CONEX-PP

Single-Axis Intelligent Stepper Motor Controller/Driver

RoHS Compliant

Newport®
Experience | Solutions

Command Interface Manual

V1.0.x

Precision Motion – *Guaranteed*™

Table of Contents

CONEX-PP
Single-Axis Intelligent Stepper Motor Controller/Driver

1.0 Introduction

1.1 Purpose

The purpose of this document is to provide the method syntax of each command to communicate with the CONEX-PP device.

1.2 Overview

The Command Interface is the wrapper class that maintains a list of CONEX-PP instruments. It exposes methods to communicate with any CONEX-PP device.

NOTE

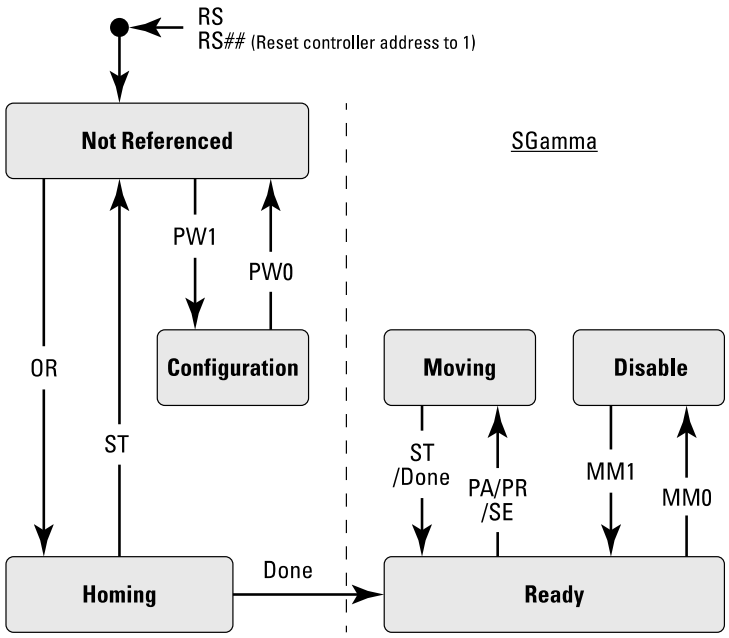
Each function name is defined with the command code "AA".

For each command function, refer to the CONEX-PP programmer's manual.

2.0 Programming

2.1 State Diagram

For a safe and consistent operation, the CONEX-PP uses 6 different operational states: Not referenced, Configuration, Homing, Ready, Disable and Moving. In each state, only specific commands are accepted by the CONEX-PP. Therefore, it is important to understand the state diagram below and to know which commands and actions cause transitions between the different states. Also see section 2.4 for command/state information:



Actions in each state when End of Runs is encountered

NOT REFERENCED:	No action.
CONFIGURATION:	No action.
HOMING:	Only check at end of HOMING and then change to NOT REFERENCED state.
MOVING:	Abort motion and then changes to NOT REFERENCED state.
READY:	Changes to NOT REFERENCED state.
DISABLE:	Changes to NOT REFERENCED state.

2.2 Command Syntax

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

Command format

nn	AA	xx

nn — Controller address, or

nothing if the issued command addresses all controllers.

AA — Command name.

xx — Parameter value, or

"**?**" to query the current value, or

nothing if the command takes no parameter.

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, a "**?**" or no suffix at all.

Blank spaces

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

2P A1.43 6

2PA1.436

Decimal separator

A dot (".") must be used as decimal separator for all numerical values.

Command terminator

Commands are executed as either of the command terminator C_R or L_F (carriage-return, ASCII 13 or line-feed, 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 command parameters are identified, all remaining characters in the input string until the first command terminator, if any, will be ignored. Commands from the PC to the CONEX-PP may still be concatenated in a single string, but each command must be separated from the next one by a carriage-return or a line-feed.

In case any error occurs, the reported error will be recorded and can be checked using the TE command. Please refer to the command set in section 2.4 for details.

2.3 Command Execution Time

The CONEX-PP controller interprets commands continuously as they are received. The typical execution time for a "tell position command" (nTP?) is about 10 ms. Here, command execution time means the between sending a command and receiving an answer.

It is important to note that a move command, that may lasts for several seconds, will not suspend the controller from further command execution. So for an efficient process flow with many move commands it is recommended to use the PT command (get time for a relative move), and 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-PP. The general command format is:

Command format

nn	AA	xx

nn — Optional or required controller address.

AA — Command name.

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, represented here as "**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.

Examples:

1VA10 sets the velocity of the controller #1 to 10 units/second (and sends nothing back).

1VA? sends back the reply "1VA10", which means: "controller #1's velocity is 10 units/second".

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

	Not Ref.	Config.	Disable	Ready	Motion	Description
AC	–	○	□	□	–	Set/Get acceleration
BA	–	○	–	–	–	Set/Get backlash compensation
BH	–	○	–	–	–	Set/Get hysteresis compensation
FR	–	○	–	–	–	Set/Get stepper motor configuration
HT	–	○	–	–	–	Set/Get HOME search type
ID	–	○	□	□	–	Set/Get stage identifier
JR	–	○	□	□	–	Set/Get jerk time
MM	–	–	●	●	–	Enter/Leave DISABLE state
OH	–	○	–	–	–	Set/Get HOME search velocity
OR	●	–	–	–	–	Execute HOME search
OT	–	○	–	–	–	Set/Get HOME search time–out
PA	–	–	–	●	–	Move absolute
PR	–	–	–	●	–	Move relative
PT	–	–	●	●	●	Get estimated duration of a relative move
PW	●	●	–	–	–	Enter/Leave CONFIGURATION state
QC	–	□	–	–	–	Set/Get idle current coefficient
QD	–	□	–	–	–	Set/Get idle current delay
QI	–	○	–	–	–	Set/Get motor's current limits
RS	●	●	●	●	●	Reset controller
RS##	●	●	●	●	●	Reset controller's address to 1
SA	–	○	–	–	–	Set/Get controller's RS–485 address
SE	–	–	–	●	–	Configure/Execute simultaneous started move
SL	–	○	□	□	–	Set/Get negative software limit
SR	–	○	□	□	–	Set/Get positive software limit
ST			–	–	●	Stop motion
TB	●	●	●	●	●	Get command error string
TE	●	●	●	●	●	Get last command error
TH	●	●	●	●	●	Get set–point position
TP	●	●	●	●	●	Get current position
TS	●	●	●	●	●	Get positioner error and controller state
VA	–	○	□	□	–	Set/Get velocity
VE	●	●	●	●	●	Get controller revision information
ZT	●	●	●	●	●	Get all axis parameters

Not Ref. Corresponds to the NOT REFERENCED state (for details see state diagram, section 2.1).

Config. Corresponds to the CONFIGURATION state.

Disable Corresponds to the DISABLE state.

Ready Corresponds to the READY state.

Motion Corresponds to the HOMING and MOVING states.

○ 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.

– Command is forbidden in this state (will memorize an error).

Grey line Command passed without preceding controller number applies to all controllers (e.g. MM0 disables all controllers).

AC — Set/Get acceleration

Usage	**Not Ref.**	**Config.**	**Disable**	**Ready**	**Motion**
	–	○	□	□	–

Syntax **xxACnn** or **xxAC?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Acceleration value.

Range

xx — **1** to **31**

nn — **> 10^{-6}** and **< 10^{12}**

Units

xx — None

nn — Preset units/s^2

Defaults

xx Missing: Error B.

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

Description

In CONFIGURATION state, this command sets the maximum acceleration value which can then be saved in the controller's non-volatile memory using the PW command. This is the maximum acceleration that can be applied to the mechanical system. It is also the default acceleration that will be used for all moves unless a lower value is set in DISABLE or READY state.

In DISABLE or READY state, this command sets the acceleration used for all subsequent moves. Its value can be up to the programmed value in CONFIGURATION state. This value is not saved in the controller's memory and will be lost after reboot.

Returns

If the sign "**?**" is used instead of **nn**, this command returns the current value for the state in which the controller is (either CONFIGURATION or DISABLE/READY).

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.

H — Execution not allowed in NOT REFERENCED state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands

JR — Set/Get jerk time.

VA — Set/Get velocity.

Example

1AC500 | *Set controller #1 acceleration to 500 units/s^2.*

1AC? | *Controller returns 1AC500.*

BA — Set/Get backlash compensation

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	–	–	–

Syntax **xxBAnn** or **xxBA?**

Parameters

Description

xx [int]	—	Controller address.
nn [float]	—	Backlash value.

Range

xx	—	**1** to **31**
nn	—	**≥ 0** and **< 10^{12}**

Units

xx	—	None
nn	—	Preset units

Defaults

xx	Missing:	Error B.
	Out of range:	Error B.
nn	Missing:	Error C.
	Out of range:	Error C.

Description The BA command sets the backlash compensation value. This is the value that the controller moves the motor in addition to the commanded distance with any move that reverses the direction of motion without changing the current position value (TP command).

The BA command helps compensating for repeatable mechanical defects that appear when reversing the direction of motion, for instance mechanical wear. The value 0 disables this function. This feature can be only used when the hysteresis compensation (BH) is disabled.

When a value different from 0 is set, the travel range of the stage is decreased by the same amount.

Returns If the sign "**?**" is used instead 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.
H	—	Execution not allowed in NOT REFERENCED state.
J	—	Execution not allowed in DISABLE state.
K	—	Execution not allowed in READY state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Rel. Commands

BH	—	Set hysteresis compensation.

Example 1BA0.005 | *Set controller #1 backlash compensation to 0.005 units.*

BH — Set/Get hysteresis compensation

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	–	–	–

Syntax **xxBHnn** or **xxBH?**

Parameters

Description
xx [int] — Controller address.
nn [float] — Hysteresis value.

Range
xx — **1** to **31**
nn — **≥ 0** and **< 10^{12}**

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

The BH command sets the hysteresis compensation value. When set to a value different than zero, the controller will issue for each move in the positive direction a move of the commanded distance plus the hysteresis compensation value, and then a second move of the hysteresis compensation value in the negative direction. This motion ensures that a final position gets always approached from the same direction and distance and helps compensating for non-repeatable mechanical defects like hysteresis or mechanical stiffness variations.

The value 0 disables this function. The BH command can not be used when the backlash compensation is enabled (BA command).

When a value different from 0 is set, the travel range of the stage is decreased by the same amount in the positive direction.

NOTE

The homing set on the positive end of run and hysteresis compensation are not compatible. Any attempt to use both features together will make the stage fail.

Returns

If the sign "**?**" is used instead 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.
H — Execution not allowed in NOT REFERENCED state.
J — Execution not allowed in DISABLE state.
K — Execution not allowed in READY state.
L — Execution not allowed in HOMING state.
M — Execution not allowed in MOVING state.

Rel. Commands
BA — Set backlash compensation.

Example 1BH0.015 | *Set controller #1 backlash compensation to 0.015 units.*

FR — Set/Get stepper motor configuration

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	–	–	–

Syntax **xxFRSnn**, **xxFRM?** or **xxFRS?**

Parameters

Description

xx [int]	—	Controller address.
Mmm [int]	—	Amount of micro-steps per full step.
Snn [float]	—	Full step displacement length in 1/1000 of unit.

Range

xx	—	**1** to **31**
mm	—	**> 0** and **≤ 2000**
nn	—	**$> 10^{-6}$** and **$< 10^{12}$**

Units

xx	—	None.
Mmm	—	None.
Snn	—	1/1000 of unit.

Defaults

xx	Missing:	Error B.
	Out of range:	Error B.
mm	Missing:	Error C.
	Out of range:	Error C.
nn	Missing:	Error C.
	Out of range:	Error C.

Description FRM: For compatibility. No effect. Always 128 µsteps.

FRS: This command sets the displacement length per full step in 1/1000 of unit.

Returns If the sign "**?**" is used instead of **mm** or **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.
H	—	Execution not allowed in NOT REFERENCED state.
J	—	Execution not allowed in DISABLE state.
K	—	Execution not allowed in READY state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Example 1FRS10 | *Set controller #1 full step value to 10 milli-units.*

HT — Set/Get HOME search type

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	–	–	–

Syntax **xxHTnn** or **xxHT?**

Parameters

Description

xx [int] — Controller address.

nn [int] — Home search type identifier.

Range

xx — **1** to **31**

nn — **1** use current position as HOME.

2 use MZ switch (mechanical zero) to detect HOME position.

4 use EoR- switch (negative end of range) to detect HOME position.

Units

xx — None.

nn — None.

Defaults

xx Missing: Error B.

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

Description This command sets the type of HOME search used with the OR command.

NOTE

The homing set on the positive end of run and hysteresis compensation are not compatible. Any attempt to use both features together will make the stage fail.

Returns If the sign "**?**" is used instead 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.

H — Execution not allowed in NOT REFERENCED state.

J — Execution not allowed in DISABLE state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands

OH — Set/Get HOME search velocity.

OR — Execute HOME search.

OT — Set HOME search time-out.

Example 1HT1 | *Set controller #1 HOME sequence to use current position.*

ID — Set/Get stage identifier

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	□	□	–

Syntax **xxIDnn** or **xxID?**

Parameters

Description

xx [int]	—	Controller address.
nn [string]	—	Stage identifier string.

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.
nn	Missing:	Error C.
	Out of range:	Error C.

Description

The ID command sets the stage identifier in the form of a character string. Any printable character can be used ; spaces are admissible only if the string is enclosed in quotes, "like this". If not, spaces and tabs are ignored.

In CONFIGURATION state, this command sets a new value for the stage identifier which can then be saved in the device's non-volatile memory with 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 stage identifier. This value is not saved in the controller’s memory and will be lost after reboot.

Returns

If the sign “**?**” is used instead of **nn**, this command returns the current identifier string for the state in which the controller is (either CONFIGURATION or DISABLE/READY).

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.
H	—	Execution not allowed in NOT REFERENCED state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Example

1ID? |	*Get stage identifier for controller #1.*
1ID URS100CC |	*Set controller #1's stage identifier to: URS100CC.*

JR — Set/Get jerk time

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	□	□	–

Syntax **xxJRnn** or **xxJR?**

Parameters

Description
- **xx** [int] — Controller address.
- **nn** [float] — Jerk time value.

Range
- **xx** — **1** to **31**
- **nn** — **> 0.001** and **< 10^{12}**

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

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

Description Jerk is the derivative of acceleration. The jerk time defines the time to reach the needed acceleration. A longer jerk time reduces stress to the mechanics and smoothes motion.

In CONFIGURATION state, this command sets the value for the maximum jerk time which can then be saved in the controller's non-volatile 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 maximum jerk time. This value is not saved in the controller's memory and will be lost after reboot.

Returns If the sign "**?**" is used instead of **nn**, this command returns the current programmed value for the state in which the controller is (either CONFIGURATION or DISABLE/READY).

Errors
- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution impossible (axis in movement).
- H — Execution not allowed in NOT REFERENCED state.
- L — Execution not allowed in HOMING state.
- M — Execution not allowed in MOVING state.

Rel. Commands
- **AC** — Set/Get acceleration.
- **VA** — Set/Get velocity.

Example 1JR0.05 | *Set controller #1 jerk time to 0.05 seconds.*

MM — Enter/Leave DISABLE state

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	–	●	●	–

Syntax **xxMMnn** or **xxMM?**

Parameters

Description

xx [int]	—	Controller address.
nn [int]	—	Whether to enter (1) or leave (0) the DISABLE state.

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 (will forward this command to all controllers).
	Out of range:	Error B.
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 is executed on all controllers.

MM0 changes the controller's state from READY to DISABLE. In DISABLE state the control loop is open and the motor is not powered.

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 (depending on the closed-loop state). Any residual error is cleared and the motor is powered.

Returns If the sign "**?**" is used instead of **nn**, this command returns the current state. Refer to the TS command section 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.
H	—	Execution not allowed in NOT REFERENCED state.
I	—	Execution not allowed in CONFIGURATION state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Rel. Commands

PW	—	Enter/leave CONFIGURATION state.

Example

MM0	|	*All controllers go to DISABLE state.*

OH — Set/Get HOME search velocity

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	–	–	–

Syntax **xxOHnn** or **xxOH?**

Parameters

Description
- **xx** [int] — Controller address.
- **nn** [float] — HOME search velocity.

Range
- **xx** — **1** to **31**
- **nn** — **> 10^{-6}** and **< 10^{12}**

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

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

Description This command sets the maximum velocity used by the controller for the HOME search.

Returns If the sign "**?**" is used instead 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.
- H — Execution not allowed in NOT REFERENCED state.
- J — Execution not allowed in DISABLE state.
- K — Execution not allowed in READY state.
- L — Execution not allowed in HOMING state.
- M — Execution not allowed in MOVING state.

Rel. Commands
- **HT** — Set/Get HOME search type.
- **OR** — Execute HOME search.
- **OT** — Set HOME search time-out.

Example 1OH50 | *Set controller #1 HOME search velocity to 50 units/s.*

OR — Execute HOME search

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	–	–	–	–

Syntax **xxOR**

Parameters

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

Range **xx** — **1** to **31**

Units **xx** — None.

Defaults
xx Missing: Error B.
Out of range: Error B.
nn Missing: Error C.
Out of range: Error C.

Description This command starts the execution of the HOME search according to the algorithm defined by the HT command.

When in NOT REFERENCED state, for instance after system start, any positioner must first be homed with the OR command before further motion commands can be executed.

The OR command is accepted only in NOT REFERENCED state and only when no hardware error is present (except end-of-runs). Refer to the TS command to get more information on the possible hardware errors.

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.
E	—	home sequence already started.
I	—	Execution not allowed in CONFIGURATION state.
J	—	Execution not allowed in DISABLE state.
K	—	Execution not allowed in READY state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Rel. Commands

HT	—	Set HOME search type.
OH	—	Set HOME search velocity.
OT	—	Set HOME search time-out.

Example 1OR | *Execute HOME search with controller #1.*

OT — Set/Get HOME search time-out

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	–	–	–

Syntax **xxOTnn** or **xxOT?**

Parameters

Description
- **xx** [int] — Controller address.
- **nn** [float] — HOME time-out.

Range
- **xx** — **1** to **31**
- **nn** — **> 1** and **< 1000**

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

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

Description This command sets the time-out value for the HOME search. When the HOME search does not finish successfully before this delay elapses, the HOME search is aborted and an error is recorded.

Returns If the sign "**?**" is used instead 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.
- H — Execution not allowed in NOT REFERENCED state.
- J — Execution not allowed in DISABLE state.
- K — Execution not allowed in READY state.
- L — Execution not allowed in HOMING state.
- M — Execution not allowed in MOVING state.

Rel. Commands
- **HT** — Set HOME search type.
- **OH** — Set HOME search velocity.
- **OR** — Execute HOME search.

Example 1OT2.2 | *Set controller #1 HOME time-out to 2.2 seconds.*

PA — Move absolute

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	–	–	●	—

Syntax **xxPAnn** or **xxPA?**

Parameters

Description

xx [int]	—	Controller address.
nn [float]	—	New absolute position.

Range

xx	—	**1** to **31**
nn	—	≥ **SL** and ≤ **SR**

Units

xx	—	None.
nn	—	Preset units.

Defaults

xx	Missing:	Error B.
	Out of range:	Error B.
nn	Missing:	Error C.
	Out of range:	Error C.

Description The PA command initiates an absolute move. When received, the positioner will move, with the predefined acceleration and velocity, to the new absolute position specified by **nn**.

The PA command is only accepted in READY state, AND when the new absolute position is higher or equal to the negative software limit (SL), AND lower or equal to the positive software limit (SR).

The controller always rounds the new target position to the closest micro-step position.

Returns If the sign "**?**" is used instead of **nn**, this command returns the target absolute 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.
H	—	Execution not allowed in NOT REFERENCED state.
I	—	Execution not allowed in CONFIGURATION state.
J	—	Execution not allowed in DISABLE state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Rel. Commands

PR	—	Move relative.
PT	—	Get motion time for a relative move.
TH	—	Get set-point position.
TP	—	Get current position.

Example 1PA2.2 | *Move positioner on controller #1 to absolute position 2.2 units.*

PR — Move relative

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	–	–	●	—

Syntax **xxPRnn** or **xxPR?**

Parameters

Description

xx [int]	—	Controller address.
nn [float]	—	Displacement.

Range

xx	—	**1** to **31**
nn	—	≥ **(SL - TP)** and ≤ **(SR - TP)**

Units

xx	—	None.
nn	—	Preset units.

Defaults

xx	Missing:	Error B.
	Out of range:	Error B.
nn	Missing:	Error C.
	Out of range:	Error C.

Description The PR command initiates a relative move. When received, the positioner will move, with the predefined acceleration and velocity, to a new absolute position **nn** units away from the current absolute position.

The PR command gets only accepted in READY state, AND when the distance of the positioner to the software limit in the same direction is longer than the commanded displacement.

The controller always rounds the new target position to the closest micro-step position.

Returns If the sign "**?**" is used instead of **nn**, this command returns the target absolute 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.
H	—	Execution not allowed in NOT REFERENCED state.
I	—	Execution not allowed in CONFIGURATION state.
J	—	Execution not allowed in DISABLE state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Rel. Commands

PA	—	Move absolute.
PT	—	Get motion time for a relative move.
TH	—	Get set-point position.
TP	—	Get current position.

Example 1PR2.2 | *Move positioner on controller #1 to a new position 2.2 units away from the current position.*

PT — Get motion time for a relative move

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	–	●	●	●

Syntax **xxPTnn**

Parameters

Description
xx [int] — Controller address.
nn [float] — Displacement.

Range
xx — **1** to **31**
nn — > $\mathbf{10^{-6}}$ and < $\mathbf{10^{12}}$

Units
xx — None.
nn — Preset units.

Defaults
xx Missing: Error B.
Out of range: Error B.
nn Missing: Error C.
Out of range: Error C.

Description The PT commands helps evaluating move times for an efficient program flow.

When receiving the PT command, the controller computes and returns the time, in seconds, necessary to execute a relative move of displacement **nn** with the current working parameters (velocity, acceleration, etc.). The controller does not execute any displacement.

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.
H — Execution not allowed in NOT REFERENCED state.
I — Execution not allowed in CONFIGURATION state.

Rel. Commands
PA — Move absolute.
PR — Move relative.
TH — Get set-point position.
TP — Get current position.

Example
1PT2.2 | *Get time to move positioner on controller #1 by 2.2 units.*
1PT0.25 | *Controller returns: 0.25 seconds.*

PW — Enter/Leave CONFIGURATION state

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	–	–	–

Syntax **xxPWnn** or **xxPW?**

Parameters

Description

xx [int]	—	Controller address.
nn [int]	—	Whether to enter (1) or leave (0) the CONFIGURATION state.

Range

xx	—	**1** to **31**
nn	—	**1**: Go from NOT REFERENCED state to CONFIGURATION state.
		0: Go from CONFIGURATION state to NOT REFERENCED state.

Units

xx	—	None.
nn	—	None.

Defaults

xx	Missing:	Error B.
	Out of range:	Error B.
nn	Missing:	Error C.
	Out of range:	Error C.

Description PW1 changes the controller's state from NOT REFERENCED to CONFIGURATION. In CONFIGURATION state, all parameter settings are saved in the controller's memory upon exiting this state and remain available after switching off the controller. In addition, some settings are only possible in CONFIGURATION state (e.g. set drive voltage, set Backlash compensation, etc.).

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 NOT REFERENCED.

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 "**?**" is used instead of **nn**, this command returns whether we are or not in the CONFIGURATION 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.
K	—	Execution not allowed in READY state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

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 the warranty.

The PW command is used to change the default configuration parameters that are stored in memory, and not working parameters that may be changed on the fly.

RS — Reset controller

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	●	●	●

Syntax **xxRS**

Parameters

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

Range **xx** — **1** to **31**

Units xx — None.

Defaults **xx** Missing: Error B.

Out of range: Error B.

Description The RS command issues a hardware reset of the controller, equivalent to a power cycle.

To go from DISABLE or READY state to CONFIGURATION state, it is also needed to reset the controller with the RS command, and then to change the controller's state with the PW1 command from NOT REFERENCED to CONFIGURATION.

Errors A — Unknown message code or floating point controller address.

B — Controller address not correct.

D — Execution not allowed.

Example 1RS | *Reset controller #1.*

RS## — Reset controller's address

Usage	Not Ref.	Config.	Disable	Ready	Motion
	○	○	○	○	○

Syntax **xxRS## or RS##**

Parameters

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

Range **xx** — **0 to 31**

Units **xx** — None.

Defaults **xx** Missing: Change to 0 (will forward this command to all controllers).

Out of range: Error B.

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

Returns

Errors A — Unknown message code or floating point controller address.

B — Controller address not correct.

D — Execution not allowed.

Rel. Commands **SA** — Set/Get controller's RS-485 address.

Example RS## | *Reset all controllers' addresses to 1.*

SA — Set/Get controller's RS422 address

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	–	–	–

Syntax **xxSAnn** or **xxSA?**

Parameters

Description

xx [int]	—	Current controller address.
nn [int]	—	New controller address (a.k.a. RS422 address).

Range

xx	—	**1**
nn	—	**1** to **31**

Units

xx	—	None.
nn	—	None.

Defaults

xx	Missing:	Error B.
	Out of range:	Error B.
nn	Missing:	Error C.
	Out of range:	Error C.

Description The SA command sets the controller's RS422 address, also known as the axis number. This address needs to be different for each CONEX-PP when connected on a RS422 communication network.

Returns If the sign "**?**" is used instead 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.
H	—	Execution not allowed in NOT REFERENCED state.
J	—	Execution not allowed in DISABLE state.
K	—	Execution not allowed in READY state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Rel. Commands **RS##** — Reset controller's address.

Example 1SA3 | *Set controller's RS422 address to 3.*

SE — Configure/Execute simultaneous started move

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	–	–	●	–

Syntax **xxSEnn**, **xxSE?** or **SE**

Parameters

Description

xx [int] — Controller address.

nn [float] — New target position.

Range

xx — **0** to **31**

nn — ≥ **SL** and ≤ **SR**

Units

xx — None.

nn — Preset units.

Defaults

xx Missing: Change to 0 (will forward this command to all controllers).

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

Description

The SE command allows starting a move on different controllers at the same time.

The command xxSEnn sets a new target position for the controller **nn**. But contrarily to the PA/PR commands, the move is not executed immediately, but only after receipt of an SE command without preceding controller number nor position value. When receiving the 2nd SE command, all controllers start moving to their target position.

The xxSEnn command is only accepted in READY 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). The controller always rounds the new target position to the closest micro-step position.

The SE command should not be confused with a synchronized move. With a synchronized move, all positioners start moving simultaneously AND complete their moves at the same time. The SE command starts a move on all controllers at the same time, but each positioner moves with its individually defined velocity and acceleration. Consequently, the different positioners do NOT complete their moves at the same time.

Returns

If the sign "**?**" is used instead of **nn**, this command returns the target position value set by the SE command, which may not be the target position set by the PA/PR commands.

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.

H — Execution not allowed in NOT REFERENCED state.

I — Execution not allowed in CONFIGURATION state.

J — Execution not allowed in DISABLE state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands

PA / PR — Move absolute / relative.

TH — Get set-point position.

TP — Get current position.

Example 1SE2.2 | *Prepare controller #1 to move to absolute position 2.2 units.*

2SE3.3 | *Prepare controller #2 to move to absolute position 3.3 units.*

SE | *All controllers start their programmed move, if any.*

SL — Set/Get negative software limit

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	□	□	–

Syntax **xxSLnn** or **xxSL?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Negative software limit.

Range

xx — **1** to **31**

nn — **> -10^{12}** and **≤ 0**

Units

xx — None.

nn — Preset units.

Defaults

xx Missing: Error B.

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

Description

In CONFIGURATION state, this command sets the negative software limit which can then be saved in the controller's non-volatile 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 than or equal to the set-point 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. For an almost infinite motion, for instance with a rotative stage, set the lowest possible value, which is -999999930400.

Returns

If the sign "**?**" is used instead 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.

H — Execution not allowed in NOT REFERENCED state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands **SR** — Set positive software limit.

Example 1SL-100 | *Set controller #1 negative software limit to –100 units.*

SR — Set/Get positive software limit

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	□	□	–

Syntax **xxSRnn** or **xxSR?**

Parameters

Description
- **xx** [int] — Controller address.
- **nn** [float] — Positive software limit.

Range
- **xx** — **1** to **31**
- **nn** — **≥ 0** and **< 10^{12}**

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

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

Description

In CONFIGURATION state, this command sets the positive software limit which can than be saved in the controller's non-volatile 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 greater or equal to the set-point 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. For an almost infinite motion, for instance with a rotative stage, set the highest possible value, which is 999999930400.

Returns

If the sign "**?**" is used instead 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.
- H — Execution not allowed in NOT REFERENCED state.
- L — Execution not allowed in HOMING state.
- M — Execution not allowed in MOVING state.

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

Example

1SR100 | *Set controller #1 positive software positive to 100 units.*

ST — Stop motion

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	–	–	–	●

Syntax **[xx]ST**

Parameters

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

Range **xx** — **0** to **31**

Units **xx** — None.

Defaults **xx** Missing: Change to 0 (will forward this command to all controllers).

Out of range: Error B.

Description The ST command is a safety feature. It stops a move in progress by decelerating the positioner immediately with the acceleration defined by the AC command until it stops.

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

Errors

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

Example ST | *Stop moves on all controllers.*

TB — Get command error string

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	●	●	●

Syntax **xxTBnn**

Parameters

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

Range **xx** — **1** to **31**

nn [char] — Error code (refer to TE command).

Units **xx** — None.

Defaults **xx** Missing: Error B.

Out of range: Error B.

nn Missing: Returns explanation of current error as a literal string.

Out of range: Error C.

Description The TB command returns a literal character 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 last command error.

Example 1TB@ | *Get explanation to error code @.*

1TB@ No error | Controller returns: @ = means no error.

TE — Get last command error

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	●	●	●

Syntax **xxTE**

Parameters

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

Range **xx** — **1** to **31**

Units **xx** — None.

Defaults **xx** Missing: Error B.

Out of range: Error B.

Description The TE command returns the currently recorded error. When a command is not executable or fails, an error is recorded. This error can be read with the TE command. After the execution of a TE command, the error buffer is erased and another TE command will return @, which means "No error". When a new command error is generated before the previous command error is read, the new command's error will overwrite the currently 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 command 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.
- E — Home sequence already started.
- G — Displacement out of limits.
- H — Command not allowed in NOT REFERENCED state.
- I — Command not allowed in CONFIGURATION state.
- J — Command not allowed in DISABLE state.
- K — Command not allowed in READY state.
- L — Command not allowed in HOMING state.
- M — Command not allowed in MOVING state.
- N — Current position out of software limit.
- S — Communication Time Out.
- U — Error during EEPROM access.
- V — Error during command execution.

TH — Get set-point position

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	●	●	●

Syntax **xxTH**

Parameters

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

Range **xx** — **1** to **31**

Units **xx** — None.

Defaults **xx** Missing: Error B.

Out of range: Error B.

Description The TH command returns the value of the set-point or theoretical position. This is the position where the positioner should be. In MOVING state, the set-point position changes according to the calculation of the motion profiler. In READY state, the set-point position is equal to the target position.

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

1TH | *Get set-point position of controller #1.*

1TH0 | *Controller returns: set-point position = 0 units.*

TP — Get current position

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	●	●	●

Syntax **xxTP**

Parameters

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

Range **xx** — **1** to **31**

Units **xx** — None.

Defaults **xx** Missing: Error B.

Out of range: Error B.

Description The TP command returns the value of the current position. This is the position where the positioner actually is. In MOVING state, this value always changes. In READY state, this value should be equal or very close to the set-point and target position.

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

Errors A — Unknown message code or floating point controller address.

B — Controller address not correct.

D — Execution not allowed

Rel. Commands **TH** — Get set-point position.

TS — Get positioner error and controller state.

Example

1TP | *Get current position of controller #1.*

1TP0 | *Controller returns: actual position = 0 units.*

TS — Get positioner error and controller state

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	●	●	●

Syntax **xxTS**

Parameters

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

Range **xx** — **1** to **31**

Units **xx** — None.

nn — None.

Defaults **xx** Missing: Error B.

Out of range: Error B.

Description The TS command returns the positioner error and the current controller state.

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

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

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001	0000

E

each bit represents one possible error (exception made of bit C1):

A				B				C				D			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Not used	Not used	Not used	Not used	Driver overheating	Driver fault	Not used	Not used	No parameters in memory	Homing time out	Not used	Newport reserved (MZ status)	RMS current limit	Not used	Positive end of run	Negative end of run

NOTE

Bit C1 (MZ status) is not an error. It is reserved for Newport technicians to diagnose the mechanical zero sensor status during customer support and servicing.

Examples:

- Error map 0000 = No errors
- Error map 0002 = Positive end of run
- Error map 0048 = Homing time out, RMS current limit

Controller states (ef):

- **0A**: NOT REFERENCED from RESET.
- **0B**: NOT REFERENCED from HOMING.
- **0C**: NOT REFERENCED from CONFIGURATION.
- **0D**: NOT REFERENCED from DISABLE.
- **0E**: NOT REFERENCED from READY.
- **0F**: NOT REFERENCED from MOVING.
- **10**: NOT REFERENCED - NO PARAMETERS IN MEMORY.
- **14**: CONFIGURATION.
- **1E**: HOMING.
- **28**: MOVING.
- **32**: READY from HOMING.
- **33**: READY from MOVING.
- **34**: READY from DISABLE.
- **3C**: DISABLE from READY.
- **3D**: DISABLE from MOVING.

NOTE

The positioner error gets updated periodically, approx. every 1 ms.

The TS command reads the positioner error and clears it at the same time (same as what the command TE does with command errors). So when launching the TS command, it is important to process the TS feedback accordingly.

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
Rel. Commands	**TE**	—	Get last command error.

Example	1TS	|	*Get error and state of controller #1.*
	1TS00000A	|	*Controller returns: no errors and state is NOT REFERENCED from reset.*

VA — Set/Get velocity

Usage	Not Ref.	Config.	Disable	Ready	Motion
	–	○	□	□	–

Syntax **xxVAnn** or **xxVA?**

Parameters

Description

xx [int]	—	Controller address.
nn [float]	—	Velocity value.

Range

xx	—	**1** to **31**
nn	—	> $\mathbf{10^{-6}}$ and < $\mathbf{10^{12}}$

Units

xx	—	None.
nn	—	Preset units/s.

Defaults

xx	Missing:	Error B.
	Out of range:	Error B.
nn	Missing:	Error C.
	Out of range:	Error C.

Description In CONFIGURATION state, this command sets the maximum (i.e. cruise) velocity value which can then be saved in the controller's non-volatile memory using the PW command. This should be the maximum velocity that can be applied to the mechanical system. It is also the default velocity that will be used for all moves unless a lower value is set in DISABLE or READY state.

In DISABLE or READY state, this command sets the velocity used for all subsequent moves. Its value can be up to the programmed value set in the CONFIGURATION state. This value is not saved in the controller's memory and will be lost after reboot.

Returns If the sign "**?**" is used instead of **nn**, this command returns the current value for the state in which the controller is (either CONFIGURATION or DISABLE/READY).

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.
H	—	Execution not allowed in NOT REFERENCED state.
L	—	Execution not allowed in HOMING state.
M	—	Execution not allowed in MOVING state.

Rel. Commands

AC	—	Set/Get acceleration.
JR	—	Set/Get jerk time.

Example 1VA50 | *Set controller #1maximum velocity to 50 units/s.*

VE — Get controller revision information

Usage	Not Ref.	Config.	Disable	Ready	Motion
	●	●	●	●	●

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.

Description This command returns the controller's firmware revision information.

Errors A — Unknown message code or floating point controller address.

B — Controller address not correct.

Example 1VE | *Get controller #1 revision information.*

1VE FC family controller 2.0.0 | Controller returns revision number

ZT — Get all configuration parameters

Usage	Not Ref.	Config.	Disable	Ready	Motion
		●	●	●	●

Syntax **xxZT**

Parameters

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

Range **xx** — **1** to **31**

Units **xx** — None.

Defaults **xx** Missing: Error B.

Out of range: Error B.

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

The ZT command allows a quick review of all current stage parameter and simplifies the configuration of Newport stages, for instance by copying all the returned values into a configuration file which can be later fed back to the stage by simply pasting its contents in the terminal emulator when the stage needs to be quickly reconfigured.

Errors A — Unknown message code or floating point controller address

B — Controller address not correct

Example

```
1ZT   |  Get controller #1 configuration data.
1PW1
1AC320.000000
1BA0.000000
…
1VA80.000000
1PW0
```

Service Form

Your Local Representative

Tel.: _____

Fax:_____

Name: _____

Company:_____

Address:_____

Country:_____

P.O. Number:_____

Item(s) Being Returned:_____

Model#: _____

Return authorization #: _____
(Please obtain prior to return of item)

Date: _____

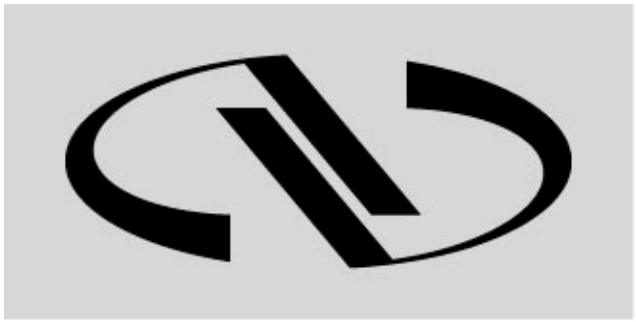
Phone Number: _____

Fax Number: _____

Serial #: _____

Description: _____

Reasons of return of goods (please list any specific problems): _____



Newport®

Experience | Solutions

Visit Newport Online at:
www.newport.com

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: 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.com

Technical Support
e-mail: tech_europe@newport.com

Service & Returns
Tel.: +33 (0)2.38.40.51.55