CONEX-CC

Single-Axis DC Motion with Controller/Driver

RoHS Compliant

Newport®

Controller Documentation

V2.0.x

Warranty

Newport Corporation warrants that this product will be free from defects in material and workmanship and will comply with Newport's published specifications at the time of sale for a period of one year from date of shipment. If found to be defective during the warranty period, the product will either be repaired or replaced at Newport's option.

To exercise this warranty, write or call your local Newport office or representative, or contact Newport headquarters in Irvine, California. You will be given prompt assistance and return instructions. Send the product, freight prepaid, to the indicated service facility. Repairs will be made and the instrument returned freight prepaid. Repaired products are warranted for the remainder of the original warranty period or 90 days, whichever occurs last.

Limitation of Warranty

The above warranties do not apply to products which have been repaired or modified without Newport's written approval, or products subjected to unusual physical, thermal or electrical stress, improper installation, misuse, abuse, accident or negligence in use, storage, transportation or handling.

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR USE. NEWPORT CORPORATION SHALL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM THE PURCHASE OR USE OF ITS PRODUCTS.

©2017 by Newport Corporation, Irvine, CA. All rights reserved.

Original instructions.

No part of this document may be reproduced or copied without the prior written approval of Newport Corporation. This document is provided for information only, and product specifications are subject to change without notice. Any change will be reflected in future publishings.

Table of Contents

1.0 System Overview 1
1.1 General Description 1
1.2 CONEX-CC 1
1.2.1 Contents of Delivery 1
1.2.2 Specifications 2
1.2.3 Dimensions 2
1.3 CONEX-PS 3
1.3.1 Specifications 3
1.3.2 Dimensions 3
1.4 System Environmental Specifications 3
1.5 Connector Identification 4
1.6 USB Communication Settings 4

2.0 Programming 5
2.1 State Diagram 5
2.2 Command Syntax 8
2.3 Command Execution Time 8
2.4 Command Set 9
AC — Set/Get acceleration 11
BA — Set/Get backlash compensation 12
BH — Set/Get hysteresis compensation 13
DV — Set/Get driver voltage 14
FD — Set/Get low pass filter cut off frequency for Kd 15
FE — Set/Get following error limit 16
FF — Set/Get friction compensation 17
HT — Set/Get HOME search type 18
ID — Set/Get stage identifier 19
JR — Set/Get jerk time 20
KD — Set/Get derivative gain 21
KI — Set/Get integral gain 22
KP — Set/Get proportional gain 23
KV — Set/Get velocity feed forward 24
MM — Enter/Leave DISABLE state 25
OH — Set/Get HOME search velocity 26
OR — Execute HOME search 27
OT — Set/Get HOME search time-out 28
PA — Move absolute 29
PR — Move relative 30

PT — Get motion time for a relative move 31
PW — Enter/Leave CONFIGURATION state 32
QI — Set/Get motor's current limits 33
RS — Reset controller 34
RS## — Reset controller's address 35
SA — Set/Get controller's RS-485 address 36
SC — Set/Get control loop state 37
SE — Configure/Execute simultaneous started move 38
SL — Set/Get negative software limit 40
SR — Set/Get positive software limit 41
ST — Stop motion 42
SU — Set/Get encoder increment value 43
TB — Get command error string 44
TE — Get last command error 45
TH — Get set-point position 46
TK — Enter/Leave TRACKING mode 47
TP — Get current position 48
TS — Get positioner error and controller state 49
VA — Set/Get velocity 51
VE — Get controller revision information 52
ZT — Get all configuration parameters 53

3.0 Connector interfaces 54
3.1 24 V Connector (Female Ø 2.1 x Ø 5.5 x 11 mm) 54
3.2 Mini-USB (Male) Connector Pinout 54

Service Form 55

Single-Axis DC Motor Controller/Driver CONEX-CC

1.0 System Overview

1.1 General Description

The CONEX-CC is a single axis motion controller/driver for DC servo motors up to 24 VDC at 0.3 Apeak. It provides a very compact and low-cost solution for driving a variety of Newport motorized stages from a PC.

Communication with the CONEX-CC 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 two letter mnemonic commands.

1.2 CONEX-CC

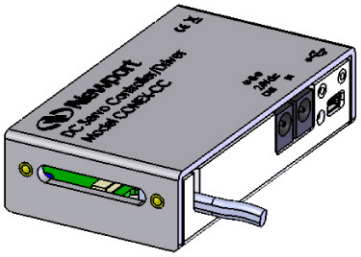
1.2.1 Contents of Delivery

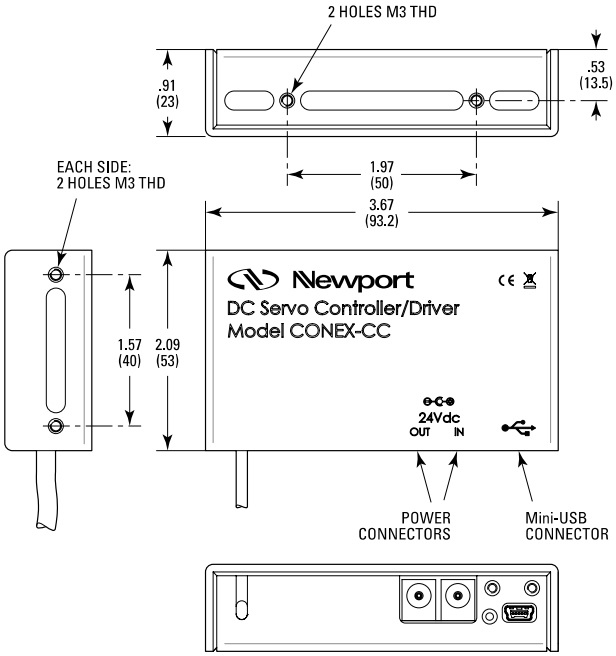
- CONEX-xxx Controller box with associated stage (cable length: 1.8m length).
- CONEX-PSC0.1 Power cable, 0.1 m length.
- CONEX-USB USB cable, 1.8 m length.
- CONEX-MOTION CD-ROM.

1.2.2 Specifications

General Description	Single-axis motion controller/driver for DC servo motors
Control Capability	DC servo motors, open or closed loop
Motor Output Power	24 VDC at 0.15 Arms, 0.3 A peak linear amplifier
Control loop	– Floating point digital PID loop with velocity and friction feed forward – 2 kHz servo rate – Backlash compensation
Motion	Point-to-point motion with S-gamma profile and jerk time control, or motion with trapezoidal profile with possible on the fly modification of the target position.
Computer interface	– USB (requires Windows™ operating system)
Programming	– 40+ intuitive, 2-letter ASCII commands – Command set includes software limits, user units, synchronized motion start, stop all
Dedicated inputs	– TTL encoder inputs for A, B, and I, max. 2 MHz rate – Forward and reverse limit, home switch
Status display	Two-color LED
Communication rate	50 Hz Max. (USB)
Internal safety feature	Watchdog timer
Consumption	+5 V (USB): < 0.5 A , +24 V (CONEX –PS): < 8 A

1.2.3 Dimensions





1.3 CONEX-PS

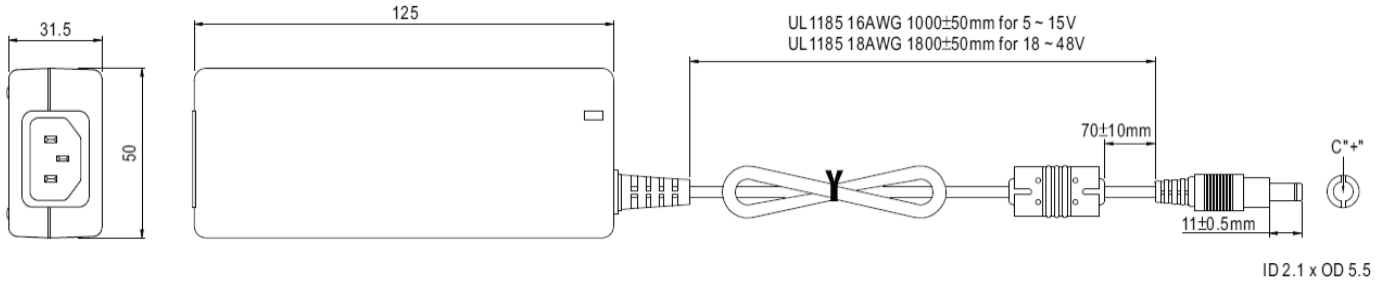
1.3.1 Specifications

AC Input	100–240 VAC, 47–63 Hz, 1.9 A
DC Output	24 V, 40 W max.
Connector	Male: Ø 2.1 x Ø 5.5 x 11 mm

NOTE

CONEX-PS can power up to 5 CONEX-CC Controller/Drivers.

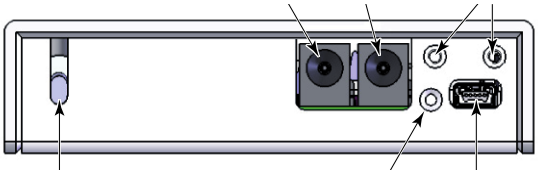
1.3.2 Dimensions



1.4 System Environmental Specifications

Operating temperature	5 °C to 40 °C
Operating humidity	20% to 85% relative humidity, non-condensing
Location	Indoor use only

1.5 Connector Identification



USB	Mini-USB connector
LED	Status LED
24 V in	Ø 2.1 x Ø 5.5 x 11 mm: Power supply input (connect to CONEX-PS)
24 V out	Ø 2.1 x Ø 5.5 x 11 mm: Power supply repeater for connecting several CONEX-CC to the same power supply
STAGE	Stage entry cable
Cable retainer	2 x M3 threaded hole to attach cable retainer

1.6 USB Communication Settings

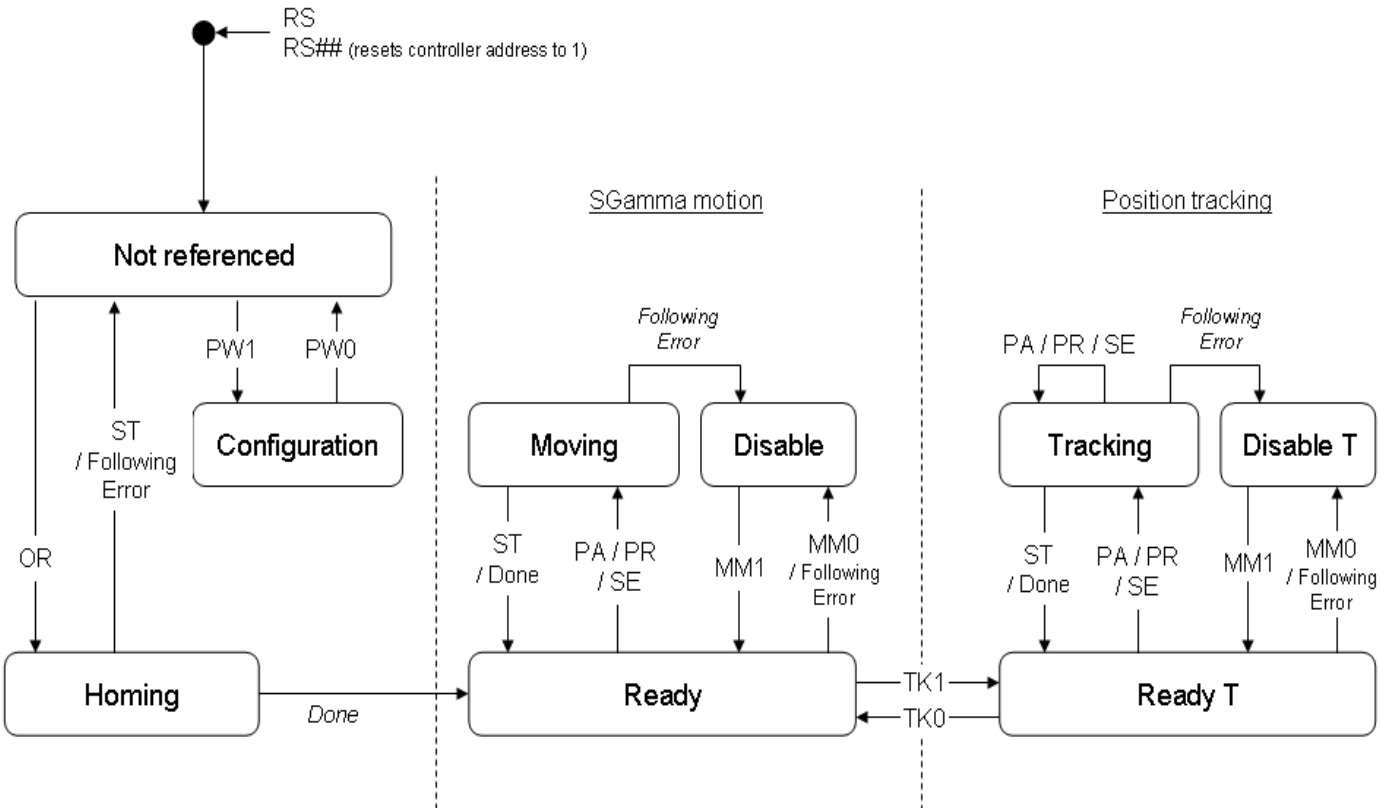
Communication parameters are preset in the CONEX-CC controller and do not require any configuration:

Bits per second	921,600
Data bits	8
Parity	None
Stop bits	1
Flow control	Xon/Xoff
Terminator	C_RL_F

2.0 Programming

2.1 State Diagram

For a safe and consistent operation, the CONEX-CC uses 9 different operation states: Not referenced, Configuration, Homing, Ready, Ready T, Disable, Disable T, Moving and Tracking. In each state, only specific commands are accepted by the CONEX-CC. Therefore, it is important to understand the state diagram below and which commands and actions cause transition between the different states. Also see section 2.4 for command/state information:



End of Runs encountered in the following state

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 change to NOT REFERENCED state.
TRACKING:	Abort motion and then change to NOT REFERENCED state.
READY:	Change to NOT REFERENCED state.
DISABLE:	Change to NOT REFERENCED state.

LED display

NOT REFERENCED:	If everything is OK then **SOLID ORANGE**.
NOT REFERENCED:	If hardware faults or wrong parameters then **SOLID RED**.
NOT REFERENCED:	If end of runs then **SLOW BLINK ORANGE**.

CONFIGURATION:	**SLOW BLINK RED**.
READY:	**SOLID GREEN**.
READY T:	**SOLID GREEN**.
DISABLE:	**SLOW BLINK GREEN**.
DISABLE T:	**SLOW BLINK GREEN.**
HOMING:	**FAST BLINK GREEN**.
MOVING:	**FAST BLINK GREEN**.
TRACKING:	**FAST BLINK GREEN**

When connecting the CONEX-CC to power, the controller must be first initialized. When the initialization is successful, the controller gets to the NOT REFERENCED state. From the NOT REFERENCED state, the controller can go to the CONFIGURATION state with the PW1 command. In CONFIGURATION stage, the CONEX-CC allows changing all stage and motor configuration parameters like maximum motor current or travel limits. The PW0 command saves all changes to the controller's memory and returns the controller back to the NOT REFERNCED state.

In Sgamma motion mode:

To execute any move commands (PA, PR), the controller must be in READY state. To get from the NOT REFERENCED state to the READY state, the positioner must be homed first with the OR command. During homing (OR command execution), the controller is in HOMING state. When the homing is successful, the controller automatically gets to the READY state. The process for homing, and which signals are looked for during homing, can be defined with the HT command.

In READY state the motor is energized and the control loop is closed (when control loop state is closed, SC1). During a move execution (PA/PR), the controller is in MOVING state and gets automatically back to the READY state when the move is completed successfully. A following error during a move changes the controller to DISABLE state. Other errors, for instance a loss of the encoder signals, may change the controller to the NOT REFERENCED state.

In DISABLE state the motor is not energized and the control loop is open. But the encoder is still read and the current position gets updated. The DISABLE state can be used for instance for manual adjustments or to make sure that no energy goes to the motor. To go from READY state to DISABLE state and vice versa, use the MM command.

To get from READY state or DISABLE state back to the NOT REFERENCED state, for instance to make some further parameter change in CONFIGURATION state, you need to reboot the controller with the RS command.

In Position Tracking mode:

The Position Tracking mode of operation is accessed using the TK1 command. The main difference with Sgamma motion mode is that it is possible to update the target position on the fly.

In this mode, the Open/Closed loop, jerktime do not apply. The other features (backlash compensation, Disable) and error handling are the similar to that of Sgamma motion mode.

2.2 Command Syntax

The CONEX-CC 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	AA	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 C_RL_F (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-CC.

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

2.3 Command Execution Time

The CONEX-CC 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 receive of the 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-CC. 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 "**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. For example, a 1VA10 sets the velocity of the controller #1 to 10 units/second. A 1VA? sends the response 1VA10.

Not every command can be executed in all states of the CONEX-CC 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	Tracking	Description
AC	–	○	□	□	–	–	Set/Get acceleration
BA	–	○	–	–	–	–	Set/Get backlash compensation
BH	–	○	–	–	–	–	Set/Get hysteresis compensation
DV	–	○	–	–	–	–	Set/Get driver voltage
FD	–	○	□	–	–	–	Set/Get low pass filter for Kd
FE	–	○	□	–	–	–	Set/Get following error limit
FF	–	○	□	–	–	–	Set/Get friction compensation
HT	–	○	–	–	–	–	Set/Get HOME search type
ID	–	○	□	□	–	–	Set/Get stage identifier
JR	–	○	□	□	–	–	Set/Get jerk time
KD	–	○	□	–	–	–	Set/Get derivative gain
KI	–	○	□	–	–	–	Set/Get integral gain
KP	–	○	□	–	–	–	Set/Get proportional gain
KV	–	○	□	–	–	–	Set/Get velocity feed forward
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 motion time for a relative move
PW	●	●	–	–	–	–	Enter/Leave CONFIGURATION state
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
SC	–	○	○	–	–	–	Set/Get control loop state
SE	–	–	–	●	–	–	Configure/Execute simultaneous started move
SL	–	○	□	□	–	–	Set/Get negative software limit
SR	–	○	□	□	–	–	Set/Get positive software limit
ST			–	–	●	●	Stop motion
SU	–	○	–	–	–	–	Set/Get encoder increment value
TB	●	●	●	●	●	●	Get command error string
TE	●	●	●	●	●	●	Get last command error
TH	●	●	●	●	●	●	Get set–point position
TK	–	–	–	●	–	–	Enter/Leave Tracking mode
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

Motion Corresponds to HOMING and MOVING state (for details see state diagram, section 2.1).

Ready Corresponds to READY and READY T states.

Disable Corresponds to DISBABLE and DISABLE T 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.

– Write command not accepted (will return an error).

Command 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	Tracking
	–	○	□	□	–	–

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.

Floating point: Error A.

nn Missing: Error C.

Out of range: Error C.

Description

In CONFIGURATION state, this command sets the maximum acceleration value which can than be saved in the controller's nonvolatile 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 the following 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 "**?**" 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.

H — Execution not allowed in NOT REFERENCED state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING state.

Rel. Commands

VA — Set 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	Tracking
	–	○	–	–	–	–

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

Parameters

Description

xx [int] — Controller address.

nn [float] — Backlash value.

Range

xx — **1** to **31**

nn — **≥ 0** and **< 1E12**

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 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 play. The value 0 disables this function. This feature can be only used when the hysteresis compensation (BH) is disabled.

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.

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.

P — Execution not allowed in TRACKING 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	Tracking
	–	○	–	–	–	–

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).

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.
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.
P — Execution not allowed in TRACKING state.

Rel. Commands **BA** — Set backlash compensation.

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

DV — Set/Get driver voltage

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

Syntax **xxDVnn** or **xxDV?**

Parameters

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

Range
xx — **1** to **31**
nn — ≥ **12** and ≤ **48**

Units
xx — None.
nn — Volts

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

Description This command sets the max. output voltage of the driver to the motor.

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.
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.
P — Execution not allowed in TRACKING state.

Rel. Commands **QI** — Set current limit.

Example 1DV48 | *Set controller #1 maximum output voltage to 48 V.*

FD — Set/Get low pass filter cut off frequency for Kd

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

Syntax **xxFDnn** or **xxFD?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Cut off frequency value.

Range

xx — **1** to **31**

nn — **> 10^{-6}** and **< 2000**

Units

xx — None.

nn — Hertz

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 value for the low pass filter cut-off frequency 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 low pass filter cut-off frequency. 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.

H — Execution not allowed in NOT REFERENCED state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING state.

Rel. Commands **SC** — Set closed loop state.

Example 1FD1500 | *Set controller #1 Kd cut-off frequency to 1500 Hz.*

FE — Set/Get following error limit

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

Syntax **xxFEnn** or **xxFE?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Following error limit value.

Range

xx — **1** to **31**

nn — **> 10^{-6}** 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

In CONFIGURATION state, this command sets the value for the maximum allowed following error 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 for the closed-loop control unless a different value is set in DISABLE state.

The following error is the most important parameter to control motion. It is the difference between the set point (or theoretical) position and the current (or encoder) position. When the current following error exceeds the maximum allowed value, a following error is issued and the controller is set to DISABLE state.

In DISABLE state, this command allows setting a new working parameter for the maximum allowed following error. 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.

H — Execution not allowed in NOT REFERENCED state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING state.

Rel. Commands

SC — Set closed loop state.

Example

1FE0.015 | *Set controller #1 following error limit to 0.015 units.*

FF — Set/Get friction compensation

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

Syntax **xxFFnn** or **xxFF?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Friction compensation value.

Range

xx — **1** to **31**

nn — **≥ 0** and **< DV**

Units

xx — None.

nn — Volt * second/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 value for the friction compensation 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 for any move unless a different value is set in DISABLE state.

The FF command helps minimizing the following error with systems that have significant friction. The value for the friction compensation is the voltage that gets added to the output voltage whenever the set point (or theoretical) velocity is different from zero. The sign of this voltage is the same as the sign of the set point velocity.

In DISABLE state, this command allows setting a new working parameter for the friction compensation. 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.

H — Execution not allowed in NOT REFERENCED state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING state.

Rel. Commands

SC — Set closed loop state.

Example

1FF0.15 | *Set controller #1 friction compensation to 0.15 V * s/units.*

HT — Set/Get HOME search type

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

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

Parameters

Description

xx [int] — Controller address.

nn [int] — Home type value.

Range

xx — **1** to **31**

nn — **0** use MZ switch and encoder Index.
1 use current position as HOME.
2 use MZ switch only.
3 use EoR- switch and encoder Index.
4 use EoR- switch only.

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 This command sets the type of HOME search used with the OR 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.
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.
P — Execution not allowed in TRACKING state.

Rel. Commands **OR** — Execute HOME search.

Example 1HT0 | *Set controller #1 HOME sequence to use MZ and encoder index.*

ID — Set/Get stage identifier

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

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

Parameters

Description
xx [int] — Controller address.
nn [float] — 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 return the stage identifier. When used with Newport ESP compatible stages (see blue label on the product), this is the identical to the Newport product name.

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.
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.
P — Execution not allowed in TRACKING state.

Example
1ID? | *Get stage identifier for controller #1.*
1ID URS100CC | *Controller returns stage identifier: URS100CC.*

JR — Set/Get jerk time

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

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

Description In CONFIGURATION state, this command sets the value for the maximum jerk time 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.

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 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 "**?**" 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 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.
P — Execution not allowed in TRACKING state.

Rel. Commands **AC** — Set positioner acceleration.

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

KD — Set/Get derivative gain

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

Syntax **xxKDnn** or **xxKD?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Derivative gain value.

Range

xx — **1** to **31**

nn — **≥ 0** and **< 10^{12}**

Units

xx — None.

nn — Volt * second/preset unit.

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 derivative 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.

H — Execution not allowed in NOT REFERENCED state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING state.

Rel. Commands

SC — Set closed loop state.

KI — Set integral gain.

KP — Set proportional gain.

KV — Set velocity feed forward.

Example

1KD0.015 | *Set controller #1 derivative gain to 0.015.*

KI — Set/Get integral gain

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

Syntax **xxKInn** or **xxKI?**

Parameters

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

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

Units
xx — None.
nn — Volt * preset unit/second.

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 integral 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.
H — Execution not allowed in NOT REFERENCED state.
K — Execution not allowed in READY state.
L — Execution not allowed in HOMING state.
M — Execution not allowed in MOVING state.
P — Execution not allowed in TRACKING state.

Rel. Commands
SC — Set closed loop state.
KD — Set derivative gain.
KP — Set proportional gain.
KV — Set velocity feed forward.

Example 1KI0.015 | *Set controller #1 integral gain to 0.015.*

KP — Set/Get proportional gain

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

Syntax **xxKPnn** or **xxKP?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Proportional gain value.

Range

xx — **1** to **31**

nn — **≥ 0** and **< 10^{12}**

Units

xx — None.

nn — Volt/preset unit

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.

H — Execution not allowed in NOT REFERENCED state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING state.

Rel. Commands

SC — Set closed loop state.

KD — Set derivative gain.

KI — Set integral gain.

KV — Set velocity feed forward.

Example

1KP0.015 | *Set controller #1 proportional gain to 0.015.*

KV — Set/Get velocity feed forward

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

Syntax **xxKVnn** or **xxKV?**

Parameters

Description

xx [int] — Controller address.

nn [float] — Velocity feed forward value.

Range

xx — **1** to **31**

nn — **≥ 0** and **< 10^{12}**

Units

xx — None.

nn — Volt * second/preset unit

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 velocity feed forward 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.

H — Execution not allowed in NOT REFERENCED state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING state.

Rel. Commands

SC — Set closed loop state.

KD — Set derivative gain.

KI — Set integral gain.

KP — Set proportional gain.

Example

1KV0.015 | *Set controller #1 velocity feed forward to 0.015.*

MM — Enter/Leave DISABLE state

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

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

Parameters

Description

xx [int] — Controller address.

nn [float] — Velocity feed forward value.

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. In DISABLE state the control loop is open and the motor is not energized . The encoder, though, is still read and the current position gets 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 (depending on the closed-loop state). The residual following error gets cleared from the buffer and the motor gets energized.

Returns

If the sign "**?**" takes place 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.

P — Execution not allowed in TRACKING 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	Tracking
	–	○	–	–	–	–

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

Parameters

Description
xx [int] — Controller address.
nn [float] — HOME high 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.
Floating point: Error A.
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 "**?**" 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.
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.
P — Execution not allowed in TRACKING state.

Rel. Commands
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	Tracking
	●	–	–	–	–	–

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.

Floating point: Error A.

nn Missing: Error C.

Out of range: Error C.

Description This command starts the execution of the HOME search as defined by the HT command.

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

The OR command gets accepted only in NOT REFERENCED state and only with no present hardware errors, except for end-of-run maybe. 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.

P — Execution not allowed in TRACKING 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	Tracking
	–	○	–	–	–	–

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

Parameters

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

Range
xx — **1** to **31**
nn — **> 1** and **< 10^3**

Units
xx — None.
nn — Seconds

Defaults
xx Missing: Error B.
Out of range: Error B.
Floating point: Error A.
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 time elapses, the HOME search will be aborted and an error gets recorded.

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.
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.
P — Execution not allowed in TRACKING 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	Tracking
	–	–	–	●	—	●

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

Parameters

Description

xx [int] — Controller address.

nn [float] — New target 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.

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, with the predefined acceleration and velocity, to the new target position specified by **nn**.

The PA command gets only accepted in READY, READY T or TRACKING states, 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).

To avoid any mismatch, the controller always rounds the new target position to the closest encoder position.

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.

H — Execution not allowed in NOT REFERENCED state.

I — Execution not allowed in CONFIGURATION state.

J — Execution not allowed in DISABLE state.

M — Execution not allowed in MOVING state.

Rel. Commands

PR — Move relative.

TH — Get set-point position.

TP — Get current position.

SU — Set encoder increment value.

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

PR — Move relative

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

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

Parameters

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

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.
Floating point: Error A.
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 target position **nn** units away from the current target position.

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

To avoid any mismatch, the controller always rounds the new target position to the closest encoder position.

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.
H — Execution not allowed in NOT REFERENCED state.
I — Execution not allowed in CONFIGURATION state.
J — Execution not allowed in DISABLE state.
M — Execution not allowed in MOVING state.

Rel. Commands
PA — Move absolute.
TH — Get set-point position.
TP — Get current position.
SU — Set encoder increment value.

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

PT — Get motion time for a relative move

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

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.
Floating point: Error A.
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 returns the time, in seconds, necessary to execute a relative move of the displacement **nn** with the current working parameters (velocity, acceleration, etc.). The controller does not execute any motion.

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.
SU — Set encoder increment value.

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	Tracking
	●	●	–	–	–	–

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

Parameters

Description

xx [int] — Controller address.

nn [float] — Velocity feed forward value.

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.

Floating point: Error A.

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 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 10 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.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

P — Execution not allowed in TRACKING 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 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.

QI — Set/Get motor's current limits

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

Syntax **xxQILnn**, **xxQIRnn**, **xxQITnn**, **xxQIL?**, **xxQIR?** or **xxQIT?**

Parameters

Description

xx [int] — Controller address.

Lmm [float]— Motor's peak current limit.

Rnn [float]— Motor's rms current limit.

Tpp [float]— Motor's rms current averaging time.

Range

xx — **1** to **31**

mm — ≥ **0.05** and ≤ **3.0**

nn — ≥ **0.05** and ≤ **1.5** and ≤ **mm**

pp — > **0.01** and ≤ **100**

Units

xx — None.

mm — Amperes.

nn — Amperes.

pp — Seconds.

Defaults

xx Missing: Error B.

Out of range: Error B.

Floating point: Error A.

mm Missing: Error C.

nn Missing: Error C.

pp Missing: Error C.

Out of range: Error C.

Description

QIL: Sets the controller's maximum or peak output current limit to the motor. When the controller detects a higher current than the peak current limit, it will generate a hardware error and a fault will be recorded.

QIR: Sets the controller's rms output current limit to the motor. The rms current limit must be lower than the peak current limit. When the controller's output current exceeds the rms current limit, it will generate a hardware error and a fault will be recorded.

QIT: Sets the controller's averaging period for rms current calculation. In general, the QIT command defines for how long time the actual motor current is allowed to exceed the rms output current limit.

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.

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.

P — Execution not allowed in TRACKING state.

Rel. Commands DV — Set driver input voltage.

Example

1QIL0.75 | *Set controller #1 current limit to 0.75 A.*

1QIR0.25 | *Set controller #1 rms current limit to 0.25 A.*

1QIT2.5 | *Set controller #1 rms averaging period to 2.5 s.*

RS — Reset controller

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

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.
Floating point: Error A.

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

To go from DISABLE or READY state to CONFIGURATION state, it is also needed to first 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	Tracking
	○	○	○	○	○	○

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

Parameters

Description **xx** [int] — Axis number.

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 RS## command resets the controller's address to 1. This address needs to be different for each CONEX-CC 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.

Example RS## | Res*et controller's address to 1.*

SA — Set/Get controller's RS-485 address

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

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

Parameters

Description

xx [int] — Axis number.

nn [int] — Controller's axis number.

Range

xx — **1**

nn — **2** to **31**

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 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 can only be sent to a controller configured for RS-232-C communication. In this configuration, the controller's address is 1. Only one controller can be configured for RS-232-C communication.

Newport recommends using the supplied utility software for all controller configurations. 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.

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

1SA3 | *Set controller's RS-485 address to 3.*

SC — Set/Get control loop state

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

Syntax **xxSCnn** or **xxSC?**

Parameters

Description

xx [int] — Controller address.

nn [int] — Closed loop state.

Range

xx — **1** to **31**

nn — **1**: CLOSED loop control.

0: OPEN loop control.

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 SC1 sets the controller to CLOSED loop control. This is the default.

SC0 sets the controller to OPEN loop control. Open loop control might be useful for defining stage parameters like friction compensation or velocity feed forward.

SC is not applicable in Tracking mode.

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.

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

KD — Set derivative gain.

KI — Set integral gain.

KP — Set proportional gain.

KV — Set velocity feed forward.

Example 1SC1 | *Set controller #1 to closed loop control.*

SE — Configure/Execute simultaneous started move

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

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.
Out of range: Error B.
Floating point: Error A.
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 different than the PA command, the move does not get executed immediately, but only after receipt of an SE command without preceding controller number and without following position value. When receiving the SE command, all controllers start a move to their new target position.

The xxSEnn command gets 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). To avoid any mismatch, the controller always rounds the new target position to the closest encoder position.

The SE command should not be confused with a synchronized move. With a synchronized move, all positioners start their motion simultaneously and have velocities, accelerations and jerk times which are limited to a rate which make all positioners start and complete their moves at the same time. The emphasis here is that they all start AND stop 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. So naturally, the different positioners don't complete their motion at the same time.

Returns If the sign "**?**" takes place of **nn**, this command returns the target position value set by the SE command, which is not necessarily the same as the target position set by the PA command.

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	**PR**	—	Move relative.
	TH	—	Get set-point position.
	TP	—	Get current position.
	SU	—	Set encoder increment value.
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	Tracking
	–	○	□	□	–	–

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.

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 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 rotation stage, set the lowest possible value, which is: -2147000000 * "encoder increment value" (see SU command). For instance if the encoder increment value is 0,0005, this limit is -1073500.

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.

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	Tracking
	–	○	□	□	–	–

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.

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 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 positive software limit. It must be larger 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 rotation stage, set the largest possible value, which is: 2147000000 * "encoder increment value" (see SU command). For instance if the encoder increment value is 0,0005, this limit is 1073500.

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.

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	Tracking
	–	–	–	–	●	●

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 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 preceding controller address stops a move in progress on controller xx. The ST command without 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.

Example

ST |	*Stop moves on all controllers.*

SU — Set/Get encoder increment value

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

Syntax **xxSUnn** or **xxSU?**

Parameters

Description
xx [int] — Controller address.
nn [float] — Equivalent units to one encoder count.

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

Units
xx — None.
nn — 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 SU command sets the value for one encoder count. It defines also the system of units for all other parameters like travel limits, velocities, accelerations, etc. Therefore, it is the first parameter to be defined for any positioner.

Example: For a positioner with an encoder resolution of 1 µm, the command xxSU0.001 sets 1 encoder count = 1 µm = 0.001 unit or 1 unit = 1 mm.

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.
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 1SU7.5e-6 | *Set controller #1 encoder increment to 7.5 * 10^{-6} units.*

TB — Get command error string

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

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.

Floating point: Error A.

nn Missing: Returns explanation of current error.

Out of range: Error C.

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

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

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.
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.
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.
P — Command not allowed in TRACKING state.
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	Tracking
	●	●	●	●	●	●

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.

Floating point: Error A.

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

Rel. Commands TP — Get current position.

Example

1TH	|	*Get set-point position of controller #1.*
1TH0	|	*Controller returns: set-point position = 0 units.*

TK — Enter/Leave TRACKING mode

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

Syntax **xxTKnn**

Parameters

Description
xx [int] — Controller address.
nn [int] — Mode

Range
xx — **1** to **31**
nn — **0** or **1**

Units
xx — None.
nn — None.

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

Description The TK command enables to enter or leave Tracking Mode.

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 — Command not allowed in DISABLE state.
L — Command not allowed in HOMING state.
M — Command not allowed in MOVING state.
P — Command not allowed in TRACKING state.

Rel. Commands TS — Get positioner error and controller state.

Example
1TK1 | *Enter tracking mode on the controller #1.*

TP — Get current position

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

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.

Floating point: Error A.

Description The TP command returns the value of the current position. This is the position where the positioner actually is according to his encoder value. 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 is completed.

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.

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

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	Tracking
	●	●	●	●	●	●

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.

Floating point: Error A.

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 in Hexadecimal. The last two characters (ef) represent the controller state.

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
ach bit represents one possible error:

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	• Not used	• Not used	• 80 W output power exceeded	• DC voltage too low	• Wrong ESP stage	• Homing time out	• Following error	• Short circuit detection	• RMS current limit	• Peak current limit	• Positive end of run	• Negative end of run

Examples:

- Error map 0000 = No errors
- Error map 0013 = Short circuit detection, Positive end of run, negative end of run
- Error map 004C = Homing time out, RMS current limit, Peak 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.
- **36**: READY T from READY.
- **37**: READY T from TRACKING.
- **38**: READY T from DISABLE T.
- **3C**: DISABLE from READY.
- **3D**: DISABLE from MOVING.
- **3E**: DISABLE from TRACKING.
- **3F**: DISABLE from READY T.
- **46**: TRACKING from READY T.
- **47**: TRACKING from TRACKING.

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.

The error "Wrong EPS stage" gets only detected during the booting of the controller. When read the error is cleared.

With no errors in the error buffer the color of the LED will change from red to either green or orange depending on the controller state.

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
Rel. Commands	**TE**	—	Get last error.
Example	1TS	|	*Get error and state of controller #1.*
	1TS00000A	|	*Controller returns: no errors and NOT REFERENCED from reset.*

VA — Set/Get velocity

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

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

Description In CONFIGURATION state, this command sets the maximum velocity value which can than be saved in the controller's nonvolatile memory using the PW command. This is 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 the following 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 "**?**" 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.
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 positioner acceleration.

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

VE — Get controller revision information

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

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-CC V2.0.0. | Controller returns revision number

ZT — Get all configuration parameters

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

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.
Floating point: Error A.

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 non Newport stages, for instance by using Hyper Terminal file transfer.

Errors A — Unknown message code or floating point controller address
B — Controller address not correct

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

Example

```
1ZT   |  Get controller #1 configuration data.
1PW1
1AC320.000000
1BA0.000000
…
1VA80.000000
1ZX3
1PW1
```

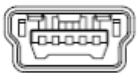
3.0 Connector interfaces

3.1 24 V Connector (Female Ø 2.1 x Ø 5.5 x 11 mm)

Pin #	Description
Center	+24 VDC
Outer	GND

3.2 Mini-USB (Male) Connector Pinout

1 2 3 4 5



USB
Mating connector:
Plug Mini-USB B 5 cts

PIN	DESCRIPTION
1	+5VdcIN Do not connect if comm connector is used
2	DATA-
3	DATA+
4	NC
5	GND

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®

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

mks

Newport® Ophir® Spectra-Physics®