FCL Series

Intelligent Stepper Motor
Linear Stages

Newport® User’s Manual
Warranty

Newport Corporation warrants this product to be free from defects in material and workmanship for a period of 1 year from the date of shipment. If found to be defective during the warranty period, the product will either be repaired or replaced at Newport’s discretion.

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

Limitation of Warranty

This warranty does not apply to defects resulting from modification or misuse of any product or part.

CAUTION

Warranty does not apply to damages resulting from:

- Incorrect usage:
  - Driven load greater than maximum specified load.
  - Stage speed higher than specified.
  - Improper grounding.
    - Connectors must be properly secured.
    - When the load on the stage represents an electrical risk, it must be connected to ground.
  - Excessive or improper cantilever loads.
- Modification of the stage or any part thereof.

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.

©2018 by Newport Corporation, Irvine, CA. All rights reserved.

Original instructions.

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

Warranty ................................................................. ii
EC Declaration of Conformity ......................................... v
Definitions and Symbols ................................................ vi
Warnings ....................................................................... vii
Cautions ....................................................................... viii

## 1.0 Introduction ......................................................... 1

## 2.0 System Overview .................................................. 2

### 2.1 General Description .............................................. 2

#### 2.1.1 Package Contents ........................................... 2

#### 2.1.2 Controller Specifications .................................... 3

#### 2.1.3 Design Details ................................................ 3

#### 2.1.4 Dimensions ................................................... 4

#### 2.1.5 Weights ......................................................... 4

#### 2.1.6 Assemblies ................................................... 5

#### 2.1.7 FC-PS40 Power Supply ..................................... 6

#### 2.1.8 USB-RS422-1.8 USB Adapter .......................... 6

#### 2.1.9 FC-CB1 Daisy Chain Communication Cable ........ 6

#### 2.1.10 EQ120 Bracket ............................................. 7

### 2.2 Characteristics .................................................... 8

#### 2.2.1 Definitions ................................................... 8

#### 2.2.2 Mechanical Specifications ................................. 9

#### 2.2.3 Load Characteristics and Stiffness ..................... 9

### 2.3 System Environmental Specifications ..................... 9

### 2.4 Connector Identification ........................................ 10

### 2.5 Communication Settings ....................................... 11

## 3.0 Getting Started ..................................................... 12

### 3.1 Mounting and Initial Setup .................................... 12

### 3.2 Connection ....................................................... 12

### 3.3 Initialization and Homing ..................................... 14

### 3.4 Daisy-Chaining .................................................. 15

### 3.5 Multi Stage Configuration .................................... 16

## 4.0 Programming ....................................................... 17

### 4.1 State Diagram .................................................... 17

### 4.2 Command Syntax ............................................... 19
4.3 Command Execution Time............................................................................................... 19
4.4 Command Set................................................................................................................... 20
AC — Set/Get acceleration ............................................................................................ 22
BA — Set/Get backlash compensation ........................................................................... 23
BH — Set/Get hysteresis compensation ........................................................................ 24
FR — Set/Get stepper motor configuration .................................................................... 25
HT — Set/Get HOME search type .................................................................................. 26
ID — Set/Get stage identifier ........................................................................................ 27
JR — Set/Get jerk time ................................................................................................... 28
MM — Enter/Leave DISABLE state.............................................................................. 29
OH — Set/Get HOME search velocity ........................................................................... 30
OR — Execute HOME search ........................................................................................ 31
OT — Set/Get HOME search time-out ........................................................................... 32
PA — Move absolute ..................................................................................................... 33
PR — Move relative ....................................................................................................... 34
PT — Get motion time for a relative move .................................................................... 35
PW — Enter/Leave CONFIGURATION state ............................................................... 36
RS — Reset controller .................................................................................................... 37
RS## — Reset controller’s address ................................................................................ 38
SA — Set/Get controller’s RS422 address ..................................................................... 39
SE — Configure/Execute simultaneous started move .................................................... 40
SL — Set/Get negative software limit ........................................................................... 41
SR — Set/Get positive software limit .......................................................................... 42
ST — Stop motion ........................................................................................................... 43
TB — Get command error string .................................................................................... 44
TE — Get last command error ....................................................................................... 45
TH — Get set-point position .......................................................................................... 46
TP — Get current position .............................................................................................. 47
TS — Get positioner error and controller state ............................................................. 48
VA — Set/Get velocity ................................................................................................... 50
VE — Get controller revision information ..................................................................... 51
ZT — Get all configuration parameters ......................................................................... 52

5.0 Maintenance ........................................................................................................... 53
5.1 Maintenance ............................................................................................................. 53
5.2 Repair ....................................................................................................................... 53
5.3 Troubleshooting ....................................................................................................... 54

Service Form ............................................................................................................. 55
EC Declaration of Conformity

EU Declaration of Conformity
following Annex II-1A
of Directive 2006/42/EC on machinery

The manufacturer:
MICRO-CONTROLE Spectra-Physics,
9, rue du bois sauvage
F-91055 Evry FRANCE

Hereby declares that the machinery:
• Description: "FCL"
• Function: Low Cost Linear Stage
• Models: FCL50 / FCL 100 / FCL 200

– the technical file of which was compiled by:
  Mr Hervé LE COINTE, Quality Director,
  MICRO-CONTROLE Spectra-Physics, Zone Industrielle - B.P.29
  F-45340 Beaune La Rolande France

– complies with all the relevant provisions of the Directive 2006/42/EC on machinery,
– complies with all the relevant provisions of the Directive 2014/30/EU relating to electromagnetic compatibility,
– complies with all the relevant provisions of the Directive 2011/65/EU relating to RoHS2.

– was designed and built in accordance with the following harmonised standards:
  • NF EN 61326-1:2013 « Electrical equipment for measurement, control and laboratory use – EMC requirements – Part 1: General requirements »
  • NF EN 55011:2010/A1:2011 Class A

– was designed and built in accordance with the following other standards:
  • NF EN 61000-4-2
  • NF EN 61000-4-3
  • NF EN 61000-4-4
  • NF EN 61000-4-5
  • NF EN 61000-4-6

ORIGINAL DECLARATION

Done in Beaune La Rolande on 16 May 2017
Hervé LE COINTE
Quality Director
Definitions and Symbols

The following terms and symbols are used in this documentation and also appear on the product where safety-related issues occur.

General Warning or Caution

The exclamation symbol 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.

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**

Warning indicates a potentially dangerous situation which can result in bodily harm or death.

---

**CAUTION**

Caution indicates a potentially hazardous situation which can result in damage to product or equipment.

---

**NOTE**

Note indicates additional information that must be considered by the user or operator.

---

European Union 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.

---

**WARNINGS AND CAUTIONS**

---

**ATTENTION**

This stage is a Class A device. In a residential environment, this device can cause electromagnetic interference. In this case, suitable measures must be taken by the user.
Warnings

WARNING
The translation of objects of all types carries potential risks for operators. Ensure the protection of operators by prohibiting access to the dangerous area and by informing the personnel of the potential risks involved.

WARNING
Do not use this stage when its motor is emitting smoke or is unusually hot to the touch or is emitting any unusual odor or noise or is in any other abnormal state.
Stop using the stage immediately, switch off the motor power by disconnecting the power supply.
After checking that smoke is no longer being emitted contact your Newport service facility and request repairs. Never attempt to repair the stage yourself as this can be dangerous.

WARNING
Make sure that this stage is not exposed to moisture and that liquid does not get into the stage.
Nevertheless, if any liquid has entered the stage, switch off the motor power by disconnecting the power supply.
Contact your Newport service facility and request repairs.

WARNING
Do not insert or drop objects into this stage, this may cause an electric shock, or lock the drive.
Do not use this stage if any foreign objects have entered the stage. Switch off the motor power by disconnecting the power supply.
Contact your Newport service facility for repairs.

WARNING
Do not place this stage in unstable locations such as on a wobbly table or sloping surface, where it may fall or tip over and cause injury.
If this stage has been dropped or the case has been damaged, switch off the motor power by disconnecting the power supply.
Contact your Newport service facility and request repairs.

WARNING
Do not attempt to modify this stage; this may cause an electric shock or downgrade its performance.

WARNING
Do not exceed the usable depth indicated for the mounting holes (see section “Dimensions”). Longer screws can damage the mechanics or cause a short-circuit.
Cautions

---

**CAUTION**

Do not place this stage in a hostile environment such as X-Rays, hard UV,... or in any vacuum environment.

---

**CAUTION**

Do not place this stage in a location affected by dust, oil fumes, steam or high humidity. This may cause an electric shock.

---

**CAUTION**

Do not leave this stage in places subject to extremely high temperatures or low temperatures. This may cause an electric shock.

- Operating temperature: +10 to +35 °C.
- Storage temperature: -10 to +40 °C (in its original packaging).

---

**CAUTION**

Do not move this stage if its motor power is on.

Make sure that cables are disconnected before moving the stage. Failure to do so may damage the cable and cause an electrical shock.

---

**CAUTION**

Be careful that the stage is not bumped when it is being carried. This may cause it to malfunction.

---

**CAUTION**

When handling this stage, always unplug the equipment from the power source for safety.

---

**CAUTION**

Contact your Newport service facility to request cleaning and specification control every year.
1.0 Introduction

This manual provides operating instructions for the intelligent stepper motor linear stage that you have purchased:

- FCL50
- FCL100
- FCL200

The FCL is a precision linear stage with an integrated stepper motor/controller. The stage can be easily controlled from a PC, using a downloadable graphical user interface (GUI). Communication with the FCL is achieved via an RS-422 serial link. A USB to RS422 adaptor can be used (requires Windows™ XP or Windows™ 7 operating system). While the GUI enables basic motion, advanced application programs can be developed via an ASCII command interface and a set of two letter mnemonic commands.
2.0 System Overview

2.1 General Description

The FCL linear stage line is an evolution of the industry proven ILS stage. Building on the robust construction of recirculating ball technology in the guide and screw, the ILS was value engineered, keeping only the features necessary for everyday precision positioning needs. This and the addition of an integrated stepper motor/controller makes the FCL the ultimate choice in user-friendly precision stages.

The FCL line incorporates similar stepper controller technology found in the popular SMC and ESP301 controllers and extended this further by integrating the electronics into the stepper motor housing, resulting in the Newport-proprietary iPP intelligent stepper motor. Similar to the successful CONEX concept for DC motors, iPP technology completely eliminates controller or driver set up. Just connect power and communication and install the GUI. Up to 4 FC series units can be daisy-chained and controlled by a single GUI, via USB (USB-RS422-1.8) and direct RS-422 (daisy chaining cable FC-CB1).

2.1.1 Package Contents

- FCLxxx Intelligent Stepper Motor Linear Stage, 50, 100 or 200 mm
- Stage test report
- FC-PS40 FC Stage Power Supply (Cable Length: 1.75 m)
- USB-RS422-1.8 USB Adapter (Cable Length: 1.8 m)
- FC-CB1 Daisy Chain Communication Cable (Cable Length: 1 m)
2.1.2 Controller Specifications

<table>
<thead>
<tr>
<th>General description</th>
<th>Single Axis Intelligent Stepper Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control capability</td>
<td>Stepper motors, open loop</td>
</tr>
<tr>
<td>Output power</td>
<td>24 VDC peak, 2.5 A peak PWM amplifier</td>
</tr>
</tbody>
</table>
| Control loop        | – PI control of motor phase current with programmable idle state  
|                     | – 2 kHz servo rate                    |
|                     | – Backlash compensation               |
| Motion              | Point-to-point motion with S-gamma profile and jerk time control. |
| Computer interface  | USB +5 V (USB): < 0.5 A , Windows Compatible |
| Programming         | – 40+ intuitive, 2-letter ASCII commands  
|                     | – Command set includes software limits, user units, synchronized motion start, stop all… |
| Communication rate  | 115 200 baud (serial link over USB) |
| Internal safety features | End of range checks, power limit checks, watchdog timer |
| Consumption         | +24 V (FC-PS40): < 1.67 A |

2.1.3 Design Details

<table>
<thead>
<tr>
<th>Base Material</th>
<th>Aluminum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor</td>
<td>Integrated iPP 2-phase step motor and controller/driver</td>
</tr>
<tr>
<td>Drive Mechanism</td>
<td>Backlash-free ball screw</td>
</tr>
<tr>
<td>Drive Screw Pitch (mm)</td>
<td>2</td>
</tr>
<tr>
<td>Bearings</td>
<td>Recirculating ball bearings</td>
</tr>
<tr>
<td>Limit Switches</td>
<td>Optical</td>
</tr>
<tr>
<td>Origin</td>
<td>Optical at center of travel</td>
</tr>
<tr>
<td>Communication</td>
<td>RS422, USB</td>
</tr>
<tr>
<td>Cable</td>
<td>Cables and power supply sold separately</td>
</tr>
</tbody>
</table>
2.1.4 Dimensions

![Dimensions Diagram]

2.1.5 Weights

<table>
<thead>
<tr>
<th>Weight [lb (kg)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCL50</td>
</tr>
<tr>
<td>FCL100</td>
</tr>
<tr>
<td>FCL200</td>
</tr>
</tbody>
</table>
2.1.6 Assemblies

Below are a few assembly examples with FC linear and rotation stages. The FC series mounting interfaces use common hole patterns, eliminating the need for adapter plates. Call for compatibility with other Newport stages.

FCL200 and FCR100 rotation stage.

2 FCL stages in XY configuration and one FCR100 rotation stage.

2 FCL stages in XZ configuration with an EQ120 bracket.
2.1.7 FC-PS40 Power Supply

**NOTE**
Each FC series stage requires a FC-PS40 power supply.

Box Dim. (L x D x H) 125 x 50 x 32 mm

**Specifications**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC Input</td>
<td>100–240 VAC, 50–60 Hz, 1 A</td>
</tr>
<tr>
<td>DC Output</td>
<td>24 V, 1.67 A, 40 W max.</td>
</tr>
<tr>
<td>Cable Length</td>
<td>1.75 m between the power supply box and the iPP driver board connector</td>
</tr>
</tbody>
</table>

2.1.8 USB-RS422-1.8 USB Adapter

Cable Length 1.8 m

2.1.9 FC-CB1 Daisy Chain Communication Cable

Cable Length 1 m
2.1.10 EQ120 Bracket
2.2 Characteristics

2.2.1 Definitions
Specifications of our products are established in reference to ISO 230 standard part II “Determination of accuracy and repeatability of positioning numerically controlled axes”.

This standard gives the definition of position uncertainty which depends on the 3 following parameters:

(Absolute) Accuracy
Difference between ideal position and real position.

On-Axis Accuracy
Difference between ideal position and real position after the compensation of linear errors.

Linear errors include: cosine errors, inaccuracy of screw or linear scale pitch, angular deviation at the measuring point (Abbe error) and thermal expansion effects. All Newport motion electronics can compensate for linear errors.

The relation between absolute accuracy and on-axis accuracy is as follows:

\[
\text{Absolute Accuracy} = \text{On-Axis Accuracy} + \text{Correction Factor} \times \text{Travel}
\]

Repeatability
Ability of a system to achieve a commanded position over many attempts.

Reversal Value (Hysteresis)
Difference between actual position values obtained for a given target position when approached from opposite directions.

Minimum Incremental Motion (MIM or Sensitivity)
The smallest increment of motion a device is capable of delivering consistently and reliably.

Resolution
The smallest increment that a motion device can theoretically move and/or detect. Resolution is not achievable, whereas MIM, is the real output of a motion system.

The testing of on-axis accuracy, repeatability, and reversal error are made systematically with test equipment in a controlled room (20 ±1 °C).

The test consists of 4 cycles in each direction, with 21 data points over the travel resulting in a total 164 data points.

Guaranteed Specifications
Guaranteed maximum performance values are verified per Newport’s A167 metrology test procedure. For more information, please consult the metrology tutorial section in the Newport catalog or at www.newport.com.
### 2.2.2 Mechanical Specifications

<table>
<thead>
<tr>
<th></th>
<th>FCL50</th>
<th>FCL100</th>
<th>FCL200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Travel Range (mm)</td>
<td>50</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>Minimum Incremental Motion (µm)</td>
<td>0.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uni-directional Repeatability (µm)</td>
<td>1.5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Bi-directional Repeatability (^{(1)}) (µm)</td>
<td>4.5 or ±2.25</td>
<td>5 or ±2.5</td>
<td>5 or ±2.5</td>
</tr>
<tr>
<td>On-Axis Accuracy (^{(1)}) (µm)</td>
<td>4 or ±2</td>
<td>5 or ±2.5</td>
<td>10 or ±5</td>
</tr>
<tr>
<td>Maximum Speed (mm/s)</td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Pitch (^{(1)}), (\alpha_y) (µrad)</td>
<td>50 or ±25</td>
<td>100 or ±50</td>
<td>200 or ±100</td>
</tr>
<tr>
<td>Yaw (^{(1)}), (\alpha_z) (µrad)</td>
<td>70 or ±35</td>
<td>100 or ±50</td>
<td>160 or ±80</td>
</tr>
<tr>
<td>MTBF (h)</td>
<td>20,000 with a 1 kg load at 50% duty cycle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^{(1)}\) Shown are peak to peak, guaranteed specifications or ± half the value as sometimes shown. The typical specifications are about 2X better than the guaranteed values.

---

**WARNING**

To achieve the guaranteed specifications stated, stages must be fixed to a plane surface with a flatness of 5 µm or better.

### 2.2.3 Load Characteristics and Stiffness

| Cz, Normal center load capacity on bearings | 250 N |
| Cx, Axial load capacity                   | 40N   |
| kax, Angular stiffness (Roll)             | 15 µrad/Nm |
| key, Angular stiffness (Pitch)            | 10 µrad/Nm |
| kaz, Angular stiffness (Yaw)              | 10 µrad/Nm |
| Q, Off-center load                        | \(Q \leq Cz/(1 + D/60)\) |
| D, Cantilever distance in mm              |       |

### 2.3 System Environmental Specifications

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

Each FC series stage includes two RS422 communication connectors and a 24 V DC input connector. RS422 connector is connected to either PC (with USB-RS422-1.8 cable) or to RS422 connector of another FC series stage (by FC-CB1). Two RS422 connectors are interchangeable: either of the two connectors can be used for PC connection or the input/output for daisy-chaining connection.

<table>
<thead>
<tr>
<th>Connector</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS422 (2) 6-Pin RS422 communication connectors (connect to USB-RS422-1.8 or to FC-CB1 cable)</td>
<td></td>
</tr>
<tr>
<td>24 V in</td>
<td>3-Pin JST PAP-03V-S (connect to FC-PS40)</td>
</tr>
<tr>
<td>Cable retainer</td>
<td>2 x M3 threaded hole to attach cable retainer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PINOUT</th>
<th>PORT 1</th>
<th>PORT 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N.C.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>+Tx</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-Tx</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-Rx</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>+Rx</td>
<td></td>
</tr>
</tbody>
</table>

---

Newport
2.5 Communication Settings

Communication parameters are preset in the FCL controller and do not require any configuration:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits per second</td>
<td>115 200</td>
</tr>
<tr>
<td>Data bits</td>
<td>8</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
<tr>
<td>Stop bits</td>
<td>1</td>
</tr>
<tr>
<td>Flow control</td>
<td>None</td>
</tr>
<tr>
<td>End of line terminator</td>
<td>CRLF</td>
</tr>
</tbody>
</table>

- Communication standard: RS-422 4-wire full duplex without handshaking.
- 120 Ω termination resistor on receiver channel.
3.0 Getting Started

3.1 Mounting and Initial Setup
To achieve the guaranteed specifications, stages must be fixed on a plane surface with a flatness of 5 μm.

![Image of stage with mounting elements]

**WARNING**
The FC series, via threaded hole (M4), must be grounded to avoid electrical disturbances generated by ground loops.

3.2 Connection

**WARNING**
Do not connect or disconnect cables to FC series stage while power is applied.

Follow the steps described below for the cable connections:
Remove the cover from integrated motor/controller housing by loosening the two screws.

Refer to the Chapter 2.1 and Chapter 2.4 for the description of cables and connectors.
Make proper connections to RS422 connector and 24 VDC power connector, using the USB-RS422-1.8 USB adapter and the FC-PS40 cable respectively.

The second RS422 connector is used only for daisy-chaining.

Connect the USB-RS422-1.8 USB to a PC first. Then, connect the FC-PS40 power supply to an electrical outlet.

**WARNING**

The Power supply cable must remain fully disconnected from electrical outlet when you plug the USB-RS422-1.8 USB adaptor cable to the PC.

When completed, assemble the cover back into integrated motor/controller housing.

To start using the FCL, refer to the FC series controller GUI manual.
3.3 Initialization and Homing

In the Initialization and Configuration area, the first button “Initialize” executes homing and changes the controller state from “Not Referenced” to “Ready”. Once it is in the “Ready” state, the “Disable” or “Enable” buttons change the controller state to “Disabled” or “Ready”.

For the different controller states, refer to section 1.3 of the FC Controller GUI Manual. For further details of the Main tab of the GUI, refer to section 3.2 of the FC Controller GUI Manual.
3.4 Daisy-Chaining

Up to 4 FC series controllers can be networked through the internal RS422 communications link. Before daisy-chaining, the controller address of each stage must be set separately via the USB and the GUI. The FC controller that will be directly connected to the PC must have its controller address set to 1 and all subsequent stages must have a different controller address set between 2 and 4. Refer to the GUI manual or command interface manual for instructions to change the controller address.

Once the controller address of each stage is set, unplug all the power cables and disconnect the USB cable from all the stages except for the FC controller that has its address set to 1. Use the daisy chain cables to connect each controller. For stages #2 thru 4, it does not matter whether you use the upper or lower RS422 connector to daisy chain.

- Only the FC controller with address 1 should be connected to the PC.
- All other FC controllers must be daisy-chained.

NOTE

Another multi-axis option is to run multiple GUI's, each with an FCL or daisy-chained FCL's.
3.5 Multi Stage Configuration

Another way to control multiple stages is opening multiple GUI’s for every FC stage. Each of the FC stages has to be discovered independently.

- Address #1 (default) for all stages
4.0 Programming

4.1 State Diagram

For a safe and consistent operation, the FCL uses 6 different operational states: Not referenced, Configuration, Homing, Ready, Disable and Moving. In each state, only specific commands are accepted by the FCL. Therefore, it is important to understand the state diagram below and to know which commands and actions cause transitions between the different states. Also see section 4.4 for command/state information:

Actions in each state when End of Runs is encountered

- **NOT REFERENCED:** No action.
- **CONFIGURATION:** No action.
- **HOMING:** Only check at end of HOMING and then change to NOT REFERENCED state.
- **MOVING:** Abort motion and then changes to NOT REFERENCED state.
- **READY:** Changes to NOT REFERENCED state.
- **DISABLE:** Changes to NOT REFERENCED state.
After connecting the FCL to a power source, the stage is in the NOT REFERENCED state and must be initialized first. When initialization is successful, the controller goes to the NOT REFERENCED state. From the NOT REFERENCED state, the controller can go to the CONFIGURATION state with the PW1 command. In the CONFIGURATION state, the FCL allows changing stage and motor configuration parameters. The PW0 command saves all changes to the controller’s memory and returns the controller back to the NOT REFERENCED state.

In the READY state, the motor is energized and ready to move. During a move execution (PA/PR), the controller is in the MOVING state and goes automatically back to the READY state when the move is completed. When errors are generated in the READY state, it changes the controller to the NOT REFERENCED state.

In the DISABLE state, the motor is not energized and move commands cannot be accepted. To go from the READY state to the DISABLE state and vice versa, use the MM command.

To go from the READY state or the DISABLE state back to the NOT REFERENCED state, for instance to make further parameter changes in the CONFIGURATION state, you must reset the controller with the RS command.

---

**CAUTION**

Changing stage and motor configuration parameters in the CONFIGURATION state is recommended only for knowledgeable persons who understand the effect of the changes. See PW command for details.
4.2 Command Syntax

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

**Command format**

```
nn   AA   xx
```

- **nn** — Controller address, or nothing if the issued command addresses all controllers.
- **AA** — Command name.
- **xx** — Parameter value, or "?" to query the current value, or nothing if the command takes no parameter.

Both upper and lower case characters are accepted. Depending on the command, it can have an optional or required prefix (nn) for the controller address and/or a suffix (xx) value, a "?" or no suffix at all.

**Blank spaces**

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

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

**Decimal separator**

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

**Command terminator**

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

After the command parameters are identified, all remaining characters in the input string until the first command terminator, if any, will be ignored. Commands from the PC to the FCL may still be concatenated in a single string, but each command must be separated from the next one by a carriage-return or a line-feed.

In case any error occurs, the reported error will be recorded and can be checked using the TE command. Please refer to the command set in section 4.4 for details.

4.3 Command Execution Time

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

It is important to note that a move command, that may lasts for several seconds, will not suspend the controller from further command execution. So for an efficient process flow with many move commands it is recommended to use the PT command (get time for a relative move), and to query the controller status (TS command) or the current position (TP command) before any further motion command is sent.
4.4 Command Set

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

**Command format**

```
nn AA xx
```

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

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

Examples:

- **1VA10** sets the velocity of the controller #1 to 10 units/second (and sends nothing back).
- **1VA?** sends back the reply "1VA10", which means: "controller #1's velocity is 10 units/second".

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

**Not Ref.**
Corresponds to the NOT REFERENCED state (for details see state diagram, section 4.1).

**Config.**
Corresponds to the CONFIGURATION state.

**Disable**
Corresponds to the DISABLE state.

**Ready**
Corresponds to the READY state.

**Motion**
Corresponds to the HOMING and MOVING states.

- **Changes configuration parameters. Those changes will be stored in the controller’s memory with the PW1 command and remain available after switching off the controller.**
- **Changes working parameters only. Those changes will get lost when switching off the controller.**
- **Accepted command.**
- **Command is forbidden in this state (will memorize an error).**

**Grey line**
Command passed without preceding controller number applies to all controllers (e.g. MM0 disables all controllers).
### AC — Set/Get acceleration

#### Syntax

<table>
<thead>
<tr>
<th>xxACnn or xxAC?</th>
</tr>
</thead>
</table>

#### Parameters

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

- **Range**
  - **xx** — 1 to 31
  - **nn** — $> 10^{-4}$ and $< 10^{12}$

- **Units**
  - **xx** — None
  - **nn** — Preset units/s²

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

#### Description

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

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

#### Returns

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

#### Errors

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

#### Rel. Commands

- **JR** — Set/Get jerk time.
- **VA** — Set/Get velocity.

#### Example

- `1AC500` — Set controller #1 acceleration to 500 units/s².
- `1AC?` — Controller returns `1AC500`.
## BA — Set/Get backlash compensation

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>○</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

`xAA[nn]` or `xBA?`

**Parameters**

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

- **Range**
  - **xx** — 1 to 31
  - **nn** — \( \geq 0 \) and \(< 10^{12} \)

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

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

**Description**

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

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

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

**Returns**

If the sign “?” is used instead of `nn`, this command returns the current programmed value.

**Errors**

- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.
- **H** — Execution not allowed in NOT REFERENCED state.
- **J** — Execution not allowed in DISABLE state.
- **K** — Execution not allowed in READY state.
- **L** — Execution not allowed in HOMING state.
- **M** — Execution not allowed in MOVING state.

**Rel. Commands**

- **BH** — Set hysteresis compensation.

**Example**

`1BA0.005` | Set controller #1 backlash compensation to 0.005 units.
BH — Set/Get hysteresis compensation

Syntax: xxBHnn or xxBH?

Parameters:
- **Description**
  - **xx** [int] — Controller address.
  - **nn** [float] — Hysteresis value.

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

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

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

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

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

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

NOTE

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

Returns:
If the sign “?” is used instead of **nn**, this command returns the current programmed value.

Errors:
- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.
- **H** — Execution not allowed in NOT REFERENCED state.
- **J** — Execution not allowed in DISABLE state.
- **K** — Execution not allowed in READY state.
- **L** — Execution not allowed in HOMING state.
- **M** — Execution not allowed in MOVING state.

Relative Commands:
- **BA** — Set backlash compensation.

Example: 1BH0.015 | Set controller #1 backlash compensation to 0.015 units.
FR — Set/Get stepper motor configuration

**Syntax**
xxFRSn, xxFRM? or xxFRS?

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int] — Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mmm [int]—</td>
<td>Amount of micro-steps per full step.</td>
</tr>
<tr>
<td>Snn [float] —</td>
<td>Full step displacement length in 1/1000 of unit.</td>
</tr>
</tbody>
</table>

**Range**
- **xx** — 1 to 31
- **mm** — > 0 and ≤ 2000
- **nn** — > 10⁻⁶ and < 10¹²

**Units**
- **xx** — None.
- **Mmm** — None.
- **Snn** — 1/1000 of unit.

**Defaults**
- **xx** Missing: Error B.
- **mm** Missing: Error C.
- **nn** Missing: Error C.

**Description**
- FRM: For compatibility. No effect. Always 128 µsteps.
- FRS: This command sets the displacement length per full step in 1/1000 of unit.

**Returns**
If the sign “?” is used instead of **mm** or **nn**, this command returns the current programmed value.

**Errors**
- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.
- **H** — Execution not allowed in NOT REFERENCED state.
- **J** — Execution not allowed in DISABLE state.
- **K** — Execution not allowed in READY state.
- **L** — Execution not allowed in HOMING state.
- **M** — Execution not allowed in MOVING state.

**Example**
1FRS10 | Set controller #1 full step value to 10 milli-units.
HT — Set/Get HOME search type

Usage

Not Ref. | Config. | Disable | Ready | Motion
---|---|---|---|---

Syntax

xxHTnn or xxHT?

Parameters

Description

xx [int] — Controller address.

nn [int] — Home search type identifier.

Range

xx — 1 to 31

nn —

1 use current position as HOME.

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

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

Units

xx — None.

nn — None.

Defaults

xx Missing: Error B.

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

Description

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

NOTE

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

Returns

If the sign “?” is used instead of nn, this command returns the current programmed value.

Errors

A — Unknown message code or floating point controller address.

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.

H — Execution not allowed in NOT REFERENCED state.

J — Execution not allowed in DISABLE state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands

OH — Set/Get HOME search velocity.

OR — Execute HOME search.

OT — Set HOME search time-out.

Example

1HT1 | Set controller #1 HOME sequence to use current position.
### ID — Set/Get stage identifier

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

`xxIDnn` or `xxID?`

**Parameters**

- **Description**: `xx` [int] — Controller address.
- **nn** [string] — Stage identifier string.

**Range**

- `xx` — 1 to 31
- `nn` — 1 to 31 ASCII characters.

**Units**

- `xx` — None
- `nn` — None

**Defaults**

- `xx` Missing: Error B.
- `nn` Missing: Error C.

**Description**

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

In CONFIGURATION state, this command sets a new value for the stage identifier which can then be saved in the device's non-volatile memory with the PW command. It is also the default value that will be used unless a different value is set in DISABLE or READY state.

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

**Returns**

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

**Errors**

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

**Example**

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

`1ID URS100CC` | Set controller #1's stage identifier to: URS100CC.
**JR — Set/Get jerk time**

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>○</td>
<td>□</td>
<td>□</td>
<td>–</td>
</tr>
</tbody>
</table>

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

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>xx</code> [int] — Controller address.</td>
</tr>
<tr>
<td><code>nn</code> [float] — Jerk time value.</td>
</tr>
</tbody>
</table>

**Range**

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

**Units**

- `xx` — None.
- `nn` — Seconds.

**Defaults**

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

**Description**

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

In CONFIGURATION state, this command sets the value for the maximum jerk time which can then be saved in the controller’s non-volatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE or READY state.

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

**Returns**

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

**Errors**

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

**Rel. Commands**

- AC — Set/Get acceleration.
- VA — Set/Get velocity.

**Example**

`1JR0.05` | Set controller #1 jerk time to 0.05 seconds.
MM — Enter/Leave DISABLE state

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**  
xxMMnn or xxMM?

**Parameters**

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

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

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

**Defaults**
- **xx** Missing: Change to 0 (will forward this command to all controllers).
- Out of range: Error B.
- **nn** Missing: Error C.
- Out of range: Error C.

**Description**
- When the MM command is sent without preceding controller number or the controller number is 0, the MM command is executed on all controllers.  
  MM0 changes the controller’s state from READY to DISABLE. In DISABLE state the control loop is open and the motor is not powered.
  MM1 changes the controller’s state from DISABLE to READY. The controller’s set point position is set equal to its current position and the control loop gets closed (depending on the closed-loop state). Any residual error is cleared and the motor is powered.

**Returns**
- If the sign “?” is used instead of **nn**, this command returns the current state. Refer to the TS command section for the list of controller states.

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

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

**Example**
- MM0 | All controllers go to DISABLE state.
### OH — Set/Get HOME search velocity

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

`xxOHnn` or `xxOH?`

**Parameters**

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

**Range**

- **xx** — 1 to 31
- **nn** — Greater than 10^-6 and less than 10^12

**Units**

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

**Defaults**

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

**Description**

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

**Returns**

If the sign “?” is used instead of `nn`, this command returns the current programmed value.

**Errors**

- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.
- **H** — Execution not allowed in NOT REFERENCED state.
- **J** — Execution not allowed in DISABLE state.
- **K** — Execution not allowed in READY state.
- **L** — Execution not allowed in HOMING state.
- **M** — Execution not allowed in MOVING state.

**Rel. Commands**

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

**Example**

`1OH50` — Set controller #1 HOME search velocity to 50 units/s.
OR — Execute HOME search

Usage

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Syntax

xxOR

Parameters

Description

xx [int] — Controller address.

Range

xx — 1 to 31

Units

xx — None.

Defaults

xx Missing: Error B.

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

Description

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

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

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

Errors

A — Unknown message code or floating point controller address.

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.

E — home sequence already started.

I — Execution not allowed in CONFIGURATION state.

J — Execution not allowed in DISABLE state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands

HT — Set HOME search type.

OH — Set HOME search velocity.

OT — Set HOME search time-out.

Example

1OR | Execute HOME search with controller #1.
OT — Set/Get HOME search time-out

**Syntax**

```
xkOTnn or xkOT?
```

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int] — Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nn [float] — HOME time-out.</td>
</tr>
<tr>
<td>Range</td>
<td>xx — 1 to 31</td>
</tr>
<tr>
<td></td>
<td>nn — &gt; 1 and &lt; 1000</td>
</tr>
<tr>
<td>Units</td>
<td>xx — None.</td>
</tr>
<tr>
<td></td>
<td>nn — Seconds</td>
</tr>
</tbody>
</table>

**Defaults**

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

**Description**

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

**Returns**

If the sign “?” is used instead of nn, this command returns the current programmed value.

**Errors**

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- H — Execution not allowed in NOT REFERENCED state.
- J — Execution not allowed in DISABLE state.
- K — Execution not allowed in READY state.
- L — Execution not allowed in HOMING state.
- M — Execution not allowed in MOVING state.

**Rel. Commands**

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

**Example**

```
1OT2.2 | Set controller #1 HOME time-out to 2.2 seconds.
```
PA — Move absolute

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

`xxPA nn` or `xxPA?`

**Parameters**

**Description**

`xx` [int] — Controller address.

`nn` [float] — New absolute position.

**Range**

`xx` — 1 to 31

`nn` — ≥ SL and ≤ SR

**Units**

`xx` — None.

`nn` — Preset units.

**Defaults**

`xx` Missing: Error B.

Out of range: Error B.

`nn` Missing: Error C.

Out of range: Error C.

**Description**

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

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

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

**Returns**

If the sign “?” is used instead of `nn`, this command returns the target absolute position value.

**Errors**

A — Unknown message code or floating point controller address.

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.

G — Target position out of limits.

H — Execution not allowed in NOT REFERENCED state.

I — Execution not allowed in CONFIGURATION state.

J — Execution not allowed in DISABLE state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

**Rel. Commands**

PR — Move relative.

PT — Get motion time for a relative move.

TH — Get set-point position.

TP — Get current position.

**Example**

1PA2.2  |  Move positioner on controller #1 to absolute position 2.2 units.
## PR — Move relative

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxxPRnn</td>
<td>or xxPR?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Parameters**

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

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

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

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

**Description**

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

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

**Returns**

- If the sign “?” is used instead of `nn`, this command returns the target absolute position value.

**Errors**

- **A** — Unknown message code or floating point controller address.
- **B** — Controller address not correct.
- **C** — Parameter missing or out of range.
- **D** — Execution not allowed.
- **G** — Displacement out of limits.
- **H** — Execution not allowed in NOT REFERENCED state.
- **I** — Execution not allowed in CONFIGURATION state.
- **J** — Execution not allowed in DISABLE state.
- **L** — Execution not allowed in HOMING state.
- **M** — Execution not allowed in MOVING state.

**Rel. Commands**

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

**Example**

```
1PR2.2  | Move positioner on controller #1 to a new position 2.2 units away from the current position.
```
## PT — Get motion time for a relative move

### Syntax

```plaintext
xxPTnn
```

### Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nn [float]</td>
<td>Displacement.</td>
</tr>
<tr>
<td>Range</td>
<td>xx</td>
<td>1 to 31</td>
</tr>
<tr>
<td></td>
<td>nn</td>
<td>&gt; $10^4$ and $10^{12}$</td>
</tr>
<tr>
<td>Units</td>
<td>xx</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>nn</td>
<td>Preset units.</td>
</tr>
</tbody>
</table>

### Defaults

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

### Description

The PT commands help evaluating move times for an efficient program flow. When receiving the PT command, the controller computes and returns the time, in seconds, necessary to execute a relative move of displacement `nn` with the current working parameters (velocity, acceleration, etc.). The controller does not execute any displacement.

### Errors

- **A** Unknown message code or floating point controller address.
- **B** Controller address not correct.
- **C** Parameter missing or out of range.
- **D** Execution not allowed.
- **H** Execution not allowed in NOT REFERENCED state.
- **I** Execution not allowed in CONFIGURATION state.

### Related Commands

- **PA** Move absolute.
- **PR** Move relative.
- **TH** Get set-point position.
- **TP** Get current position.

### Example

- `1PT2.2` | *Get time to move positioner on controller #1 by 2.2 units.*
- `1PT0.25` | *Controller returns: 0.25 seconds.*
PW — Enter/Leave CONFIGURATION state

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Syntax: `xxPWnn` or `xxPW?`

Parameters:

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

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

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

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

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

PW0 checks all stage parameters, and if they are acceptable, saves them in the flash memory of the controller. After that, it changes the controller’s state from CONFIGURATION to NOT REFERENCED.

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

Returns:
If the sign “?” is used instead of `nn`, this command returns whether we are or not in the CONFIGURATION state.

Errors:
- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- J — Execution not allowed in DISABLE state.
- K — Execution not allowed in READY state.
- L — Execution not allowed in HOMING state.
- M — Execution not allowed in MOVING state.

Related Commands:
MM — Enter/Leave DISABLE state.

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

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

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

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

**Syntax**

xxRS

**Parameters**

**Description**

xx [int] — Controller address.

**Range**

xx — 1 to 31

**Units**

xx — None.

**Defaults**

xx Missing: Error B.

Out of range: Error B.

**Description**

The RS command issues a hardware reset of the controller, equivalent to a power cycle. To go from DISABLE or READY state to CONFIGURATION state, it is also needed to reset the controller with the RS command, and then to change the controller’s state with the PW1 command from NOT REFERENCED to CONFIGURATION.

**Errors**

A — Unknown message code or floating point controller address.

B — Controller address not correct.

D — Execution not allowed.

**Example**

1RS | Reset controller #1.
RS## — Reset controller’s address

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

**Parameters**

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

**Range**
xx — 0 to 31

**Units**
xx — None.

**Defaults**
xx Missing: Change to 0 (will forward this command to all controllers).
Out of range: Error B.

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

**Returns**

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

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

**Example**
RS## | Reset all controllers’ addresses to 1.
**SA — Set/Get controller’s RS422 address**

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**
xxSAnn or xxSA?

**Parameters**

**Description**

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

**Range**

| xx | — | 1 |
| nn | — | 1 to 31 |

**Units**

| xx | — | None. |
| nn | — | None. |

**Defaults**

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

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

**Returns**
If the sign “?” is used instead of nn, this command returns the current programmed value.

**Errors**

| A | — | Unknown message code or floating point controller address. |
| B | — | Controller address not correct. |
| C | — | Parameter missing or out of range. |
| D | — | Execution not allowed. |
| H | — | Execution not allowed in NOT REFERENCED state. |
| J | — | Execution not allowed in DISABLE state. |
| K | — | Execution not allowed in READY state. |
| L | — | Execution not allowed in HOMING state. |
| M | — | Execution not allowed in MOVING state. |

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

**Example**

1SA3 | Set controller’s RS422 address to 3.
SE — Configure/Execute simultaneous started move

Usage
Not Ref. Config. Disable Ready Motion

- - - ● -

Syntax  xxSEnn, xxSE? or SE

Parameters

Description  xx [int] — Controller address.

nn [float] — New target position.

Range  xx — 0 to 31

nn — ≥ SL and ≤ SR

Units  xx — None.

nn — Preset units.

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

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

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

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

The xxSEnn command is only accepted in READY state, AND when the new target position is higher or equal to the negative software limit (SL), AND lower or equal to the positive software limit (SR). The controller always rounds the new target position to the closest micro-step position.

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

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

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

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.

H — Execution not allowed in NOT REFERENCED state.

I — Execution not allowed in CONFIGURATION state.

J — Execution not allowed in DISABLE state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands  PA / PR — Move absolute / relative.

TH — Get set-point position.

TP — Get current position.

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

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

SE | All controllers start their programmed move, if any.
SL — Set/Get negative software limit

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>○</td>
<td>□</td>
<td>□</td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

xxSLnn or xxSL?

**Parameters**

**Description**

- xx [int] — Controller address.
- nn [float] — Negative software limit.

**Range**

- xx — 1 to 31
- nn — $> -10^{12}$ and $\leq 0$

**Units**

- xx — None.
- nn — Preset units.

**Defaults**

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

**Description**

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

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

The software limits are useful to limit the travel range of a positioner. There is no possibility to disable software limits. For an almost infinite motion, for instance with a rotative stage, set the lowest possible value, which is -999999930400.

**Returns**

If the sign “?” is used instead of nn, this command returns the current programmed value.

**Errors**

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

**Rel. Commands**

SR — Set positive software limit.

**Example**

1SL-100 | Set controller #1 negative software limit to –100 units.
SR — Set/Get positive software limit

Syntax: xxSRnn or xxSR?

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

Description:
- In CONFIGURATION state, this command sets the positive software limit which can then be saved in the controller’s non-volatile memory using the PW command. It is also the default value that will be used unless a different value is set in DISABLE or READY state.
- In DISABLE or READY state, this command allows setting a new working parameter for the positive software limit. It must be greater or equal to the set-point position. This value is not saved in the controller’s memory and will be lost after reboot.
- The software limits are useful to limit the travel range of a positioner. There is no possibility to disable software limits. For an almost infinite motion, for instance with a rotative stage, set the highest possible value, which is 999999930400.

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

Returns:
- If the sign “?” is used instead of **nn**, this command returns the current programmed value.

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

Related Commands:
- SL — Set negative software limit.

Example:
- 1SR100 | Set controller #1 positive software positive to 100 units.
ST — Stop motion

Syntax

```
[xx]ST
```

Parameters

- **Description**: xx [int] — Controller address.
- **Range**: xx — 0 to 31
- **Units**: xx — None.
- **Defaults**: xx Missing: Change to 0 (will forward this command to all controllers).
  Out of range: Error B.

Description

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

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

Errors

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

Example

```
ST | Stop moves on all controllers.
```
TB — Get command error string

<table>
<thead>
<tr>
<th>Syntax</th>
<th>xxTBnn</th>
</tr>
</thead>
</table>

**Parameters**
- **Description**: xx [int] — Controller address.
- **Range**: xx — 1 to 31
- **Units**: xx — None.
- **Defaults**: xx Missing: Error B.
  - Out of range: Error B.
- **nn [char]** — Error code (refer to TE command).
- **Units**: nn — None.
- **Defaults**: nn Missing: Returns explanation of current error as a literal string.
  - Out of range: Error C.

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

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

**Rel. Commands**: TE — Get last command error.

**Example**

1TB@ | Get explanation to error code @.

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

Syntax: xxTE

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

Description:
The TE command returns the currently recorded error. When a command is not executable or fails, an error is recorded. This error can be read with the TE command. After the execution of a TE command, the error buffer is erased and another TE command will return @, which means "No error". When a new command error is generated before the previous command error is read, the new command's error will overwrite the currently memorized error.

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

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

Related Commands:
- TB — Get command error string.

Example:

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

List of errors and corresponding strings (see TB command):
- @ — No error.
- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Command not allowed.
- E — Home sequence already started.
- G — Displacement out of limits.
- H — Command not allowed in NOT REFERENCED state.
- I — Command not allowed in CONFIGURATION state.
- J — Command not allowed in DISABLE state.
- K — Command not allowed in READY state.
- L — Command not allowed in HOMING state.
- M — Command not allowed in MOVING state.
- N — Current position out of software limit.
- S — Communication Time Out.
- U — Error during EEPROM access.
- V — Error during command execution.
TH — Get set-point position

Usage

Not Ref. Config. Disable Ready Motion

● ● ● ● ●

Syntax

xxTH

Parameters

Description

xx [int] — Controller address.

Range

xx — 1 to 31

Units

xx — None.

Defaults

xx Missing: Error B.

Out of range: Error B.

Description

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

Errors

A — Unknown message code or floating point controller address.

B — Controller address not correct.

D — Execution not allowed.

Rel. Commands

TP — Get current position.

Example

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

ITH0 | Controller returns: set-point position = 0 units.
**TP — Get current position**

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Parameters**

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

**Description**

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

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

**Errors**

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

**Rel. Commands**

- **TH** — Get set-point position.
- **TS** — Get positioner error and controller state.

**Example**

- **1TP** | *Get current position of controller #1.*
- **1TP0** | *Controller returns: actual position = 0 units.*
TS — Get positioner error and controller state

Syntax: `xxTS`

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

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

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

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

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

NOTE

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

Examples:
- Error map 0000 = No errors
- Error map 0002 = Positive end of run
- Error map 0048 = Homing time out, RMS current limit
**Controller states (cf):**

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

---

**NOTE**

The positioner error gets updated periodically, approx. every 1 ms. The TS command reads the positioner error and clears it at the same time (same as what the command TE does with command errors). So when launching the TS command, it is important to process the TS feedback accordingly.

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

**Rel. Commands**

- **TE**: Get last command error.

**Example**

```plaintext
1TS        | Get error and state of controller #1.
1TS000000A | Controller returns: no errors and state is NOT REFERENCED from reset.
```
### VA — Set/Get velocity

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxVAnn or xxVA?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameters</td>
<td>xx [int] — Controller address.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>nn [float] — Velocity value.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>In CONFIGURATION state, this command sets the maximum (i.e. cruise) velocity value which can then be saved in the controller’s non-volatile memory using the PW command. This should be the maximum velocity that can be applied to the mechanical system. It is also the default velocity that will be used for all moves unless a lower value is set in DISABLE or READY state.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>In DISABLE or READY state, this command sets the velocity used for all subsequent moves. Its value can be up to the programmed value set in the CONFIGURATION state. This value is not saved in the controller’s memory and will be lost after reboot.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Returns</td>
<td>If the sign “?” is used instead of nn, this command returns the current value for the state in which the controller is (either CONFIGURATION or DISABLE/READY).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Errors</td>
<td>A — Unknown message code or floating point controller address.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B — Controller address not correct.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>C — Parameter missing or out of range.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>D — Execution not allowed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H — Execution not allowed in NOT REFERENCED state.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>L — Execution not allowed in HOMING state.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M — Execution not allowed in MOVING state.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rel. Commands</td>
<td>AC — Set/Get acceleration.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>JR — Set/Get jerk time.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>1VA50</td>
<td>Set controller #1maximum velocity to 50 units/s.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### VE — Get controller revision information

<table>
<thead>
<tr>
<th>Usage</th>
<th>Not Ref.</th>
<th>Config.</th>
<th>Disable</th>
<th>Ready</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>xxVE</td>
<td><strong>xx</strong> [int] — Controller address.</td>
<td><strong>nn</strong> [string] — Action.</td>
<td><strong>xx</strong> — 1 to 31</td>
<td><strong>xx</strong> — None.</td>
</tr>
</tbody>
</table>

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

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

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

**Example**

1VE | Get controller #1 revision information.

1VE FC family controller 2.0.0 | Controller returns revision number
ZT — Get all configuration parameters

Syntax xxZT

Parameters

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

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

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

Example

1ZT | Get controller #1 configuration data.
1PW1
1AC320.000000
1BA0.000000
...
1VA80.000000
1PW0
5.0 Maintenance

5.1 Maintenance

The FCL series stages require no particular maintenance and there are no user-serviceable parts or user adjustments to be made. However, as with other precision mechanical stages, care must be taken for handling, operation and storage.

CAUTION

FCL stages must be used or stocked in a clean environment to avoid dust, humidity, solvents or other substances.

RECOMMENDATION

It is recommended to return your stage to Newport's After Sales Service after every 18 months of use for lubrication and inspection.

If your FCL stage is mounted on a workstation and cannot be easily removed, please contact Newport's After Sales Service for further instructions.

5.2 Repair

CAUTION

Never attempt to disassemble any part of the stage unless it is described in this manual.

Disassembly of components, unless instructed by Newport, may lead to failure or malfunction of the stage.

If you observe any malfunction in your stage, please contact us immediately to arrange a repair service.

CAUTION

Any attempt to disassemble or repair the stage without authorization will void your warranty.
5.3 Troubleshooting

A list of the most common problems and their corrective actions is provided in the table below. Use this as a reference but remember that problems may be related to other operator errors and may be resolved by other simple solutions.

<table>
<thead>
<tr>
<th>Problem (Error Message)</th>
<th>Cause</th>
<th>Corrective Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please verify the stage is powered, then relaunch the user interface</td>
<td>Power cord not plugged in.</td>
<td>Plug the power cord in the appropriate outlet. Verify the 3-pin connector is seated properly.</td>
</tr>
<tr>
<td></td>
<td>Power supply cable was connected to outlet before the USB cable was connected to PC.</td>
<td>The power supply cable must remain fully disconnected from the outlet when the USB cable is connected to the PC. Unplug both the USB cable and the power supply cable. Then, make the USB connection first.</td>
</tr>
<tr>
<td>Bad connection</td>
<td></td>
<td>Check the integrity of cables and make sure each connector is fully secured.</td>
</tr>
<tr>
<td>Access to com port is denied.</td>
<td>Com port of the PC may be already used by another program.</td>
<td>Make sure to terminate all other programs (ie. LabVIEW driver) before launching the GUI.</td>
</tr>
<tr>
<td>Applet launching unsuccessful. Access to the log file denied.</td>
<td>When you are not the Administrator of the PC, Window 7 disables writing in the Program Files folder.</td>
<td>Obtain the full Admin right in the PC. Or simply copy the “Motion Control” folder from “Program Files” to “My Documents” folder.</td>
</tr>
</tbody>
</table>
Service Form

Name: ____________________________________________  Return authorization #: ________________________________________

Company: ____________________________________________

Address: ____________________________________________

Country: ____________________________________________

P.O. Number: ________________________________________

Item(s) Being Returned: ____________________________________________

Model#: ____________________________________________  Serial #: _______________________________________________

Description: ________________________________________________________________________________________________________

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

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________

__________________________________________________________________________________________________________________
North America & Asia
Newport Corporation
1791 Deere Ave.
Irvine, CA 92606, USA
Sales
Tel.: (800) 222-6440
e-mail: sales@newport.com
Technical Support
Tel.: (800) 222-6440
e-mail: tech@newport.com
Service, RMAs & Returns
Tel.: (800) 222-6440
e-mail: 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