Table of Contents

1.0 System Overview ................................................................................................................. 1
  1.1 General Description ............................................................................................................... 1
  1.2 Part Numbers ..................................................................................................................... 1
    1.2.1 CONEX-AGAP ............................................................................................................. 1
    1.2.2 Accessories .................................................................................................................. 1
  1.3 CONEX-AGAP ..................................................................................................................... 2
    1.3.1 Delivered Items .......................................................................................................... 2
    1.3.2 Specifications ............................................................................................................. 2
    1.3.3 Dimensions .................................................................................................................. 3
  1.4 System Environmental Specifications ................................................................................. 3
  1.5 Connector Identification ...................................................................................................... 3
  1.6 USB Communication Settings ............................................................................................ 4

2.0 Programming .......................................................................................................................... 5
  2.1 State Diagram ...................................................................................................................... 5
  2.2 Command Syntax ............................................................................................................... 7
  2.3 Command Execution Time ................................................................................................. 7
  2.4 Command Set ..................................................................................................................... 7
    DB[a] — Set/Get corrector deadband .................................................................................. 9
    DD[a] — Set/Get deadband settling time .......................................................................... 10
    ID — Set/Get stage identifier .......................................................................................... 11
    JA[a] — Jog motion ........................................................................................................... 12
    KI[a] — Set/Get integral gain .......................................................................................... 13
    KP[a] — Set/Get proportional gain .................................................................................. 14
    KY[a] — Set/Get calibration coefficients ......................................................................... 15
    KZ[a] — Set/Get calibration coefficients ......................................................................... 16
    LF — Set/Get low pass filter frequency .......................................................................... 17
    MM — Enter/Leave DISABLE state .................................................................................. 18
    PA[a] — Move absolute ................................................................................................... 19
    PR[a] — Move relative ..................................................................................................... 20
    PW — Enter/Leave CONFIGURATION state ..................................................................... 21
    RS — Reset controller ...................................................................................................... 22
    RS## — Reset controller’s address .................................................................................. 23
    SA — Set/Get controller’s RS-485 address ....................................................................... 24
    SL[a] — Set/Get negative software limit .......................................................................... 25
    SR[a] — Set/Get positive software limit .......................................................................... 26
    ST — Stop motion ............................................................................................................. 27
    SU — Set/Get system resolution ...................................................................................... 28
TB — Get command error string ................................................................. 29
TE — Get last command error ................................................................. 30
TH[a] — Get target position .................................................................. 31
TP[a] — Get current position ................................................................. 32
TS — Get positioner error and controller state .................................... 33
VE — Get controller revision information ............................................. 35
XR[a] — Step motion ........................................................................... 36
XU[a] — Set/Get step motion size ......................................................... 37
ZT — Get all configuration parameters ............................................... 38

3.0 Connector interfaces ........................................................................ 39
3.1 USB (Male mini-USB) ........................................................................ 39

Service Form ......................................................................................... 41
CONEX-AGAP
Agilis-D Controller with Strain Gages Feedback

1.0 System Overview

1.1 General Description
The CONEX-AGAP is a two-axis motion controller/driver for piezo actuator with Strain Gages Feedback. It provides a very compact and low-cost solution for driving a variety of Newport Agilis-type piezo stages from a PC.

Communication with the CONEX-AGAP is achieved via an USB port (requires Windows™ operating system). A Windows™ based software enables basic motion. Advanced application programming is simplified by an ASCII command interface and a set of three-letter mnemonic commands.

1.2 Part Numbers

1.2.1 CONEX-AGAP

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONEX-AG-M100-D</td>
<td>CONEX-AGAP controller with mirror mount.</td>
</tr>
</tbody>
</table>

1.2.2 Accessories

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONEX-USB</td>
<td>USB cable, 1.8 m length</td>
</tr>
<tr>
<td>CONEX-BP</td>
<td>Base plate to attach up to 6 CONEX controllers</td>
</tr>
</tbody>
</table>
1.3 CONEX-AGAP

1.3.1 Delivered Items

- CONEX-AG-M100D Controller box with stage (cable length: 1 m)
- CONEX-USB USB cable, 1.8 m length
- CONEX-MOTION CD-Rom

1.3.2 Specifications

<table>
<thead>
<tr>
<th>General Description</th>
<th>Agilis controller with strain gages feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Capability</td>
<td>Piezo motors, open or closed loop</td>
</tr>
<tr>
<td>Piezo Output Voltage</td>
<td>35 Vpeak</td>
</tr>
<tr>
<td>Control loop</td>
<td>– Digital PI loop</td>
</tr>
<tr>
<td></td>
<td>– 50 Hz servo rate</td>
</tr>
<tr>
<td>Motion</td>
<td>Absolute and relative motion in open or closed loop</td>
</tr>
<tr>
<td>Computer interface</td>
<td>– USB (requires Windows™ operating system)</td>
</tr>
<tr>
<td>Programming</td>
<td>– 25+ intuitive, 2- or 3-letter ASCII commands</td>
</tr>
<tr>
<td></td>
<td>– Command set includes software limits…</td>
</tr>
<tr>
<td>Dedicated inputs</td>
<td>– Analog signals from gages</td>
</tr>
<tr>
<td>Status display</td>
<td>Two color LED</td>
</tr>
<tr>
<td>Communication rate</td>
<td>50 Hz Max. (USB)</td>
</tr>
<tr>
<td>Internal safety feature</td>
<td>Watchdog timer</td>
</tr>
<tr>
<td>Consumption</td>
<td>+5V (USB): &lt; 0.5 A</td>
</tr>
</tbody>
</table>
1.3.3 Dimensions

1.4 System Environmental Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating temperature</td>
<td>15 °C to 35 °C</td>
</tr>
<tr>
<td>Operating humidity</td>
<td>20% to 85% relative humidity, non-condensing</td>
</tr>
<tr>
<td>Location</td>
<td>Indoor use only</td>
</tr>
</tbody>
</table>

1.5 Connector Identification

<table>
<thead>
<tr>
<th>Connector</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB</td>
<td>mini USB connector</td>
</tr>
<tr>
<td>LED</td>
<td>Status LED</td>
</tr>
<tr>
<td>STAGE</td>
<td>Stage entry cable</td>
</tr>
<tr>
<td>Cable retainer</td>
<td>2 x M3 threaded hole to attach cable retainer</td>
</tr>
</tbody>
</table>
1.6 USB Communication Settings

Communication parameters are preset in the CONEX-AGAP controller and do not require any configuration:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits per second</td>
<td>921,600</td>
</tr>
<tr>
<td>Data bits</td>
<td>8</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
<tr>
<td>Stop bits</td>
<td>1</td>
</tr>
<tr>
<td>Flow control</td>
<td>Xon/Xoff</td>
</tr>
<tr>
<td>Terminator</td>
<td>CRLF</td>
</tr>
</tbody>
</table>
2.0 Programming

2.1 State Diagram

For a safe and consistent operation, the CONEX-AGAP uses 6 different operational states: Configuration, Ready OL, Disable and Moving CL, Stepping OL and Jogging OL. In each state, only specific commands are accepted by the CONEX-AGAP. Therefore, it is important to understand the state diagram below and which commands and actions cause transitions between the different states. See section 2.4 for additional command/state information:

**LED display:**

- **CONFIGURATION:** SLOW BLINKING RED.
- **READY OL:** SOLID GREEN.
- **DISABLE:** SLOW BLINKING GREEN.
- **MOVING CL:** FAST BLINKING GREEN.
- **STEPPING OL:** FAST BLINKING GREEN.
- **JOGGING OL:** FAST BLINKING GREEN.
When powering the CONEX-AGAP, the controller starts initialization. When the initialization is successful, the controller goes to the READY OL state. The controller can go to the CONFIGURATION state using the PW1 command. In the CONFIGURATION state, the CONEX-AGAP allows changes to all configuration parameters, like travel limits or controller address. The PW0 command saves all changes to the controller’s memory and returns the controller back to the DISABLE states.

To execute move commands PA[a], PR[a], the controller must be in the READY OL or MOVING CL states. To get from the DISABLE state to the READY OL state, the positioner must be enabled first with the MM1 command.

In the READY OL state, the control loop is open. During a move execution (PA/PR), the loop is closed, the controller is in the MOVING CL state and goes automatically back to the READY OL state when the move is completed. A time out error during a move changes the controller to the DISABLE state.

In the DISABLE state, the control loop is open. But the encoder is still read and the current position gets updated. To go from the READY state to the DISABLE state and vice versa, use the MM command. Going to DISABLE state is allowed for compatibility with other Newport products.
2.2 Command Syntax

The CONEX-AGAP is a command driven controller. The general format of a command is a two-letter ASCII character preceded and followed by parameters specific to the command:

**Command format:**

```
nn AAA xx
```

- **nn** — Optional or required controller address.
- **AA** — Command name.
- **xx** — Optional or required value or “?” to query current value.

Both, upper and lower case characters are accepted. Depending on the command, it can have an optional or required prefix (nn) for the controller address and/or a suffix (xx) value or a “?”.

**Blank spaces**

Blanks are allowed and ignored in any position, including inside a numerical value. The following two commands are equivalent, but the first example might be confusing and uses more memory:

- 2P A1.43 6
- 2PA1.436

**Decimal separator**

A dot (“.”) is used as decimal separator for all numerical values.

**Command terminator**

Commands are executed as the command terminator CRLF (carriage-return line-feed, ASCII 13 and ASCII 10) is received. The controller will analyze the received string. If the command is valid and its parameters are in the specified range, it will be executed. Otherwise it will memorize an error.

After the execution of the command, all remaining characters in the input string, if any, will be ignored. In particular, it is not possible to concatenate several commands on a single string from the PC to the CONEX-AGAP.

Each command will handle the memorization of related errors that can be accessed with the TE command properly. Please refer to the command set in section 2.4 for details.

2.3 Command Execution Time

The CONEX-AGAP controller interprets commands continuously as received. The typical execution time for a "tell position command" (nTP?) is about 10 ms. Here, command execution time means the time from sending the command until receipt of the answer.

It is important to note that a move command that may last for several seconds will not suspend the controller from further command execution. For an efficient process flow with many move commands, it is recommended to query the controller status (TS command) or the current position (TP command) before any further motion command is sent.

2.4 Command Set

This section describes the supported two-letter ASCII commands used to configure and operate the CONEX-AGAP. The general command format is:
Command format:

| nn | AA | a | xx |

nn — Optional or required controller address.
AA — Command name.
a — Optional axis reference (U or V)
xx — Optional or required value or “?” to query current value.

Most commands can be used to set a value (in that case the command name is followed by the value “xx”) or to query the current value (in that case the command name is followed by a “?”). When querying a value, the controller responds with the command it received followed by the queried value.

Not every command can be executed in all states of the CONEX-AGAP and some commands have different meanings in different states. It is therefore important to understand the state diagram of the controller, see section 2.1.

<table>
<thead>
<tr>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Moving</th>
<th>Stepping</th>
<th>Jogging</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB[a]</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get corrector deadband</td>
</tr>
<tr>
<td>DD[a]</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get deadband settling time</td>
</tr>
<tr>
<td>ID</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get stage identifier</td>
</tr>
<tr>
<td>JA[a]</td>
<td>–</td>
<td>–</td>
<td>●</td>
<td>–</td>
<td>–</td>
<td>Move jogging</td>
</tr>
<tr>
<td>KI[a]</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get integral gain</td>
</tr>
<tr>
<td>KP[a]</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get proportional gain</td>
</tr>
<tr>
<td>KY</td>
<td>○</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Set/Get calibration coefficients</td>
</tr>
<tr>
<td>LF</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get low pass filter frequency</td>
</tr>
<tr>
<td>MM</td>
<td>–</td>
<td>●</td>
<td>●</td>
<td>–</td>
<td>–</td>
<td>Leave DISABLE state</td>
</tr>
<tr>
<td>PA[a]</td>
<td>–</td>
<td>–</td>
<td>●</td>
<td>●</td>
<td>–</td>
<td>Move absolute</td>
</tr>
<tr>
<td>PR[a]</td>
<td>–</td>
<td>–</td>
<td>●</td>
<td>–</td>
<td>–</td>
<td>Move relative</td>
</tr>
<tr>
<td>PW</td>
<td>●</td>
<td>–</td>
<td>●</td>
<td>–</td>
<td>–</td>
<td>Enter/Leave CONFIGURATION state</td>
</tr>
<tr>
<td>RS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Reset controller</td>
</tr>
<tr>
<td>RS##</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Reset controller’s address to 1</td>
</tr>
<tr>
<td>SA</td>
<td>○</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Set/Get controller’s RS-485 address</td>
</tr>
<tr>
<td>SL[a]</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get negative software limit</td>
</tr>
<tr>
<td>SR[a]</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get positive software limit</td>
</tr>
<tr>
<td>ST[a]</td>
<td>–</td>
<td>–</td>
<td>●</td>
<td>●</td>
<td>–</td>
<td>Stop motion</td>
</tr>
<tr>
<td>SU</td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get encoder resolution</td>
</tr>
<tr>
<td>TB</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>–</td>
<td>Get command error string</td>
</tr>
<tr>
<td>TE</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>–</td>
<td>●</td>
<td>Get last command error</td>
</tr>
<tr>
<td>TH[a]</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>–</td>
<td>Get target position</td>
</tr>
<tr>
<td>TP[a]</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get current position</td>
</tr>
<tr>
<td>TS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get positioner error and controller state</td>
</tr>
<tr>
<td>VE</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get controller revision information</td>
</tr>
<tr>
<td>XR[a]</td>
<td>–</td>
<td>–</td>
<td>●</td>
<td>–</td>
<td>–</td>
<td>Move stepping</td>
</tr>
<tr>
<td>XU[a]</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>Set/Get step size for STEPPING OL state</td>
</tr>
<tr>
<td>ZT</td>
<td>●</td>
<td>●</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Get all controller parameters</td>
</tr>
</tbody>
</table>

○ Changes configuration parameters. Those changes will be stored in the controller’s memory with the PW1 command and remain available after switching off the controller.
□ Changes working parameters only. Those changes will get lost when switching off the controller.
● Accepted command.
– Write command not accepted (will return an error).

Command: Command passed without preceding controller number applies to all controllers (e.g. ST stops all controllers).
### DB[a] — Set/Get corrector deadband

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**: xxDB[a]nn or xxDB[a]?

**Parameters**

- **Description**: xx [int] — Controller address.
- **a** [char] — Axe reference.
- **nn** [int] — Deadband value.

**Range**

- **xx** — 1 to 31
- **a** — U or V
- **nn** — 0 to 0.005

**Units**

- **xx** — None.
- **nn** — Deg.

**Defaults**

- **xx**: Missing: Error B.
- **nn**: Out of range: Error B.
- **Floating point**: Error A.

**Description**: The deadband parameter defines an area, around a set position, in which the controller will consider it the end of the closed loop motion.

**Errors**

- **A**: Unknown message code or floating point controller address.
- **B**: Controller address not correct.
- **D**: Execution not allowed.
- **V**: Unknown axe reference.

**Rel. Commands**: DD[a] — Set/Get deadband settling time.

**Example**: 1DB0.00075 | Set controller #1 deadband to 0.75 mdeg.
## DD[a] — Set/Get deadband settling time

### Syntax

xxDD[a]nn or xxDD[a]?

### Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>a [char]</th>
<th>nn [int]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller address</td>
<td>xx</td>
<td>a</td>
<td>nn</td>
</tr>
<tr>
<td>Axe reference</td>
<td></td>
<td>U or V</td>
<td></td>
</tr>
<tr>
<td>Timer value</td>
<td></td>
<td></td>
<td>0 to 10^4</td>
</tr>
</tbody>
</table>

### Range

- xx: 1 to 31
- a: U or V
- nn: 0 to 10^4

### Units

- xx: None.

### Defaults

- xx: Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.

### Description

This command sets the deadband settling time. It corresponds to the number of corrector cycle after which, when the current position is below the deadband value from the target position, the system considers it has reached the target position.

### Errors

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- D — Execution not allowed.
- V — Unknown axe reference.

### Related Commands

- DB[a] — Set/Get corrector deadband.

### Example

1DD10 | Set controller #1 timer to 10 control loop period.
ID — Set/Get stage identifier

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Syntax

- xxIDnn or xxID?

Parameters

- **Description**
  - **xx** [int] — Controller address.
  - **nn** [char] — Stage model number.

- **Range**
  - **xx** — 1 to 31
  - **nn** — 1 to 31 ASCII characters.

- **Units**
  - **xx** — None
  - **nn** — None

- **Defaults**
  - **xx** Missing: Error B.
  - Out of range: Error B.
  - Floating point: Error A.
  - **nn** Missing: Error C.
  - Out of range: Error C.

Description

The ID? command returns the product name. In CONFIGURATION mode, this command allows changing the controller identifier.

Returns

If the sign “?” takes place of **nn**, this command returns the current programmed value.

Errors

- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.

Related Commands

- **ZT** — Get configuration parameters.

Example

1ID? | Get stage identifier for controller #1.

1ID CONEX-AGAP | Controller returns product name: CONEX-AGAP.
**JA[a] — Jog motion**

**Usage**
- Config.
- Disable
- Ready OL
- Moving CL
- Stepping OL
- Jogging OL

**Syntax**
xxJA[a]nn or xxJA[a]?

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int] — Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn [float]</td>
<td>Percentage of full speed.</td>
</tr>
</tbody>
</table>

**Range**
- xx — 1 to 31
- a — U or V
- nn — -100 to 100

**Units**
- xx — None.
- nn — % Full speed.

**Defaults**
- xx Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.

**Description**
If in READY state, the JA command sets the controller in JOGGING state, and make a relative motion with a speed, i.e.: a set of pulse amplitude and frequency, which follows a law between 0 and 100% as shown on the figure below.

Both axes can be in jog motion at the same time with different speed values. A speed of 0 stops the motion but does not take the controller out of the JOGGING state. The use of ST command returns to the controller to READY state.

**Errors**
- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- I — Execution not allowed in CONFIGURATION state.
- J — Execution not allowed in DISABLED state.
- M — Execution not allowed in Motion states.
- V — Unknown axe reference.

**Rel. Commands**
- TP[a] — Get current position
- ST — Stop motion

**Example**
1JAU50.35 | Set controller #1 speed at 50.35% of full speed on axe U.

**NOTE**
Not to scale.
## KI[a] — Set/Get integral gain

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>

**Syntax**

xxKI[a]nn or xxKI[a]?

**Parameters**

**Description**

- **xx** [int] — Controller address.
- **a** [char] — Axe reference.
- **nn** [int] — Integral gain.

**Range**

- **xx** — 0 to 31
- **a** — U or V
- **nn** — > 0.

**Units**

- **xx** — None.
- **nn** — None.

**Defaults**

- **xx** Missing: Change to 0.
- **nn** Missing: Error C.
- Out of range: Error C.

**Description**

In CONFIGURATION state, this command sets the integral gain of the PID control loop which can then be saved in the controller’s nonvolatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE state.

In DISABLE state, this command allows setting a new working parameter for the derivative gain. This value is not saved in the controller’s memory and will be lost after reboot.

**Returns**

If the sign “?” takes place of **nn**, this command returns the current programmed value.

**Errors**

- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.
- **M** — Execution not allowed in Motion states.
- **V** — Unknown axe reference.

**Rel. Commands**

<table>
<thead>
<tr>
<th>KP</th>
<th>Set/Get proportional gain.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF</td>
<td>Set/Get low pass filter frequency.</td>
</tr>
</tbody>
</table>

**Example**

1KIU5 | Set the controller #1U axe integral gain to 5
1KIU?

1KIU5
**KP[a] — Set/Get proportional gain**

**Syntax**

xxKP[a]nn or xxKP[a]?

**Parameters**

- **Description**
  - xx [int] — Controller address.
  - nn [float] — Proportional gain.

- **Range**
  - xx — 1 to 31
  - a — U or V
  - nn — > 0

- **Units**
  - xx — None.
  - nn — Preset units.

- **Defaults**
  - xx Missing: Error B.
  - Out of range: Error B.
  - Floating point: Error A.
  - nn Missing: Error C.
  - Out of range: Error C.

**Description**

In CONFIGURATION state, this command sets the proportional gain of the PID control loop which can than be saved in the controller’s nonvolatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE state.

In DISABLE state, this command allows setting a new working parameter for the derivative gain. This value is not saved in the controller’s memory and will be lost after reboot.

**Returns**

If the sign “?” takes place of nn, this command returns the current programmed value.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- M — Execution not allowed in Motion states.
- V — Unknown axe reference.

**Rel. Commands**

- KI — Set/Get integral gain.
- LF — Set/Get low pass filter frequency.

**Example**

- 1KPU5 | Set the controller #1U axe proportional gain to 5
- 1KPU?

1KPU5
### KY[a] — Set/Get calibration coefficients

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>○</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

#### Syntax

xxKY[a]nn or xx KY?

#### Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int] —</th>
<th>a [char] —</th>
<th>nn [float] —</th>
</tr>
</thead>
</table>

#### Range

<table>
<thead>
<tr>
<th>xx [char] —</th>
<th>1 to 31</th>
</tr>
</thead>
<tbody>
<tr>
<td>a [char] —</td>
<td>F or T or C</td>
</tr>
</tbody>
</table>

#### Units

<table>
<thead>
<tr>
<th>xx [char] —</th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn —</td>
<td>Preset units.</td>
</tr>
</tbody>
</table>

#### Defaults

<table>
<thead>
<tr>
<th>xx [char] —</th>
<th>Missing: Error B.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a [char] —</td>
<td>Out of range: Error B.</td>
</tr>
<tr>
<td>nn [float] —</td>
<td>Floating point: Error A.</td>
</tr>
</tbody>
</table>

#### Description

The KY command is used to set the calibration coefficients. Those are factory set values. Users should not modify those parameters.

#### Returns

If the sign “?” takes place of nn, this command returns the calibration values.

#### Errors

A — Unknown message code or floating point controller address.

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.
**KZ[a] — Set/Get calibration coefficients**

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>

**Syntax**

\[xxKZ[a]nn\] or \[xxKZ?\]

**Parameters**

**Description**

- \[xx\] [int] — Controller address.
- \[a\] [char] — Coefficient reference.
- \[nn\] [float] — Calibration value.

**Range**

- \[xx\] [int] — 1 to 31
- \[a\] [char] — F or T or C.

**Units**

- \[xx\] [int] — None.
- \[nn\] [float] — None.

**Defaults**

- Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.
- Missing: Error C.
- Out of range: Error C.

**Description**

The KZ command is used to set the calibration coefficients. Those are factory set values. Users should not modify those parameters.

**Returns**

If the sign “?” takes place of \[nn\], this command returns the calibration values.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- J — Execution not allowed in DISABLE state.
- M — Execution not allowed in Motion states.
LF — Set/Get low pass filter frequency

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

**Syntax**  
xxLFnn or LF?

**Parameters**

**Description**  
xx [int] — Controller address.

nn [float] — Frequency.

**Range**  
xx — 1 to 31

nn — >0

**Units**  
xx — None.

nn — Hertz.

**Defaults**  
xx Missing: Error B.

Out of range: Error B.

Floating point: Error A.

**Description**  
The LF command sets or gets the digital low pass filter frequency.

**Returns**  
If the sign “?” takes place of nn, this command returns the current programmed value.

**Errors**

A — Unknown message code or floating point controller address.

B — Controller address not correct.

D — Execution not allowed.

**Rel. Commands**

KP[α] — Set/Get proportional gain.

KI[α] — Set/Get integral gain.

**Example**

1LF5  | Set the controller #1 low pass filter frequency to 5Hz.
**MM — Enter/Leave DISABLE state**

**Usage**

<table>
<thead>
<tr>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>●</td>
<td>●</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

`xxMMnn` or `xxMM?`

**Parameters**

- **Description**
  - `xx` [int] — Controller address.
  - `nn` [int] — State change direction.

- **Range**
  - `xx` — 0 to 31
  - `nn` — 0 changes state from READY to DISABLE.
  - 1 changes state from DISABLE to READY.

- **Units**
  - `xx` — None.
  - `nn` — None.

- **Defaults**
  - `xx` Missing: Change to 0.
  - Out of range: Error B.
  - Floating point: Error A.
  - `nn` Missing: Error C.
  - Out of range: Error C.

**Description**

When the MM command is sent without preceding controller number or the controller number is 0, the MM command gets executed on all controllers.

MM0 changes the controller’s state from READY to DISABLE. The current position gets still updated.

MM1 changes the controller’s state from DISABLE to READY. The controller’s set point position is set equal to its current position and the control loop gets closed.

**Returns**

If the sign “?” takes place of `nn`, this command returns the current controller state (ef). Refer to the TS command for the list of controller states.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- I — Execution not allowed in CONFIGURATION state.
- M — Execution not allowed in Motion states.

**Rel. Commands**

- PW — Enter/leave CONFIGURATION state.

**Example**

```
1MM1 | The controller #1 goes to READY state.
1MM?
1MM32
```
PA\,[a] — Move absolute

**Syntax**

xxPA\,[a]\,nn  or  xxPA\,[a]?  

**Parameters**

- **Description**
  - xx [int] — Controller address.
  - a [char] — Axe reference
  - nn [float] — New target position.

- **Range**
  - xx — 1 to 31
  - a — U or V
  - nn — > SL and < SR

- **Units**
  - xx — None.
  - a — U or V
  - nn — Preset units.

- **Defaults**
  - xx Missing: Error B.
  - Out of range: Error B.
  - Floating point: Error A.
  - nn Missing: Error C.
  - Out of range: Error C.

**Description**
The PA command initiates an absolute move. When received, the positioner will move to the new target position specified by nn.
The PA command gets only accepted in READY or MOVING state, AND when the new target position is higher or equal to the negative software limit (SL), AND lower or equal to the positive software limit (SR).

**Returns**
If the sign “?” takes place of nn, this command returns the target position value.

**Errors**
- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- G — Target position out of limits.
- I — Execution not allowed in CONFIGURATION state.
- J — Execution not allowed in DISABLE state.
- V — Unknown axe reference.

**Rel. Commands**
- PR — Move relative.
- TH — Get target position.
- TP — Get current position.

**Example**

1PAV0.2  |  Move positioner on controller #1 to absolute position 0.2 units.
## PR\([a]\) — Move relative

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving OL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

**Syntax** \(xx\)PR\([a]\)\(nn\)

**Parameters**

**Description**

- \(xx\) [int] — Controller address.
- \(a\) [char] — Axe reference
- \(nn\) [float] — Displacement.

**Range**

- \(xx\) — 1 to 31
- \(a\) — U or V
- \(nn\) — > SL and < SR

**Units**

- \(xx\) — None.
- \(nn\) — Preset units.

**Defaults**

- \(xx\) Missing: Error B.
- \(nn\) Missing: Error C.

**Description**

The PR command initiates a relative move. When received, the positioner will move to a new target position \(nn\) units away from the current target position.

The PR command gets only accepted in READY or MOVING state, AND when the distance of the positioner to the end of runs is larger than the commanded displacement.

**Returns**

- If the sign “?” takes place of \(nn\), this command returns the target position value.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- G — Displacement out of limits.
- I — Execution not allowed in CONFIGURATION state.
- J — Execution not allowed in DISABLE state.
- V — Unknown axe reference.

**Rel. Commands**

- PA — Move absolute.
- TH — Get target position.
- TP — Get current position.

**Example**

\(1\)PRU0.2 | Move positioner on controller #1 to a new position 0.2 units away from the current target position.
## PW — Enter/Leave CONFIGURATION state

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
</table>

Syntax: `xxPWnn` or `xxPW?`

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>nn [float]</th>
</tr>
</thead>
</table>

- Controller address.
- Mode.

**Range**

- `xx`: 1 to 31
- `nn`: 1: Go from READY state to CONFIGURATION state.
  0: Go from CONFIGURATION state to READY state.

**Units**

- `xx`: None.
- `nn`: None.

**Defaults**

- Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.
- Missing: Error C.
- Out of range: Error C.

**Description**

PW1 changes the controller’s state from READY to CONFIGURATION. In Configuration state all parameter settings are saved in the controller’s memory and remain available after switching off the controller.

PW0 checks all stage parameters, and if they are acceptable, saves them in the flash memory of the controller. After that, it changes the controller’s state from CONFIGURATION to READY.

The execution of a PW0 command may take up to 5 seconds. During that time the controller will not respond to any other command.

**Returns**

- If the sign “?” takes place of `nn`, this command returns the current state.

**Errors**

- A: Unknown message code or floating point controller address.
- B: Controller address not correct.
- C: Parameter missing or out of range.
- D: Execution not allowed.
- J: Execution not allowed in DISABLE state.
- M: Execution not allowed in Motion states.

**Rel. Commands**

- MM: Enter/Leave DISABLE state.

**Example**

1PW1 | Changes controller #1 to CONFIGURATION state.

---

### NOTE

The PW command is limited to 100 writes. Unit failure due to excessive use of the PW command is not covered by warranty.

The PW command is used to change the configuration parameters that are stored in memory, and not parameters that are needed to be changed on the fly.
### RS — Reset controller

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
</table>

**Syntax**: `xxRS`

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Range</strong></td>
<td>xx</td>
<td>1 to 31</td>
</tr>
<tr>
<td><strong>Units</strong></td>
<td>xx</td>
<td>None.</td>
</tr>
<tr>
<td><strong>Defaults</strong></td>
<td>xx</td>
<td>Missing: Error B.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Out of range: Error B.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Floating point: Error A.</td>
</tr>
</tbody>
</table>

**Description**: The RS command issues a hardware reset of the controller, equivalent to a power-up.

**Errors**

<table>
<thead>
<tr>
<th>A</th>
<th>Unknown message code or floating point controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Controller address not correct.</td>
</tr>
<tr>
<td>D</td>
<td>Execution not allowed.</td>
</tr>
</tbody>
</table>

**Example**: `1RS` | *Reset controller #1.*
### RS## — Reset controller’s address

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxRS## or RS##</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Parameters**

**Description**

xx [int] — Controller address.

**Range**

xx — 1 to 31

**Units**

xx — None.

**Defaults**

xx Missing: Change to 0.

Out of range: Error B.

Floating point: Error A.

**Description**

The RS## command resets the controller’s address to 1. This address needs to be different for each CONEX devices when connected on a RS-485 communication network.

* The minimum endurance of the memory used to store parameters is of 100 write cycles. Users should limit the use of RS## command.

**Errors**

A — Unknown message code or floating point controller address.

B — Controller address not correct.

D — Execution not allowed.

**Example**

RS## | Reset controller’s address to 1.
SA — Set/Get controller’s RS-485 address

**Usage**

<table>
<thead>
<tr>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

`xxSAnn` or `xxSA?`

**Parameters**

**Description**

- **xx** [int] — Controller address.
- **nn** [int] — Controller new address.

**Range**

- xx — 1 to 31
- nn — 1 to 31 and ≠ xx

**Units**

- xx — None.
- nn — None.

**Defaults**

- xx Missing: Error B.
- nn Missing: Error C.
- Out of range: Error C.
- Floating point: Error A.

**Description**

The SA command sets the controller’s RS-485 address. This address is ONLY used when the controller is configured for RS-485 communication.

The SA command is of practical use only when not using this software.

**Returns**

If the sign “?” takes place of **nn**, this command returns the current programmed value.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- J — Execution not allowed in DISABLE state.
- M — Execution not allowed in motion states.

**Example**

`1SA3` | Set controller’s RS-485 address to 3.

`3SA?` | Get the controller address

`3SA3`
**SL[a] — Set/Get negative software limit**

### Syntax

```
xSL[a]nn or xSL[a]?
```

### Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nn [float]</td>
<td>Negative software limit.</td>
</tr>
</tbody>
</table>

**Range**

- xx: 1 to 31
- a: U or V
- nn: ≥ -1 and ≤ 0

**Units**

- xx: None.
- nn: Deg.

### Defaults

- xx: Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.
- nn: Missing: Error C.
- Out of range: Error C.

### Description

In **CONFIGURATION** state, this command sets the negative software limit which can than be saved in the controller’s nonvolatile memory using the PW command. It is also the default value that will be used unless a different value is set in **DISABLE** or **READY** state.

In **DISABLE** or **READY** state, this command allows setting a new working parameter for the negative software limit. It must be lower or equal to the target position. This value is not saved in the controller’s memory and will be lost after reboot.

The software limits are useful to limit the travel range of a positioner. There is no possibility to disable software limits.

### Returns

If the sign “?” takes place of nn, this command returns the current programmed value.

### Errors

- A: Unknown message code or floating point controller address.
- B: Controller address not correct.
- C: Parameter missing or out of range.
- D: Execution not allowed.
- M: Execution not allowed in Motion states.
- V: Unknown axe reference.

### Rel. Commands

- SR: Set positive software limit.

### Example

```
1SLV-0.5
```

Set controller #1 negative software limit to –0.5 units for axe V.
SR[a] — Set/Get positive software limit

### Usage

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>○</td>
<td>□</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Syntax

xxSR[a]nn or xxSR[a]?

### Parameters

**Description**

- **xx** [int] — Controller address.
- **a** [char] — Axe reference.
- **nn** [float] — Positive software limit.

**Range**

- **xx** — 1 to 31
- **a** — U or V
- **nn** — ≥ 0 and ≤ 1

**Units**

- **xx** — None.
- **nn** — Deg.

**Defaults**

- **xx** Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.
- **nn** Missing: Error C.
- Out of range: Error C.

### Description

- In CONFIGURATION state, this command sets the positive software limit which can then be saved in the controller’s nonvolatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE or READY state.

- In DISABLE or READY state, this command allows setting a new working parameter for the positive software limit. It must be larger or equal to the target position. This value is not saved in the controller’s memory and will be lost after reboot.

- The software limits are useful to limit the travel range of a positioner. There is no possibility to disable software limits.

### Returns

- If the sign “?” takes place of **nn**, this command returns the current programmed value.

### Errors

- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.
- **M** — Execution not allowed in Motion states.
- **V** — Unknown axe reference.

### Related Commands

- **SL** — Set negative software limit.

### Example

1SRU0.75 | Set controller #1 positive software positive to 0.75 units for axe U.
ST — Stop motion

Usage  Config.  Disable  Ready OL  Moving CL  Stepping OL  Jogging OL
       —       —       —       ●       ●       ●

Syntax  [xx]ST

Parameters

Description  xx [int] — Controller address.

Range  xx — 0 to 31

Units  xx — None.

Defaults  xx  Missing: Change to 0.

Out of range: Error B.

Floating point: Error A.

Description  The xxST command with preceding controller address stops a move in progress on controller xx. The ST command without preceding controller address stops the moves on ALL controllers.

On the AGAP controllers, this command stops both U and V axes at the same time. The new target positions for both axes are set to the current positions.

Errors  A — Unknown message code or floating point controller address.
        B — Controller address not correct.
        D — Execution not allowed.
        I — Execution not allowed in CONFIGURATION state.
        J — Execution not allowed in DISABLED state.
        K — Execution not allowed in READY state.

Example  ST  |  Stop moves on all controllers.
CONEX-AGAP

Agilis-D Controller with Strain Gages Feedback

SU — Set/Get system resolution

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>☐</td>
<td>☐</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

xxSU\(\text{nn}\) or \(\text{SU}\)?

**Parameters**

**Description**

- xx [int] — Controller address.
- nn [float] — Resolution.

**Range**

- xx — 1 to 31
- nn — >0

**Units**

- xx — None.
- nn — Deg.

**Defaults**

- xx — Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.

**Description**

The SU command sets or gets the resolution of the system. The device determines the position using analog signals and rounds it to the encoder resolution set using this command.

**Returns**

If the sign “?” takes place of nn, this command returns the current programmed value.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- D — Execution not allowed.

**Rel. Commands**

- DB — Set/Get corrector deadband.

**Example**

1SU0.0005 — Set the controller #1 resolution to 0.5mdeg.
TB — Get command error string

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxTBnn</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Parameters

**Description**
- **xx** [int] — Controller address.
- **nn** [char] — Error code (refer to TE command).

**Range**
- **xx** — 1 to 31

**Units**
- **xx** — None.

**Defaults**
- **xx** Missing: Error B.
- **nn** Missing: Returns explanation of current error.

**Description**
The TB command returns a string that explains the meaning of the error code **nn** (see TE command for complete list).

**Errors**
- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.

**Rel. Commands**
- **TE** — Get error code.

**Example**

```
1TB@  | Get explanation to error code @.

1TB@ No error | Controller returns: @ = means no error.
```
## TE — Get last command error

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxTE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Parameters**

- **Description**: xx [int] — Controller address.
- **Range**: xx — 1 to 31
- **Units**: xx — None.
- **Defaults**:
  - Missing: Error B.
  - Out of range: Error B.
  - Floating point: Error A.

**Description**
The TE command returns the currently memorized error. When a command is not executable, it memorizes an error. This error can be read with the TE command. After the execution of a TE command, the error buffer gets erased and another TE command will return @, means no error. When a new command error is generated before the previous command error is read, the new command error will overwrite the current memorized error.

For a safe program flow it is recommended to always query the command error after each command execution.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- D — Execution not allowed.

**Rel. Commands**

- TB — Get error string.

**Example**

- 1TE | Get last error memorized on controller #1.
  - Controller returns: 1TE@, means no error.

List of errors and corresponding strings (see TB command):

- @ — No error.
- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Command not allowed.
- G — Displacement out of limits.
- I — Command not allowed in CONFIGURATION state.
- J — Command not allowed in DISABLE state.
- K — Command not allowed in READY state.
- M — Command not allowed in motion states.
- N — Current position out of software limit.
- S — Communication Time Out.
- U — Error during EEPROM access.
- V — Unknown axe reference.
TH\[a\] — Get target position

Syntax xxTH[a] or xxTH

Parameters

Description xx [int] — Controller address.

Range xx — 1 to 31
a — U or V

Units xx — None.

Defaults xx Missing: Error B.
Out of range: Error B.
Floating point: Error A.

Description The TH command returns the value of the unrounded target position. This is the position where the positioner should be. The target position rounded to the device resolution is given by the commands PR[a]? and PA[a]?

Errors A — Unknown message code or floating point controller address.
B — Controller address not correct.
D — Execution not allowed.

Rel. Commands TP — Get current position.

Example 1THU | Get target position of axe U of controller #1.

1THU0.0023512 | Controller returns: target position for axe U = 0.0023512 units.
TP\,[a] — Get current position

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxTP,[a] or xxTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int] — Controller address.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Range</th>
<th>xx — 1 to 31</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>U or V</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Units</th>
<th>xx — None.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Defaults</th>
<th>xx Missing:</th>
<th>Error B.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out of range:</td>
<td>Error B.</td>
<td></td>
</tr>
<tr>
<td>Floating point:</td>
<td>Error A.</td>
<td></td>
</tr>
</tbody>
</table>

Description

The TP command returns the value of the current position. This is the position where the positioner actually is according to its encoder value. In MOVING state, this value always changes. In READY state, this value should be equal or very close to the target position.

Together with the TS command, the TP command helps evaluating whether a motion is completed.

Errors

<table>
<thead>
<tr>
<th>A</th>
<th>Unknown message code or floating point controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Controller address not correct.</td>
</tr>
<tr>
<td>D</td>
<td>Execution not allowed.</td>
</tr>
</tbody>
</table>

Rel. Commands

| TH — Get target position. |

Example

1TPU | Get current position of axe U of controller #1. |
1TPU0 | Controller returns: actual position for axe U = 0 units. |
TS — Get positioner error and controller state

Usage  Config. Disable Ready OL Moving CL Stepping OL Jogging OL

Syntax  xxTS

Parameters

Description  xx [int] — Controller address.

Range  xx — 1 to 31

Units  xx — None.

Defaults  xx  Missing: Error B.

Floating point: Error A.

Description  The TS command returns the positioner error and the current controller state. The motion time out flag is always set with one of the two-associated following error.

Returns  The TS command returns six characters (1TSabcdef). The first 4 characters (abcd) represent the positioner error in Hexadecimal. The last two characters (ef) represent the controller state.

Error code (abcd): Convert each hexadecimal to a binary:

<table>
<thead>
<tr>
<th>F</th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
<th>A</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>1110</td>
<td>1101</td>
<td>1100</td>
<td>1011</td>
<td>1010</td>
<td>1001</td>
<td>1000</td>
<td>0111</td>
<td>0110</td>
<td>0101</td>
<td>0100</td>
<td>0011</td>
<td>0010</td>
<td>0001</td>
<td>0000</td>
</tr>
</tbody>
</table>

Each bit represents one possible error:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Not used</td>
<td>Not used</td>
<td>Not used</td>
<td>Not used</td>
</tr>
</tbody>
</table>

```
A  B  C  D
1 1 1 1
Not used Not used Not used Not used
```

Motion Time out

Examples:
- Error map 0000 = No errors
- Error map 0020 = Motion time out.
Controller states (ef):
- 14: CONFIGURATION.
- 28: MOVING CL.
- 29: STEPPING OL.
- 32: READY from Reset.
- 33: READY from MOVING CL.
- 34: READY from DISABLE.
- 35: READY from JOGGING OL
- 36: READY from STEPPING OL.
- 3C: DISABLE from READY OL.
- 3D: DISABLE from MOVING CL.
- 46: JOGGING OL.

NOTES
THE ERROR BUFFER GETS UPDATED PERIODICALLY, APPROX. EVERY 1 MS.
THE TS COMMAND READS THE ERROR BUFFER AND CLEARS THE ERROR BUFFER AT THE SAME TIME (SAME AS FOR COMMANDS TE, TB). SO WHEN LAUNCHING THE TS COMMAND, IT IS IMPORTANT TO PROCESS THE TS FEEDBACK ACCORDINGLY

<table>
<thead>
<tr>
<th>Errors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Unknown message code or floating point controller address.</td>
</tr>
<tr>
<td>B</td>
<td>Controller address not correct.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rel. Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE</td>
<td>Get last error.</td>
</tr>
</tbody>
</table>

Example

<table>
<thead>
<tr>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1TS</td>
<td>Get error and state of controller #1.</td>
</tr>
<tr>
<td>1TS000032</td>
<td>Controller returns: no errors and READY from reset.</td>
</tr>
</tbody>
</table>
VE — Get controller revision information

Usage

<table>
<thead>
<tr>
<th></th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Syntax

xxVE

Parameters

Description

xx [int] — Controller address.

nn [string] — Action.

Range

xx — 1 to 31

Units

xx — None.

Defaults

xx Missing: Error B.

Out of range: Error B.

Floating point: Error A.

Description

This command returns the controller’s revision information.

Errors

A — Unknown message code or floating point controller address.

B — Controller address not correct.

Rel. Commands

TP — Get current position.

Example

1VE | Get controller #1 revision information.

1VE CONEX-AGAP V1.0.0. | Controller returns revision number
**XR[a] — Step motion**

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

xxXR[a]nn

**Parameters**

- **Description**
  - xx [int] — Controller address.
  - nn [int] — Number of steps.

- **Range**
  - xx — 1 to 31
  - a — U or V
  - nn — ≥ -10⁶ and ≤ 10⁶

- **Units**
  - xx — None.

- **Defaults**
  - xx Missing: Error B.
  - Out of range: Error B.
  - Floating point: Error A.

**Description**

Starts a relative move of nn steps with step amplitude defined by the XU command. At the end of the motion, the target position of the axe takes the value of the current axe position.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- I — Execution not allowed in CONFIGURATION state.
- J — Execution not allowed in DISABLED state.
- M — Execution not allowed in Motion states.
- V — Unknown axe reference.

**Rel. Commands**

- TP — Get current position.
- XU[a] — Set/Get step motion size.

**Example**

1XRU100 | Set controller #1 step number on axe U.
**XU[a] — Set/Get step motion size**

**Syntax**  
`xxXU[a]nn` or `xxXU[a]?`

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a [char]</td>
<td>U or V</td>
<td>Axe reference</td>
</tr>
<tr>
<td>nn [int]</td>
<td>-50 to +50</td>
<td>Step size.</td>
</tr>
</tbody>
</table>

**Range**

- `xx` 1 to 31
- `a` U or V
- `nn` -50 to +50

**Units** None.

**Defaults**

- Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.

**Description**

Sets the step amplitude (step size) in positive or negative direction. If the parameter is positive, it will set the step amplitude in the forward direction. If the parameter is negative, it will set the step amplitude in the backward direction.

**NOTES**

The step amplitude is a relative measure. The step amplitude corresponds to the amplitude of the electrical signal sent to the Agilis motor. There is no linear correlation between the step amplitude and the effective motion size. In particular, too low a setting for the step amplitude may result in no output motion. Also, the same step amplitude setting for forward and backward direction may result in different size motion steps. Also, the motion step size corresponding to a step amplitude setting may vary by position, load, and throughout the life time of the product. The step amplitude setting is not stored after power down. The default value after power-up is 35.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- M — Execution not allowed in Motion states.
- V — Unknown axe reference.

**Rel. Commands**

- TP — Get current position.
- XR[a] — Step motion.

**Example**

- `1XUU20` | Set controller #1 step size to 20 on axe U.
### ZT — Get all configuration parameters

<table>
<thead>
<tr>
<th>Usage</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready OL</th>
<th>Moving CL</th>
<th>Stepping OL</th>
<th>Jogging OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxZT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Parameters

**Description**

- `xx` [int] — Controller address.

**Range**

- `xx` — 1 to 31

**Units**

- `xx` — None.

**Defaults**

- Missing: Error B.
- Out of range: Error B.
- Floating point: Error A.

#### Description

The ZT command returns the list of all current configuration parameters.

#### Errors

- **A** — Unknown message code or floating point controller address
- **B** — Controller address not correct
- **M** — Execution not allowed in Motion states.

#### Rel. Commands

- **TE** — Get error code.

#### Example

```sql
1ZT | Get controller #1 configuration data.
1PW1
1IDAG-M100D
1SA1
1SLU-1
1SRU1...
1PW0
```
3.0 Connector interfaces

3.1 USB (Male mini-USB)

1 2 3 4 5

USB
Mating connector:
Plug Mini-USB B 5 cts

<table>
<thead>
<tr>
<th>PIN</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+5VdcIN</td>
</tr>
<tr>
<td></td>
<td>Do not connect if comm connector is used</td>
</tr>
<tr>
<td>2</td>
<td>DATA-</td>
</tr>
<tr>
<td>3</td>
<td>DATA+</td>
</tr>
<tr>
<td>4</td>
<td>NC</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
</tr>
</tbody>
</table>
Service Form

Your Local Representative
Tel.: ___________________
Fax: ___________________

Name: __________________________________________________  Return authorization #: ___________________________________
(Please obtain prior to return of item)

Company: _______________________________________________

Address: _______________________________________________

Country: _______________________________________________

P.O. Number: ___________________________________________

Date: __________________________________________________

Fax Number: ___________________________________________

Item(s) Being Returned: __________________________________

Model#: _______________________________________________

Serial #: _______________________________________________

Description:__________________________________________________________________________________________________________

Reasons of return of goods (please list any specific problems):
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
____________________________________________________________________________________________________________________
North America & Asia
Newport Corporation
1791 Deere Ave.
Irvine, CA 92606, USA
Sales
Tel.: (800) 222-6440
e-mail: sales@newport.com
Technical Support
Tel.: (800) 222-6440
e-mail: tech@newport.com
Service, RMAs & Returns
Tel.: (800) 222-6440
e-mail: rma.service@newport.com

Europe
MICRO-CONTROLE Spectra-Physics S.A.S
9, rue du Bois Sauvage
91055 Évry CEDEX
France
Sales
Tel.: +33 (0)1.60.91.68.68
e-mail: france@newport-fr.com
Technical Support
e-mail: tech_europe@newport.com
Service & Returns
Tel.: +33 (0)2.38.40.51.55