active
84	PI_CNTR_MOTOR_IS_OFF	Motor is off
85	PI_CNTR_ONLY_IN_MACRO	Macro-only command
86	PI_CNTR_JOYSTICK_UNKNOWN_AXIS	Invalid joystick axis
87	PI_CNTR_JOYSTICK_UNKNOWN_ID	Joystick unknown
88	PI_CNTR_REF_MODE_IS_ON	Move without referenced stage
89	PI_CNTR_NOT_ALLOWED_IN_CURRENT_MOTION_MODE	Command not allowed in current motion mode
100	PI_LABVIEW_ERROR	PI LabVIEW driver reports error. See source control for details.
200	PI_CNTR_NO_AXIS	No stage connected to axis
201	PI_CNTR_NO_AXIS_PARAM_FILE	File with axis parameters not found
202	PI_CNTR_INVALID_AXIS_PARAM_FILE	Invalid axis parameter file
203	PI_CNTR_NO_AXIS_PARAM_BACKUP	Backup file with axis parameters not found
204	PI_CNTR_RESERVED_204	PI internal error code 204
205	PI_CNTR_SMO_WITH_SERVO_ON	SMO with servo on
206	PI_CNTR_UUDECODE_INCOMPLETE_HEADER	uudecode: incomplete header

207	PI_CNTR_UUDECODE_NOTHING_TO_DECODE	uudecode: nothing to decode
208	PI_CNTR_UUDECODE_ILLEGAL_FORMAT	uudecode: illegal UUE format
209	PI_CNTR_CRC32_ERROR	CRC32 error
210	PI_CNTR_ILLEGAL_FILENAME	Illegal file name (must be 8-0 format)
211	PI_CNTR_FILE_NOT_FOUND	File not found on controller
212	PI_CNTR_FILE_WRITE_ERROR	Error writing file on controller
213	PI_CNTR_DTR_HINDERS_VELOCITY_CHANGE	VEL command not allowed in DTR Command Mode
214	PI_CNTR_POSITION_UNKNOWN	Position calculations failed
215	PI_CNTR_CONN_POSSIBLY_BROKEN	The connection between controller and stage may be broken
216	PI_CNTR_ON_LIMIT_SWITCH	The connected stage has driven into a limit switch, call CLR to resume operation
217	PI_CNTR_UNEXPECTED_STRUT_STOP	Strut test command failed because of an unexpected strut stop
218	PI_CNTR_POSITION_BASED_ON_ESTIMATION	While MOV! is running position can only be estimated!
219	PI_CNTR_POSITION_BASED_ON_INTERPOLATION	Position was calculated during MOV motion
230	PI_CNTR_INVALID_HANDLE	Invalid handle
231	PI_CNTR_NO_BIOS_FOUND	No bios found
232	PI_CNTR_SAVE_SYS_CFG_FAILED	Save system configuration failed
233	PI_CNTR_LOAD_SYS_CFG_FAILED	Load system configuration failed
301	PI_CNTR_SEND_BUFFER_OVERFLOW	Send buffer overflow
302	PI_CNTR_VOLTAGE_OUT_OF_LIMITS	Voltage out of limits
303	PI_CNTR_OPEN_LOOP_MOTION_SET_WHEN_SERVO_ON	Open-loop motion attempted when servo ON

304	PI_CNTR_RECEIVING_BUFFER_OVERFLOW	Received command is too long
305	PI_CNTR_EEPROM_ERROR	Error while reading/writing EEPROM
306	PI_CNTR_I2C_ERROR	Error on I2C bus
307	PI_CNTR_RECEIVING_TIMEOUT	Timeout while receiving command
308	PI_CNTR_TIMEOUT	A lengthy operation has not finished in the expected time
309	PI_CNTR_MACRO_OUT_OF_SPACE	Insufficient space to store macro
310	PI_CNTR_EUI_OLDVERSION_CFGDATA	Configuration data has old version number
311	PI_CNTR_EUI_INVALID_CFGDATA	Invalid configuration data
333	PI_CNTR_HARDWARE_ERROR	Internal hardware error
400	PI_CNTR_WAV_INDEX_ERROR	Wave generator index error
401	PI_CNTR_WAV_NOT_DEFINED	Wave table not defined
402	PI_CNTR_WAV_TYPE_NOT_SUPPORTED	Wave type not supported
403	PI_CNTR_WAV_LENGTH_EXCEEDS_LIMIT	Wave length exceeds limit
404	PI_CNTR_WAV_PARAMETER_NR	Wave parameter number error
405	PI_CNTR_WAV_PARAMETER_OUT_OF_LIMIT	Wave parameter out of range
406	PI_CNTR_WGO_BIT_NOT_SUPPORTED	WGO command bit not supported
555	PI_CNTR_UNKNOWN_ERROR	BasMac: unknown controller error
601	PI_CNTR_NOT_ENOUGH_MEMORY	not enough memory
602	PI_CNTR_HW_VOLTAGE_ERROR	hardware voltage error
603	PI_CNTR_HW_TEMPERATURE_ERROR	hardware temperature out of range
1000	PI_CNTR_TOO_MANY_NESTED_MACROS	Too many nested macros

1001	PI_CNTR_MACRO_ALREADY_DEFINED	Macro already defined
1002	PI_CNTR_NO_MACRO_RECORDING	Macro recording not activated
1003	PI_CNTR_INVALID_MAC_PARAM	Invalid parameter for MAC
1004	PI_CNTR_RESERVED_1004	PI internal error code 1004
1005	PI_CNTR_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
2000	PI_CNTR_ALREADY_HAS_SERIAL_NUMBER	Controller already has a serial number
4000	PI_CNTR_SECTOR_ERASE_FAILED	Sector erase failed
4001	PI_CNTR_FLASH_PROGRAM_FAILED	Flash program failed
4002	PI_CNTR_FLASH_READ_FAILED	Flash read failed
4003	PI_CNTR_HW_MATCHCODE_ERROR	HW match code missing/invalid
4004	PI_CNTR_FW_MATCHCODE_ERROR	FW match code missing/invalid
4005	PI_CNTR_HW_VERSION_ERROR	HW version missing/invalid
4006	PI_CNTR_FW_VERSION_ERROR	FW version missing/invalid
4007	PI_CNTR_FW_UPDATE_ERROR	FW update failed
5200	PI_CNTR_AXIS_NOT_CONFIGURED	Axis must be configured for this action

Interface Errors

0	COM_NO_ERROR	No error occurred during function call
-1	COM_ERROR	Error during com operation (could not be specified)
-2	SEND_ERROR	Error while sending data

-3	REC_ERROR	Error while receiving data
-4	NOT_CONNECTED_ERROR	Not connected (no port with given ID open)
-5	COM_BUFFER_OVERFLOW	Buffer overflow
-6	CONNECTION_FAILED	Error while opening port
-7	COM_TIMEOUT	Timeout error
-8	COM_MULTILINE_RESPONSE	There are more lines waiting in buffer
-9	COM_INVALID_ID	There is no interface or DLL handle with the given ID
-10	COM_NOTIFY_EVENT_ERROR	Event/message for notification could not be opened
-11	COM_NOT_IMPLEMENTED	Function not supported by this interface type
-12	COM_ECHO_ERROR	Error while sending "echoed" data
-13	COM_GPIB_EDVR	IEEE488: System error
-14	COM_GPIB_ECIC	IEEE488: Function requires GPIB board to be CIC
-15	COM_GPIB_ENOL	IEEE488: Write function detected no listeners
-16	COM_GPIB_EADR	IEEE488: Interface board not addressed correctly
-17	COM_GPIB_EARG	IEEE488: Invalid argument to function call
-18	COM_GPIB_ESAC	IEEE488: Function requires GPIB board to be SAC
-19	COM_GPIB_EABO	IEEE488: I/O operation aborted
-20	COM_GPIB_ENEB	IEEE488: Interface board not found
-21	COM_GPIB_EDMA	IEEE488: Error performing DMA
-22	COM_GPIB_EOIP	IEEE488: I/O operation started before previous operation completed
-23	COM_GPIB_ECAP	IEEE488: No capability for intended operation

-24	COM_GPIB_EFSO	IEEE488: File system operation error
-25	COM_GPIB_EBUS	IEEE488: Command error during device call
-26	COM_GPIB_ESTB	IEEE488: Serial poll-status byte lost
-27	COM_GPIB_ESRQ	IEEE488: SRQ remains asserted
-28	COM_GPIB_ETAB	IEEE488: Return buffer full
-29	COM_GPIB_ELCK	IEEE488: Address or board locked
-30	COM_RS_INVALID_DATA_BITS	RS-232: 5 data bits with 2 stop bits is an invalid combination, as is 6, 7, or 8 data bits with 1.5 stop bits
-31	COM_ERROR_RS_SETTINGS	RS-232: Error configuring the COM port
-32	COM_INTERNAL_RESOURCES_ERROR	Error dealing with internal system resources (events, threads, ...)
-33	COM_DLL_FUNC_ERROR	A DLL or one of the required functions could not be loaded
-34	COM_FTDIUSB_INVALID_HANDLE	FTDIUSB: invalid handle
-35	COM_FTDIUSB_DEVICE_NOT_FOUND	FTDIUSB: device not found
-36	COM_FTDIUSB_DEVICE_NOT_OPENED	FTDIUSB: device not opened
-37	COM_FTDIUSB_IO_ERROR	FTDIUSB: IO error
-38	COM_FTDIUSB_INSUFFICIENT_RESOURCES	FTDIUSB: insufficient resources
-39	COM_FTDIUSB_INVALID_PARAMETER	FTDIUSB: invalid parameter
-40	COM_FTDIUSB_INVALID_BAUD_RATE	FTDIUSB: invalid baud rate
-41	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_ERASE	FTDIUSB: device not opened for erase
-42	COM_FTDIUSB_DEVICE_NOT_OPENED_FOR_WRITE	FTDIUSB: device not opened for write
-43	COM_FTDIUSB_FAILED_TO_WRITE_DEVICE	FTDIUSB: failed to write device

-44	COM_FTDIUSB_EEPROM_READ_FAILED	FTDIUSB: EEPROM read failed
-45	COM_FTDIUSB_EEPROM_WRITE_FAILED	FTDIUSB: EEPROM write failed
-46	COM_FTDIUSB_EEPROM_ERASE_FAILED	FTDIUSB: EEPROM erase failed
-47	COM_FTDIUSB_EEPROM_NOT_PRESENT	FTDIUSB: EEPROM not present
-48	COM_FTDIUSB_EEPROM_NOT_PROGRAMMED	FTDIUSB: EEPROM not programmed
-49	COM_FTDIUSB_INVALID_ARGS	FTDIUSB: invalid arguments
-50	COM_FTDIUSB_NOT_SUPPORTED	FTDIUSB: not supported
-51	COM_FTDIUSB_OTHER_ERROR	FTDIUSB: other error
-52	COM_PORT_ALREADY_OPEN	Error while opening the COM port: was already open
-53	COM_PORT_CHECKSUM_ERROR	Checksum error in received data from COM port
-54	COM_SOCKET_NOT_READY	Socket not ready, you should call the function again
-55	COM_SOCKET_PORT_IN_USE	Port is used by another socket
-56	COM_SOCKET_NOT_CONNECTED	Socket not connected (or not valid)
-57	COM_SOCKET_TERMINATED	Connection terminated (by peer)
-58	COM_SOCKET_NO_RESPONSE	Can't connect to peer
-59	COM_SOCKET_INTERRUPTED	Operation was interrupted by a nonblocked signal
-60	COM_PCI_INVALID_ID	No device with this ID is present
-61	COM_PCI_ACCESS_DENIED	Driver could not be opened (on Vista: run as administrator!)

DLL Errors

-1001	PI_UNKNOWN_AXIS_IDENTIFIER	Unknown axis identifier

-1002	PI_NR_NAV_OUT_OF_RANGE	Number for NAV out of range--must be in [1,10000]
-1003	PI_INVALID_SGA	Invalid value for SGA--must be one of 1, 10, 100, 1000
-1004	PI_UNEXPECTED_RESPONSE	Controller sent unexpected response
-1005	PI_NO_MANUAL_PAD	No manual control pad installed, calls to SMA and related commands are not allowed
-1006	PI_INVALID_MANUAL_PAD_KNOB	Invalid number for manual control pad knob
-1007	PI_INVALID_MANUAL_PAD_AXIS	Axis not currently controlled by a manual control pad
-1008	PI_CONTROLLER_BUSY	Controller is busy with some lengthy operation (e.g. reference move, fast scan algorithm)
-1009	PI_THREAD_ERROR	Internal error--could not start thread
-1010	PI_IN_MACRO_MODE	Controller is (already) in macro mode--command not valid in macro mode
-1011	PI_NOT_IN_MACRO_MODE	Controller not in macro mode--command not valid unless macro mode active
-1012	PI_MACRO_FILE_ERROR	Could not open file to write or read macro
-1013	PI_NO_MACRO_OR_EMPTY	No macro with given name on controller, or macro is empty
-1014	PI_MACRO_EDITOR_ERROR	Internal error in macro editor
-1015	PI_INVALID_ARGUMENT	One or more arguments given to function is invalid (empty string, index out of range, ...)
-1016	PI_AXIS_ALREADY_EXISTS	Axis identifier is already in use by a connected stage
-1017	PI_INVALID_AXIS_IDENTIFIER	Invalid axis identifier
-1018	PI_COM_ARRAY_ERROR	Could not access array data in COM server
-1019	PI_COM_ARRAY_RANGE_ERROR	Range of array does not fit the number of parameters
-1020	PI_INVALID_SPA_CMD_ID	Invalid parameter ID given to SPA or SPA?

-1021	PI_NR_AVG_OUT_OF_RANGE	Number for AVG out of range--must be >0
-1022	PI_WAV_SAMPLES_OUT_OF_RANGE	Incorrect number of samples given to WAV
-1023	PI_WAV_FAILED	Generation of wave failed
-1024	PI_MOTION_ERROR	Motion error while axis in motion, call CLR to resume operation
-1025	PI_RUNNING_MACRO	Controller is (already) running a macro
-1026	PI_PZT_CONFIG_FAILED	Configuration of PZT stage or amplifier failed
-1027	PI_PZT_CONFIG_INVALID_PARAMS	Current settings are not valid for desired configuration
-1028	PI_UNKNOWN_CHANNEL_IDENTIFIER	Unknown channel identifier
-1029	PI_WAVE_PARAM_FILE_ERROR	Error while reading/writing wave generator parameter file
-1030	PI_UNKNOWN_WAVE_SET	Could not find description of wave form. Maybe WG.INI is missing?
-1031	PI_WAVE_EDITOR_FUNC_NOT_LOADED	The WGWaveEditor DLL function was not found at startup
-1032	PI_USER_CANCELLED	The user cancelled a dialog
-1033	PI_C844_ERROR	Error from C-844 Controller
-1034	PI_DLL_NOT_LOADED	DLL necessary to call function not loaded, or function not found in DLL
-1035	PI_PARAMETER_FILE_PROTECTED	The open parameter file is protected and cannot be edited
-1036	PI_NO_PARAMETER_FILE_OPENED	There is no parameter file open
-1037	PI_STAGE_DOES_NOT_EXIST	Selected stage does not exist
-1038	PI_PARAMETER_FILE_ALREADY_OPENED	There is already a parameter file open. Close it before opening a new file
-1039	PI_PARAMETER_FILE_OPEN_ERROR	Could not open parameter file
-1040	PI_INVALID_CONTROLLER_VERSION	The version of the connected controller is invalid

-1041	PI_PARAM_SET_ERROR	Parameter could not be set with SPA--parameter not defined for this controller!
-1042	PI_NUMBER_OF_POSSIBLE_WAVES_EXCEEDED	The maximum number of wave definitions has been exceeded
-1043	PI_NUMBER_OF_POSSIBLE_GENERATORS_EXCEEDED	The maximum number of wave generators has been exceeded
-1044	PI_NO_WAVE_FOR_AXIS_DEFINED	No wave defined for specified axis
-1045	PI_CANT_STOP_OR_START_WAV	Wave output to axis already stopped/started
-1046	PI_REFERENCE_ERROR	Not all axes could be referenced
-1047	PI_REQUIRED_WAVE_NOT_FOUND	Could not find parameter set required by frequency relation
-1048	PI_INVALID_SPP_CMD_ID	Command ID given to SPP or SPP? is not valid
-1049	PI_STAGE_NAME_ISNT_UNIQUE	A stage name given to CST is not unique
-1050	PI_FILE_TRANSFER_BEGIN_MISSING	A uuencoded file transferred did not start with "begin" followed by the proper filename
-1051	PI_FILE_TRANSFER_ERROR_TEMP_FILE	Could not create/read file on host PC
-1052	PI_FILE_TRANSFER_CRC_ERROR	Checksum error when transferring a file to/from the controller
-1053	PI_COULDNT_FIND_PISTAGES_DAT	The PiStages.dat database could not be found. This file is required to connect a stage with the CST command
-1054	PI_NO_WAVE_RUNNING	No wave being output to specified axis
-1055	PI_INVALID_PASSWORD	Invalid password
-1056	PI_OPM_COM_ERROR	Error during communication with OPM (Optical Power Meter), maybe no OPM connected
-1057	PI_WAVE_EDITOR_WRONG_PARAMNUM	WaveEditor: Error during wave creation, incorrect number of parameters
-1058	PI_WAVE_EDITOR_FREQUENCY_OUT_OF_RANGE	WaveEditor: Frequency out of range

-1059	PI_WAVE_EDITOR_WRONG_IP_VALUE	WaveEditor: Error during wave creation, incorrect index for integer parameter
-1060	PI_WAVE_EDITOR_WRONG_DP_VALUE	WaveEditor: Error during wave creation, incorrect index for floating point parameter
-1061	PI_WAVE_EDITOR_WRONG_ITEM_VALUE	WaveEditor: Error during wave creation, could not calculate value
-1062	PI_WAVE_EDITOR_MISSING_GRAPH_COMPONENT	WaveEditor: Graph display component not installed
-1063	PI_EXT_PROFILE_UNALLOWED_CMD	User Profile Mode: Command is not allowed, check for required preparatory commands
-1064	PI_EXT_PROFILE_EXPECTING_MOTION_ERROR	User Profile Mode: First target position in User Profile is too far from current position
-1065	PI_EXT_PROFILE_ACTIVE	Controller is (already) in User Profile Mode
-1066	PI_EXT_PROFILE_INDEX_OUT_OF_RANGE	User Profile Mode: Block or Data Set index out of allowed range
-1067	PI_PROFILE_GENERATOR_NO_PROFILE	ProfileGenerator: No profile has been created yet
-1068	PI_PROFILE_GENERATOR_OUT_OF_LIMITS	ProfileGenerator: Generated profile exceeds limits of one or both axes
-1069	PI_PROFILE_GENERATOR_UNKNOWN_PARAMETER	ProfileGenerator: Unknown parameter ID in Set/Get Parameter command
-1070	PI_PROFILE_GENERATOR_PAR_OUT_OF_RANGE	ProfileGenerator: Parameter out of allowed range
-1071	PI_EXT_PROFILE_OUT_OF_MEMORY	User Profile Mode: Out of memory
-1072	PI_EXT_PROFILE_WRONG_CLUSTER	User Profile Mode: Cluster is not assigned to this axis
-1073	PI_UNKNOWN_CLUSTER_IDENTIFIER	Unknown cluster identifier
-1074	PI_INVALID_DEVICE_DRIVER_VERSION	The installed device driver doesn't match the required version. Please see the documentation to determine the required device driver version.

-1075	PI_INVALID_LIBRARY_VERSION	The library used doesn't match the required version. Please see the documentation to determine the required library version.
-1076	PI_INTERFACE_LOCKED	The interface is currently locked by another function. Please try again later.

11 Controller Parameters

11.1 Parameter Handling

To adapt the E-517 to your application, you can modify parameter values. The parameters available depend on the controller firmware. With HPA? (p. 157) you can obtain a list of all available parameters with information about each (e.g. short descriptions). The volatile and non-volatile memory parameter values can be read with the SPA? (p. 188) or SEP? (p. 184) commands, respectively.
Note that many parameters are "protected" by higher command levels, as indicated in the "Command Level" column in the "Parameter Overview" table (p. 252). By going to command level 1 using the CCL command (p. 139), it is possible to change level-1 parameters. Parameters with level 2 or higher are reserved for service personnel.

Using the "general" modification commands SPA, RPA, SEP and WPA, all parameters for which the currently active command level has write permission can be changed in volatile memory (SPA (p. 185), RPA (p. 179)) or in non-volatile memory (SEP (p. 183), WPA (p. 229)). It is recommended that any modifications be first made with SPA, and when the controller runs well, saved using WPA. If you change the current interface settings with SPA, it may be necessary to close the current connection and re-open it with the new settings.

In addition to the "general" modification commands, there are commands which change certain specific parameters. All the commands listed below, except of IFS, change the parameter value only in volatile memory, and WPA must be used to save the value to non-volatile memory. IFS changes and saves the interface parameters directly in non-volatile memory only.

DFH (p. 146) ("User Origin", ID 0x07010200)
IFC (p. 159) (interface parameters: "RS-232 Baud Rate", ID 0x11000400, "GPIB Address", ID 0x11000900, "IP Address", ID 0x11000600, "IP Configuration", ID 0x11000800, "IP Mask", ID 0x11000700)
RTR (p. 180) ("Table Rate" for data recording, ID 0x16000000)
SAI (p. 181) ("Axis Name", i.e. the axis identifier, ID 0x07000600)
VEL (p. 205) ("Servo Loop Slew-Rate", ID 0x07000200)
VMA (p. 207) ("Output Voltage High Limit", ID 0x0C000001)
VMI (p. 208) ("Output Voltage Low Limit", ID 0x0C000000)
WAV (p. 209) (with CFG wave type; "Wave Offset", ID 0x1300010B, and "Wave Generator Table Rate", ID 0x13000109)
WGC (p. 220) ("Wave Generator Cycles", ID 0x13000003)
WOS (p. 228) ("Wave Offset", ID 0x1300010B)
WTR (p. 231) ("Wave Generator Table Rate", ID 0x13000109)

IFS (p. 163) (the same interface parameters as IFC, but in non-volatile memory)

!

CAUTION

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

It is strongly recommended to save the parameter values of the E-517 to a file on the host PC before you make any changes. This way the original settings can be restored if the new parameter settings will not prove satisfactory. To save the parameter values and to load them back to the E-517, use the *Device Parameter Configuration* window of PIMikroMove™. See "Create Backup File for Controller Parameters" (p. 12) for more information.

NOTE

The PIMikroMove™ host software gives access to parameter values in a more convenient way. Use its *Device Parameter Configuration* window to check/edit the individual parameters. See the PIMikroMove™ manual for more information.

Each parameter refers to one of the following item types (see the "Item Type Concerned" column in the table below):

- Whole system (E-517 interface / display module)
- Hardware components of the E-517
- Logical axes
- Input signal channels (sensor channels)
- Output signal channels (piezo channels)
- Wave generators
- Wave tables
- Data recorder tables
- Digital output lines

The "Max. No. of Items" column shows the maximum number of items for which the parameter is used. Example: With an E-517.i3, for parameter 0x02000200 the number of items is "3" which means that this parameter has different values for each of the 3 input signal channels (= sensor channels). For parameters which refer to the whole system the maximum number of items is always 1. See "Accessible Items and Their Identifiers" (p. 50) for the item identifiers to use with SPA, SEP or WPA when changing/saving parameter values or when asking for parameter values with the SPA? or SEP? commands.

Values stored in non-volatile memory are power-on defaults, so that the system can be used in the desired way immediately. Note that PI records the data files of every E-517 controller calibrated at the factory for easy restoration of original settings should that ever be necessary.

NOTES

With the E-517, the WPA command saves also the settings made by the following commands, although they are no parameters:

CSV (p. 141) ("Set GCS Syntax Version", selects E-517 or E-516 GCS syntax)
CTO (p. 142) ("Set Configuration Of Trigger Output")
DRC (p. 150) ("Set Data Recorder Configuration")
NLM (p. 173) ("Set Low Position Soft Limit", limits the axis travel range in closed-loop operation)
PLM (p. 177) ("Set High Position Soft Limit", limits the axis travel range in closed-loop operation)
VCO (p. 204) ("Set Velocity Control Mode (On/Off)", activates/deactivates velocity limitation)

The settings of the E-517 are also valid if the device is switched to the E-516 GCS syntax version. Not all parameters of the E-517 are present as E-516 parameters, and the IDs of the available parameters differ from the E-517 parameter IDs. See the E-516 documentation for available parameters.

11.2 Parameter Overview

See "Parameter Handling" (p. 250) for the meaning of the individual columns.

The content of the "Max. No. of Items" column is to be interpreted as follows:

- For "Item Type Concerned" = "Input Signal Channel" and "Output Signal Channel", it gives the number of channels.
  In principle, input signal channels are the sensor channels and output signal channels the piezo channels of the system, each with IDs from 1 to 3. The only exception are the parameters regarding the monitor signals for sensor position and piezo voltage (0x04000xxx): these signals are all input for the E-517. Hence the parameters refer to input signal channels which comprise sensor channels (IDs 1 to 3) as well as piezo channels (IDs 4 to 6).

- For "Item Type Concerned" = "Logical Axis", it gives the number of axes

- For "Item Type Concerned" = "Wave Generator", it gives the number of wave generators

- For "Item Type Concerned" = "Wave Table", it gives the number of wave tables

- For "Item Type Concerned" = "Data Recorder Table", it gives the number of data recorder tables

- For "Item Type Concerned" = "Digital Output Line", it gives the number of digital output lines

- For "Item Type Concerned" = "Hardware Component", it gives the number of internal boards of which an E-517 consists (1 = main board, 2 = display board).

Parameter ID (hexadecimal)	Command Level for write access	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description	Notes
0x02000000	1	Input Signal Channel	3	INT	Sensor enable	0 = Disabled 1 = Enabled Enables/disables also the corresponding axis and piezo channel
0x02000200	0	Input Signal Channel	3	FLOAT	Sensor correction 0 order (offset)	
0x02000300	0	Input Signal Channel	3	FLOAT	Sensor correction 1st order (gain)	

Parameter ID (hexadecimal)	Command Level for write access	Item Type Concerned	Max. No. of Items	Data Type	Parameter Description	Notes
0x04000500	2	Input Signal Channel	6	FLOAT	ADC gain	Parameters apply to the monitor signals for sensor position and piezo voltage
0x04000600	2	Input Signal Channel	6	FLOAT	ADC offset	
0x04000700	2	Input Signal Channel	6	FLOAT	HW gain	
0x04000800	2	Input Signal Channel	6	FLOAT	HW offset	
0X04000E00	1	Input Signal Channel	6	CHAR	LCD unit	
0X04000E01	1	Input Signal Channel	6	INT	LCD format	
0x05000000	0	Input Signal Channel	3	INT	Digital filter type	0 = No Filter 1 = IIR Filter 2 = FIR filter 99 = User
0x05000001	0	Input Signal Channel	3	FLOAT	Digital filter Bandwidth	
0x05000002	0	Input Signal Channel	3	INT	Digital filter order	
0x05000101	0	Input Signal Channel	3	FLOAT	User filter parameter A0	
0x05000102	0	Input Signal Channel	3	FLOAT	User filter parameter A1	
0x05000103	0	Input Signal Channel	3	FLOAT	User filter parameter B0	
0x05000104	0	Input Signal Channel	3	FLOAT	User filter parameter B1	
0x05000105	0	Input Signal Channel	3	FLOAT	User filter parameter B2	
0x07000000	2	Logical Axis	3	FLOAT	Range min limit	
0x07000001	2	Logical Axis	3	FLOAT	Range max limit	
0x07000200	0	Logical Axis	3	FLOAT	Servo loop slew rate [axis unit/s]	
0x07000500	3	Logical Axis	3	FLOAT	Position from sensor 1	
0x07000501	3	Logical Axis	3	FLOAT	Position from sensor 2	
0x07000502	3	Logical Axis	3	FLOAT	Position from sensor 3	
0x07000600	0	Logical Axis	3	CHAR	Axis name	
0x07000601	0	Logical Axis	3	CHAR	Axis unit	
0x07000900	0	Logical Axis	3	FLOAT	Tolerance	
0x07000C01	0	Logical Axis	3	FLOAT	Default voltage	
0x07010200	0	Logical Axis	3	FLOAT	User origin	
0x07010600	0	Logical Axis	3	INT	Enable SW On Target Signal	0 = Disabled 1 = Enabled
0x07030100	3	Logical Axis	3	INT	Axis servo mode selection	0 = Hardware 1 = Firmware

Parameter ID (hexadecimal)	Command Level for write access	Item Type Con-cerned	Max. No. of Items	Data Type	Parameter Description	Notes
0x09000000	3	Logical Axis	3	FLOAT	Driving with piezo 1	
0x09000001	3	Logical Axis	3	FLOAT	Driving with piezo 2	
0x09000002	3	Logical Axis	3	FLOAT	Driving with piezo 3	
0x0A000010	2	Output Signal Channel	3	FLOAT	DAC offset	
0x0A000020	2	Output Signal Channel	3	FLOAT	DAC gain	
0x0B000003	0	Output Signal Channel	3	FLOAT	Gain	
0x0B000007	2	Output Signal Channel	3	FLOAT	Min voltage	
0x0B000008	2	Output Signal Channel	3	FLOAT	Max voltage	
0x0B00000A	0	Output Signal Channel	3	FLOAT	Offset	
0x0C000000	0	Output Signal Channel	3	FLOAT	Output voltage low limit [V]	
0x0C000001	0	Output Signal Channel	3	FLOAT	Output voltage high limit [V]	
0x0D000000	2	System	1	CHAR	Device serial number	
0x0D000100	2	Hardware Component	2	CHAR	Hardware serial number	
0x0D000200	2	Hardware Component	2	CHAR	Hardware name	
0x0D000400	2	Hardware Component	2	INT	Hardware revision	
0x0D000600	2	System	1	INT	Device ID	
0x0E000100	3	System	1	FLOAT	Sensor sampling time	
0x0E000200	3	System	1	FLOAT	Servo update time	

Parameter ID (hexadecimal)	Command Level for write access	Item Type Con-cerned	Max. No. of Items	Data Type	Parameter Description	Notes
0x0E000900	0	System	1	INT	Pulse width (µs) for all trigger output (10, 20, ... , up to 150 µs)	
0x0E000B03	3	System	1	INT	Number of sensor channels	
0x0E000B04	3	System	1	INT	Number of piezo channels	
0x0E000B05	3	System	1	INT	Number of trigger outputs	
0x0E000D00	0	System	1	INT	LCD brightness (10~31)	
0x0E000D01	0	System	1	INT	LCD contrast (5~31)	
0x11000400	0	System	1	INT	RS232 Baudrate	
0x11000600	0	System	1	CHAR	Ethernet IP address	
0x11000700	0	System	1	CHAR	Ethernet IP mask	
0x11000800	0	System	1	INT	Ethernet IP configuration	
0x11000900	0	System	1	INT	GPIB address	
0x11000B00	2	System	1	CHAR	Ethernet IP MAC address	
0x12000001	2	System	1	INT	GPIB enable	
0x13000003	0	Wave Generator	3	INT	Wave generator cycles	
0x13000004	3	System	1	INT	Max Wave Points	
0x13000109	0	Wave Generator	3	INT	Wave generator table rate	
0x1300010A	3	System	1	INT	Number of wave tables	
0X1300010B	0	Wave Generator	3	FLOAT	Wave offset	
0x13000201	3	Wave Table	3	INT	Max Wave Points of table i	
0x16000000	0	System	1	INT	Table rate	

Parameter ID (hexadecimal)	Command Level for write access	Item Type Con-cerned	Max. No. of Items	Data Type	Parameter Description	Notes
0x16000100	3	System	1	INT	Maximum number of channels	
0x16000200	3	System	1	INT	Maximum record points	
0x16000201	3	Data Recorder Table	3	INT	Maximum record points of table i	
0x18000100	0	Digital Output Line	3	INT	Number of trigger cycles	Only applied to trigger output in conjunction with the wave generator

12 Troubleshooting

Communication with controller does not work

Communication cable is wrong or defective

⇒ Check cable. Does it work properly with another device?

For RS-232, a null-modem cable must be used.

For TCP/IP connections: Connect the controller to an Ethernet connector in the PC using the included cross-over network cable. When connecting to a network hub or router, it might be necessary to use a straight-through network cable.

Communications has not been established yet

⇒ TCP/IP communication: If no DHCP server is available on the network or if a point-to-point connection between host PC and controller is being used, after power-on or reboot it might take a period of about 30 seconds before communication is possible.

The interface is not configured correctly

⇒ With the RS-232 interface, check port and baud rate (depending on your controller, the baud rate may be set via DIP switches on the front panel or via a controller parameter). It is recommended that the host PC have a "genuine" RS-232 interface on board. If the host PC uses a USB-to-serial adapter instead, data loss could occur during communication, especially when transferring large amounts of data.

⇒ With the GPIB (IEEE 488) interface, check the device address.

⇒ 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-517 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. Presently, only one port (50000) is available on the E-517 so that only one application at a time can use the TCP/IP connection.

⇒ The first time you connect over the USB interface, be sure you are logged on the PC as a user having administrator rights. After the E-517 is powered on, a message will appear on the PC screen saying that new hardware has been detected. Follow the on-screen instructions and insert the E-517 CD. The required FTDI hardware drivers are found in the \USB_Serial_Driver directory.

Controller was power-cycled or rebooted

⇒ With TCP/IP and USB connections, communication can not be maintained after the E-517 is power-cycled or rebooted. The connection must then be closed and reopened.

Another program is using the interface

⇒ Close the other program.

Specific software has problems

⇒ See if the system works with some other software, e.g. a terminal or development environment. You can, for example, test the communication by simply starting a terminal program, e.g. PI Terminal, and entering commands like *IDN? or HLP?. Note that multi-character commands are transferred as terminated by a LF (line feed) character and are executed only after the LF is received.

Stage does not move

Cable not connected properly

⇒ Check the connecting cable(s)

Stage or stage cable is defective

⇒ Exchange stage with a working stage to test a new combination of controller and stage:

With E-7xx controllers, this is only possible with stages which are equipped with ID-chips.
With E-861 controllers, the encoder hardware of closed-loop systems must be identical or the parameters of the GEMAC interpolation circuit must be adapted, see "GEMAC Parameter Adjustment" for more information.

Wrong command or wrong syntax

⇒ Check the error code with the ERR? command (p. 153). "Error Codes" (p. 234) gives the complete error reference.

Wrong axis commanded

⇒ Check if the correct axis identifier is used and if the commanded axis is that of the desired stage (axis identifier also required with single-axis systems!)

Move commands, macros or wave generator commands provoke errors and are ignored

⇒ The applicable control sources for the axis motion depend on the current control mode of a piezo channel (OFFLINE or ONLINE mode, see "Control Modes" (p. 23) for details).

In OFFLINE mode, move commands (received via interface or from a running macro), wave generator output and trackball target settings are ignored and may provoke an error message.

In ONLINE mode, the axis motion can be commanded by multiple sources (see "Control Value Generation" (p. 53) for details): move commands received via interface or from a running macro (SVA (p. 191), SVR (p. 195), MOV (p. 170), MVR (p. 172), GOH (p. 154), IMP (p. 165), STE (p. 189)), trackball settings on the CHANNEL SETTING screen of the E-517 display, wave generator output.

While move commands and trackball settings can be used in parallel to command axis motion, they are not accepted when the wave generator is running for the axis.

Macros can run in OFFLINE and ONLINE mode, but move commands from macros are only accepted in ONLINE mode and only when the wave generator is not running.

When the E-517 is in OFFLINE mode, the wave generator output can not be started. If the wave generator output was started in ONLINE mode, it remains active when switching to OFFLINE mode, but it is not used as control value. This means that it will not cause axis motion until switching back to ONLINE mode.

Incorrect control mode of the piezo channel

⇒ Check the current control mode using the ONL? command (p. 175) or on the main screen of the E-517 display. The current control mode of a piezo channel determines the applicable control sources for the output voltage and hence for the axis motion. See "Control Modes" (p. 23) for details.

Incorrect configuration

⇒ Check the parameter settings on the E-517 with the SPA? (p. 188) and SEP? (p. 184) commands.

Incorrect GCS syntax version

⇒ Check the current GCS syntax version using the CSV? command (p. 141) or on the main screen of the E-517 display ("E517" or "E516", in the top right corner of the display). The current GCS syntax version determines the applicable GCS commands and software tools (except for the PITerminal, the software on the E-517 CD can not be used when the E-517 is switched to the E-516 GCS syntax version). See "Select Command Set Version" (p. 44) for details.

The high voltage output of the piezo control electronics is deactivated, while the communication with the E-517 is still possible

⇒ The high voltage output of the piezo control electronics may be deactivated automatically when an internal temperature sensor detects overheating. To reactivate the high voltage output, let the device cool down and power-cycle or reboot the E-517.
Note that the wave generator output will continue even if the high voltage output is deactivated, i.e. if a certain number of output cycles was set, they may have already finished when the high voltage output is reactivated.

How to avoid overheating:
Keep the ambient temperature at a noncritical value: Note that the difference between ambient temperature and internal temperature of the piezo control electronics normally is about 20 Centigrade (36 Fahrenheit) degrees.

Place the system in a location with adequate ventilation. Allow at least 10 cm (4 inches) clearance from the top and the rear of the unit and 5 cm (2 inches) from each side. If this is not possible, keep the ambient temperature low. Never cover the ventilation openings of the piezo control electronics as this will impede ventilation.
When using the wave generator, it is recommended to reduce the frequency and/or the amplitude and/or the output duration to avoid overheating. See "Wave Generator" (p. 84) for more information.

Unsatisfactory system performance

The sensor values are not reliable, and the whole system is instable.

⇒ Only thermally stable systems can have the best performance. For a thermally stable system, power on the E-517 at least one hour before you start working with it.

Custom software accessing PI drivers does not run.

Wrong combination of driver routines/VIs

⇒ Check if system runs with Terminal program. If yes read the software manual and compare sample code from the E-517 CD to check the necessary driver routines.

13 Customer Service

Call your PI representative or write to info@pi.ws; please have the following information about your system ready:

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

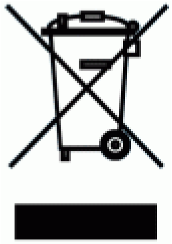
14 Old Equipment Disposal

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

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

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

Physik Instrumente (PI) GmbH & Co. KG
Auf der Römerstr. 1
76228 Karlsruhe, Germany



15 Technical Data

15.1 Specifications

	E-517.i1	**E-517.i3**
Function	Digital operation module	Digital operation module
Channels	1	3
Processor	DSP 60 MHz	DSP 60 MHz
Sampling rate, sensor	25 kHz, 8-times oversampling	25 kHz, 8-times oversampling
Thermal drift	Stability: 0.2 mV	Stability: 0.2 mV
Linearity @ nominal range	0.01%	0.01%
Resolution	DAC: 24 bit, ±12 V ADC: 18 bit, sampling	DAC: 24 bit, ±12 V ADC: 18 bit, sampling
Interfaces and operation		
Interfaces/communication	Ethernet (TCP/IP), USB, RS-232, IEEE 488	Ethernet (TCP/IP), USB, RS-232, IEEE 488
I/O ports	1 trigger input 1 trigger output 5 V MDR14 connector	3 trigger inputs 3 trigger outputs 5 V MDR14 connector
Command set	PI General Command Set (GCS)	PI General Command Set (GCS)
User software	PIMikroMove™	PIMikroMove™
Software drivers	Lab VIEW drivers, Windows and Linux Libraries (DLL)	Lab VIEW drivers, Windows and Linux Libraries (DLL)
Supported functionality	Wave generator, data recorder, macro programming	Wave generator, data recorder, macro programming
Display	LCD display for monitor signals (position and voltage), states and trackball menus	LCD display for monitor signals (position and voltage), states and trackball menus
Manual control	Operation via trackball	Operation via trackball

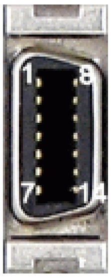
	E-517.i1	E-517.i3
Miscellaneous		
Operating temperature range	+5° to +50° C	+5° to +50° C
Dimensions	21HP/3U	21HP/3U
Mass	0.37 kg	0.37 kg
Operating voltage	E-500 system	E-500 system

15.2 Pin Assignments

15.2.1 Digital In/Out Socket

Connector type: MDR14

Pin	Signal	Function
1	GND	GND
2	nc	not connected
3	output	DIO_O3 (TTL, this digital output line can be configured with the CTO (p. 142), TWS (p. 201) and WGO (p. 221) commands for triggering tasks, identifier is 3); with E-517.i3 only
4	output	DIO_O2 (TTL, this digital output line can be configured with CTO, TWS and WGO for triggering tasks, identifier is 2); with E-517.i3 only
5	output	DIO_O1 (TTL, this digital output line can be configured with CTO, TWS and WGO for triggering tasks, identifier is 1)
6	nc	not connected
7	nc	not connected
8	reserved	reserved 2
9	reserved	reserved 1
10	nc	not connected
11	input	DIO_I3 (TTL, digital input line 3, for start and synchronization of wave generator 3 , see WGO (p. 221) command); with E-517.i3 only
12	input	DIO_I2 (TTL, digital input line 2, for start and synchronization of wave generator 2, see WGO); with E-517.i3 only
13	input	DIO_I1 (TTL, digital input line 1, for start and synchronization of wave generator 1, see WGO)
14	reserved	reserved

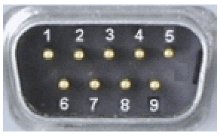


The signal state of the DIO_I1 to DIO_I3 input lines can be queried with the DIO? command (p. 149).

15.2.2 RS-232 Socket

Connector type: Sub-D 9 pin (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



15.2.3 Main Connector

Connector type: 32-pin connector, DIN 41612

Row	Pin a	Pin c
1	AGND	AGND
2	IN: ch1	OUT: ch1
3	IN: ch2	OUT: ch2
4	IN: ch3	OUT: ch3
5	AGND (analog)	AGND (analog)
6	IN: Sensor ch1	OUT: PZT ch1
7	IN: Sensor ch2*	OUT: PZT ch2*
8	IN: Sensor ch3*	OUT: PZT ch3*
9	AGND (analog)	AGND (analog)
10	nc	nc
11	nc	nc
12	nc	nc
13	VC/EC ch1 (servo mode)	Overflow ch1
14	VC/EC ch2* (servo mode)	Overflow ch2*
15	VC/EC ch3* (servo mode)	Overflow ch3*
16	On Target ch1	nc
17	On Target ch2*	nc
18	On Target ch3*	nc
19	nc	nc
20	nc	nc
21	nc	nc
22	nc	nc
23	Sync_100k	Sync_200k
24	nc	nc
25	reserved	nc
26	internal use (+27 V)	nc
27	internal use (+130V)	internal use (-33 V)
28	-15 V	-15 V
29	+15 V	+15 V
30	+5 V (VCC)	+5V (VCC)
31	GND (VCC)	GND (VCC)
32	AGND (analog)	AGND (analog)

nc - no connection
AGND - Analog Ground, isolated from the GND(VCC) (pins 31ac)
* on E-517.i1, lines for channels 2 and 3 should not be used

NOTE

This connector is the interface to the analog piezo control electronics. Normally, it is not accessible because the E-517 comes installed in the chassis of the analog piezo control electronics.