CONEX-PP

Single-Axis Intelligent Stepper Motor Controller/Driver

Newport® Command Interface Manual

V1.0.x
Table of Contents

1.0 Introduction .................................................................................................. 1
  1.1 Purpose ........................................................................................................ 1
  1.2 Overview ...................................................................................................... 1

2.0 Programming ................................................................................................ 2
  2.1 State Diagram ............................................................................................ 2
  2.2 Command Syntax ....................................................................................... 3
  2.3 Command Execution Time ......................................................................... 3
  2.4 Command Set ............................................................................................. 4
      AC — Set/Get acceleration ............................................................................ 6
      BA — Set/Get backlash compensation .......................................................... 7
      BH — Set/Get hysteresis compensation ........................................................ 8
      FR — Set/Get stepper motor configuration .................................................. 9
      HT — Set/Get HOME search type ............................................................... 10
      ID — Set/Get stage identifier .................................................................... 11
      JR — Set/Get jerk time .............................................................................. 12
      MM — Enter/Leave DISABLE state ............................................................. 13
      OH — Set/Get HOME search velocity ......................................................... 14
      OR — Execute HOME search .................................................................... 15
      OT — Set/Get HOME search time-out ........................................................ 16
      PA — Move absolute .................................................................................. 17
      PR — Move relative .................................................................................... 18
      PT — Get motion time for a relative move .................................................. 19
      PW — Enter/Leave CONFIGURATION state .............................................. 20
      RS — Reset controller .............................................................................. 21
      RS## — Reset controller’s address ............................................................... 22
      SA — Set/Get controller’s RS422 address ................................................... 23
      SE — Configure/Execute simultaneous started move .................................. 24
      SL — Set/Get negative software limit .......................................................... 26
      SR — Set/Get positive software limit .......................................................... 27
      ST — Stop motion ....................................................................................... 28
      TB — Get command error string ................................................................ 29
      TE — Get last command error ................................................................... 30
      TH — Get set-point position ...................................................................... 31
      TP — Get current position ........................................................................ 32
      TS — Get positioner error and controller state ........................................... 33
VA — Set/Get velocity ........................................................................................................ 35
VE — Get controller revision information ........................................................................... 36
ZT — Get all configuration parameters ............................................................................. 37

Service Form ....................................................................................................................... 39
1.0 Introduction

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

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

NOTE
Each function name is defined with the command code “AA”.
For each command function, refer to the CONEX-PP programmer’s manual.
2.0 Programming

2.1 State Diagram

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

Actions in each state when End of Runs is encountered

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

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

Command format

```
nn AA xx
```

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

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

Blank spaces

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

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

Decimal separator

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

Command terminator

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

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

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

2.3 Command Execution Time

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

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

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

**Command format**

```
nn  A  x
```

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

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

Examples:

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

Not every command can be executed in all states of the CONEX-PP and some commands have different meaning in different states. It is therefore important to understand the state diagram of the controller, see section 2.1.
<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>●</td>
<td>Reset controller</td>
</tr>
<tr>
<td>RS##</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Reset controller’s address to 1</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>●</td>
<td>Get command error string</td>
</tr>
<tr>
<td>TE</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get last command error</td>
</tr>
<tr>
<td>TH</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get set–point position</td>
</tr>
<tr>
<td>TP</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get current position</td>
</tr>
<tr>
<td>TS</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get positioner error and controller state</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>●</td>
<td>Get controller revision information</td>
</tr>
<tr>
<td>ZT</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Get all axis parameters</td>
</tr>
</tbody>
</table>

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

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

**Disable**
Corresponds to the DISABLE state.

**Ready**
Corresponds to the READY state.

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

- **□** Changes configuration parameters. Those changes will be stored in the controller’s memory with the PW1 command and remain available after switching off the controller.

- **●** Changes working parameters only. Those changes will get lost when switching off the controller.

- **–** Accepted command.

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

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

Usage

<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  

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

Parameters

**Description**

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

**Range**

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

**Units**

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

Syntax xxBAnn or xxBA?

Parameters

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

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

| Units | xx | — | None |
| nn | — | Preset units |

Defaults

| xx | Missing: Error B. |
| nn | 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

\[ \text{xBHnn} \] or \[ \text{xBH?} \]

Parameters

- **Description**
  - **xx** [int] — Controller address.
  - **nn** [float] — Hysteresis 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.
  - Floating point: Error A.
  - **nn** Missing: Error C.
  - Out of range: Error C.

Description

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

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

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

NOTE

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

Returns

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

Errors

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

Rel. Commands

BA — Set backlash compensation.

Example

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

<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>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Syntax**

xxFRSnn, xxFRM? or xxFRS?

**Parameters**

**Description**

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

**Range**

- **xx** — 1 to 31
- **mm** — > 0 and ≤ 2000
- **nn** — > 10⁴ and < 10¹²

**Units**

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

**Defaults**

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

**Description**

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

**Returns**

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

**Errors**

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

**Example**

IFRS10 | *Set controller #1 full step value to 10 milli-units.*
HT — Set/Get HOME search type

Syntax xxHTnn or xxHT?

Parameters

Description xx [int] — Controller address.

nn [int] — Home search type identifier.

Range xx — 1 to 31

nn —

1 use current position as HOME.

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

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

Units xx — None.

nn — None.

Defaults xx Missing: Error B.

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

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

NOTE

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

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

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

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.

H — Execution not allowed in NOT REFERENCED state.

J — Execution not allowed in DISABLE state.

K — Execution not allowed in READY state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands OH — Set/Get HOME search velocity.

OR — Execute HOME search.

OT — Set HOME search time-out.

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

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

Syntax  xxIDnn or xxID?

Parameters

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

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

Units  xx — None
  nn — None

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

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

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

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

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

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

Example  1ID? | Get stage identifier for controller #1.
         1ID URS100CC | Set controller #1’s stage identifier to: URS100CC.
**JR — Set/Get jerk time**

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

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

**Range**
- **xx** — 1 to 31
- **nn** — > 0.001 and < 10\(^2\)

**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

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int] — Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nn [int] — Whether to enter (1) or leave (0) the DISABLE state.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Range xx</th>
<th>— 0 to 31</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>— 0 changes state from READY to DISABLE.</td>
</tr>
<tr>
<td></td>
<td>1 changes state from DISABLE to READY.</td>
</tr>
</tbody>
</table>

### Units

<table>
<thead>
<tr>
<th>xx</th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>None.</td>
</tr>
</tbody>
</table>

### Defaults

<table>
<thead>
<tr>
<th>xx</th>
<th>Missing: Change to 0 (will forward this command to all controllers).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Out of range: Error B.</td>
</tr>
<tr>
<td>nn</td>
<td>Missing: Error C.</td>
</tr>
<tr>
<td></td>
<td>Out of range: Error C.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>A</th>
<th>Unknown message code or floating point controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Controller address not correct.</td>
</tr>
<tr>
<td>C</td>
<td>Parameter missing or out of range.</td>
</tr>
<tr>
<td>D</td>
<td>Execution not allowed.</td>
</tr>
<tr>
<td>H</td>
<td>Execution not allowed in NOT REFERENCED state.</td>
</tr>
<tr>
<td>I</td>
<td>Execution not allowed in CONFIGURATION state.</td>
</tr>
<tr>
<td>L</td>
<td>Execution not allowed in HOMING state.</td>
</tr>
<tr>
<td>M</td>
<td>Execution not allowed in MOVING state.</td>
</tr>
</tbody>
</table>

### Rel. Commands

PW — Enter/leave CONFIGURATION state.

### Example

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

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>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Syntax

xxOHnn or xxOH?

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>HOME search velocity.</td>
</tr>
</tbody>
</table>

Range

<table>
<thead>
<tr>
<th>xx</th>
<th>1 to 31</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>&gt; 10^4 and &lt; 10^12</td>
</tr>
</tbody>
</table>

Units

<table>
<thead>
<tr>
<th>xx</th>
<th>None.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>Preset units/s.</td>
</tr>
</tbody>
</table>

Defaults

<table>
<thead>
<tr>
<th>xx</th>
<th>Missing: Error B.</th>
</tr>
</thead>
<tbody>
<tr>
<td>nn</td>
<td>Out of range: Error B.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>nn</th>
<th>Missing: Error C.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Out of range: Error C.</td>
</tr>
</tbody>
</table>

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

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.

- Missing: Error C.

- Out of range: Error C.

Description

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

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

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

Errors

- A — Unknown message code or floating point controller address.
- B — Controller address not correct.
- C — Parameter missing or out of range.
- D — Execution not allowed.
- E — Home sequence already started.
- I — Execution not allowed in CONFIGURATION state.
- J — Execution not allowed in DISABLE state.
- K — Execution not allowed in READY state.
- L — Execution not allowed in HOMING state.
- M — Execution not allowed in MOVING state.

Rel. Commands

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

Example

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

Syntax: `xxOTnn` or `xxOT?`

Parameters:
- **Description:**
  - `xx` [int] — Controller address.
- **Range:**
  - `xx` — 1 to 31
  - `nn` — > 1 and < 1000
- **Units:**
  - `xx` — None.
  - `nn` — Seconds
- **Defaults:**
  - 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.

Related 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

Syntax: \texttt{xxPAnn} or \texttt{xxPA?}

Parameters:

\begin{itemize}
  \item **Description**: \texttt{xx} [int] — Controller address.
  \item **Range**\texttt{nn} [float] — New absolute position.
  \item **Units**: \texttt{xx} — None.
  \item **Defaults**: \texttt{nn} — Preset units.
\end{itemize}

Defaults:

\begin{itemize}
  \item Missing: Error B.
  \item Out of range: Error C.
\end{itemize}

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 \texttt{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 \texttt{nn}, this command returns the target absolute position value.

Errors:

\begin{itemize}
  \item A — Unknown message code or floating point controller address.
  \item B — Controller address not correct.
  \item C — Parameter missing or out of range.
  \item D — Execution not allowed.
  \item G — Target position out of limits.
  \item H — Execution not allowed in NOT REFERENCED state.
  \item I — Execution not allowed in CONFIGURATION state.
  \item J — Execution not allowed in DISABLE state.
  \item L — Execution not allowed in HOMING state.
  \item M — Execution not allowed in MOVING state.
\end{itemize}

Rel. Commands:

\begin{itemize}
  \item PR — Move relative.
  \item PT — Get motion time for a relative move.
  \item TH — Get set-point position.
  \item TP — Get current position.
\end{itemize}

Example: \texttt{1PA2.2} | \textit{Move positioner on controller \#1 to absolute position 2.2 units.}
PR — Move relative

Syntax xxxPRnn or xxxPR?

Parameters

Description

xx [int] — Controller address.

nn [float] — Displacement.

Range

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

Units

xx — None.

nn — Preset units.

Defaults

xx Missing: Error B.

Out of range: Error B.

nn Missing: Error C.

Out of range: Error C.

Description

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

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

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

Returns

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

Errors

A — Unknown message code or floating point controller address.

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.

G — Displacement out of limits.

H — Execution not allowed in NOT REFERENCED state.

I — Execution not allowed in CONFIGURATION state.

J — Execution not allowed in DISABLE state.

L — Execution not allowed in HOMING state.

M — Execution not allowed in MOVING state.

Rel. Commands

PA — Move absolute.

PT — Get motion time for a relative move.

TH — Get set-point position.

TP — Get current position.

Example

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

Syntax  
xPTnn

Parameters

Description  
xx [int] — Controller address.

nn [float] — Displacement.

Range  
xx — 1 to 31

nn — > 10^6 and < 10^12

Units  
xx — None.

nn — Preset units.

Defaults  
xx — Missing: Error B.

Out of range: Error B.

nn — Missing: Error C.

Out of range: Error C.

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

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

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

B — Controller address not correct.

C — Parameter missing or out of range.

D — Execution not allowed.

H — Execution not allowed in NOT REFERENCED state.

I — Execution not allowed in CONFIGURATION state.

Rel. Commands  
PA — Move absolute.

PR — Move relative.

TH — Get set-point position.

TP — Get current position.

Example  
1PT2.2  |  Get time to move positioner on controller #1 by 2.2 units.

1PT0.25  |  Controller returns: 0.25 seconds.
PW — Enter/Leave CONFIGURATION state

<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>xxPWnn   or xxPW?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>nn [int]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Controller address.</td>
<td>Whether to enter (1) or leave (0) the CONFIGURATION state.</td>
</tr>
<tr>
<td>Range</td>
<td>xx — 1 to 31</td>
<td>nn — 1: Go from NOT REFERENCED state to CONFIGURATION state. 0: Go from CONFIGURATION state to NOT REFERENCED state.</td>
</tr>
<tr>
<td>Units</td>
<td>xx — None.</td>
<td>nn — None.</td>
</tr>
<tr>
<td>Defaults</td>
<td>xx Missing: Error B.</td>
<td>nn Missing: Error C.</td>
</tr>
</tbody>
</table>

Description

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

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

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

Returns

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

Errors

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

Rel. Commands

MM — Enter/Leave DISABLE state.

Example

1PW1 | Changes controller #1 to CONFIGURATION state.

NOTE

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

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

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

<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>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
</tr>
</tbody>
</table>

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

**Parameters**

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

**Range**  
**xx** — 0 to 31

**Units**  
**xx** — None.

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

Out of range: Error B.

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

**Returns**

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

B — Controller address not correct.

D — Execution not allowed.

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

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

## Syntax

```
xxSAnn or xxSA?
```

## Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>New controller address (a.k.a. RS422 address).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>xx</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>nn</td>
<td>1 to 31</td>
</tr>
<tr>
<td>Units</td>
<td>xx</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td>nn</td>
<td>None.</td>
</tr>
</tbody>
</table>

## Defaults

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

## Description

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

## Returns

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

## Errors

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

## Related Commands

- RS## — Reset controller's address.

## Example

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

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

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

| Description  | xx [int] — Controller address. |
|             | nn [float] — Positive software limit. |
| Range       | xx — 1 to 31 |
|             | nn — ≥ 0 and < 10¹² |
| Units       | xx — None. |
|             | nn — Preset units. |

Defaults

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

Description

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

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

The software limits are useful to limit the travel range of a positioner. There is no possibility to disable software limits. For an almost infinite motion, for instance with a rotative stage, set the highest possible value, which is 99999993040.

Returns

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

Errors

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

Rel. Commands

SL — Set negative software limit.

Example

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

## 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

Syntax: \texttt{xTnn}

Parameters:

- **xx**: [int] — Controller address.
- **nn**: [char] — Error code (refer to TE command).

Defaults:
- xx: Missing: Error B.
- Out of range: Error B.
- nn: Missing: Returns explanation of current error as a literal string.
- Out of range: Error C.

Description:
The TB command returns a literal character string that explains the meaning of the error code \texttt{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.

Related 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:
- **xx** [int] — Controller address.
- **Range**: xx — 1 to 31
- **Units**: xx — None.
- **Defaults**:
  - 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**

**Syntax**

```
xxTH
```

**Parameters**

**Description**

```
xx [int] — Controller address.
```

**Range**

```
xx — 1 to 31
```

**Units**

```
xx — None.
```

**Defaults**

```
x   Missing: Error B.
     Out of range: Error B.
```

**Description**

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

**Errors**

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

**Rel. Commands**

```
TP — Get current position.
```

**Example**

```
1TH | Get set-point position of controller #1.
1TH0 | Controller returns: set-point position = 0 units.
```
## TP — Get current position

<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**

```
xTP
```

**Parameters**

<table>
<thead>
<tr>
<th>Description</th>
<th>xx [int]</th>
<th>Controller address.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>xx</td>
<td>1 to 31</td>
</tr>
<tr>
<td>Units</td>
<td>xx</td>
<td>None.</td>
</tr>
<tr>
<td>Defaults</td>
<td>xx</td>
<td>Missing: Error B.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Out of range: Error B.</td>
</tr>
</tbody>
</table>

**Description**

The TP command returns the value of the current position. This is the position where the positioner actually is. In MOVING state, this value always changes. In READY state, this value should be equal or very close to the set-point and target position. Together with the TS command, the TP command helps evaluating whether a motion has completed.

**Errors**

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

**Rel. Commands**

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

**Example**

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

1TP0 | Controller returns: actual position = 0 units.
```
**TS — Get positioner error and controller state**

**Usage**
- Not Ref.
- Config.
- Disable
- Ready
- Motion

**Syntax**
```
xxTS
```

**Parameters**

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

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

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

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

<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>1011</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>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 1</td>
<td>Not used</td>
<td>Not used</td>
<td>Not used</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Not used</td>
<td>Driver overheating</td>
<td>Driver fault</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Not used</td>
<td>Not used</td>
<td>Not used</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Not used</td>
<td>No parameters in memory</td>
<td>Not used</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Not used</td>
<td>Homing time out</td>
<td>Not used</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Not used</td>
<td>Newpport reserved (MZ status)</td>
<td>Not used</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Not used</td>
<td>RMS current limit</td>
<td>Not used</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>Positive end of run</td>
<td>Negative end of run</td>
<td></td>
</tr>
</tbody>
</table>

**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>

<table>
<thead>
<tr>
<th>Rel. Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TE</td>
<td>Get last command error.</td>
</tr>
</tbody>
</table>

**Example**

1TS | Get error and state of controller #1.

1TS00000A | Controller returns: no errors and state is NOT REFERENCED from reset.
**VA — Set/Get velocity**

<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**

`xxVA nn` or `xxVA?`

**Parameters**

**Description**

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

**Range**

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

**Units**

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

**Defaults**

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

**Description**

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

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

**Returns**

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

**Errors**

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

**Rel. Commands**

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

**Example**

`1VA50` | *Set controller #1 maximum velocity to 50 units/s.*
VE — Get controller revision information

Syntax: xxVE

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

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

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

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

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

Usage

<table>
<thead>
<tr>
<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>
</tr>
</tbody>
</table>

Syntax

xxZT

Parameters

Description

xx [int] — Controller address.

Range

xx — 1 to 31

Units

xx — None.

Defaults

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

Description

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

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

Errors

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

Example

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

Your Local Representative
Tel.: __________________
Fax: ___________________

Name: _________________________________________________  Return authorization #: ____________________________________
Company:_______________________________________________
(Please obtain prior to return of item)
Address: ________________________________________________  Date: __________________________________________________
Country: ________________________________________________  Phone Number: _________________________________________
P.O. Number: ____________________________________________  Fax Number: __________________________________________

Item(s) Being Returned: ____________________________________  Serial #: _________________________________________
Model#: ________________________________________________

Description: ____________________________________________________________________________________________
Reasons of return of goods (please list any specific problems): __________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
________________________________________________________________________________________________________
Visit Newport Online at:
www.newport.com

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

Europe
MICRO-CONTROLE Spectra-Physics S.A.S
9, rue du Bois Sauvage
91055 Évry CEDEX
France
Sales
Tel.: +33 (0)1.60.91.68.68
e-mail: france@newport.com
Technical Support
e-mail: tech_europe@newport.com
Service & Returns
Tel.: +33 (0)2.38.40.51.55