# Super Agilis Series

## CONEX-SAG Controller with SAG-xxxx Stages


Newport®

User's Manual

V1.0.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.



Original instructions.

# Table of Contents

# EU Declaration of Conformity

## EU Declaration of Conformity

Application of Council Directive(s): ☒ 2014/30/EU Electromagnetic Compatibility Directive (EMCD)
☐ 2014/35/EU Low Voltage Directive (LVD)
☐ 2014/68/EU European Pressure Equipment Directive (PED)
☒ 2011/65/EU Restriction of Hazardous Substances Directive (RoHS)

Standard(s) to which conformity is declared: ☒EN 61326-1:2013 (EMC); ☐EN 61326-2-3:2013 (EMC);
☐EN 61010-1:2010 (Safety);
☐PED Module SEP (Sound Engineering Practices); 2014

**Emissions:**
☒CISPR 11:2015 Industrial, Scientific and Medical Equipment Radio-Frequency Disturbance Characteristics - Limits and Methods of Measurement
☐IEC 61000-3-2:2014 EMC/Limits for Harmonic Current Emission (1)
☐IEC 61000-3-3:2013 EMC/Limitations of Voltage Fluctuations and Flicker in Low-Voltage Supply Systems (2)
☐IEC 61000-3-3:2013 EMC/Limitations of Voltage Fluctuations and Flicker in Low-Voltage Supply Systems (3)
☒EN 55011: 2009+A1:2010 (4)
☐FCC 47 CFR Part 18: 2015

**Immunity:**
☒IEC 61000-4-2:2008 EMC/Electrostatic Discharge Immunity Test
☒IEC 61000-4-3:2006 2006+AMD1:2007+AMD2:2010 EMC/Radiated Radio - Frequency Electromagnetic Field Immunity Test
☒IEC 61000-4-4:2012 EMC/Electrical Fast Transient/Burst Immunity Test
☐IEC 61000-4-5:2014 EMC/Surge Immunity Test (3)
☒IEC 61000-4-6:2013 EMC/Conducted Disturbances induced by Radio Frequency Fields Immunity Test
☐IEC 61000-4-8:2009 EMC/Power Frequency Magnetic Field Immunity Test
☐IEC 61000-4-11:2004 EMC/Voltage Dips, Short Interruptions and Variations Immunity Test (5)
☐IEC 61000-4-34:2005+AMD1:2009 EMC/Voltage Dips, Short Interruptions and Variations Immunity Test (5)

**Manufacturers Name: MICRO-CONTROLE Spectra-Physics. EVRY, France.**

Importer's Name & Location:

Equipment Type/Description: **CONTROLLER CONEX SUPER AGILIS**

Model Number(s) (6): CONEX-SAG-LS16, CONEX-SAG-LS32, CONEX-SAG-LS48
CONEX-SAG-LS16P, CONEX-SAG-LS32P, CONEX-SAG-LS48P

MKS confirms that, with respect to the products listed above, it believes it is in conformity with the selected European Union harmonization legislation. MKS product conforms to the above Directive(s) and Standard(s) only when installed in accordance with manufacturer's specifications. This declaration has been issued under the sole responsibility of the manufacturer.

Date: 16/04/2019

**Le Cointe Hervé**
**Quality Director**

1) Applicable to AC powered product only. Class B
2) Applicable to AC powered product; DC powered connections must not connect to a D.C. distribution network; I/O Signal and Control Lines must be less than 30m and not exit the building.
3) Applicable to AC powered product; DC powered connections and may connect to a D.C. distribution network.
4) Class A, Group 2
5) Applicable to AC powered product only.
6) Compliance of the above model numbers requires the use of a braided shielded cable properly terminated at both ends – if so noted in the MKS Instruction Manual.

MKS Instruments, Inc.
Andover, MA USA

MKS CONFIDENTIAL

Document Number: MKS-CR-1197
Revision: B

Preface

## Confidentiality & Proprietary Rights

**Reservation of Title**

The Newport Programs and all materials furnished or produced in connection with them ("Related Materials") contain trade secrets of Newport and are for use only in the manner expressly permitted. Newport claims and reserves all rights and benefits afforded under law in the Programs provided by Newport Corporation.

Newport shall retain full ownership of Intellectual Property Rights in and to all development, process, align or assembly technologies developed and other derivative work that may be developed by Newport. Customer shall not challenge, or cause any third party to challenge, the rights of Newport.

**Preservation of Secrecy and Confidentiality and Restrictions to Access**

Customer shall protect the Newport Programs and Related Materials as trade secrets of Newport, and shall devote its best efforts to ensure that all its personnel protect the Newport Programs as trade secrets of Newport Corporation. Customer shall not at any time disclose Newport's trade secrets to any other person, firm, organization, or employee that does not need (consistent with Customer's right of use hereunder) to obtain access to the Newport Programs and Related Materials. These restrictions shall not apply to information (1) generally known to the public or obtainable from public sources; (2) readily apparent from the keyboard operations, visual display, or output reports of the Programs; (3) previously in the possession of Customer or subsequently developed or acquired without reliance on the Newport Programs; or (4) approved by Newport for release without restriction.

## Sales, Tech Support & Service

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

**Sales Germany**
Tel.: +49 (0) 61 51 /708 – 0
e-mail: germany@newport.com

**Sales UK**
Tel.: +44 (0)1635.521757
e-mail: uk@newport.com

**Technical Support**
e-mail: tech_europe@newport.com

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

## Service Information

The user should not attempt any maintenance or service of the Super Agilis Controller/Driver and its accessories beyond the procedures outlined in this manual. Any problem that cannot be resolved should be referred to Newport Corporation. When calling Newport regarding a problem, please provide the Tech Support representative with the following information:

- Your contact information.
- System serial number or original order number.
- Description of problem.
- Environment in which the system is used.
- State of the system before the problem.
- Frequency and repeatability of problem.
- Can the product continue to operate with this problem?
- Can you identify anything that may have caused the problem?

## Newport Corporation RMA Procedures

Any Super Agilis Controller being returned to Newport must have been assigned an RMA number by Newport. Assignment of the RMA requires the item serial number.

## Packaging

The Super Agilis Controller being returned under an RMA must be securely packaged for shipment. If possible, reuse the original factory packaging.


# CONEX-SAG Controller with SAG-xxxx Stages Super Agilis Series

## 1.0 Introduction

### 1.1 Definitions and Symbols

The following terms and symbols are used in this documentation and also appear on the Super Agilis Controller/Driver where safety-related issues occur.

#### 1.1.1 General Warning or Caution

*Figure 1: General Warning or Caution Symbol.*

The Exclamation Symbol in Figure 1 may appear in Warning and Caution tables in this document. This symbol designates an area where personal injury or damage to the equipment is possible.

#### 1.1.2 Electric Shock

*Figure 2: Electrical Shock Symbol.*

The Electrical Shock Symbol in Figure 2 may appear on labels affixed to the Super Agilis Controller/Driver. This symbol indicates a hazard arising from dangerous voltage. Any mishandling could result in irreparable damage to the equipment, in personal injury, or death.

#### 1.1.3 European Union CE Mark

*Figure 3: CE Mark.*

The presence of the CE Mark on Newport Corporation equipment means that it has been designed, tested and certified as complying with all applicable European Union (CE) regulations and recommendations.

## 1.2 Warnings and Cautions

The following are definitions of the Warnings, Cautions and Notes that may be used in this manual to call attention to important information regarding personal safety, safety and preservation of the equipment, or important tips.

**WARNING**

**Situation has the potential to cause bodily harm or death.**

**CAUTION**

**Situation has the potential to cause damage to property or equipment.**

**NOTE**

**Additional information the user or operator should consider.**

## 1.3 General Warnings and Cautions

The following general safety precautions must be observed during all phases of operation of this equipment.

Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment.

- Heed all warnings on the unit and in the operating instructions.
- To prevent damage to the equipment, read the instructions in this manual.
- Only plug the power supply to a grounded power outlet.
- Assure that the power supply is properly grounded to earth ground through the grounding lead of the AC power connector
- Route power cords and cables where they are not likely to be damaged.
- Disconnect or do not plug in the AC power cord in the following circumstances:
  - If the AC power cord or any other attached cables are frayed or damaged.
  - If the power plug or receptacle is damaged.
  - If the unit is exposed to rain or excessive moisture, or liquids are spilled on it.
  - If the unit has been dropped or the case is damaged.
  - If the user suspects service or repair is required.
- Keep air vents free of dirt and dust.
- Keep liquids away from unit.
- Do not expose equipment to excessive moisture (>85% humidity)
- Do not operate this equipment in an explosive atmosphere.
- Disconnect power before cleaning the Controller/Driver unit. Do not use liquid or aerosol cleaners.
- Do not open the CONEX-SAG controller. There are no user-serviceable parts inside.
- Return equipment to Newport Corporation for service and repair.
- Dangerous voltages associated with the 100-240 VAC power supply are present inside the power supply. To avoid injury, do not touch exposed connections or components while power is on.
- Follow precautions for static-sensitive devices when handling electronic circuits.

# 2.0 System Overview

## 2.1 General Description

The CONEX-SAG is a single axis motion controller/driver for piezo actuator with encoder 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-SAG 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.

### 2.1.1 Encoder Theory of Operation

The Agilis stages use a Sine/Cosine encoder which is interpolated to give high resolution position information with low noise. It also presents high positioning repeatability.

### 2.1.2 Closed Loop technology

The Conex closed loop algorithm for Agilis stages is a simplified version of the typical DC servo loop algorithm. Although a motion profile is not generated in the algorithm, the Conex controller still closes a loop based on the error.

The closed loop motions are divided into 3 phases: jogging, shifting and scanning.

#### 2.1.2.1 Closed Loop Jogging phase

When a new target is requested or when the position error is larger than the output deadband (DB and DDS commands), the motion enter in jogging phase to move the stage to a shifted target position (SSD command).

- For firmware version inferior to v.0.17.0:

The larger the error, the faster the stage is commanded to move to reduce the error. The stage could start at the full system speed of 10KHz if the stage is far from the desired position.

- For firmware version superior or equal to v.0.17.0:

The stage follow a trapezoid motion profile according to VA (velocity) and AC (acceleration) commands

The Jogging phase ends when the stage is within a specified range about shifted target position. This range is called the shift deadband and is set by SSP and SSN commands.

If the stage is inside the shift deadband within SST*10 ms, the closed-loop motion enters in shifting phase.

#### 2.1.2.2 Closed Loop Shifting phase

In this phase the stages move slowly to the target position by increasing the piezzo voltage untils the stage.

The Shifting phase ends when the stage is within a specified range about desired target position. This range is called the deadband and is set by DBP and DBN commands.

If the stage is inside the deadband within DDT*10 ms, the closed-loop motion enters in scanning phase and the controller state is set to ready-closed-loop.

#### 2.1.2.3 Closed Loop Scanning phase

In this phase the motion is considered as ended but the position error is continuously corrected by adjusting the piezo voltage with an integral servo coefficient (SSK).

If a position error superior to output deadband (DB and DDS commands) is detected, the motions phase change to jogging.

#### 2.1.2.4 Closed Loop Deadband

- Whith scanning mode enabled (SSK > 0)

| | JOGGING | SHIFTING | SCANNING |
|---|---|---|---|
| Entry Deadband +<br>Entry Deadband - | SSD + SSP<br>SSD + SSN | DBP<br>DBN | DBP<br>DBN |
| Exit Deadband +<br>Exit Deadband - | N/A | DBP * DDS+<br>SSD - SSN * DDS- | DBP * DDS+<br>DBN * DDS- |

- With scanning mode disabled (SSK = 0)

| | JOGGING | SHIFTING | SCANNING |
|---|---|---|---|
| Entry Deadband +<br>Entry Deadband - | DBP<br>DBN | N/A | N/A |
| Exit Deadband +<br>Exit Deadband - | DBP * DDS+<br>DBN * DDS- | N/A | N/A |

#### 2.1.2.5 Example


## 2.2 Security features

### 2.2.1 Temperature sensor

The controller has an embedded sensor to monitor the temperature.

If the temperature exceeds 85 °C, the motion is stopped and an error flag is set.

All motion requests will be refused by the controller until the temperature falls below 85 °C.

The current temperature of the controller can be read with the RT command.

### 2.2.2 Motion Timeout

If a motion last more than motion timeout (configured with MT command), the motion is stopped and an error flag is set.

All motion requests will be refused by the controller until the error flag is read with the TS command.

An update of the velocity (JA) or target position (PA/PR) during the motion will reset the timeout counter.

### 2.2.3 Hard stop detection

This secutity feature is available only for closed loop stages.

During a motion, if an obstacle is blocking the stages, the controller will detect it by monitoring the current stage velocity.

If the velocity is below the minimum velocity (configured with TOD command) during more than the motor stall timeout (configured with TOT command), the motion is stopped and an error flag is set.

All motion requests will be refused by the controller until the error flag is read with the TS command.

## 2.3 Part Numbers

| Product | Description |
|---|---|
| **CONEX-SAG** | Super Agilis Motion Controller/Driver for Agilis stages. |
| **CONEX-PS** | 40 W power supply for SA Controller. |
| **CONEX-USB** | USB cable, 1.8 m length |
| **CONEX-BP** | Base plate to attach up to 6 CONEX controllers |

## 2.4 Super Agilis CONEX-SAG Controller


### 2.4.1 Contents of Delivery

- Super Agilis Controller box.

### 2.4.2 Specifications

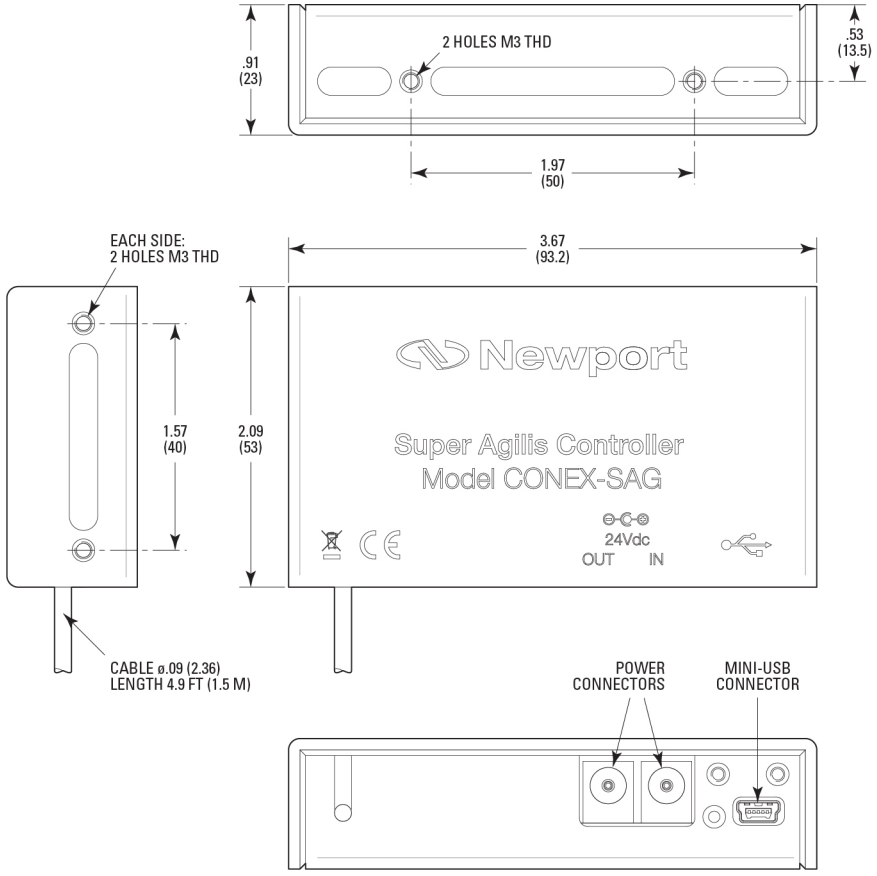
| | |
|---|---|
| General Description | Super Agilis controller |
| Control Capability | Piezo motors, closed loop |
| Piezo Output | 48Vpeak<br>10 kHz max |
| Control loop | Digital PI loop<br>100 Hz servo rate for firmware version inferior to v0.17.0<br>1 KHz servo rate for firmware version superior to v0.17.0 |
| Motion | Absolute and relative motion |
| Computer interface | USB Virtual COM port |
| Programming | 25+ intuitive, 2 or 3 letter ASCII commands |
| Communication rate | 50 Hz Max. (USB) |
| Status display | Two color LED |
| Power Supply | +5V (USB):<br>Consumption : < 0.3 Arms<br>Operating voltage : 4.75V to 5.25V<br><br>+24V (POWER) :<br>Consumption : < 1.5 Arms<br>Operating voltage : 23V to 25V<br><br>It is mandatory to keep the +5V and +24V power supplies independant one to the other to insure the voltages are kept into the recommended operating values during hot plugging phases. |

### 2.4.3 USB Communication Settings

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

| | |
|---|---|
| Bits per second | 57,600 |
| Data bits | 8 |
| Parity | None |
| Stop bits | 1 |
| Flow control | None |
| Terminator | $C_RL_F$ |

### 2.4.4 Dimensions

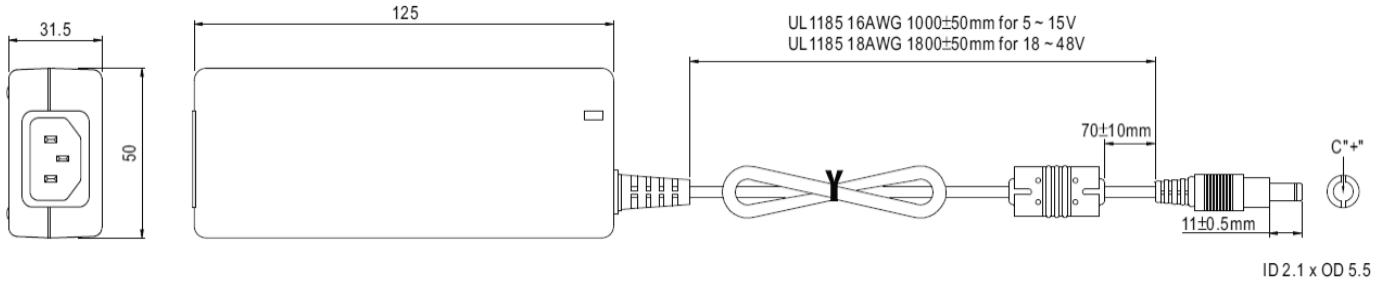



## 2.5 CONEX-PS

### 2.5.1 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 |

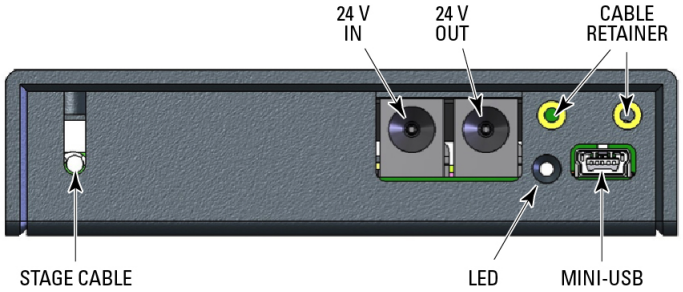
### 2.5.2 Dimensions




## 2.6 Environmental Specifications

| | |
|---|---|
| Operating temperature | 5 °C to 40 °C |
| Operating humidity | <85% relative humidity, non-condensing |
| Storage temperature | 0 °C to 60 °C<br>RH <85% relative humidity, non-condensing |
| Installation category | II |
| Pollution degree | 2 |
| Use location | Indoor use only |

## 2.7 Connector Identification




| | |
|---|---|
| USB | mini USB connector |
| LED | Status LED |
| POWER IN | 24V supply input |
| POWER OUT | 24V supply out |
| STAGE | Stage entry cable |
| Cable retainer | 2 x M3 threaded hole to attach cable retainer |

# 3.0 Getting Started

This section guides the user through the proper set-up of the Super Agilis Controller motion control system.

Carefully unpack and visually inspect the Agilis Stage, Super Agilis Controller and CONEX-PS power supply for any damage. Place all components on a flat and clean surface.

## 3.1 Connections

To power the system, follow the sequence below:

1. Connect the CONEX-USB to the Super Agilis Controller.
2. Connect the CONEX-USB to a PC.
3. Connect the CONEX-PS to the Super Agilis Controller (DC IN connector).
4. Connect the CONEX-PS to power.

The Super Agilis Controller is automatically powered on. Wait for the boot time (few seconds). Super Agilis Controller LED should turn solid orange

## 3.2 Communication Settings

### 3.2.1 Requirements

Provided Agilis GUI and Dll are designed for Personal Computers running Windows 7, 8.1 and 10 platforms (32 bit & 64 bit).

### 3.2.2 PC USB Communication

Using the USB cable (provided) connect Super Agilis Controller (USB connector) to your PC.

If not already installed Windows will ask for the compatible driver. Follow the procedure below to install it.

**NOTE**

**To install the USB driver, consult the USB Driver Installation Manual of our DL Controller on our website and proceed in the same way.**

### 3.2.3 Configuring the Controller

Consult the **Super Agilis Controller GUI Manual** (chapter 3.0) to set parameters.

#### 3.2.3.1 Modified payload

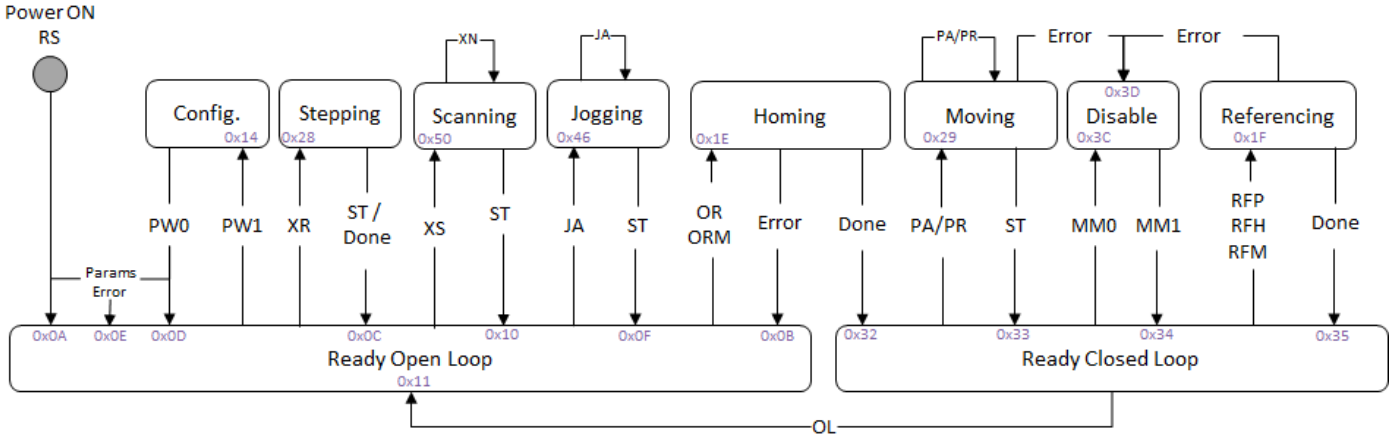
All Super Agilis Controller parameters have been set in the factory per the associated stage without any payload. In case of significant load added to the stage, 2 parameters must be adjusted depending on the payload placed on the stage:

- The KP parameter through the "KP" command.
- The KI parameter through the "KI" command.
- The KO parameter through the "KO" command.
- The DB parameter through the "DB" command.

# 4.0 Programming

## 4.1 State Diagram

For a safe and consistent operation, the Agilis Controller uses 10 different operation states: Configuration, Ready Open Loop, Stepping, Scanning, Jogging, Homing, Ready Closed Loop, Disable, Moving and Referencing. In each state, only specific commands are accepted by the Agilis Controller. Therefore, it is important to understand the state diagram below and which commands and actions cause transition between the different states. Also see chapter 5.5 for command/state information.




**LED display:**

| | RED | ORANGE | GREEN |
|---|---|---|---|
| **SOLID** | READY OPEN LOOP with FAULT or ERROR CONFIG | READY OPEN LOOP | READY CLOSED LOOP |
| **SLOW BLINK** | CONFIG | HOMING | DISABLE |
| **FAST BLINK** | - | STEPPING JOGGING SCANNING | MOVING REFERENCING |

READY OPEN LOOP: If hardware faults or wrong parameters then SOLID RED.

READY OPEN LOOP: If everithing is OK then SOLID ORANGE.

CONFIGURATION : SLOW BLINK RED.

READY CLOSED LOOP : SOLID GREEN.

DISABLE : SLOW BLINK GREEN.

HOMING : SLOW BLINK ORANGE.

MOVING : FAST BLINK GREEN.

STEPPING : FAST BLINK ORANGE.

SCANNING : FAST BLINK ORANGE.

JOGGING : FAST BLINK ORANGE.

When connecting the Super Agilis Controller to power, it starts in READY OPEN LOOP state. From this state, the Controller can go to the CONFIGURATION state with the PW1 command. In CONFIGURATION stage, the Super Agilis Controller allows changing all stage configuration parameters like proportional gain of the closed loop. The PW0 command saves all changes to the Controller's memory and returns the Controller back to the READY OPEN LOOP state.

In READY OPEN LOOP state, the controller can make open loop motion. The motions are executed with the commands (XR, JA, XS).

To execute move commands (PA, PR) in closed loop, the Controller must be in READY CLOSED LOOP state. To get from the READY OPEN LOOP state to the READY CLOSED LOOP state, the positioner must homed 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 CLOSED LOOP state. The position after homing is either the position related to the current phase angle of the encoder signal when the OR0 is used, or, if the ORMxx is used, the current position is given by xx (xx in mm). The stage adjusts its position, if necessary, by making a motion to match the phase of the encoder.

The process for referencing, and which mechanical ends of runs are looked for during referencing, can be defined with the HT command.

The referencing can either go to a mechanical limit (RFH), measure the current position by going to a limit and returning to the original position (RFP). This is usefull if the power was removed and the current position information was lost. The command RFMnn moves to a hard stop and goes to the positin indicated by the parameter nn.

To get from READY CLOSED LOOP state to READY OPEN LOOP state, for instance to make some further parameter change in CONFIGURATION state, you can either reboot the Controller with the RS command (working parameters are restore to default) or use the OL command (working parameters are maintained).

## 4.2 Command Syntax

The Super Agilis Controller 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 or AAA | xx |
|---|---|---|

**nn** — Axis number (optional).

**AA or AAA** — Command name.

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

Both, upper and lower case characters are accepted.

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, 1KP300 sets the proportional gain of the axis #1 to 300 units. 1KP? sends the response 1KP300.

Not every command can be executed in all states of the Agilis Controller and some commands have different meaning in different states. It is therefore important to understand the state diagram of the Controller, see section 4.1.

**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:

PA1.43 6

PA1.436

**Decimal separator**

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

**Command terminator**

Commands are executed as the command terminator $C_R$ or $L_F$ (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 Super Agilis Controller.

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 0 for details.

## 4.3 Command Execution Time

The Agilis Controller interprets commands continuously as received. The typical execution time for a "tell position command" (TP?) is inferior to 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, which may last for several seconds, will not suspend the Controller from further command execution. So, for an efficient process flow with many move commands it is to query the Controller status (TS command) or the current position (TP command) before any further motion command is sent.

## 4.4 Command Set

This section describes the supported two-letter ASCII commands used to configure and operate the Agilis Controller.

| | CONF | RDY OL | RDY CL | STEP. | JOG | SCAN | MOV. | REF. | DISABLE | OL/CL | Description |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AC | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Set/Get Acceleration |
| DB | ○ | □ | | | | | | | □ | CL | Set/Get deadband |
| DD | ○ | □ | | | | | | | □ | CL | Set/Get deadband entry/exit conditions |
| HT | ○ | □ | | | | | | | □ | CL | Set/Get referencing config |
| ID | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Get stage identifier |
| IF | ● | ● | ● | ● | ● | ● | ● | ● | ● | CL | Get interpolation factor |
| FS | ○ | | | | | | | | | BOTH | Restore/Set factory setting |
| JA | | ● | | | ● | | | | | BOTH | Move jogging |
| KF | ○ | □ | | | | | | | □ | CL | Set/Get feedforward gain |
| KI | ○ | □ | | | | | | | □ | CL | Set/Get integral gain |
| KP | ○ | □ | | | | | | | □ | CL | Set/Get proportional gain |
| KS | ○ | □ | | | | | | | □ | CL | Set/Get integral saturation |
| KO | ○ | □ | | | | | | | □ | CL | Set/Get offset friction |
| MM | | | ● | | | | | | ● | CL | Enter/leave DISABLE state |
| MS | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Get motion status |
| MT | ○ | □ | | | | | | | □ | BOTH | Set/Get motion timeout |
| OL | | | ● | | | | | | | CL | Go to READY OL state |
| OR | | ● | | | | | | | | CL | Go to READY CL state |
| PA | | | ● | | | | ● | | | CL | Move absolute |
| PR | | | ● | | | | ● | | | CL | Move relative |
| PW | ● | ● | | | | | | | | BOTH | Enter /Leave CONFIG state |
| RF | | | ● | | | | | | | CL | Execute REFERENCING |
| RS | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Reset Controller |
| RT | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Read controller temperature |
| SA | ○ | | | | | | | | | BOTH | Set/Get axis number |
| SL | ○ | □ | | | | | | | □ | CL | Set/Get stage left travel range |
| SR | ○ | □ | | | | | | | □ | CL | Set/Get stage right travel range |
| SS | ○ | □ | | | | | | | □ | CL | Set/Get scanning setting |
| ST | | | | ● | ● | ● | ● | ● | | BOTH | Stop motion |
| SU | ○ | | | | | | | | | CL | Set/Get encoder pitch |
| TB | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Get error string |
| TE | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Get last error |
| TH | ● | ● | ● | ● | ● | ● | ● | ● | ● | CL | Get target position |
| TO | ○ | □ | | | | | | | □ | CL | Set/Get motor stall timeout |
| TP | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Get current position |
| TS | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Get Controller status |
| VA | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Set/Get Velocity |
| VE | ● | ● | ● | ● | ● | ● | ● | ● | ● | BOTH | Get Controller version |
| XF | ○ | □ | | | | | | | | BOTH | Set/Get OL step frequency |
| XN | | | | | | ● | | | | BOTH | Set/Get scanning command |
| XR | | ● | | | | | | | | BOTH | Move stepping |
| XS | | ● | | | | | | | | BOTH | Go to SCANNING state |
| XU | ○ | □ | | | | | | | | BOTH | Set/Get OL step amplitude |
| ZT | ● | ● | | | | | | | ● | BOTH | Get all axis 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.

**Blank:** Not accepted command (will return an error).

**NOTE**

**Use ZT command to get all actual parameters value.**

**Command read is accepted in all states.**

# AC — Set/Get Acceleration

**Usage** ■ Config ■ Ready OL ■ Stepping ■ Jogging ■ Scanning
■ Ready CL ■ Referencing ■ Moving ■ Disable

**Compatibility** □ Open-Loop
■ Closed-Loop

**Syntax** **AC**nn or **AC**?

**Parameters**

**Description** **nn** [float] — Acceleration

**Range** **nn** — ≥ 1.5 and ≤ 1500

**Units** **nn** — mm/s²

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** The AC command set the acceleration used by the profiler during closed loop motion.
This command is not available for firmware version inferior to v0.17.0

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

**Errors** — Unknown message code.
— Execution not allowed.

**Rel. Commands** VA — Set/Get Velocity
PA — Move absolute.
PR — Move relative.

**Example**
```
AC500   | set acceleration to 500mm/s²
AC?     | Get acceleration.
        | Controller returns : AC500
```

# DB — Set deadband

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax** **DB**nn,pp or **DB**?

**Parameters**

**Description**
**nn** [float] — Negative deadband.
**pp** [float] — Positive deadband.

**Range**
**nn** — ≥ -0.05 and <0
**pp** — >0 and ≤+0.05

**Units**
**nn** — Preset unit.
**pp** — Preset unit.

**Defaults**
**nn** Missing: Error.
**pp** Missing: Error.
Out of range: Error.

**Description**
In CONFIGURATION state, this command sets deadband of the PI control loop which can 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 OPEN LOOP states.

In DISABLE or READY OPEN LOOP states, this command allows setting a new working parameter for deadband. 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**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands**
DD — Set deadband entry/exit conditions
KI — Set integral gain.
KP — Set proportional gain.

**Example**

| | |
|---|---|
| DB-0.000015, 0.00001 | Set negative deadband to 15e-6 units. |
| | Set positive deadband to 10e-6 units. |
| DB? | Get deaband |
| | Controller returns "DB-1.5e-5, 1e-5" |

# DD — Set deadband entry/exit conditions

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax** **DDS**nn,pp or **DDS**?
**DDT**qq or **DDT**?
**DDX**rr or **DDX**?

**Parameters**

**Description**

| | | |
|---|---|---|
| **nn** [int] | — | Deadband exit positive coefficient. |
| **pp** [int] | — | Deadband exit negative coefficient. |
| **qq** [int] | — | Deadband entry timer |
| **rr** [int] | — | Deadband exit timer |

**Range** **nn-rr** — >0 and ≤100

**Units** **nn-rr** — Preset unit.

**Defaults** **nn-rr** Missing: Error.
Out of range: Error.

**Description** In CONFIGURATION state, this command sets deadband entry/exit conditions which can 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 OPEN LOOP states.

The entry timer value defines how many successives positions have to be detected in the deadband before validating the end of motion.

The exit coeffients define how much the position error has to be superior to deadband to trig a motion

The exit timer value defines how many successives positions have to be detected outside the deadband to trig a motion (available from firmware version v0.17.0)

In DISABLE or READY OPEN LOOP states, this command allows setting a new working parameter for deadband entry/exit conditions. 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**

| | | |
|---|---|---|
| | — | Unknown message code. |
| | — | Parameter missing or out of range. |
| | — | Execution not allowed. |
| | — | Execution not allowed in XXXX state. |

**Rel. Commands**

| | | |
|---|---|---|
| DB | — | Set deadband |
| KI | — | Set integral gain. |
| KP | — | Set proportional gain. |

**Example**

| | | |
|---|---|---|
| DDT4 | \| | position has to be in deadband during 4 corrector period to validate motion |
| DDS-3, 4 | \| | position error superior to 4*deadband+ generates a move |
| | \| | position error inferior to 3*deadband- generates a move |

# HT — Set/Get referencing configuration

| | | | | | | |
|---|---|---|---|---|---|---|
| **Usage** | ■ Config | ■ Ready OL | ☐ Stepping | ☐ Jogging | ☐ Scanning |
| | | ☐ Ready CL | ☐ Referencing | ☐ Moving | ■ Disable |
| **Compatibility** | ☐ Open-Loop | | | | |
| | ■ Closed-Loop | | | | |

**Syntax** **HT**nn, or **HT**?

**Parameters**

**Description** **nn** [int] — Home type value.

**Range** **nn** — 3 use positive mechanical end of run

4 use negative mechanical end of run

**Units** **nn** — None.

**Defaults** **nn** Missing: Error.

Out of range: Error.

**Description** This command sets the HOME type that will be used when RF command will be executed.

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

**Errors** — Unknown message code.

— Parameter missing or out of range.

— Execution not allowed.

— Execution not allowed in XXXX state.

**Rel. Commands** RF — Execute referencing.

**Example** HT3 | Use EoR- switch for HOME.

# ID —Get stage identifier

**Usage**
■ Config ■ Ready OL ■ Stepping ■ Jogging ■ Scanning
■ Ready CL ■ Referencing ■ Moving ■ Disable

**Compatibility**
■ Open-Loop
■ Closed-Loop

**Syntax** **ID**?

**Parameters**

**Description** **nn** [string] — Stage identifier.

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

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** The ID? command return the stage identifier. This is the Newport product name.

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

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** ZT — Get memory configuration.

**Example** ID? | Get stage identifier.
| Controller returns "Super Agilis V1.0.0"

# IF — Get interpolation factor

**Usage**
■ Config ■ Ready OL ■ Stepping ■ Jogging ■ Scanning
■ Ready CL ■ Referencing ■ Moving ■ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax** **IF**?

**Description** The IF? command return the interpolation factor.

The encoder position resolution is : 0.25 * SU / IF

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands**
ZT — Get memory configuration.
SU — Set/Get encoder pitch

**Example**
IF? | Get interpolation factor.
| Controller returns 7987
SU? | Get encoder pitch
| Controller returns 0.0798742
| encoder resolution = 0.25 * 0.0798742 / 7987 = 2.5nm

# FS — Factory settings modification/restoration

**Usage** ■ Config ☐ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **FSM**nn, or **FSM**?
**FSR**

**Parameters**

**Description** **nn** [string] — Password for factory settings modification.

**Range** **nn** — Unique password.

**Units** **nn** — None.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** **FSM**: This command sends the password to allow factory settings modifications. If the password is correct, all parameters modification in CONFIGURATION state changes the factory settings of the Controller. The factory settings modifications rights are lost after leaving the CONFIGURATION state.

**FSR**: This command restores all parameters to factory settings.

**Returns** If the sign "?" takes place of nn, this command returns 1 if the Controller have the factory settings modifications rights, else it returns 0.

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** PW — Enter/Leave CONFIGURATION state.

**Example** FSR | Restore all parameters to factory settings.

# JA — Move jogging

**Usage** ☐ Config ■ Ready OL ☐ Stepping ■ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **JA**nn, or **JA**?

**Parameters**

**Description** **nn** [int] — Jogging mode.

**Range** **nn** — ≥-4 and ≤4.

**Units** **nn** — Mode.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** The JA command initiates a motion in Jogging mode.
-4 — Negative direction, 10 000 steps/s at max. step amplitude.
-3 — Negative direction, 5 000 steps/s at max. step amplitude.
-2 — Negative direction, 1 000 step/s at max. step amplitude.
-1 — Negative direction, 50 steps/s at defined step amplitude.
0 — No move, stay in JOGGING state. (use ST command to go to READY OL state)
1 — Positive direction, 50 steps/s at defined step amplitude.
2 — Positive direction, 1 000 steps/s at max. step amplitude.
3 — Positive direction, 5 000 steps/s at max. step amplitude.
4 — Positive direction, 10 000 steps/s at max. step amplitude.

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

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** ST — Stop the current motion.

**Example** JA4 | Set jogging mode to 4.

# KF — Set feed-forward gain

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax** **KF**nn1,nn2,nn3,nn4 or **KF**?

**Parameters**

**Description**
**nn1** [float] — Feedforward gain value for -12mm/s velocity.
**nn2** [float] — Feedforward gain value for -6mm/s velocity.
**nn3** [float] — Feedforward gain value for +6mm/s velocity.
**nn4** [float] — Feedforward gain value for +12mm/s velocity.

**Range** **nn** — ≥0 and <10.

**Units** **nn** — none.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description**
In CONFIGURATION state, this command sets the integral gain of the feedforward control loop which can 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 OPEN LOOP states.

In DISABLE or READY OPEN LOOP states, this command allows setting a new working parameter for the feedforward gain. This value is not saved in the Controller's memory and will be lost after reboot.

This command is not available for firmware version inferior to v0.17.0

The feedwoard KF gain is variable and can be set with 4 points (-12, -6, 6 and 12 mm/s)

- KF is constant at nn1 for setpoint velocity from -15 to -12mm/s
- KF is interpolated between nn1 and nn2 for setpoint velocity from -12 to -6mm/s
- KF is constant at nn2 for setpoint velocity from -6 to 0mm/s
- KF is constant at nn3 for setpoint velocity from 0 to 6mm/s
- KF is interpolated between nn3 and nn4 for setpoint velocity from 6 to 12mm/s
- KF is constant at nn4 for setpoint velocity from 12 to 15mm/s

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

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands**
DB — Set/Get deadband.
KP — Set/Get proportional gain.

**Example**

| | |
|---|---|
| KF1,1,1,1 | Set feedforward gain to 1 (constant) |
| KF 0.7,1.4,1.0,0.7 | Set feedforward gain to be attenuated at 0.7 at high velocity boosted at 1.4 at low negative velocity |

# KI — Set integral gain

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax** **KI**nn, or **KI**?

**Parameters**

**Description** **nn** [float] — Integral gain value.

**Range** **nn** — ≥0 and <1E12.

**Units** **nn** — none.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** In CONFIGURATION state, this command sets the integral gain of the PI control loop which can 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 OPEN LOOP states.

In DISABLE or READY OPEN LOOP states, this command allows setting a new working parameter for the integral 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** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** DB — Set/Get deadband.
KP — Set/Get proportional gain.

**Example** KI7800 | Set integral gain to 7800.

# KP — Set proportional gain

**Usage**
■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax** **KP**nn, or **KP**?

**Parameters**

**Description** **nn** [float] — Proportional gain value.

**Range** **nn** — ≥0 and <1E12.

**Units** **nn** — none.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** In CONFIGURATION state, this command sets the proportional gain of the PI 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 or READY OPEN LOOP states.

In DISABLE or READY OPEN LOOP states, this command allows setting a new working parameter for the proportional 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**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands**
DB — Set/Get deadband.
KI — Set/Get integral gain.

**Example** KP356 | Set proportional gain to 356.

# KS — Set integral saturation

**Usage**
■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax** **KS**nn, or **KS**?

**Parameters**

**Description** **nn** [float] — Integral saturation value.

**Range** **nn** — ≥0 and <14.7.

**Units** **nn** — mm/s.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** In CONFIGURATION state, this command sets the integral saturation of the PI 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 or READY OPEN LOOP states.

In DISABLE or READY OPEN LOOP states, this command allows setting a new working parameter for the saturation value. This value is not saved in the Controller's memory and will be lost after reboot.

This command is not available for firmware version inferior to v0.17.0

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

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands**
DB — Set/Get deadband.
KI — Set/Get integral gain.

**Example** KS1.5 | Set integral saturation to 1.5mm/s

# KO — Set/Get offset friction

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax** **KO**nn,pp or **KO**?

**Parameters**

**Description**

| | | | |
|---|---|---|---|
| **nn** [int] | — | Offset negative friction. |
| **pp** [int] | — | Offset positive friction. |

**Range**

| | | |
|---|---|---|
| **nn** | — | >-100 and <0 |
| **pp** | — | >0 and <100 |

**Units**

| | | |
|---|---|---|
| **nn** | — | % |
| **pp** | — | % |

**Defaults**

| | | |
|---|---|---|
| **nn** | Missing: | Error. |
| **pp** | Missing: | Error. |
| | Out of range: | Error. |

**Description** The Agilis controller produces pulses on the piezo, to generate motion, which scale from 0V to maximum pulse voltage as defined in the specification table. The KO command is used to configure the offset of the pulse generated in closed loop motion: ie: the minimum voltage of the pulse to be generated.

In CONFIGURATION state, this command sets the offset of the output pulse 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 OPEN LOOP states.

In DISABLE or READY OL states, this command allows setting a new working parameter for the offset. 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 in the form KO negative value, positive value.

**Errors**

| | | |
|---|---|---|
| | — | Unknown message code. |
| | — | Parameter missing or out of range. |
| | — | Execution not allowed. |
| | — | Execution not allowed in XXXX state. |

**Rel. Commands**

| | | |
|---|---|---|
| KI | — | Set integral gain. |
| KP | — | Set proportional gain. |

**Example**

| | | |
|---|---|---|
| KO-5,10 | \| | Set the positive offset friction to 10% of full scale pulse. |
| | \| | Set the negative offset friction to 5% of full scale pulse. |
| KO? | \| | Get offset friction |
| | | Controller returns "KO-5, 10" |

# MM — Enter/Leave DISABLE state

**Usage**
☐ Config ☐ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
■ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax** **MM**nn, or **MM**?

**Parameters**

**Description** **nn** [int] — direction.

**Range** **nn** — 0 change from READY CLOSED LOOP to DISABLE state.
1 change from DISABLE to READY CLOSED LOOP state.

**Units** **nn** — None.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** MM0 changes the Controller's state from READY to DISABLE. In DISABLE state the control loop is open. The encoder, though, is still read and the current position gets updated.

MM1 changes the Controller's state from DISABLE to READY CLOSED LOOP. The Controller's set point position is set equal to its current position and the control loop gets closed. The residual following error gets cleared from the buffer.

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

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** PW — Enter/leave CONFIGURATION state.

**Example** MM0 | Controllers goes to DISABLE state.

# MS — Get motion status

**Usage**
- ■ Config
- ■ Ready OL
- ■ Ready CL
- ■ Stepping
- ■ Referencing
- ■ Jogging
- ■ Moving
- ■ Scanning
- ■ Disable

**Compatibility**
- ■ Open-Loop
- ■ Closed-Loop

**Syntax** **MS**?

**Description** The MS? command returns 1 if the stage is moving, else it returns 0.

**Errors**

| | | |
|---|---|---|
| | — | Unknown message code. |
| | — | Execution not allowed. |

**Rel. Commands**

| | | |
|---|---|---|
| JA | — | Move jogging |
| PA | — | Move absolute. |
| PR | — | Move relative. |
| XR | — | move stepping |

**Example**

| | |
|---|---|
| JA4 | \| start jogging |
| MS? | \| Get motion status. |
| | \| Controller returns : MS1 |
| JA0 | \| stop jogging |
| MS? | \| Get motion status. |
| | \| Controller returns : MS0 |

# MT — Set/Get motion timeout

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **MT**nn or **MT**?

**Parameters**

| | | | |
|---|---|---|---|
| **Description** | **nn** [float] | — | timeout value |
| **Range** | **nn** | — | >0 and <200 |
| **Units** | **nn** | — | seconds |
| **Defaults** | **nn** Missing: | Error. | |
| | Out of range: | Error. | |

**Description** In CONFIGURATION state, this command sets the motion timeout 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 OPEN LOOP states.

In DISABLE or READY OL states, this command allows setting a new working parameter for the motion timeout. This value is not saved in the Controller's memory and will be lost after reboot.

Motion timeout is applied in MOVING and JOGGING states.

In JOGGING state the motion timeout is:

b. 1 * MT for JA4/JA-4
c. 3 * MT for JA3/JA-3
d. 10 * MT for JA2/JA-2
e. 500 * MT for JA1/JA-1

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

**Errors**

| | | |
|---|---|---|
| — | Unknown message code. |
| — | Parameter missing or out of range. |
| — | Execution not allowed. |
| — | Execution not allowed in XXXX state. |

**Rel. Commands**

| | | |
|---|---|---|
| JA | — | Move jogging |
| PA | — | Move absolute. |
| PR | — | Move relative. |
| XR | — | move stepping |

**Example** MT10 | Set the motion timeout to 10 seconds.

# OL — Go to READY OPEN LOOP state

**Usage**
☐ Config ☐ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
■ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax** **OL**.

**Parameters** None.

**Description** This command changes the current state from READY CLOSED LOOP to the state READY OPEN LOOP.

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

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

**Example**
OL | Go to READY OL state.

# OR — Go to READY CLOSED LOOP state

**Usage**
☐ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax**
**OR**
**ORM**nn

**Description**
**nn** [float] — Preset position

**Range**
**nn** — SL to SR

**Units**
**nn** — Units.

**Defaults**
**nn** Missing: Error.
Out of range: Error.

**Description**
This command goes from READY OPEN LOOP to READY CLOSED LOOP.

The OR command closes the control loop without modification of current position.

The ORM command sets the position specified by nn. If the requested position is do not match the current phase of the stage, a motion is initiated so that the current phase of the encoder matches the requested position.

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands**
OL — Go to READY OL state.

**Example**
OR | Go to READY CLOSED LOOP state.

# PA — Move absolute

| | | | | | |
|---|---|---|---|---|---|
| **Usage** | ☐ Config | ☐ Ready OL | ☐ Stepping | ☐ Jogging | ☐ Scanning |
| | | ■ Ready CL | ☐ Referencing | ■ Moving | ☐ Disable |
| **Compatibility** | ☐ Open-Loop | | | | |
| | ■ Closed-Loop | | | | |

**Syntax** **PA**nn, or **PA**?

**Parameters**

**Description** **nn** [float] — New target position.

**Range** **nn** — SL to SR.

**Units** **nn** — Preset units.

**Defaults** **nn** Missing: Error.

Out of range: Error.

**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 CLOSED LOOP or MOVING states.

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** — Unknown message code.

— Parameter missing or out of range.

— Execution not allowed.

— Target position out of limits.

— Execution not allowed in XXXX state.

**Rel. Commands** PR — Move relative.

TH — Tell set point position.

TP — Tell current position.

**Example** PA2.2 | Move to 2.2 units absolute position.

# PR — Move relative

| | | | | | |
|---|---|---|---|---|---|
| **Usage** | ☐ Config | ☐ Ready OL | ☐ Stepping | ☐ Jogging | ☐ Scanning |
| | | ■ Ready CL | ☐ Referencing | ■ Moving | ☐ Disable |
| **Compatibility** | ☐ Open-Loop | | | | |
| | ■ Closed-Loop | | | | |

**Syntax** **PR**nn, or **PR**?

**Parameters**

**Description** **nn** [float] — Displacement.

**Range** **nn** — SL to SR

**Units** **nn** — Preset units.

**Defaults** **nn** Missing: Error.

Out of range: Error.

**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 CLOSED LOOP or MOVING states.

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**

— Unknown message code.

— Parameter missing or out of range.

— Execution not allowed.

— Target position out of limits.

— Execution not allowed in XXXX state.

**Rel. Commands**

PA — Move absolute.

TH — Tell set point position.

TP — Tell current position.

**Example** PR2.2 | Move 2.2 units from target position.

# PW — Enter/Leave configuration state

**Usage**
■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility**
■ Open-Loop
■ Closed-Loop

**Syntax** **PW**nn, or **PW**?

**Parameters**

**Description** **nn** [int] — Configuration mode.

**Range** **nn** — 1 change from READY OPEN LOOP to CONFIGURATION.
0 change from CONFIGURATION to READY OPEN LOOP.

**Units** **nn** — None.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** PW1 changes the Controller's state from READY OPEN LOOP 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 controller address, 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 READY OPEN LOOP.

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**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

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

**Example** PW1 | Enters CONFIGURATION state.

# RF — Execute the referencing sequence

**Usage**
☐ Config ☐ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
■ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility**
■ Open-Loop
■ Closed-Loop

**Syntax**
**RFH**
**RFP**
**RFM**nn
**RFS**?

**Parameters**

**Description** **nn** [float] — Reference position.

**Range** **nn** — SL to SR.

**Units** **nn** — Preset Unit.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description**
The RF command executes the referencing sequence:

**RFH**: Move to mechanical end of run defined by HT, and take this position as reference.

After the sequence current position equals SR or SL according to HT value.

**RFP**: Move to mechanical end of run defined by HT, take this position as reference, and move back to previous position.

After the physical position is the same than before the sequence, but the current position is referenced to SR or SL according to HT value.

**RFM**: Move to mechanical end of run defined by HT, take this position as reference, and move to position indicated by the parameter nn.

**RFS?**: Returns 1 if the stage is referenced and 0 otherwise.

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Example**
RFS? | Controller returns 0
RFH | Move to the reference position to the mechanical limit defined by HT.
RFS? | Controller returns 1

# RS — Reset controller

**Usage**
- ■ Config
- ■ Ready OL
- ■ Ready CL
- ■ Stepping
- ■ Referencing
- ■ Jogging
- ■ Moving
- ■ Scanning
- ■ Disable

**Compatibility**
- ■ Open-Loop
- ■ Closed-Loop

**Syntax** **RS**.

**Parameters** None.

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

**Errors**
- — Unknown message code.
- — Execution not allowed.

**Example**

RS | Reset Controller.

# RT — Tell temperature

**Usage**
- ■ Config
- ■ Ready OL
- ■ Ready CL
- ■ Stepping
- ■ Referencing
- ■ Jogging
- ■ Moving
- ■ Scanning
- ■ Disable

**Compatibility**
- ■ Open-Loop
- ■ Closed-Loop

**Syntax** **RT**

**Parameters** None.

**Description** The RT command returns the current temperature of the processor in °C.

**Errors**
- — Unknown message code.
- — Execution not allowed.

**Example**

| | | |
|---|---|---|
| RT | \| | Tell temperature |
| RT50 | \| | Controller returns 50 °C. |

# SA — Set/Get axis number

**Usage** ■ Config ☐ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **SA**nn, or **SA**?

**Parameters**

**Description** **nn** [int] — Axis number

**Range** **nn** — 1 to 31

**Units** **nn** — None

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** obsolete command kept for backward compability.

For firmware versions inferior to v1.0.0, The controller responds only to commands sent with this address.

For firmware version starting from v1.0.0, this command have no effect, the controller responds to all addresses. SA can still be used to store axis number information even if it has no effect in firmware.

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

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** **RF** — Execute the referencing sequence.

**Example (v0.x.x)**

| | | |
|---|---|---|
| SA2 | \| | Set the axis number to 2 |
| 1TP? | \| | Controller doesnt respond |
| 2TP? | \| | Controller responds with position |

**Example (v1.x.x)**

| | | |
|---|---|---|
| SA2 | \| | Set the axis number to 2 |
| 1TP? | \| | Controller responds with position |
| 2TP? | \| | Controller responds with position |

# SL — Set/Get stage left travel range

**Usage**
■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax** **SL**nn, or **SL**?

**Parameters**

**Description** **nn** [float] — Stage left travel range.

**Range** **nn** — ≤0 and >-1E12.

**Units** **nn** — Preset units.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** In CONFIGURATION state, this command sets the stage left travel range which can than be saved in the Controller's nonvolatile memory using the PW command.

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

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** **RF** — Execute the referencing sequence.
**SR** — Set/get stage right travel range

**Example** SL-16 | Set stage left travel range to -16 units.

# SR — Set/Get stage right travel range

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax** **SR**nn, or **SR**?

**Parameters**

**Description** **nn** [float] — Stage right travel range.

**Range** **nn** — ≥0 and <1E12.

**Units** **nn** — Preset units.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** In CONFIGURATION state, this command sets the stage right travel range which can than be saved in the Controller's nonvolatile memory using the PW command.

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

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** **RF** — Execute the referencing sequence.
**SL** — Set/get stage left travel range

**Example** SR16 | Set stage left travel range to 16 units.

# SS — Set/Get scanning settings

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax**

| | |
|---|---|
| **SSD**nn1 | or **SSD**? |
| **SSI**nn2 | or **SSI**? |
| **SSK**nn3,mm3 | or **SSK**? |
| **SSP**nn4 | or **SSP**? |
| **SSN**nn5 | or **SSN**? |
| **SST**nn6 | or **SST**? |

**Parameters**

**Description**

| | | |
|---|---|---|
| **nn1** [float] | — | Scanning shift distance |
| **nn2** [float] | — | Scanning integral initialization gain |
| **nn3** [float] | — | Shifting integral coefficient |
| **mm3** [float] | — | Scanning integral coefficient |
| **nn4** [float] | — | Scanning shift positive deadband |
| **nn5** [float] | — | Scanning shift negative deadband |
| **nn6** [int] | — | Scanning shift deadband timer |

**Range**

| | | |
|---|---|---|
| **nn1** | — | ≤ 0 and ≥-0.0005 |
| **nn2** | — | ≥ 0 and ≤2 |
| **nn3,mm3** | — | ≥ 0 and <50000 |
| **nn4,nn5** | — | <0 and >-0.0005 |
| **nn6** | — | >0 and <100 |

**Units** **nn** — Preset units.

**Defaults**

| | | |
|---|---|---|
| **nn** | Missing: | Error. |
| | Out of range: | Error. |

**Description** In closed-loop motion are separated in 3 phases:

JOGGING:

The stage move to the target position shifted by SSD distance

Once the position is detected SST times in the shift deadband defined by SSP/SSN, the controller switch to SHIFTING phase.

SHIFTING:

In this phase the stage moves slowly to the target position by increasing the piezo voltage with SSK integrale servo coefficient (1st parameter). The integrale is initialized to a value that is multiplied by SSI.

Once the position is detected DDT times in the deadband defined by DB, the controller switch to SCANNING phase, and the motion is ended (go back to ready CL state). The servo integral coefficient is also switched to 2nd SSK parameter

SCANNING:

In this phase the piezo voltage is continuously controlled with integrale servo (SSK 2nd parameter).

If the position is detected outside the exit deadband (see DDS) the controller go back to JOGGING state with shifted target.

For firmware version inferior to v0.17.0, SSK have only one parameter and the same integral coefficient is used for SHIFTING and SCANNING

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

**Errors**
- — Unknown message code.
- — Parameter missing or out of range.
- — Execution not allowed.
- — Execution not allowed in XXXX state.

**Rel. Commands**
- **DB** — Set/Get deadband
- **DD** — Set/Get deadband entry/exit conditions

**Example**

| | | |
|---|---|---|
| SSD-0.0002 | \| | Set shift distance to -0.0002 preset units |
| SSP0.00015 | \| | Set positive shift deadband to 0.00015 preset units |
| SSN-0.0002 | \| | Set negative shift deadband to -0.0002 preset units |
| SSK3000 | \| | Set scanning KI to 3000 |
| SSI0.9 | \| | Set scanning integral initialization gain to 0.9 |

# ST — Stop motion

| | | | | | |
|---|---|---|---|---|---|
| **Usage** | ☐ Config | ☐ Ready OL | ■ Stepping | ■ Jogging | ■ Scanning |
| | | ☐ Ready CL | ■ Referencing | ■ Moving | ☐ Disable |
| **Compatibility** | ■ Open-Loop | | | | |
| | ■ Closed-Loop | | | | |

**Syntax** **ST**.

**Parameters** None.

**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.

**Errors**
- — Unknown message code.
- — Execution not allowed.
- — Execution not allowed in XXXX state.

**Example**

| | | |
|---|---|---|
| ST | \| | Stop motion. |

# SU — Set/Get encoder pitch

| | | | | | | |
|---|---|---|---|---|---|---|
| **Usage** | ■ Config | ☐ Ready OL | ☐ Stepping | ☐ Jogging | ☐ Scanning | |
| | | ☐ Ready CL | ☐ Referencing | ☐ Moving | ☐ Disable | |
| **Compatibility** | ☐ Open-Loop | | | | | |
| | ■ Closed-Loop | | | | | |

**Syntax** **SU**nn or **SU**?

**Parameters**

**Description** **nn** [float] — Encoder scale pitch.

**Range** **nn** — ≥0 and <1E12.

**Units** **nn** — Preset units.

**Defaults** **nn** Missing: Error.

Out of range: Error.

**Description** In CONFIGURATION state, this command sets the encoder pitch which can than be saved in the Controller's nonvolatile memory using the PW command.

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

**Errors** — Unknown message code.

— Parameter missing or out of range.

— Execution not allowed.

— Execution not allowed in XXXX state.

**Rel. Commands** IF — Get the interpolation factor.

**Example** SU0.079 | Set encoder scale pitch to 0.079 units.

# TB — Tell error string

**Usage**
■ Config ■ Ready OL ■ Stepping ■ Jogging ■ Scanning
■ Ready CL ■ Referencing ■ Moving ■ Disable

**Compatibility**
■ Open-Loop
■ Closed-Loop

**Syntax** **TB**nn.

**Parameters**

**Description** **nn** [int] — Error code.

**Range** **nn** — 0 to ERROR CODE.

**Units** **nn** — None.

**Defaults** **nn** Missing: Error.
Out of range: Error.

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

**Errors**
— Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.

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

**Example**
TB@ | Tell explanation of error code @.
TB@ No error | Controller returns No error.

# TE — Tell last error

**Usage**
- ■ Config
- ■ Ready OL
- ■ Ready CL
- ■ Stepping
- ■ Referencing
- ■ Jogging
- ■ Moving
- ■ Scanning
- ■ Disable

**Compatibility**
- ■ Open-Loop
- ■ Closed-Loop

**Syntax** **TE**.

**Parameters** None.

**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**
— Unknown message code.
— Execution not allowed.

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

**Example**
TE | Tell last error code.
TE@ | Controller returns @.

**List of errors codes :.**

A : Unknown Message Code.

B : Axis Number not correct

C : Parameter out of Limits.

D : Function Execution not Allowed.

E : Voltage ERROR.

F : Function Execution not Allowed in SCANNING mode.

G : Function Execution not Allowed in JOGGING mode.

H : Function Execution not Allowed in READY OPEN LOOP mode.

I : Function Execution not Allowed in CONFIGURATION mode.

J : Function Execution not Allowed in DISABLE mode.

K : Function Execution not Allowed in READY CLOSED LOOP mode.

L : Function Execution not Allowed in HOMING/REFERENCING mode.

M : Function Execution not Allowed in MOVING mode.

N : Function Execution not Allowed in STEPPING mode.

O : Function Execution not Allowed in NO ENCODER mode.

P : Function Execution not Allowed in ENCODER mode.

S : Communication ERROR.

U : Error during EEPROM access.

# TH — Tell target position

**Usage**
■ Config ■ Ready OL ■ Stepping ■ Jogging ■ Scanning
■ Ready CL ■ Referencing ■ Moving ■ Disable

**Compatibility**
☐ Open-Loop
■ Closed-Loop

**Syntax** **TH**.

**Parameters** None.

**Description** The TH command returns the value of the target position.

**Errors**
— Unknown message code.
— Execution not allowed.

**Rel. Commands** TP — Tell current position.

**Example**
TH | Tell target position.
TH0 | Controller returns 0.

# TO — Set/Get motor stall timeout

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ■ Disable

**Compatibility** ☐ Open-Loop
■ Closed-Loop

**Syntax** **TOT**nn1 or **TOT**?
**TOD**nn2 or **TOD**?

**Parameters**

**Description**

| | | | |
|---|---|---|---|
| **nn1** [float] | — | timeout value |
| **nn2** [float] | — | min velocity value |

**Range**

| | | |
|---|---|---|
| **nn1** | — | >0 and <200 |
| **nn2** | — | 0.001 to 0.015 |

**Units**

| | | |
|---|---|---|
| **nn1** | — | seconds |
| **nn2** | — | preset units / seconds |

**Defaults**

| | | |
|---|---|---|
| **nn1** Missing: | Error. | |
| Out of range: | Error. | |

**Description** In CONFIGURATION state, this command sets the motor stall timeout 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 OPEN LOOP states.

In DISABLE or READY OL states, this command allows setting a new working parameter for the motor stall timeout. This value is not saved in the Controller's memory and will be lost after reboot.

Motor stall timeout stops the motion when velocity is under TOD during TOT seconds.

The purpose of this timeout is to stop automaticaly the motion when end of run is reached.

Motor stall timeout is applied during MOVING and JOGGING states, except for JA1/-1

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

**Errors**

| | | |
|---|---|---|
| | — | Unknown message code. |
| | — | Parameter missing or out of range. |
| | — | Execution not allowed. |
| | — | Execution not allowed in XXXX state. |

**Rel. Commands**

| | | |
|---|---|---|
| JA | — | Move jogging |
| PA | — | Move absolute. |
| PR | — | Move relative. |
| XR | — | move stepping |

**Example**

| | | |
|---|---|---|
| MT10 | \| | Set the motion timeout to 10 seconds. |

# TP — Tell current position

**Usage**
- ■ Config
- ■ Ready OL
- ■ Ready CL
- ■ Stepping
- ■ Referencing
- ■ Jogging
- ■ Moving
- ■ Scanning
- ■ Disable

**Compatibility**
- ■ Open-Loop
- ■ Closed-Loop

**Syntax** **TP**.

**Parameters** None.

**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 CLOSED LOOP 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.

If the stage doesn't have an encoder, TP return a step counter instead of the position.

**Errors**
- — Unknown message code.
- — Execution not allowed.

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

**Example**

| | | |
|---|---|---|
| TP | \| | Tell current position. |
| TP0 | \| | Controller returns 0. |

# TS — Get positioner error and controller state

**Usage** ■ Config ■ Ready OL ■ Stepping ■ Jogging ■ Scanning ■ Ready CL ■ Referencing ■ Moving ■ Disable

**Compatibility** ■ Open-Loop ■ Closed-Loop

**Syntax** **TS**.

**Parameters** None.

**Description** The TS command returns the error bits and the current Controller state.

**Returns** The TS command returns 8 characters (1TSabcdef).

The first 4 characters (abcd) represent the error bits in Hexadecimal.

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

| Error bits | |
|---|---|
| 0001 | Not used |
| 0002 | Not used |
| 0004 | Not used |
| 0008 | Not used |
| 0010 | Bit motor stall timeout |
| 0020 | Bit time out motion |
| 0040 | Bit time out homing |
| 0080 | Bit bad memory parameters |
| 0100 | Bit supply voltage too low |
| 0200 | Bit internal error |
| 0400 | Bit memory problem |
| 0800 | Bit over temperature |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

| Controller state | |
|---|---|
| 0A | READY OPEN LOOP: after reset |
| 0B | READY OPEN LOOP: after HOMING state |
| 0C | READY OPEN LOOP: after STEPPING state |
| 0D | READY OPEN LOOP: after CONFIGURATION state |
| 0E | READY OPEN LOOP: after with no parameters |
| 0F | READY OPEN LOOP: after JOGGING state |
| 10 | READY OPEN LOOP: after SCANNING state |
| 11 | READY OPEN LOOP: after READY CLOSED LOOP state |
| 14 | CONFIGURATION |
| 1E | HOMING |
| 1F | REFERENCING |
| 28 | MOVING OPEN LOOP (OL) |
| 29 | MOVING CLOSED LOOP (CL) |
| 32 | READY CLOSED LOOP: after HOMING state |
| 33 | READY CLOSED LOOP: after MOVING CL state |
| 34 | READY CLOSED LOOP: after DISABLE state |
| 35 | READY CLOSED LOOP: after REFERENCING state |
| 3C | DISABLE: after READY CLOSED LOOP state |
| 3D | DISABLE: after MOVING CL state |
| 46 | JOGGING |
| 50 | SCANNING |

**Errors** — Unknown message code.

**Rel. Commands** TE — Tell last error.

**Example** TS | Tell current status & errors.

TS000033 | Controller returns :
READY CLOSED LOOP: after MOVING CL state.

# VA — Set/Get Velocity

**Usage** ■ Config ■ Ready OL ■ Stepping ■ Jogging ■ Scanning
■ Ready CL ■ Referencing ■ Moving ■ Disable

**Compatibility** □ Open-Loop
■ Closed-Loop

**Syntax** **VA**nn or **VA**?

**Parameters**

**Description** **nn** [float] — Velocity

**Range** **nn** — ≥ 0.6 and ≤ 15

**Units** **nn** — mm/s

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** The VA command set the velocity used by the profiler during closed loop motion.
This command is not available for firmware version inferior to v0.17.0

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

**Errors** — Unknown message code.
— Execution not allowed.

**Rel. Commands** AC — Set/Get Acceleration
PA — Move absolute.
PR — Move relative.

**Example**
```
VA5      | set velocity to 5mm/s
VA?      | Get velocity.
         | Controller returns : VA5
```

# VE — Tell controller revision information

**Usage**
- ■ Config
- ■ Ready OL
- ■ Ready CL
- ■ Stepping
- ■ Referencing
- ■ Jogging
- ■ Moving
- ■ Scanning
- ■ Disable

**Compatibility**
- ■ Open-Loop
- ■ Closed-Loop

**Syntax** **VE** or **VE?**.

**Parameters** None.

**Description** This command returns the Controller's revision information.

**Errors** —Unknown message code.

**Rel. Commands** TP — Tell current position.

**Example**
```
VE |  Get Controller revision information.
   |  Controller returns VE Super Agilis Controller version 1.0.
```

# XF — Set/Get stepping frequency

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **XF**nn, or **XF**?

**Parameters**

**Description** **nn** [float] — Stepping frequency.

**Range** **nn** — ≥1 and ≤10 000.

**Units** **nn** — Hertz (Hz).

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** The XF command is used to read or modify the stepping frequency.

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

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** XR — Move stepping.
XU — Set/Get Open Loop step size.
TP — Tell current position.

**Example**

| | | |
|---|---|---|
| XF3000 | \| | Set the stepping frequency to 3000 Hz. |
| XU-60,50 | \| | Set the stepping negative amplitude to 60% |
| | \| | Set the stepping positive amplitude to 50% |
| XR1000 | \| | Generate 1000 positive pulses |
| XR-1000 | \| | Generate 1000 negative pulses |

# XN — Set/Get scanning command

**Usage** ☐ Config ☐ Ready OL ☐ Stepping ☐ Jogging ■ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **XN**nn or **XN**?

**Parameters**

**Description** **nn** [float] — scanning command.

**Range** **nn** [float] — 0 to 96.

**Units** **nn** — percent.

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** The XN command set the piezo voltage command in the SCANNING state.
The XN parameter is a percentage of the maximum voltage (48V)

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

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** XS — Go to scanning state.
ST — Stop motion

**Example** XS | Go to SCANNING state
XN20 | Set scanning command to 20%

# XR — Move stepping

**Usage** ☐ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **XR**nn

**Parameters**

| | | | |
|---|---|---|---|
| **Description** | **nn** [int] | — | Step count. |
| **Range** | **nn** [int] | — | int32. |
| **Units** | **nn** | — | Preset units. |
| **Defaults** | **nn** Missing: | | Error. |
| | Out of range: | | Error. |

**Description** The XR command initiates a stepping motion. When received, the positioner will generate a number of pulses specified by nn.

The XR command gets only accepted in READY OPEN LOOP state.

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

**Errors**

| | | |
|---|---|---|
| — | Unknown message code. |
| — | Parameter missing or out of range. |
| — | Execution not allowed. |
| — | Execution not allowed in XXXX state. |

**Rel. Commands**

| | | |
|---|---|---|
| XF | — | Set/Get stepping frequency. |
| XU | — | Set/Get Open Loop step size. |
| TP | — | Tell current position. |

**Example**

| | | |
|---|---|---|
| XF3000 | \| | Set the stepping frequency to 3000 Hz. |
| XU-60,50 | \| | Set the stepping negative amplitude to 60% |
| | \| | Set the stepping positive amplitude to 50% |
| XR1000 | \| | Generate 1000 positive pulses |
| XR-1000 | \| | Generate 1000 negative pulses |

# XS — Go to SCANNING state

**Usage**
☐ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility**
■ Open-Loop
■ Closed-Loop

**Syntax** **XS**

**Parameters** None.

**Description** The XS command changes the Controller's state from READY OPEN LOOP to SCANNING. The scanning command can then be changed with XN command to drive directly the piezo voltage at a constant value between 0V and 48V, allowing fine positioning in the piezo travel range (approx.. 2µm).

The XS command gets only accepted in READY OPEN LOOP state.

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

**Errors**
— Unknown message code.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands**
XN — Set/Get scanning command.
ST — Stop motion

**Example**
XS | Enter scanning mode.
XN15 | Set the scanning command to 15%
XN85 | Set the scanning command to 85%
ST | Leave scanning mode

# XU — Set/Get Open Loop step size

**Usage** ■ Config ■ Ready OL ☐ Stepping ☐ Jogging ☐ Scanning
☐ Ready CL ☐ Referencing ☐ Moving ☐ Disable

**Compatibility** ■ Open-Loop
■ Closed-Loop

**Syntax** **XU**nn, or **XU**?

**Parameters**

**Description** **nn** [int] — Step size.

**Range** **nn** — >-100 and <0 for negative pulses.
— >0 and <100 for positive pulses.

**Units** **nn** — %

**Defaults** **nn** Missing: Error.
Out of range: Error.

**Description** The XU commands enables to change the current open loop step size.

The variable step size is available only for XF frequencies up to 1 kHz. For higher frequencies the step size is always 100%

**Warning: if the step size is too small, the stage may not move or move in the wrong direction.**

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

**Errors** — Unknown message code.
— Parameter missing or out of range.
— Execution not allowed.
— Execution not allowed in XXXX state.

**Rel. Commands** XR — Move stepping.
XF — Set/Get stepping frequency.
TP — Tell current position.

**Example**

| | |
|---|---|
| XF3000 | Set the stepping frequency to 3000 Hz. |
| XU-60,50 | Set the stepping negative amplitude to 60% |
| | Set the stepping positive amplitude to 50% |
| XR1000 | Generate 1000 positive pulses |
| XR-1000 | Generate 1000 negative pulses |
| XU? | Get open loop step size. |
| | Controller returns “XU-60, 50” |

# ZT — List all configuration parameters

| | | | | | |
|---|---|---|---|---|---|
| **Usage** | ■ Config | ■ Ready OL | ☐ Stepping | ☐ Jogging | ☐ Scanning |
| | | ☐ Ready CL | ☐ Referencing | ☐ Moving | ■ Disable |
| **Compatibility** | ■ Open-Loop | | | | |
| | ■ Closed-Loop | | | | |

**Syntax** **ZT**.

**Parameters** None.

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

**Errors**
— Unknown message code.
— Execution not allowed in XXXX state.

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

**Example**

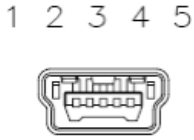
```
ZT |  Tell configuration parameters
PW1
…
SR32
…
…
PW0.
```

# 5.0 Connector Pinout

## 5.1 Power Connectors (Female Ø 2.1 x Ø 5.5 x 11 mm)

| Pin # | Description |
|---|---|
| Center | +24 VDC |
| Outer | GND |

## 5.2 USB Connector (USB Mini-B Receptacle)




| Pin | Description |
|---|---|
| 1 | N.C. |
| 2 | USBDM |
| 3 | USBDP |
| 4 | N.C. |
| 5 | Ground |
| 22 | Ground |
| 23 | N.C. |
| 24 | N.C. |
| 25 | N.C. |

# 6.0 Troubleshooting

Here is a list of possible errors and the recommended corresponding actions to be taken prior to contact Newport.

| "TS" Error | Description | Actions |
|---|---|---|
| 00001 | Not implemented in SAG series stages | |
| 00002 | | |
| 00004 | | |
| 00008 | | |
| 00010 | The stage has reached a hard limit | • Check that the stage is referenced<br>• Check software limits (SR/SL)<br>• Increase MO<br>• Restore Factory settings |
| 00020 | The requested motion didn't end after MT seconds | • Increase MT<br>• Increase deadband for quicker stabilization (but lower accuracy)<br>• Restore Factory settings |
| 00040 | Bit time out homing | • Restore Factory settings |
| 00080 | Bit bad memory parameters | • Restore Factory settings |
| 00100 | Power supply voltage lower than 42V caused Or defective power supply | • Check power supply |
| 00200 | Bit internal error | |
| 00400 | Bit memory problem | |
| 00800 | Bit over temperature | • Add time between moves |

# 7.0 Maintenance and Service

## 7.1 Enclosure Cleaning

The Super Agilis Controller/Driver should only be cleaned with a lightly damped cloth or sponge with a soapy water solution. Do not use an acetone or alcohol solution, this will damage the finish of the enclosure.

## 7.2 Obtaining Service

The Super Agilis Controller/Driver contains no user serviceable parts. To obtain information regarding factory service, contact Newport Corporation or your Newport representative. Please have the following information available:

- Instrument model number.
- Description of the problem.

If the instrument is to be returned to Newport Corporation, you will be given a Return Number, which you should reference in your shipping documents.

Complete a copy of the Service Form as represented on the next page and include it with your shipment.

# 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®