CONEX-AGAP

Agilis-D Controller
with Strain Gages Feedback

Newport
Experience | Solutions

Controller
Documentation

Firmware V1.0.x

Table of Contents

CONEX-AGAP
Agilis-D Controller with Strain Gages Feedback

1.0   System Overview

1.1   General Description

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

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

1.2   Part Numbers

1.2.1   CONEX-AGAP

Product	Description
CONEX-AG-M100-D	CONEX-AGAP controller with mirror mount.

1.2.2   Accessories

CONEX-USB	USB cable, 1.8 m length
CONEX-BP	Base plate to attach up to 6 CONEX controllers

1.3   CONEX-AGAP

1.3.1   Delivered Items

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

1.3.2   Specifications

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

1.3.3   Dimensions
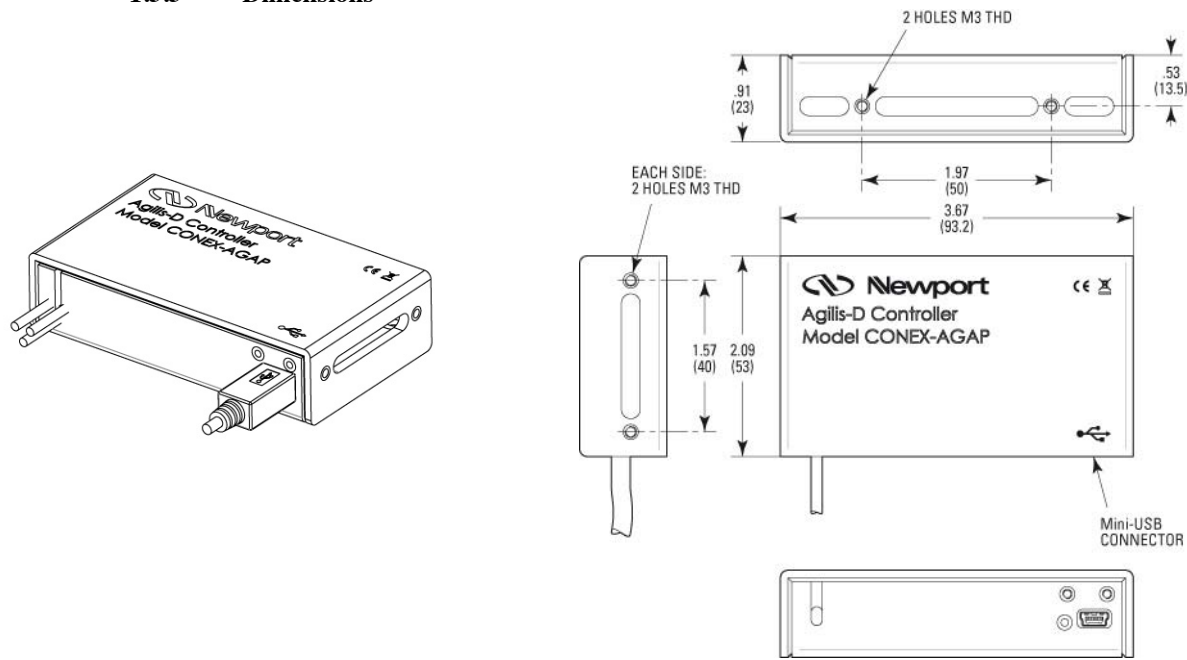
1.4   System Environmental Specifications

Operating temperature	15 °C to 35 °C
Operating humidity	20% to 85% relative humidity, non-condensing
Location	Indoor use only

1.5   Connector Identification
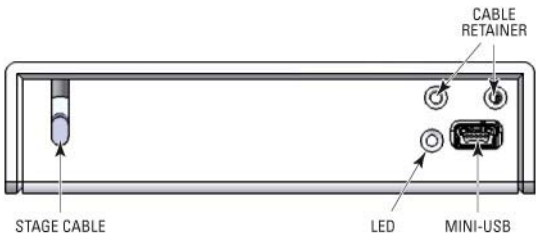
USB	mini USB connector
LED	Status LED
STAGE	Stage entry cable
Cable retainer	2 x M3 threaded hole to attach cable retainer

1.6   USB Communication Settings

Communication parameters are preset in the CONEX-AGAP 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	$C_R L_F$

2.0 Programming

2.1 State Diagram

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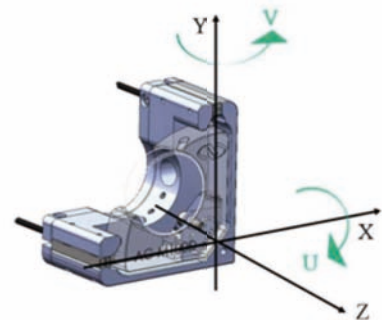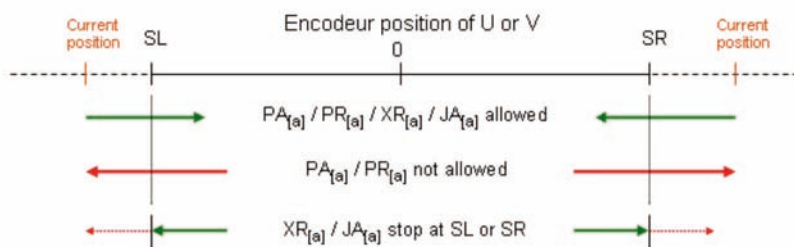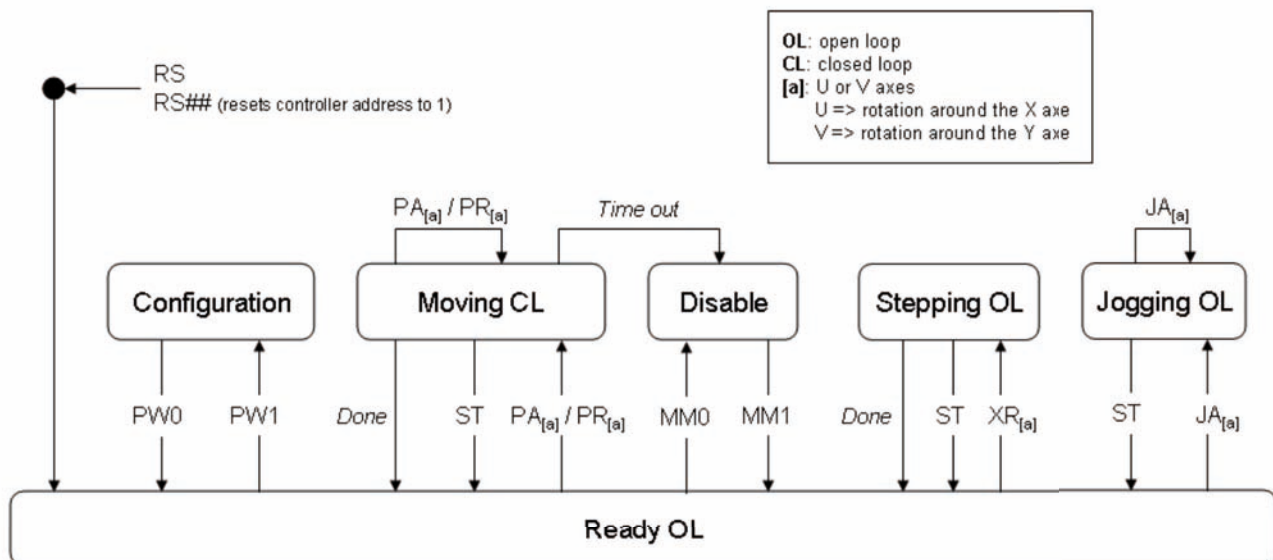
LED display:

CONFIGURATION: SLOW BLINKING RED.

READY OL: SOLID GREEN.

DISABLE: SLOW BLINKING GREEN.

MOVING CL: FAST BLINKING GREEN.

STEPPING OL: FAST BLINKING GREEN.

JOGGING OL: FAST BLINKING GREEN.

When powering the CONEX-AGAP, the controller starts initialization. When the initialization is successful, the controller goes to the READY OL state. The controller can go to the CONFIGURATION state using the PW1 command. In the CONFIGURATION state, the CONEX-AGAP allows changes to all configuration parameters, like travel limits or controller address. The PW0 command saves all changes to the controller's memory and returns the controller back to the DISABLE states.

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

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

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

2.2   Command Syntax

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

Command format:

nn	**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-AGAP

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

2.3   Command Execution Time

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

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

2.4   Command Set

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

Command format:

nn	**AA**	a	xx

nn — Optional or required controller address.

AA — Command name.

a — Optional axis reference (U or V)

xx — Optional or required value or "**?**" to query current value.

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

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

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

○   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.

-       Not accepted command (will return an error).

Command: Command passed without preceding controller number applies to all controllers (e.g. ST stops all controllers).

DB[a] — Set/Get corrector deadband

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

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

Parameters

Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference.
	nn [int]	—	Deadband value.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	**0** to **0.005**
Units	**xx**	—	None.
	nn	—	Deg.
Defaults	**xx** Missing:		Error B.
	Out of range:		Error B.
	Floating point:		Error A.

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

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	D	—	Execution not allowed.
	V	—	Unknown axe reference.
Rel. Commands	**DD[a]**	—	Set/Get deadband settling time.
Example	1DB0.00075		*Set controller #1 deadband to 0.75 mdeg.*

DD[a] — Set/Get deadband settling time

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

Syntax	**xxDD[a]nn or xxDD[a]?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference.
	nn [int]	—	Timer value.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	**0** to $\mathbf{10^4}$
Units	**xx**	—	None.
Defaults	**xx** Missing:		Error B.
	Out of range:		Error B.
	Floating point:		Error A.
Description	This command sets the deadband settling time. It corresponds to the number of corrector cycle after which, when the current position is below the deadband value from the target position, the system considers it has reached the target position.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	D	—	Execution not allowed.
	V	—	Unknown axe reference.
Rel. Commands	**DB[a]**	—	Set/Get corrector deadband.
Example	1DD10		*Set controller #1 timer to 10 control loop period.*

ID — Set/Get stage identifier

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

Syntax	**xxIDnn** or **xxID?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	nn [char]	—	Stage model number.
Range	**xx**	—	**1** to **31**
	nn	—	**1** to **31** ASCII characters.
Units	**xx**	—	None
	nn	—	None
Defaults	**xx** Missing:		Error B.
	Out of range:		Error B.
	Floating point:		Error A.
	nn Missing:		Error C.
	Out of range:		Error C.
Description	The ID? command returns the product name. In CONFIGURATION mode, this command allows changing the controller identifier.		
Returns	If the sign "**?**" takes place of **nn**, this command returns the current programmed value.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.
Rel. Commands	**ZT**	—	Get configuration parameters.
Example	1ID? I		*Get stage identifier for controller #1.*

1ID CONEX-AGAP I Controller returns product name: CONEX-AGAP.

$JA_{[a]}$ — Jog motion

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

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

Parameters

Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference.
	nn [float]	—	Percentage of full speed.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	-**100** to **100**
Units	**xx**	—	None.
	nn	—	% Full speed.
Defaults	**xx**	Missing:	Error B.
		Out of range:	Error B.
		Floating point:	Error A.

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

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

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	I	—	Execution not allowed in CONFIGURATION state.
	J	—	Execution not allowed in DISABLED state.
	M	—	Execution not allowed in Motion states.
	V	—	Unknown axe reference.
Rel. Commands	**TP[a]**	—	Get current position
	ST	—	Stop motion
Example	1JAU50.35		*Set controller #1 speed at 50.35% of full speed on axe U.*

NOTE

Not to scale.

Newport.
Experience | Solutions
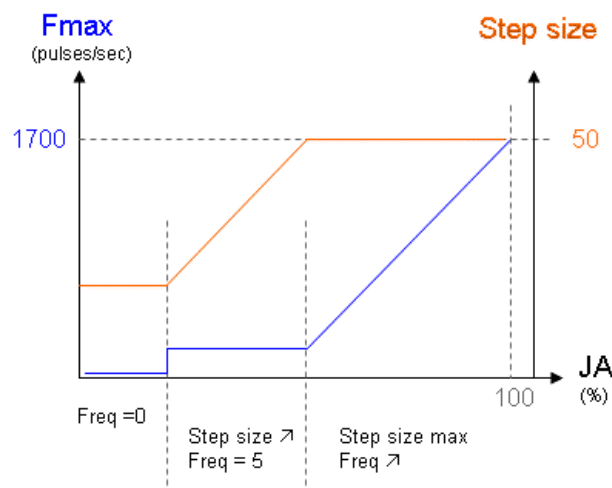
KI[a] — Set/Get integral gain

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

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

Parameters

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

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

Units			
	xx	—	None.
	nn	—	None.

Defaults			
	xx	Missing:	Change to 0.
		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.
	M	—	Execution not allowed in Motion states.
	V	—	Unknown axe reference.

Rel. Commands			
	KP	—	Set/Get proportional gain.
	LF	—	Set/Get low pass filter frequency.

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

KP[a] — Set/Get proportional gain

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

Syntax	**xxKP[a]nn** or **xxKP[a]?**	
Parameters		
Description	**xx** [int] —	Controller address.
	a [char] —	Axe reference.
	nn [float] —	Proportional gain.
Range	**xx** —	**1** to **31**
	a —	**U** or **V**
	nn —	**> 0**
Units	**xx** —	None.
	nn —	Preset units.
Defaults	**xx** Missing:	Error B.
	Out of range:	Error B.
	Floating point:	Error A.
	nn Missing:	Error C.
	Out of range:	Error C.

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

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

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

Errors	A —	Unknown message code or floating point controller address.
	B —	Controller address not correct.
	C —	Parameter missing or out of range.
	D —	Execution not allowed.
	M —	Execution not allowed in Motion states.
	V —	Unknown axe reference.
Rel. Commands	**KI** —	Set/Get integral gain.
	LF —	Set/Get low pass filter frequency.
Example	1KPU5	*Set the controller #1U axe proportional gain to 5*
	1KPU?	
	1KPU5	

Newport
Experience | Solutions

KY[a] — Set/Get calibration coefficients

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

Syntax	**xxKY[a]nn** or **xx KY?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Coefficient reference.
	nn [float]	—	Calibration value.
Range	**xx**	—	**1** to **31**
	a [char]	—	**F** or **T** or **C**
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 KY command is used to set the calibration coefficients. Those are factory set values. Users should not modify those parameters.		
Returns	If the sign "**?**" takes place of **nn**, this command returns the calibration values.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.

KZ[a] — Set/Get calibration coefficients

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

Syntax	**xxKZ[a]nn** or **xxKZ?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Coefficient reference.
	nn [float]	—	Calibration value.
Range	**xx**	—	**1** to **31**
	a [char]	—	**F** or **T** or **C**.
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 KZ command is used to set the calibration coefficients. Those are factory set values. Users should not modify those parameters.		
Returns	If the sign "**?**" takes place of **nn**, this command returns the calibration values.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.
	J	—	Execution not allowed in DISABLE state.
	M	—	Execution not allowed in Motion states.

LF — Set/Get low pass filter frequency

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

Syntax	**xxLFnn** or **LF?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	nn [float]	—	Frequency.
Range	**xx**	—	**1** to **31**
	nn	—	>0
Units	xx	—	None.
	nn	—	Hertz.
Defaults	**xx** Missing:		Error B.
	Out of range:		Error B.
	Floating point:		Error A.
Description	The LF command sets or gets the digital low pass filter frequency.		
Returns	If the sign "**?**" takes place of **nn**, this command returns the current programmed value.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	D	—	Execution not allowed.
Rel. Commands	**KP[a]**	—	Set/Get proportional gain.
	KI[a]	—	Set/Get integral gain.
Example	1LF5		Set the *controller #1 low pass filter frequency to 5Hz*.

MM — Enter/Leave DISABLE state

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

Syntax	**xxMMnn** or **xxMM?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	nn [int]	—	State change direction.
Range	**xx**	—	**0** to **31**
	nn	—	**0** changes state from READY to DISABLE.
			1 changes state from DISABLE to READY.
Units	**xx**	—	None.
	nn	—	None.
Defaults	**xx** Missing:	Change to 0.	
	Out of range:	Error B.	
	Floating point:	Error A.	
	nn Missing:	Error C.	
	Out of range:	Error C.	

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

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

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

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

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.
	I	—	Execution not allowed in CONFIGURATION state.
	M	—	Execution not allowed in Motion states.
Rel. Commands	PW	—	Enter/leave CONFIGURATION state.
Example	1MM1		*The controller #1 goes to READY state.*
	1MM?		
		1MM32	

PA[a] — Move absolute

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

Syntax	**xxPA[a]nn** or **xxPA[a]?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference
	nn [float]	—	New target position.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	**> SL** and **< SR**
Units	**xx**	—	None.
	a	—	**U** or **V**
	nn	—	Preset units.
Defaults	**xx** Missing:	Error B.	
	Out of range:	Error B.	
	Floating point:	Error A.	
	nn Missing:	Error C.	
	Out of range:	Error C.	

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

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.
	G	—	Target position out of limits.
	I	—	Execution not allowed in CONFIGURATION state.
	J	—	Execution not allowed in DISABLE state.
	V	—	Unknown axe reference.
Rel. Commands	**PR**	—	Move relative.
	TH	—	Get target position.
	TP	—	Get current position.
Example	1PAV0.2		*Move positioner on controller #1 to absolute position 0.2 units.*

$PR_{[a]}$ — Move relative

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

Syntax	**xxPR[a]nn**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference
	nn [float]	—	Displacement.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	**> SL** and **< SR**
Units	**xx**	—	None.
	nn	—	Preset units.
Defaults	**xx** Missing:		Error B.
	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 to a new target position **nn** units away from the current target position.
	The PR command gets only accepted in READY or MOVING state, AND when the distance of the positioner to the end of runs is larger than the commanded displacement.
Returns	If the sign "**?**" takes place of **nn**, this command returns the target position value.

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.
	G	—	Displacement out of limits.
	I	—	Execution not allowed in CONFIGURATION state.
	J	—	Execution not allowed in DISABLE state.
	V	—	Unknown axe reference.
Rel. Commands	PA	—	Move absolute.
	TH	—	Get target position.
	TP	—	Get current position.
Example	1PRU0.2		*Move positioner on controller #1 to a new position 0.2 units away from the current target position.*

PW — Enter/Leave CONFIGURATION state

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

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

Parameters

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

　　　　　　　nn [float]   —   Mode.

Range   **xx**   —   **1** to **31**

　　　　nn   —   **1**: Go from READY state to CONFIGURATION state.

　　　　　　　　　　0: Go from CONFIGURATION state to READY state.

Units   **xx**   —   None.

　　　　nn   —   None.

Defaults   **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 READY to CONFIGURATION. In Configuration state all parameter settings are saved in the controller's memory and remain available after switching off the controller.

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

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

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

Returns   If the sign "**?**" takes place of **nn**, this command returns the current state.

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

　　　　B   —   Controller address not correct.

　　　　C   —   Parameter missing or out of range.

　　　　D   —   Execution not allowed.

　　　　J   —   Execution not allowed in DISABLE state.

　　　　M   —   Execution not allowed in Motion states.

Rel. Commands   **MM**   —   Enter/Leave DISABLE state.

Example   1PW1 | *Changes controller #1 to CONFIGURATION state.*

RS — Reset controller

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

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

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

Parameters

Description   xx [int]   —   Controller address.

Range   xx   —   **1** to **31**

Units   xx   —   None.

Defaults   xx   Missing: Change to 0.

   Out of range: Error B.

Floating point: Error A.

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

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

Returns

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

B   —   Controller address not correct.

D   —   Execution not allowed.

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

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

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

Syntax	**xxSAnn** or **xxSA?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	nn [int]	—	Controller new address.
Range	**xx**	—	**1** to **31**
	nn	—	**1** to **31** and **≠ xx**
Units	**xx**	—	None.
	nn	—	None.
Defaults	**xx** Missing:	Error B.	
	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 is of practical use only when not using this software.

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

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

Example	1SA3		*Set controller's RS-485 address to 3.*
	3SA?		*Get the controller address*
		3SA3	

SL[a] — Set/Get negative software limit

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

Syntax	**xxSL[a]nn** or **xxSL[a]?**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference.
	nn [float]	—	Negative software limit.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	**≥ -1** and **≤ 0**
Units	**xx**	—	None.
	nn	—	Deg.
Defaults	**xx** Missing:		Error B.
	Out of range:		Error B.
	Floating point:		Error A.
	nn Missing:		Error C.
	Out of range:		Error C.

Description	In CONFIGURATION state, this command sets the negative software limit which can than be saved in the controller's nonvolatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE or READY state.
	In DISABLE or READY state, this command allows setting a new working parameter for the negative software limit. It must be lower or equal to the target position. This value is not saved in the controller's memory and will be lost after reboot.
	The software limits are useful to limit the travel range of a positioner. There is no possibility to disable software limits.
Returns	If the sign "**?**" takes place of **nn**, this command returns the current programmed value.

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.
	M	—	Execution not allowed in Motion states.
	V	—	Unknown axe reference.
Rel. Commands	**SR**	—	Set positive software limit.
Example	1SLV-0.5		*Set controller #1 negative software limit to –0.5 units for axe V.*

SR[a] — Set/Get positive software limit

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

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

Parameters

Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference.
	nn [float]	—	Positive software limit.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	**≥ 0** and **≤ 1**
Units	**xx**	—	None.
	nn	—	Deg.
Defaults	**xx** Missing:	Error B.
	Out of range:	Error B.
	Floating point:	Error A.
	nn Missing:	Error C.
	Out of range:	Error C.

Description In CONFIGURATION state, this command sets the positive software limit which can 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 target position. This value is not saved in the controller's memory and will be lost after reboot.

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

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

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	C	—	Parameter missing or out of range.
	D	—	Execution not allowed.
	M	—	Execution not allowed in Motion states.
	V	—	Unknown axe reference.
Rel. Commands	SL	—	Set negative software limit.
Example	1SRU0.75		*Set controller #1 positive software positive to 0.75 units for axe U.*

ST — Stop motion

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

Syntax	[xx]ST		
Parameters			
Description	**xx** [int]	—	Controller address.
Range	xx	—	**0** to **31**
Units	xx	—	None.
Defaults	xx	Missing:	Change to 0.
		Out of range:	Error B.
		Floating point:	Error A.

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

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

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

Example	ST		*Stop moves on all controllers*.

SU — Set/Get system resolution

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	○	□	□	●	●	●

Syntax   **xxSUnn** or **SU?**

Parameters

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

  nn [float]   —   Resolution.

Range   xx   —   **1** to **31**

  nn   —   >0

Units   xx   —   None.

  nn   —   Deg.

Defaults   **xx**   Missing: Error B.

   Out of range: Error B.

   Floating point: Error A.

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

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

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

  B   —   Controller address not correct.

  D   —   Execution not allowed.

Rel. Commands   **DB**   —   Set/Get corrector deadband.

Example   1SU0.0005 | Set the *controller #1 resolution to 0.5mdeg*.

TB — Get command error string

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

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

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.
G	—	Displacement out of limits.
I	—	Command not allowed in CONFIGURATION state.
J	—	Command not allowed in DISABLE state.
K	—	Command not allowed in READY state.
M	—	Command not allowed in motion states.
N	—	Current position out of software limit.
S	—	Communication Time Out.
U	—	Error during EEPROM access.
V	—	Unknown axe reference.

TH_[a] — Get target position

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

Syntax	**xxTH[a]** or **xxTH**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference.
Range	xx	—	**1** to **31**
	a	—	**U** or **V**
Units	xx	—	None.
Defaults	xx Missing:		Error B.
	Out of range:		Error B.
	Floating point:		Error A.
Description	The TH command returns the value of the unrounded target position. This is the position where the positioner should be. The target position rounded to the device resolution is given by the commands PR[a]? and PA[a]?.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	D	—	Execution not allowed.
Rel. Commands	TP	—	Get current position.
Example	1THU ⏐		*Get target position of axe U of controller #1.*
	1THU0.0023512⏐		*Controller returns: target position for axe U = 0.0023512 units.*

TP_[a] — Get current position

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

Syntax   **xxTP[a]** or **xxTP**

Parameters

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

   a [char]   —   Axe reference.

Range   xx   —   **1** to **31**

   a   —   **U** or **V**

Units   xx   —   None.

Defaults   xx   Missing: Error B.

   Out of range: Error B.

   Floating point: Error A.

Description   The 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 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.

Rel. Commands   **TH**   —   Get target position.

Example   1TPU | *Get current position of axe U of controller #1.*

   1TPU0 | *Controller returns: actual position for axe U = 0 units.*

Newport.
Experience | Solutions

TS — Get positioner error and controller state

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

Syntax	**xxTS**	
Parameters		
Description	**xx** [int] —	Controller address.
Range	**xx** —	**1** to **31**
Units	**xx** —	None.
	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. The motion time out flag is always set with one of the two-associated following error.

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

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

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	• Not used	• Not used	• Not used	• Not used	• Motion Time out	• Not used	• Not used	• Not used	• Not used	• Not used

Examples:

- Error map 0000 = No errors
- Error map 0020 = Motion time out.

Controller states (ef):

- **14**: CONFIGURATION.
- **28**: MOVING CL.
- **29**: STEPPING OL.
- **32**: READY from Reset.
- **33**: READY from MOVING CL.
- **34**: READY from DISABLE.
- **35**: READY from JOGGING OL
- **36**: READY from STEPPING OL.
- **3C**: DISABLE from READY OL.
- **3D**: DISABLE from MOVING CL.
- **46**: JOGGING OL.

NOTES

THE ERROR BUFFER GETS UPDATED PERIODICALLY, APPROX. EVERY 1 MS.

THE TS COMMAND READS THE ERROR BUFFER AND CLEARS THE ERROR BUFFER AT THE SAME TIME (SAME AS FOR COMMANDS TE, TB). SO WHEN LAUNCHING THE TS COMMAND, IT IS IMPORTANT TO PROCESS THE TS FEEDBACK ACCORDINGLY

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.*
	1TS000032	∣	*Controller returns: no errors and READY from reset.*

Newport
Experience | Solutions

VE — Get controller revision information

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

Syntax	**xxVE**		
Parameters			
Description	**xx** [int]	—	Controller address.
	nn [string]	—	Action.
Range	xx	—	**1** to **31**
Units	xx	—	None.
Defaults	xx	Missing:	Error B.
		Out of range:	Error B.
		Floating point:	Error A.
Description	This command returns the controller's revision information.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
Rel. Commands	TP	—	Get current position.
Example	1VE		*Get controller #1 revision information.*

1VE CONEX-AGAP V1.0.0. | *Controller returns revision number*

$XR_{[a]}$ — Step motion

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

Syntax	**xxXR[a]nn**		
Parameters			
Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference.
	nn [int]	—	Number of steps.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	$\geq \mathbf{-10^6}$ and $\leq \mathbf{10^6}$
Units	**xx**	—	None.
Defaults	**xx** Missing:		Error B.
	Out of range:		Error B.
	Floating point:		Error A.
Description	Starts a relative move of nn steps with step amplitude defined by the XU command. At the end of the motion, the target position of the axe takes the value of the current axe position.		
Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	I	—	Execution not allowed in CONFIGURATION state.
	J	—	Execution not allowed in DISABLED state.
	M	—	Execution not allowed in Motion states.
	V	—	Unknown axe reference.
Rel. Commands	**TP**	—	Get current position.
	XU[a]	—	Set/Get step motion size.
Example	1XRU100		*Set controller #1 step number on axe U.*

XU_[a] — Set/Get step motion size

Usage	Config.	Disable	Ready OL	Moving CL	Stepping OL	Jogging OL
	□	□	□	–	●	–

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

Parameters

Description	**xx** [int]	—	Controller address.
	a [char]	—	Axe reference
	nn [int]	—	Step size.
Range	**xx**	—	**1** to **31**
	a	—	**U** or **V**
	nn	—	**-50** to **+50**
Units	**xx**	—	None.

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

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

NOTES

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

Errors	A	—	Unknown message code or floating point controller address.
	B	—	Controller address not correct.
	M	—	Execution not allowed in Motion states.
	V	—	Unknown axe reference.
Rel. Commands	**TP**	—	Get current position.
	XR[a]	—	Step motion.
Example	1XUU20		*Set controller #1 step size to 20 on axe U.*

ZT — Get all configuration parameters

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

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.

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

Rel. Commands   TE   —   Get error code.

Example	1ZT		*Get controller #1 configuration data.*
	1PW1		
	1IDAG-M100D		
	1SA1		
	1SLU-1		
	1SRU1		
	…		
	1PW0		

3.0 Connector interfaces

3.1 USB (Male mini-USB)

```
     1 2 3 4 5
```

USB
Mating connector:
Plug Mini-USB B 5 cts

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

Service Form

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): _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____
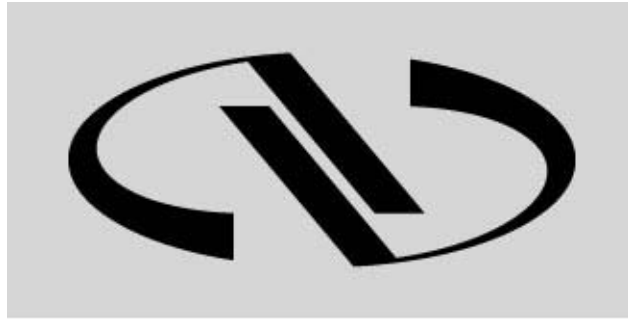
Newport®

Experience | Solutions

North America & Asia
Newport Corporation
1791 Deere Ave.
Irvine, CA 92606, USA

Sales
Tel.: (800) 222-6440
e-mail: sales@newport.com

Technical Support
Tel.: (800) 222-6440
e-mail: tech@newport.com

Service, RMAs & Returns
Tel.: (800) 222-6440
e-mail: rma.service@newport.com

Europe
MICRO-CONTROLE Spectra-Physics S.A.S
1, rue Jules Guesde – Bât. B
ZI Bois de l'Épine – BP189
91006 Evry Cedex
France

Sales
Tel.: +33 (0)1.60.91.68.68
e-mail: france@newport-fr.com

Technical Support
e-mail: tech_europe@newport.com

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