
Coordinate Systems for Hexapod Microrobots

When working with a Hexapod microrobot ("Hexapod"), it may be necessary for certain applications to define custom coordinate systems and use these instead of default coordinate systems. This Technical Note describes the fundamental principles for working with user-defined coordinate systems and the commands provided for this purpose.

Other applicable documents: Manual for the C-887 Hexapod controller, manual for the used H-xxx Hexapod microrobot

Contents

Overview of Working with User-Defined Coordinate Systems.....	2
Groups and Types of Coordinate Systems, HEXAPOD Coordinate System.....	2
The Work-and-Tool Concept	3
Applying the Work-and-Tool Concept.....	3
Consideration of Coordinate Systems of the KSD and KSF Type from the "Work" and "Tool" Perspective.....	4
Defining, Copying and Deleting Coordinate Systems	5
Linking Coordinate Systems.....	5
Enabling/Disabling Coordinate Systems.....	6
Backing Up and Restoring Settings	7
Commanding Motion	7
PC Software from PI.....	8
Examples of User-Defined Coordinate Systems.....	9
Optics: The Optical Axis is to be the Z Axis.....	9
Optics: Scanning in an Inclined Plane.....	10
User-Defined "Home" Position	10
Applying the Work-and-Tool Concept.....	11
GCS Commands	11
Command Overview	12
Naming Conventions for Coordinate Systems.....	13
Command Descriptions.....	13

Overview of Working with User-Defined Coordinate Systems

Basic steps:

1. Define a coordinate system suitable for the application concerned
2. Optional: Link the defined coordinate system to other existing coordinate systems so that their properties are inherited by the new coordinate systems
3. Enable the defined coordinate system so that its properties become effective for the work with the Hexapod
4. Optional: Save the definition and the activation state of the coordinate system so that they are retained when the Hexapod controller is switched off or rebooted
5. Command motion

If necessary, the default settings for coordinate systems can be restored.

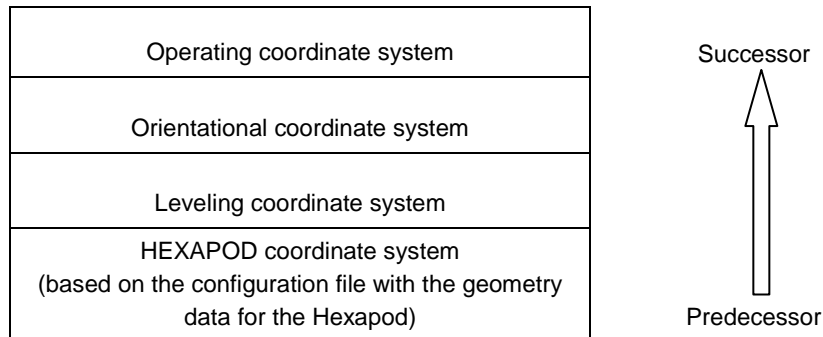
Groups and Types of Coordinate Systems, HEXAPOD Coordinate System

The various coordinate system types can be divided into groups. Precisely one coordinate system or precisely one coordinate system combination is enabled for each group. The types and coordinate systems which **cannot** be defined by the user appear in the table below in **bold**.

Group	Types	Purpose	Default Settings*
Operating coordinate system	ZERO , KSD, KSF When applying the work-and-tool concept (p. 3): KSW ("work" coordinate system), KST ("tool" coordinate system) In this case, a combination of enabled work and enabled tool coordinate system is used; for more details, see p. 3	Adapting the position display, the direction of motion and the center of rotation for the moving platform of the Hexapod to the application concerned	ZERO coordinate system of the ZERO type is enabled
Oriental coordinate system	KSB(PI) , KSB(USER)	Permanent modification to the direction of the X and/or Y and/or Z axes (e.g., if Z always has to point towards the default X axis)	PI_Base coordinate system of the KSB(PI) type is enabled
Leveling coordinate system	KLD(PI) , KLF(PI) , KLD(USER), KLF(USER)	Permanent correction of errors in the Hexapod alignment (e.g., due to faulty mounting)	PI_Levelling coordinate system of the KLD(PI) type is enabled

*Default settings are defined by PI and can be restored at any time, see „Backing Up and Restoring Settings“ (p. 7).

Each coordinate system is part of at least one chain. Basic structure of chains:



The HEXAPOD coordinate system, which does not belong to any of the aforementioned groups, is the starting point for all coordinate system chains and as such dictates the fundamental properties of all coordinate systems. HEXAPOD is based on the configuration file with the geometry data for the Hexapod. The dimensional drawings in the manuals for the H-xxx Hexapod microrobots each show the position of the HEXAPOD coordinate system.

The properties of coordinate system and pivot point described in the manuals for the H-xxx Hexapod microrobots, and the behavior for translations and rotations which is described there corresponds to the properties and the behavior when the ZERO operating coordinate system is enabled (default setting).

Based on HEXAPOD, the orientational and leveling coordinate systems adapt fundamental properties of the enabled operating coordinate system, and in most applications their custom definition and enabling is not necessary at all or only once.

Default and user-defined coordinate systems are **always right-handed** systems. It is **not** possible to convert a right-handed system to a left-handed system.

The Work-and-Tool Concept

Applying the Work-and-Tool Concept

The work-and-tool concept uses a combination of two enabled operating coordinate systems ("work" coordinate system and "tool" coordinate system). Generally, the combination consists of one enabled coordinate system of the KST (tool coordinate system) type and one of the KSW (work coordinate system) type.

The work-and-tool concept is applied when a coordinate system of the KST and/or KSW type is enabled. The current position of the moving platform of the Hexapod is then considered the position of the tool coordinate system in the work coordinate system (query with the POS? command, see the manual of the Hexapod controller).

Work coordinate systems of the KSW type and tool coordinate systems of the KST type can be placed and aligned in space arbitrarily.

The X, Y, and Z axes of the tool coordinate system are always permanently connected to the moving platform of the Hexapod, i.e., the tool coordinate system moves together with the platform.

The X, Y, and Z axes of the work coordinate system are always spatially-fixed, i.e., the work coordinate system does **not** move when the platform of the Hexapod moves.

The center of rotation always lies at the origin of the tool coordinate system and it therefore moves just as the tool coordinate system with the platform.

A combination of work and tool coordinate systems is created in the volatile memory when a coordinate system of the KST or KSW type is enabled. The combinations remain in the volatile memory even if the coordinate

systems of the KST or KSW type contained therein are no longer enabled, but are still present.

The work-and-tool concept is applied even when only a coordinate system of the KST **or** the KSW type is enabled, i.e., a coordinate system of the other type is **not** enabled. An automatically generated substitute is used for the missing type. The placement and alignment of the axes of the coordinate system which is generated and used as a substitute, correspond to the ones of the ZERO operating coordinate system (offset values for the position of the axes $X = Y = Z = U = V = W = 0$). When the properties of coordinate system combinations are queried using the KLC? command (p. 17), the coordinate systems generated and used as a substitute are therefore contained in the response under the name "Zero".

Consideration of Coordinate Systems of the KSD and KSF Type from the "Work" and "Tool" Perspective

The user-defined coordinate systems of the KSD (p. 30) and KSF (p. 32) types can be best understood if we imagine that a work coordinate system and a tool coordinate system is always present, even if the work-and-tool concept is not applied. **Here too, the current position of the moving platform of the Hexapod has to be considered the position of the tool coordinate system in the work coordinate system.**

The default settings are considered from this perspective first:

- In the default settings, i.e., when the ZERO operating coordinate system is enabled, both the work and tool coordinate systems are congruent with ZERO, if the platform of the Hexapod has the current position $X = Y = Z = U = V = W = 0$. During motion, the tool coordinate system moves with the platform of the Hexapod whereas the work coordinate system is spatially-fixed and remains congruent with ZERO.

Consideration of operating coordinate systems of the KSD and KSF types from the "work" and "tool" perspective:

- When an operating coordinate system of the KSD type is defined, the work **and** the tool coordinate systems are simultaneously positioned and aligned in space according to the offset values given. After enabling an operating coordinate system of the KSD type, the work and the tool coordinate systems are congruent exactly when the moving platform of the Hexapod is positioned at $X = Y = Z = U = V = W = 0$. During motion, the tool coordinate system moves with the platform of the Hexapod as usual whereas the work coordinate system remains spatially-fixed in the position and alignment defined by KSD.
- When an operating coordinate system of the KSF type is defined, the work coordinate system is placed and aligned in space so as to be congruent with the tool coordinate system which is currently placed and aligned in space according to the previous motions of the platform of the Hexapod. After enabling an operating coordinate system of the KSF type, the current position of the platform of the Hexapod is thus displayed as $X = Y = Z = U = V = W = 0$, provided that no motions have been commanded between the definition of the coordinate system and its enabling. During motion, the tool coordinate system moves with the platform of the Hexapod as usual, whereas the work coordinate system remains spatially-fixed in the position and alignment defined by KSF.

In principle, the center of rotation here also lies at the origin of the tool coordinate system. When the enabled operating coordinate system is the ZERO coordinate system or a coordinate system of the KSF type, the center of rotation can additionally be moved from the origin of the tool coordinate system in X and/or Y and/or Z direction with the SPI command (the center of rotation that can be moved using SPI is also referred to as "pivot point"). Changing the coordinates of the center of rotation using SPI is only possible if the current position of the platform is $U = V = W = 0$. Since the center of rotation either lies directly at the origin of the tool coordinate system or it is moved using SPI with fixed offset values from the origin, it always moves with the platform just as the tool coordinate system.

Defining, Copying and Deleting Coordinate Systems

Coordinate systems can be defined in the volatile memory using the following commands:

Coordinate System Type	Command	Description, see
KSD	KSD	p. 30
KSF	KSF	p. 32
KST	KST	p. 33
KSW	KSW	p. 35
KSB(USER)	KSB	p. 29
KLD(USER)	KLD	p. 20
KLF(USER)	KLF	p. 22

A copy of a coordinate system can be generated in the volatile memory using the KCP command (p. 14).

A coordinate system that is not enabled or is not linked to the enabled coordinate system as a predecessor can be deleted from the volatile memory using the KRM command (p. 28). When deleting a coordinate system, its direct predecessor and its successor are linked to one another.

Using the KLS? command (p. 25), the properties of the coordinate systems in the volatile memory can be queried.

Linking Coordinate Systems

Coordinate systems can be linked to one another using the KLN command (p. 23).

Details on the structure of chains and the applicability of KLN:

- The HEXAPOD coordinate system is always based on the configuration file with the geometry data for the Hexapod and it is the starting point for all coordinate system chains.
- The PI_Levelling leveling coordinate system is always the successor to HEXAPOD.
- By default, the successor to the PI_Levelling leveling coordinate system is the PI_Base orientational coordinate system.
 PI_Levelling is automatically set as predecessor, when a leveling coordinate system of the KLD(USER) or KLF(USER) type is defined.
 When a leveling coordinate system of the KLD(USER) or KLF(USER) type is enabled, PI_Base is automatically linked to this coordinate system as a successor, and no longer directly to PI_Levelling.
- By default, the successor to the PI_Base orientational coordinate system is the ZERO operating coordinate system.
 When an orientational coordinate system of the KSB(USER) type is defined, it automatically has the enabled orientational coordinate system as its predecessor (by default this is PI_Base). Multiple orientational coordinate systems of the KSB(USER) type can be linked to one another using KLN, whereby PI_Base always remains the predecessor of the respective chain.
 When an orientational coordinate system of the KSB(USER) type is enabled, ZERO is automatically linked to this coordinate system as successor.
- When an operating coordinate system of the KSD, KSF, KST or KSW type is defined for the first time, the predecessor is automatically the ZERO coordinate system. Multiple operating coordinate systems of the KSD, KSF, KST or KSW type can be linked to one another using KLN, whereby ZERO always

remains the predecessor of the respective chain.

- A coordinate system cannot be linked to itself.
- Although ring connections of at least two coordinate systems can be formed, they cannot be enabled using KEN (p. 14).

Enabling/Disabling Coordinate Systems

Precisely one coordinate system or precisely one coordinate system combination is enabled from the groups described in "Groups and Types of Coordinate Systems" (p. 2).

Only the properties of enabled coordinate systems are effective for the work with the Hexapod.

Enabled coordinate systems cannot be changed or deleted. If a coordinate system is linked as predecessor to the enabled coordinate system, it is not enabled itself, however, it cannot be changed or deleted. In this Technical Note, enabled coordinate systems and coordinate systems that are predecessors of the enabled coordinate system are also referred to as "coordinate systems used". Enabled coordinate systems can be queried using the KEN? (p. 16) and the KET? (p. 16) commands. The response to the KLS? command (p. 25) also lists, along with the properties of the coordinate systems, whether the coordinate systems are currently in use, i.e., whether they are enabled or are predecessors of the enabled coordinate system (value "Used = """).

The chain of the predecessors of the enabled coordinate system cannot be changed.

The KEN command (p. 14) can be used to enable coordinate systems of the following types:
 KSD, KSF, KSW, KST, KSB(USER), KSB(PI), KLF(USER), KLF(PI), KLD(USER), KLD(PI), ZERO

When using KEN to enable coordinate systems, a motion is not initiated, but the display for the current position of the moving platform of the Hexapod is changed.

When a KST or KSW type coordinate system is enabled, a combination of work and tool coordinate systems is created in the volatile memory. The text below therefore also refers to "enabling a coordinate system combination". Using the KLC? command (p. 17), the properties of the coordinate system combinations in the volatile memory can be queried.

By enabling a coordinate system for one of the coordinate system groups (operating, orientational, or leveling coordinate systems), the coordinate system or the coordinate system combination that was previously enabled for this group is simultaneously disabled.

Sending KEN ZERO re-enables the ZERO operating coordinate system that is enabled by default. When the command level 1 is active (see description of the CCL command in the manual for the Hexapod controller), KEN ZERO also re-enables the PI_Levelling leveling coordinate system which was enabled by default, but not the PI_Base orientational coordinate system which was enabled by default (this can be re-enabled by sending KEN PI_Base). By sending DPA SKS (p. 13), all coordinate systems enabled by default can be re-enabled independently of the currently active command level.

Depending on the coordinate system type, when a coordinate system or a coordinate system combination is enabled, further settings, besides the ones of the offset values for the positions of the X, Y, Z, U, V, W axes, become effective:

- For the ZERO and KSF types:
 SPI: pivot point coordinates R, S, T
- For the ZERO, KSD and KSF types and also for coordinate system combinations of the KSW/KST or ZERO/KST or KSW/ZERO types:
 - NLM: Limit for the low end of the axis travel range ("soft limit")
 - PLM: Limit for the high end of the axis travel range ("soft limit")

-
-
- SSL: Activation state of the soft limits of the axis
 - SST: Step size for motions initiated by a manual control unit

The settings for the enabled coordinate system can be changed using the corresponding commands (SPI, NLM, PLM, SSL, SST; see the manual for the Hexapod controller).

Backing Up and Restoring Settings

Coordinate systems and coordinate system combinations are always defined, enabled, copied, linked, or deleted in the volatile memory of the Hexapod controller. Unless backed up, the current settings are lost when the Hexapod controller is switched off or rebooted.

To perform the backup, the currently valid settings for coordinate systems can be written to the nonvolatile memory of the Hexapod controller using the WPA command (p. 40). The settings saved using WPA are automatically loaded from the nonvolatile memory to the volatile memory when the Hexapod controller is switched on or rebooted.

Saving using WPA does **not** overwrite the default settings.

The default settings can be restored using the DPA command (p. 13) or by sending KEN ZERO, see also "Enabling/Disabling Coordinate Systems" (p. 6).

Commanding Motion

Motions of the platform of the Hexapod can always be considered motions of the tool coordinate system in the work coordinate system, see also "Consideration of Coordinate Systems of the KSD and KSF Type from the "Work" and "Tool" Perspective" (p. 4). The alignment and placement of the work and tool coordinate system depends on the enabled operating coordinate system or the coordinate system combination.

Absolute motions of the platform of the Hexapod, which are commanded using MOV (for a description, see the manual for the Hexapod controller), are therefore also always regarded as absolute motions of the tool coordinate system in the work coordinate system. Using MVR (for a description, see the manual of the Hexapod controller) corresponding relative motions can be commanded.

Relative motions of the platform of the Hexapod in the tool coordinate system can always be commanded using MRT (p. 37).

Relative motions of the platform of the Hexapod in the work coordinate system can always be commanded using MRW (p. 38).

The pivot point defined using the SPI command (for a description, see the manual for the Hexapod controller), can be used for rotations or can be changed only if the ZERO coordinate system or a coordinate system of the KSF type is enabled as the operating coordinate system.

By using the TRA? command (p. 39), you can query the maximum absolute position that would be commandable if the platform of the Hexapod were to move along a given direction vector.

The current position of the platform of the Hexapod and hence the position of the tool coordinate system in the work coordinate system can be queried using the POS? command (for a description, see the manual for the Hexapod controller).

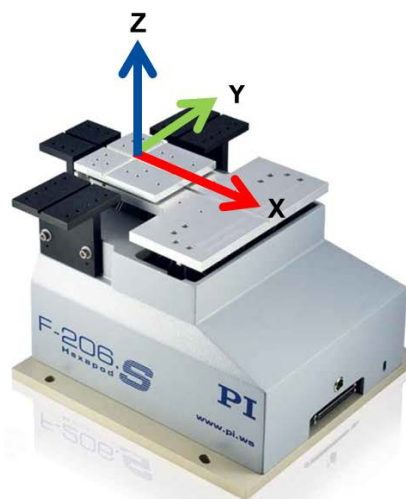
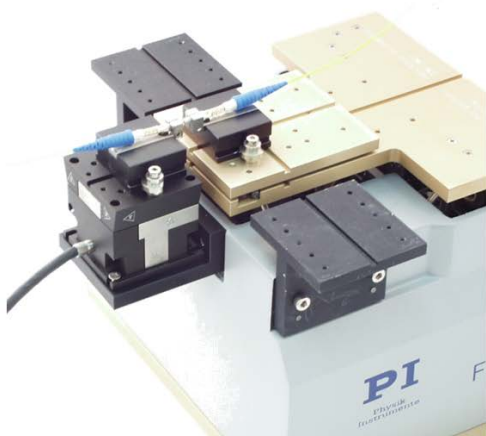
PC Software from PI

The PC software from PI (incl. PIMikroMove®, PI GCS 2 Dynamic Library, GCS LabVIEW driver set, ***Hexapod Simulation Software*** simulation program) has been modified for working with user-defined coordinate systems.

The modifications in PIMikroMove® concern primarily the ***Hexapod Platform*** window and the ***Hexapod 3D View*** tab card. In addition, the Show ***Hexapod Simulation Tool*** menu item is available which launches the ***Hexapod Simulation Software*** simulation program. The settings made in the ***Hexapod Simulation Software*** simulation program can be sent to the Hexapod controller using PIMikroMove®.

Examples of User-Defined Coordinate Systems

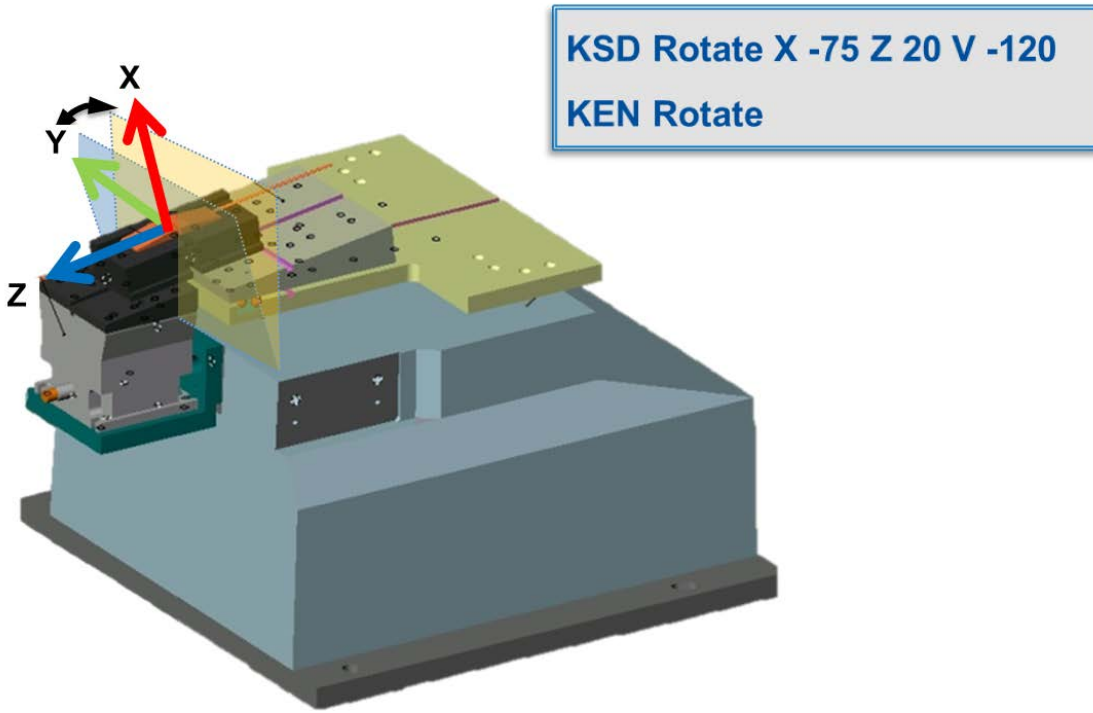
Optics: The Optical Axis is to be the Z Axis



KSB Base V -90
KEN Base

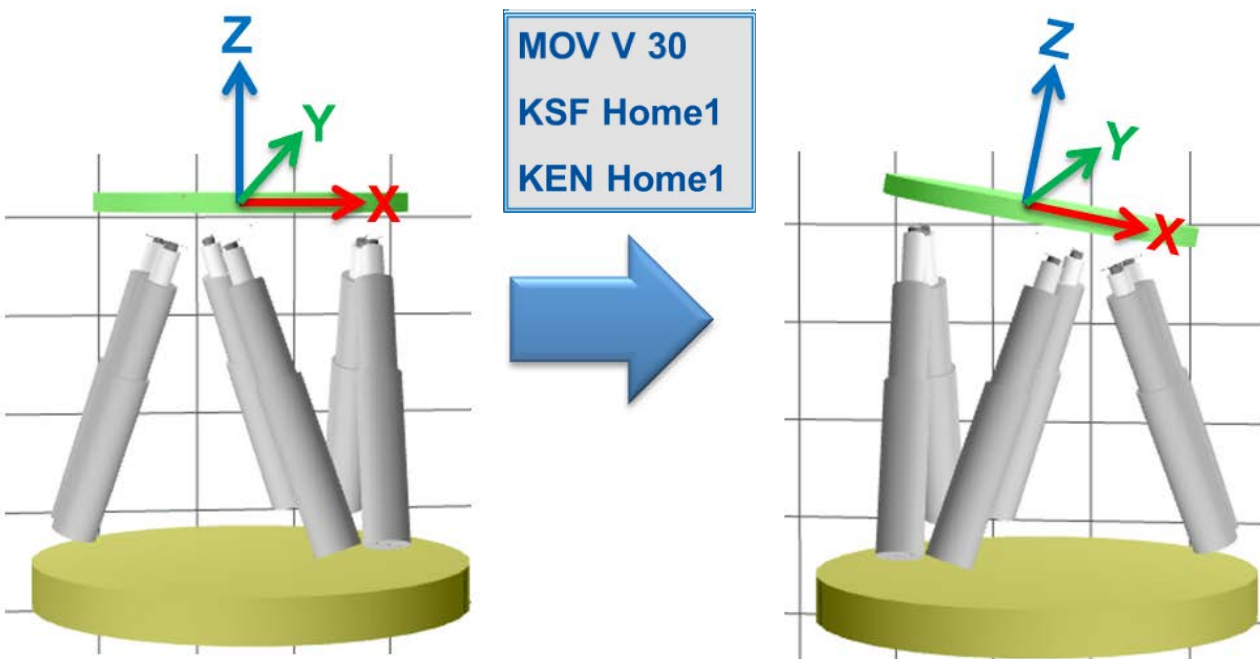


Optics: Scanning in an Inclined Plane



User-Defined "Home" Position

The platform was moved using MOV to the new position before sending KSF.

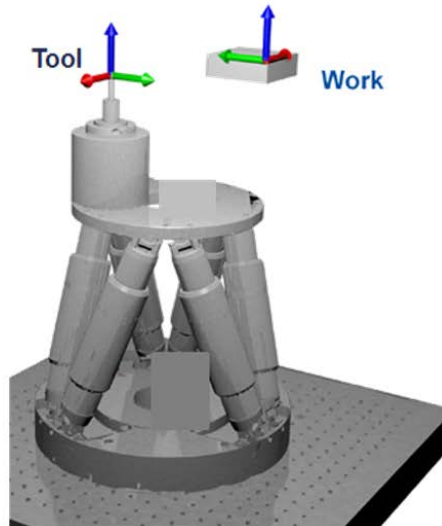


Applying the Work-and-Tool Concept

Carried tool interacts with spatially-fixed workpiece, e.g., during milling, gluing or cutting

- Work coordinate system and tool coordinate system are defined and enabled.
- The motion using MRT advances the tool.
- The motion using MRW occurs along the workpiece in order to machine the workpiece.

KST Tool X 10 Z 40 W -15
KSW Work X 20 Y -10 W 165
KEN Tool
KEN Work
MRT Z 5
MRW X 10



GCS Commands

The commands described below were introduced for working with user-defined coordinate systems. The commands make it possible, among other things, to define, enable, link and save coordinate systems.

The commands described in this Technical Note complement the PI General Command Set (PI GCS) which is described in the manual for the C-887 Hexapod controller. The manual for the Hexapod controller also contains basic information on the syntax used in the command descriptions and on the PI GCS syntax.

Command Overview

Command	Format	Description
DPA (p. 13)	DPA <Pswd> [{<ItemID> <PamID>}]	Reset Volatile Memory Settings To Default
KCP (p. 14)	KCP <CSNameSource> <CSNameCopy>	Copy Coordinate System
KEN (p. 14)	KEN <CSName>	Enable Coordinate System
KEN? (p. 16)	KEN? [{<CSName>}]	Get Enabled Coordinate Systems
KET? (p. 16)	KET? [{<CSType>}]	Get Enabled Coordinate System Types
KLC? (p. 17)	KLC? [<CSName1>[<CSName2>[<Item1>[<Item2>]]]]]	Get Properties Of Work-And-Tool Combinations
KLD (p. 20)	KLD <CSName> [{<AxisID> <Offset>}]	Define Leveling Coordinate System By Specifying Values
KLF (p. 22)	KLF <CSName>	Define Leveling Coordinate System At Current Position
KLN (p. 23)	KLN <CSName1> <CSName2>	Link Coordinate Systems
KLN? (p. 25)	KLN? [{<CSName>}]	Get Coordinate System Chains
KLS? (p. 25)	KLS? [<CSName>[<Item1>[<Item2>]]]	Get Coordinate System Properties
KLT? (p. 27)	KLT? [<StartCS> [<EndCS>]]	Get Offsets Resulting From A Chain
KRM (p. 28)	KRM <CSName>	Remove Coordinate System
KSB (p. 29)	KSB <CSName> [{<AxisID> <Angle>}]	Define Orientational Coordinate System
KSD (p. 30)	KSD <CSName> [{<AxisID> <Offset>}]	Define Operating Coordinate System By Specifying Values
KSF (p. 32)	KSF <CSName>	Define Operating Coordinate System At Current Position
KST (p. 33)	KST <CSName> [{<AxisID> <Offset>}]	Define "Tool" Operating Coordinate System
KSW (p. 35)	KSW <CSName> [{<AxisID> <Offset>}]	Define "Work" Operating Coordinate System
MRT (p. 37)	MRT {<AxisID> <Distance>}	Set Target Relative In Tool Coordinate System
MRW (p. 38)	MRW {<AxisID> <Distance>}	Set Target Relative In Work Coordinate System
TRA? (p. 39)	TRA? {<AxisID> <Component>}	Get Maximum Commandable Position For Direction Vector
WPA (p. 40)	WPA <Pswd> [{<ItemID> <PamID>}]	Save Settings To Non-Volatile Memory

Naming Conventions for Coordinate Systems

Permissible characters: 1234567890ABCDEFGHIJKLMNQRSTUvwxyz_

The number of characters is unlimited.

The name must start with an alphabetic character.

Reserved names which must not be used for defining, copying or deleting: HEXAPOD, PI_LEVELLING, PI_BASE, ZERO, 0, NULL, XML, KLF, KLF(USER), KLF(PI), KLD, KLD(USER), KLD(PI), KSB, KSB(USER), KSB(PI), KSD, KSF, KST, KSW

Each name can exist only once. Any existing coordinate system not in use will be overwritten when a coordinate system with the same name is created (defining, generating a copy). For details, see KLD (p. 20), KLF (p. 22), KSB (p. 29), KSD (p. 30), KSF (p. 32), KST (p. 33) and KSW (p. 35).

Command Descriptions

DPA (Reset Volatile Memory Settings To Default)

Description:	Restores the defaults settings for parameters and parameter-independent settings in the volatile memory.
Format:	DPA <Pswd> [{<ItemID> <PamID>}]
Arguments:	<p><Pswd> is the password for resetting the volatile memory. See below for details.</p> <p><ItemID> is the item for which a parameter is to be reset. See below for details.</p> <p><PamID> is the parameter ID, it can be written in hexadecimal or decimal format. See below for details.</p>
Response:	None
Troubleshooting:	Invalid item identifier, wrong parameter ID
Notes:	<p>With the C-887, DPA can be used only to restore the default parameter-independent settings for coordinate systems.</p> <p>Valid password: SKS</p> <p>When using the SKS password, <ItemID> and <PamID> are not needed.</p>

Default settings which are restored using DPA SKS:

- Orientational coordinate system (KSB() type): PI_BASE is enabled
- Leveling coordinate system (KLD() or KLF() type): PI_Levelling is enabled
- ZERO operating coordinate system is enabled and based on PI_BASE and PI_Levelling
- Pivot point (see SPI), soft limits of axes (see NLM, PLM and SSL), step size for motions initiated by a manual control unit (see SST)

The default settings loaded using DPA are independent of the settings in the nonvolatile memory, which can be overwritten using WPA (p. 40). The settings in the nonvolatile memory (saved using WPA) are automatically loaded to the volatile memory when the C-887 is switched on or rebooted.

KCP (Copy Coordinate System)

Description: Generates a copy of a coordinate system.

Format: KCP <CSNameSource> <CSNameCopy>

Arguments: <CSNameSource> is the name of the coordinate system of which a copy is to be generated. PI_Base, PI_Levelling, ZERO and HEXAPOD cannot be copied.
 <CSNameCopy> is the name of the copy of the coordinate system.

Response: None

Notes: PI_Base, PI_Levelling, ZERO and HEXAPOD cannot be copied.

Options for creating the copy:

- <CSNameCopy> is a new name. The copy is created as a new coordinate system with this name.
- <CSNameCopy> is the name of an existing coordinate system that is not in use. This overwrites the coordinate system.

The linking to the predecessor in a chain of coordinate systems is copied. The linking to successors is **not** copied.

The copy is generated in the volatile memory. Using WPA, the copy can be written to the nonvolatile memory.

KEN (Enable Coordinate System)

Description: Enables the given coordinate system. The scope of the settings that are influenced by enabling depends on the coordinate system type, see below.

Position values for the moving platform of the Hexapod (query using POS?) refer to the enabled operating coordinate system.

When applying the work-and-tool concept:

- The work-and-tool concept uses a combination of two enabled operating coordinate systems. Generally, the combination consists of one active coordinate system of the KST type and one of the KSW type. If a coordinate system is enabled for only one of the two types, a substitute is automatically used for the other type, for details see p. 3.
- The current position of the moving platform of the Hexapod queried using POS? can be considered the position of the tool coordinate system in the work coordinate system.

Enabling coordinate systems using KEN does not initiate a motion.

Format: KEN <CSName>

Arguments: <CSName> is the name of the coordinate system to be enabled.

Response: None

Notes: Before enabling a coordinate system, KEN checks that the definition of this coordinate system and its linking are correct (for details, see p. 5). If the coordinate system is not correctly defined, it is not enabled.

KEN can be used to enable coordinate systems of the following types: KSD, KSF, KSW, KST, KSB(USER), KSB(PI), KLF(USER), KLF(PI), KLD(USER), KLD(PI), ZERO

Precisely one coordinate system or precisely one coordinate system combination is enabled from the following groups of coordinate systems:

- Operating coordinate system: A coordinate system of ZERO, or KSF, or KSD type or - for the work-and-tool concept - a combination of coordinate systems of the KSW/KST, or ZERO/KST, or KSW/ZERO type
- Orientational coordinate system (KSB (PI) or KSB(USER) type)
- Leveling coordinate system (of the KLD(USER), or KLD(PI), or KLF(USER), or KLF(PI) type)

By enabling a coordinate system for one of the groups, the coordinate system or the coordinate system combination that was previously enabled for this group is simultaneously disabled.

Sending KEN ZERO re-enables the ZERO operating coordinate system that is enabled by default. When the command level 1 is active (see description of the CCL command in the manual for the Hexapod controller), KEN ZERO also re-enables the PI_Levelling leveling coordinate system which was enabled by default, but not the PI_Base orientational coordinate system which was enabled by default (this can be re-enabled by sending KEN PI_Base). By sending DPA SKS (p. 13), all coordinate systems enabled by default can be re-enabled independently of the currently active command level.

The enabled operating coordinate system specifies values for the following settings (for the work-and-tool concept, the values are specified using the combination of two operating coordinate systems):

- NLM: Limit for the low end of the axis travel range ("soft limit")
- PLM: Limit for the high end of the axis travel range ("soft limit")
- SSL: Activation state of the soft limits of the axis
- SPI: Coordinates of the pivot point (only for coordinate systems of the KSF and ZERO types)

- SST: Step size for motions initiated by a manual control unit

Before enabling leveling and orientational coordinate systems (of the KLD(), KLF() and KSB() types), it is necessary to switch to command level 1 (see CCL).

The coordinate systems are enabled and disabled in the volatile memory. Using WPA, the activation state can be written to the nonvolatile memory.

KEN? (Get Enabled Coordinate Systems)

Description: Lists the names of the enabled coordinate systems and displays their type.

Format: KEN? [{<CSName>}]

Arguments: <CSName> is the name of an enabled coordinate system. Illegal: ZERO.
When <CSName> is omitted, all enabled coordinate systems are listed.

Response: {<CSName>="<CSType>}

where

<CSType> gives the coordinate system type.

Notes: KEN? queries the volatile memory.

When the ZERO operating coordinate system is enabled, it is **not** displayed in the response to KEN? and the response contains only the following:

- The enabled leveling coordinate system, i.e., a coordinate system of the KLD(PI), or KLD(USER), or KLF(PI), or KLF(USER) type
- The enabled orientational coordinate system, i.e., a coordinate system of the KSB(PI) or KSB(USER) type

If <CSName> is given in the query and the corresponding coordinate system is not enabled, the C-887 sends an empty response and sets an error (error code query using ERR?).

KET? (Get Enabled Coordinate System Types)

Description: Lists the enabled coordinate system types and displays the names of the corresponding coordinate systems.

Format: KET? [{<CSType>}]

Arguments: <CSType> is an enabled coordinate system type. Possible values: KSW, KST, KSF, KSD, KLD(PI), KLD(USER), KLF(PI), KLF(USER), KSB(PI), KSB(USER)

When <CSType> is omitted, all enabled coordinate system types are listed.

Response: {<CSType>="<CSName>}

where

<CSName> gives the coordinate system name.

Notes: KET? queries the volatile memory.

When the operating coordinate system of the ZERO type is enabled, this type is **not** displayed in the response to KET?, and the response contains only the following:

- The type of the enabled leveling coordinate system, i.e., KLD(PI) or KLD(USER) or KLF(PI) or KLF(USER)
- The type of the enabled orientational coordinate system, i.e., KSB(PI) or KSB(USER)

If <CSType> is given in the query and no coordinate system of the corresponding type is enabled, the C-887 sends an empty response and sets an error (error code query using ERR?).

KLC? (Get Properties Of Work-And-Tool Combinations)

Description: Lists the properties of the coordinate system combinations for the work-and-tool concept that are present in the volatile memory.

The work-and-tool concept uses a combination of two enabled operating coordinate systems. Generally, the combination consists of one active coordinate system of the KST type and one of the KSW type. If a coordinate system is enabled for only one of the two types, a substitute is automatically used for the other type. Coordinate systems used as a substitute are listed under the name "Zero" in the response to KLC?, for details see p. 3.

A combination is created in the volatile memory when a coordinate system of the KST or KSW type is enabled. The combinations remain in the volatile memory even if the coordinate systems of the KST or KSW type contained therein are no longer enabled.

When a coordinate system of the KST or KSW type is deleted using KRM, the response to KLC? no longer lists the combinations in which this coordinate system was involved.

Using WPA, the combinations in the volatile memory can be written to the nonvolatile memory.

Using KLS?, the properties of the coordinate systems in the volatile memory can be queried.

Format: KLC? [<CSName1>[<CSName2>[<Item1>[<Item2>]]]]

Arguments: <CSName1> is the name of a coordinate system of the KST or KSW type which is part of a combination in the volatile memory.

<CSName2> is the name of a coordinate system of the KST or KSW type which is part of a combination in the volatile memory.

<Item1> is a property of the axes of the controller for the queried combination of coordinate systems. Possible values:

- NLM: Limit for the low end of the axis travel range ("soft limit")
- PLM: Limit for the high end of the axis travel range ("soft limit")
- SSL: Activation state of the soft limits of the axis
- SST: Step size for motions initiated by a manual control unit

The settings for the properties of the currently enabled combination can be changed using the corresponding commands and saved using WPA.

<Item2> is an axis of the controller, possible values: X, Y, Z, U, V, W

If the properties of all coordinate system combinations are to be listed, all arguments are omitted.

Response: <String>

<String> contains information, in XML, on the coordinate systems in the volatile memory.

The response structure depends on the number of arguments in the command sent. Possible responses, depending on the number of arguments in the example of coordinate systems Node1, Node2, Node3 and Node4:

Sent

KLC?

Response:

```
<CombinedCoordinateSystem>[SP][LF]
```

```
[TAB] <NODE1.NODE2 Name="NODE1.NODE2" Work="NODE1" Tool="NODE2">[SP][LF]
```

```
[TAB] [TAB] <NLM X="-3.0" ... W="-5.0"/>[SP][LF]
```

```
[TAB] [TAB] <PLM X="3.0" ... W="5.0"/>[SP][LF]
```

```
[TAB] [TAB] <SSL X="1" Y="0" ... W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.1" ... W="0.2"/>[SP][LF]
[TAB] </NODE1.NODE2>[SP][LF]
[TAB] [TAB] <NODE1.NODE4 Name="NODE1.NODE4" Work="NODE1" Tool="NODE4"> [SP][LF]
[TAB] [TAB] <NLM X="-4.0" ... W="-3.4"/>[SP][LF]
[TAB] [TAB] <PLM X="2.0" ... W="2.0"/>[SP][LF]
[TAB] [TAB] <SSL X="1" Y="0" ... W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.2" ... W="0.15"/>[SP][LF]
[TAB] </NODE1.NODE4>[SP][LF]
...
</CombinedCoordinateSystem>[LF]
```

Sent

KLC? Node1

Response:

```
<CombinedCoordinateSystem>[SP][LF]
[TAB] <NODE1.NODE2 Name="NODE1.NODE2" Work="NODE1" Tool="NODE2"> [SP][LF]
[TAB] [TAB] <NLM X="-3.0" ... W="-5.0"/>[SP][LF]
[TAB] [TAB] <PLM X="3.0" ... W="5.0"/>[SP][LF]
[TAB] [TAB] <SSL X="1" Y="0" ... W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.1" ... W="0.2"/>[SP][LF]
[TAB] </NODE1.NODE2 >[SP][LF]
[TAB] <NODE1.NODE4 Name="NODE1.NODE4" Work="NODE1" Tool="NODE4"> [SP][LF]
[TAB] [TAB] <NLM X="-1.0" ... W="-7.0"/>[SP][LF]
[TAB] [TAB] <PLM X="1.1" ... W="10.0"/>[SP][LF]
[TAB] [TAB] <SSL X="1" Y="0" ... W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.2" ... W="0.21"/>[SP][LF]
[TAB] </NODE1.NODE4 >[SP][LF]
...
</CombinedCoordinateSystem>[LF]
```

Sent

KLC? Node1 Node2

Response:

```
<CombinedCoordinateSystem>[SP][LF]
[TAB] <NODE1.NODE2 Name="NODE1.NODE2" Work="NODE1" Tool="NODE2"> [SP][LF]
[TAB] [TAB] <NLM X="-3.0" ... W="-5.0"/>[SP][LF]
[TAB] [TAB] <PLM X="3.0" ... W="5.0"/>[SP][LF]
[TAB] [TAB] <<SSL X="1" Y="0" ... W="1"/> [SP][LF]
[TAB] [TAB] <SST X="0.1" ... W="0.2"/>[SP][LF]
[TAB] </NODE1.NODE2 >[SP][LF]
</CombinedCoordinateSystem>[LF]
```

Sent

KLC? Node1 Node2 PLM

Response:

<PLM X="3.0" ... W="5.0"/>[LF]

Sent

KLC? Node1 Node2 PLM X

Response:

X = 3.0 [LF]

KLD (Define Leveling Coordinate System By Specifying Values)

Description: Defines a leveling coordinate system of the KLD(USER) type for permanently correcting errors in the Hexapod alignment (e.g., faulty mounting).

The leveling coordinate system is defined on the basis of measurements (e.g., using an interferometer) and corrects the linear displacement (X, Y, Z axes) and axis inclination (U, V, W axes) of the moving platform of the Hexapod.

If the linear displacement and axis inclination cannot be measured: Use KLF to define a leveling coordinate system.

The coordinate system is defined in the volatile memory. Using WPA, the definition can be written to the nonvolatile memory.

Format: KLD <CSName> [{<AxisID> <Offset>}]

Arguments: <CSName> is the name of the coordinate system to be defined.
 <AxisID> is one axis of the controller. Possible values: X, Y, Z, U, V, W.
 <Offset> is an offset which is added after the reference move to the current position value of the axis; in physical units.
 For axes not given in the KLD command, the offset is set to zero.

Response: None

Notes: Before defining a leveling coordinate system, it is necessary to switch to command level 1 (see CCL).

Options for defining a coordinate system using KLD:

- <CSName> is a new name. The leveling coordinate system is created under this new name.
- <CSName> is the name of an existing coordinate system that is not in use. The coordinate system is overwritten by the definition created with KLD.

Recommended procedure for defining and enabling a leveling coordinate system of the KLD(USER) type:

1. Perform a reference move (see FRF)
2. Measure the discrepancy between the position and orientation of the moving platform of the Hexapod and the position and orientation for which $X = 0$, $Y = 0$, $Z = 0$, $U = 0$, $V = 0$, $W = 0$ is applicable in future (measurement using an external measuring instrument)
3. Switch to command level 1 by sending CCL 1 advanced
4. Define the leveling coordinate system using KLD by giving the measured discrepancies for the axes of the moving platform (offset values)
5. Enable the leveling coordinate system (see KEN)
6. Optional: Define the behavior after the reference move by setting the **Behavior After Reference Move** (ID 0x07030401) and **Target For Motion After Reference Move** (ID 0x07030402) parameters. In this way, for example, the platform can automatically be moved to the zero position after the reference move.
 - Value of the parameter 0x07030401 = 0: Moving platform remains in the reference position after the reference move, i.e., at the reference point switches of the Hexapod struts
 - Value of parameter 0x07030401 = 1: After the reference move, the moving platform moves to the target position which is given by parameter 0x07030402
7. Save the settings by sending WPA SKS

The offset values which are displayed in the response to KLS? for the coordinate systems of the KLD(USER) type result by means of recalculation from all currently enabled coordinate systems. Thus, they can change when changing the enabled coordinate systems. The offset values listed in the response to KLT?, however, refer in each case to the given predecessor in the chain and are thus independent of the currently enabled coordinate systems.

The following applies for leveling coordinate systems defined using KLD:

- The leveling coordinate system is always the direct successor to the default PI_Levelling leveling coordinate system (automatic linking).
- The leveling coordinate system **cannot** be linked to other coordinate systems using KLN

DPA SKS re-enables independently of the currently enabled command level the default PI_Levelling leveling coordinate system, for details see KEN.

KLF (Define Leveling Coordinate System At Current Position)

Description: Defines a leveling coordinate system of the KLF(USER) type for permanently correcting errors in the Hexapod alignment (e.g., faulty mounting).

To define the leveling coordinate system, the moving platform after the reference move is commanded into the position and orientation for which $X = 0$, $Y = 0$, $Z = 0$, $U = 0$, $V = 0$, $W = 0$ is applicable in future. Sending KLF defines a coordinate system with offset values which are added, after the reference move, to the current position values for the axes; in physical units.

If the linear displacement (X, Y, Z axes) and axis inclination (U, V, W axes) are to be measured: Use KLD to define a leveling coordinate system.

The coordinate system is defined in the volatile memory. Using WPA, the definition can be written to the nonvolatile memory.

Format: KLF <CSName>

Arguments: <CSName> is the name of the coordinate system to be defined.

Response: None

Notes: Before defining a leveling coordinate system, it is necessary to switch to command level 1 (see CCL).

Options for defining a coordinate system using KLF:

- <CSName> is a new name. The leveling coordinate system is created under this new name.
- <CSName> is the name of an existing coordinate system that is not in use. The coordinate system is overwritten by the definition created with KLF.

The definition using KLF is possible only when the Hexapod is not in motion.

Recommended procedure for defining and enabling a leveling coordinate system of the KLF(USER) type:

1. Perform a reference move (see FRF)
2. Approach the position and orientation of the moving platform of the Hexapod for which $X = 0$, $Y = 0$, $Z = 0$, $U = 0$, $V = 0$, $W = 0$ is applicable in future
3. Switch to command level 1 by sending CCL 1 advanced
4. Define the leveling coordinate system using KLF
5. Enable the leveling coordinate system (see KEN)
6. Optional: Define the behavior after the reference move by setting the **Behavior After Reference Move** (ID 0x07030401) and **Target For Motion After**

Reference Move (ID 0x07030402) parameters. In this way, for example, the platform can automatically be moved to the zero position after the reference move.

- Value of the parameter 0x07030401 = 0: Moving platform remains in the reference position after the reference move, i.e., at the reference point switches of the Hexapod struts
- Value of parameter 0x07030401 = 1: After the reference move, the moving platform moves to the target position which is given by parameter 0x07030402

7. Save the settings by sending WPA SKS

The offset values which are displayed in the response to KLS? for the coordinate systems of the KLF(USER) type result by means of recalculation from all currently enabled coordinate systems. Thus, they can change when changing the enabled coordinate systems. The offset values listed in the response to KLT?, however, refer in each case to the given predecessor in the chain and are thus independent of the currently enabled coordinate systems.

The following applies for leveling coordinate systems defined using KLF:

- The leveling coordinate system is always the direct successor to the default PI_Levelling leveling coordinate system (automatic linking).
- The leveling coordinate system **cannot** be linked to other coordinate systems using KLN

DPA SKS re-enables independently of the currently enabled command level the default PI_Levelling leveling coordinate system, for details see KEN.

KLN (Link Coordinate Systems)

Description:	Links two coordinate systems to create a chain of predecessor and successor.
Format:	KLN <CSName1> <CSName2>
Arguments:	<CSName1> is the name of the coordinate system which is to be attached in the chain as successor to <CSName2>. <CSName2> is the name of the coordinate system which is to be the predecessor of <CSName1> in the chain.
Response:	None
Notes:	Each coordinate system is part of at least one chain. For the basic structure of coordinate system chains, see "Groups and Types of Coordinate Systems" (p. 2) and "Linking of Coordinate Systems" (p. 5).

By default, the following coordinate systems are linked to form a chain:

- The HEXAPOD coordinate system, which is based on the configuration file with the geometric data of the Hexapod, is the "origin" of all chains and the predecessor of the PI_Levelling leveling coordinate system (fixed linking)
- PI_Levelling is the predecessor of the PI_Base orientational coordinate system
- PI_Base is the predecessor of the ZERO operating coordinate system

The following applies for linked coordinate systems:

- The actual offset values for the position of the X, Y, Z, U, V, W axes result from the offset values for the predecessors linked to a coordinate system (for details, see KLT?).
- Each coordinate system has precisely one predecessor and can have at least one successor.
- When a coordinate system is enabled, all predecessors in the chain are likewise in use and cannot be deleted or overwritten.
- When a coordinate system (unused) is deleted, its predecessor and successor are linked to one another in the chain.

Limitations for creating chains using KLN:

- A coordinate system cannot be linked to itself.
- Although ring connections from at least two coordinate systems can be formed, they **cannot** be enabled using KEN.
- Using KLN, a coordinate system in use **cannot** be attached as successor to another coordinate system.
- Using KLN, a coordinate system (not in use) can be attached as successor to a coordinate system in use.
- Before linking orientational coordinate systems of the KSB(USER) type as successor, it is necessary to switch to command level 1 (see CCL).
- Using KLN, coordinate systems of the KLD(PI), KLF(PI), KLD(USER), KLF(USER) types and the HEXAPOD coordinate system **cannot** be linked.
- Using KLN, the PI_Base coordinate system **cannot** be attached as successor to another coordinate system.
- Using KLN, coordinate systems of the KSB(USER) type can only be linked to other coordinate systems of the KSB(USER) type or as successor to the PI_Base coordinate system.
- Using KLN, the ZERO coordinate system **cannot** be attached as successor to another coordinate system.

Using KLN, coordinate systems are linked in the volatile memory. Using WPA, the linking can be written to the nonvolatile memory.

KLN? (Get Coordinate System Chains)

- Description:** Lists the components of the existing coordinate system chains.
- Each coordinate system is part of at least one chain. For the basic structure of coordinate system chains, see "Linking of Coordinate Systems" (p. 5).
- Format:** KLN? [{<CSName>}]
- Arguments:** <CSName> is the name of a coordinate system, the predecessors of which are to be listed in the chain.
- If the predecessors of all coordinate systems are to be listed, <CSName> is omitted.
- Response:** {<CSName>="<String>}
- where
- <String> contains the names of the coordinate system predecessors in the chain. The "origin" of the chain is always given at the end of the line
- Notes:** To ensure a clear overview, only the predecessors up to the ZERO coordinate system are listed for operating coordinate systems of the KSD, KSF, KSW and KST types. The predecessors of ZERO, however, can be specifically queried and are also contained as a separate line in the response if <CSName> is omitted from the query.

KLS? (Get Coordinate System Properties)

- Description:** Lists the properties of the coordinate systems that are present in the volatile memory.
- Using WPA, the coordinate systems in the volatile memory can be written to the nonvolatile memory.
- The properties of the coordinate system combinations for the work-and-tool concept that are present in the volatile memory can be queried using KLC?.
- Format:** KLS? [<CSName>[<Item1>[<Item2>]]]
- Arguments:** <CSName> is the name of a coordinate system. Illegal: HEXAPOD.
- <Item1> is a property of the axes of the controller. Possible values:
- For all types of coordinate systems:
- POS: Offset for positions of the axes
- Only for coordinate systems of the KSD, KSF and ZERO types:

- NLM: Limit for the low end of the axis travel range ("soft limit")
- PLM: Limit for the high end of the axis travel range ("soft limit")
- SSL: Activation state of the soft limits of the axis
- SST: Step size for motions initiated by a manual control unit

Only for coordinate systems of the KSF and ZERO types:

- SPI: Coordinates of the pivot point

<Item2> is an axis of the controller, possible values: X, Y, Z, U, V, W

If the properties of all coordinate systems are to be listed, all arguments are omitted.

Response:

<String>

<String> contains information, in XML, on the coordinate systems in the volatile memory.

The <String> structure depends on the number of arguments in the command sent.

If no argument or only the name of a coordinate system is given, <String> lists, for each coordinate system contained in the response, the data applicable for <Item1> and <Item2> as well as the information below:

- Name of the coordinate system (Name="")
- Name of the direct predecessor of the coordinate system in the chain (Parent="")
- Current use of the coordinate system;
 - Used="True": The coordinate system is in use, i.e., it is either enabled itself, or it is a predecessor of the enabled coordinate system in its chain.
 - Used="False": The coordinate system is not used.
- Coordinate system type (Type="")

Note:

The offset values which are displayed in the response to KLS? for the leveling coordinate systems of the KLD(USER) and KLF(USER) types result by means of recalculation from all currently enabled coordinate systems. Thus, they can change when changing the enabled coordinate systems.

Example:

The example below shows the response to KLS? for the coordinate systems existing by default.

KLS?

<SingleCoordinateSystem>

<ZERO Name="ZERO" Parent="PI_BASE" Used="True" Type="ZERO">

<POS X="0.000000" Y="0.000000" Z="0.000000" U="0.000000" V="0.000000" W="0.000000"/>

<NLM X="-10.000100" Y="-10.000100" Z="-10.000100" U="-1.000100" V="-1.000100" W="-1.000100"/>

<PLM X="10.000100" Y="10.000100" Z="10.000100" U="1.000100" V="1.000100" W="1.000100"/>

<SSL X="1" Y="1" Z="1" U="1" V="1" W="1"/>

<SPI R="0.000000" S="0.000000" T="481.000000"/>

```

<SST X="0.010000" Y="0.010000" Z="0.010000" U="0.010000" V="0.010000" W="0.010000"/>
</ZERO>
<PI_BASE Name="PI_BASE" Parent="PI_LEVELLING" Used="True" Type="KSB(PI)">
<POS X="0.000000" Y="0.000000" Z="0.000000" U="0.000000" V="0.000000" W="0.000000"/>
</PI_BASE>
<PI_LEVELLING Name="PI_LEVELLING" Parent="HEXAPOD" Used="True" Type="KLD(PI)">
<POS X="0.000000" Y="0.000000" Z="0.000000" U="0.000000" V="0.000000" W="0.000000"/>
</PI_LEVELLING></SingleCoordinateSystem>

```

KLT? (Get Offsets Resulting From A Chain)

Description: Lists, for a coordinate system, the actual offset values for the position of the X, Y, Z, U, V, W axes which result from the offset values of the predecessors linked to this coordinate system.

For operating coordinate systems, only the operating coordinate systems linked as predecessors up to the ZERO coordinate system are included in the calculation.

For orientational coordinate systems, only the orientational coordinate systems linked as predecessors up to the HEXAPOD coordinate system are included in the calculation.

For leveling coordinate systems, only the leveling coordinate systems linked as predecessors up to the HEXAPOD coordinate system are included in the calculation.

Format: KLT? [<StartCS> [<EndCS>]]

Arguments: <StartCS> is the name of the coordinate system for which the offset values resulting from its predecessors are to be queried.

<EndCS> is the name of a coordinate system linked as a predecessor of <StartCS>, which is to be used as the starting point for the offset calculation. If <EndCS> is omitted, the starting point for the calculation depends on the value for <StartCS>:

- <StartCS> is an operating coordinate system (KSD, KSF, KST, KSW or ZERO type): Starting point is ZERO
- <StartCS> is an orientational or leveling coordinate system (KSB(PI), KSB(USER), KLD(PI), KLD(USER), KLF(PI), KLF(USER) types): Starting point is HEXAPOD

If the resultant offset values for all coordinate systems are to be listed, all arguments are omitted.

Response: <String>

<String> contains, for each queried coordinate system, a line with the following data:

- Name: The name of the coordinate system for which the resultant offset values are listed.
- EndCoordinateSystem: The name of the coordinate system which was used as the starting point for calculating the offset values.
- X, Y, Z, U, V, W: Resultant offset value for the corresponding axis.

Note: KLT? queries the volatile memory.

Using KLS?, the properties of the coordinate systems in the volatile memory can be queried.

Example: The example below shows the response to KLT? for the coordinate systems existing by default.

KLT?

Name=ZERO	EndCoordinateSystem=ZERO	X=0.000000	Y=0.000000	Z=0.000000	U=0.000000	V=0.000000	W=0.000000
Name=PI_BASE	EndCoordinateSystem=HEXAPOD	X=0.000000	Y=0.000000	Z=0.000000	U=0.000000	V=0.000000	W=0.000000
Name=PI_LEVELLING	EndCoordinateSystem=HEXAPOD	X=0.000000	Y=0.000000	Z=0.000000	U=0.000000	V=0.000000	W=0.000000

KRM (Remove Coordinate System)

Description: Deletes a coordinate system.

Format: KRM <CSName>

Arguments: <CSName> is the name of the coordinate system to be deleted.

Response: None

Notes: A coordinate system in use (i.e., it is enabled itself or linked to the enabled coordinate system as a predecessor) **cannot** be deleted.

When a coordinate system (unused) is deleted, its predecessor and its successor are linked to one another in the coordinate system chain.

Before deleting a leveling coordinate system of the KLD(USER), KLF(USER) and KSB(USER) types, it is necessary to switch to command level 1 (see CCL).

The coordinate system is deleted from the volatile memory. Using WPA, the deletion can be transferred to the nonvolatile memory.

KSB (Define Orientational Coordinate System)

Description:	<p>Defines an orientational coordinate system of the KSB(USER) type for permanently changing the direction of the X, and/or Y, and/or Z axes.</p> <p>The direction of the axes is changed by rotating the coordinate system in 90° increments as follows:</p> <ul style="list-style-type: none"> • Rotating around X, i.e., angular value given for U, changes the direction of the Y and Z axes • Rotating around Y, i.e., angular value given for V, changes the direction of the X and Z axes • Rotating around Z, i.e., angular value given for W, changes the direction of the X and Y axes <p>The coordinate system is defined in the volatile memory. Using WPA, the definition can be written to the nonvolatile memory.</p>
Format:	KSB <CSName> [{<AxisID> <Angle>}]
Arguments:	<p><CSName> is the name of the coordinate system to be defined.</p> <p><AxisID> is one axis of the controller. Possible values: U, V, W.</p> <p><Angle> is the angle around which the axis is to be rotated. Possible values: 0, 90, 180, 270, -90, -180, -270 (unit: Degree).</p> <p>For axes not given in the KSB command, the angle is set to zero.</p>
Response:	None
Notes:	<p>Before defining an orientational coordinate system, it is necessary to switch to command level 1 (see CCL).</p>

Options for defining a coordinate system using KSB:

- <CSName> is a new name. The orientational coordinate system is newly created under this name and automatically has the enabled orientational coordinate system as its predecessor
- <CSName> is the name of an existing coordinate system that is not in use. The coordinate system is overwritten by the definition using KSB and automatically attached as successor to the enabled orientational coordinate system

Using KLN, orientational coordinate systems of the KSB(USER) type can be linked to other coordinate systems of the KSB(USER) type or attached as successor to the PI_Base coordinate system.

Orientational coordinate systems of the KSB(USER) type can be enabled using KEN. DPA SKS re-enables the PI_Base orientational coordinate system that is enabled by default, for details see KEN.

KSD (Define Operating Coordinate System By Specifying Values)

Description: Defines an operating coordinate system of the KSD type.

A coordinate system of the KSD type can be placed and aligned in space arbitrarily. For more details, see the "Work-and-Tool Concept" (p. 3) and in particular the "Consideration of Coordinate Systems of the KSD and KSF Type from the "Work" and "Tool" Perspective" (p. 4).

The placement of the coordinate system in space is defined by giving offset values for the X, Y, Z axes. The alignment of the coordinate system is defined by giving offset values for the U, V, W axes:

- Rotating around X, i.e., offset given for U, changes the direction of the Y and Z axes
- Rotating around Y, i.e., offset given for V, changes the direction of the X and Z axes
- Rotating around Z, i.e., offset given for W, changes the direction of the X and Y axes

If an operating coordinate system of the KSD type has been enabled using KEN:

- The coordinate system determines the position values for the moving platform of the Hexapod (query of the current position using POS?).
- The coordinates of the center of rotation **cannot** be changed using SPI and the pivot point defined using SPI is **not** used.

If a coordinate system of the KSD type is not enabled itself, but it nonetheless is linked to the enabled operating coordinate system as a predecessor, its offset values are included in the calculation of the offset values for the enabled operating coordinate system (see KLT?).

The coordinate system is defined in the volatile memory. Using WPA, the definition can be written to the nonvolatile memory.

Format: KSD <CSName> [{<AxisID> <Offset>}]

Arguments: <CSName> is the name of the coordinate system to be defined.
 <AxisID> is one axis of the controller. Possible values: X, Y, Z, U, V, W.
 <Offset> is an offset value for placing or aligning the axis; in physical units.
 For axes not given in the KSD command, the offset is set to zero.

Response: None

Notes: When an operating coordinate system of the KSD type is enabled:

- In addition to the offset values for the X, Y, Z, U, V, W axes, the coordinate system specifies values for the following settings:
 - NLM: Limit for the low end of the axis travel range ("soft limit")
 - PLM: Limit for the high end of the axis travel range ("soft limit")
 - SSL: Activation state of the soft limits of the axis
 - SST: Step size for motions initiated by a manual control unit
- The current settings can be changed using the corresponding commands.
- The current settings can be queried using KLS?.
- The current settings are saved for the coordinate system using WPA.
- For initiating motions, see "Commanding Motion" (p. 7).

Options for defining a coordinate system using KSD:

- <CSName> is a new name. The operating coordinate system is newly created under this name and automatically has the ZERO operating coordinate system as its predecessor.
- <CSName> is the name of an existing coordinate system that is not in use. The coordinate system is overwritten by the definition created with KSD.
 - If the overwritten coordinate system was also of the KSD type, the chain with its predecessor and successor is retained.
 - If the overwritten coordinate system was not of the KSD type, its type is changed to KSD and it is attached to ZERO as the new predecessor.

Using KLN, operating coordinate systems of the KSD, KSF, KST, KSW types can be linked to other operating coordinate systems of these types or reattached as successor to the ZERO coordinate system.

KEN ZERO and DPA SKS re-enable the ZERO operating coordinate system that is enabled by default, for details see KEN.

KSF (Define Operating Coordinate System At Current Position)

Description:	<p>Defines an operating coordinate system of the KSF type at the current position of the moving platform of the Hexapod.</p> <p>For more details, see the "Work-and-Tool Concept" (p. 3) and in particular the "Consideration of Coordinate Systems of the KSD and KSF Type from the "Work" and "Tool" Perspective" (p. 4).</p> <p>If an operating coordinate system of the KSF type was enabled using KEN, it determines the position values for the moving platform of the Hexapod (query of the current position using POS?), and the pivot point defined using SPI is used as center of rotations.</p> <p>If a coordinate system of the KSF type is not enabled itself, but it nonetheless is linked to the enabled operating coordinate system as a predecessor, its offset values are included in the calculation of the offset values for the enabled operating coordinate system (see KLT?).</p> <p>The definition using KSF is possible only when the Hexapod is not in motion.</p> <p>The coordinate system is defined in the volatile memory. Using WPA, the definition can be written to the nonvolatile memory.</p>
Format:	KSF <CSName>
Arguments:	<CSName> is the name of the coordinate system to be defined.
Response:	None
Notes:	<p>If a coordinate system of the KSF type is defined, its pivot point coordinates (R, S, T, see SPI) are set to the values valid for the currently enabled operating coordinate system. If the enabled operating coordinate system does not support the pivot point set using SPI (KSD, KST/KSW types), the pivot point coordinates of the KSF coordinate system are set to $R = S = T = 0$.</p>

When an operating coordinate system of the KSF type is enabled:

- In addition to the offset values for the X, Y, Z, U, V, W axes, the coordinate system specifies values for the following settings:
 - NLM: Limit for the low end of the axis travel range ("soft limit")
 - PLM: Limit for the high end of the axis travel range ("soft limit")
 - SSL: Activation state of the soft limits of the axis
 - SPI: Pivot point coordinates R, S, T
 - SST: Step size for motions initiated by a manual control unit

- The current settings can be changed using the corresponding commands. The pivot point coordinates can only be changed using SPI if $U = V = W = 0$ applies for the current position of the platform.
- The current settings can be queried using KLS?.
- The current settings are saved for the coordinate system using WPA.
- For initiating motions, see "Commanding Motion" (p. 7).

Options for defining a coordinate system using KSF:

- <CSName> is a new name. The operating coordinate system is newly created under this name and automatically has the ZERO operating coordinate system as its predecessor.
- <CSName> is the name of an existing coordinate system that is not in use. The coordinate system is overwritten by the definition using KSF and automatically attached as successor to the ZERO operating coordinate system.

Using KLN, operating coordinate systems of the KSD, KSF, KST, KSW types can be linked to other operating coordinate systems of these types or reattached as successor to the ZERO coordinate system.

KEN ZERO and DPA SKS re-enable the ZERO operating coordinate system that is enabled by default, for details see KEN.

KST (Define "Tool" Operating Coordinate System)

Description: Defines an operating coordinate system of the KST type ("tool coordinate system") for the work-and-tool concept. For more information, see "The Work-and-Tool Concept" (p. 3).

The placement of the tool coordinate system is defined by giving offset values for the X, Y, Z axes. The alignment of the tool coordinate system is defined by giving offset values for the U, V, W axes:

- Rotating around X, i.e., offset given for U, changes the direction of the Y and Z axes
- Rotating around Y, i.e., offset given for V, changes the direction of the X and Z axes
- Rotating around Z, i.e., offset given for W, changes the direction of the X and Y axes

When an operating coordinate system of the KST type is enabled, the work-and-tool concept is used and the following applies:

- In addition to the tool coordinate system, a work coordinate system must be enabled. When no work coordinate system of the KSW type is enabled, an

automatically generated work coordinate system is used as a substitute, for details see p. 3.

- The coordinates of the center of rotation **cannot** be changed using SPI and the pivot point defined using SPI is **not** used.

If a coordinate system of the KST type is not enabled itself, but it nonetheless is linked to the enabled operating coordinate system as a predecessor, its offset values are included in the calculation of the offset values for the enabled operating coordinate system (see KLT?).

The coordinate system is defined in the volatile memory. Using WPA, the definition can be written to the nonvolatile memory.

Format: KST <CSName> [{<AxisID> <Offset>}]

Arguments: <CSName> is the name of the coordinate system to be defined.
 <AxisID> is one axis of the controller. Possible values: X, Y, Z, U, V, W.
 <Offset> is an offset value for placing or aligning the axis; in physical units.
 For axes not given in the KST command, the offset is set to zero.

Response: None

Notes: When a combination of coordinate systems of the KSW/KST, or ZERO/KST, or KSW/ZERO types is enabled:

- In addition to the offset values for the X, Y, Z, U, V, W axes, the combination specifies values for the following settings:
 - NLM: Limit for the low end of the axis travel range ("soft limit")
 - PLM: Limit for the high end of the axis travel range ("soft limit")
 - SSL: Activation state of the soft limits of the axis
 - SST: Step size for motions initiated by a manual control unit
- The current settings can be changed using the corresponding commands.
- The current settings can be queried using KLC?.
- The current settings are saved using WPA for the coordinate system combination.
- For initiating motions, see "Commanding Motion" (p. 7).

Options for defining a coordinate system using KST:

- <CSName> is a new name. The operating coordinate system is newly created under this name and automatically has the ZERO operating coordinate system as its predecessor.
- <CSName> is the name of an existing coordinate system that is not in use. The coordinate system is overwritten by the definition created with KST.
 - If the overwritten coordinate system was also of the KST type, the chain

with its predecessor and successor is retained.

- If the overwritten coordinate system was not of the KST type, its type is changed to KST and it is attached to ZERO as the new predecessor.

Using KLN, operating coordinate systems of the KSD, KSF, KST, KSW types can be linked to other operating coordinate systems of these types or reattached as successor to the ZERO coordinate system.

KEN ZERO and DPA SKS re-enable the ZERO operating coordinate system that is enabled by default, for details see KEN.

KSW (Define "Work" Operating Coordinate System)

Description: Defines an operating coordinate system of the KSW type ("work coordinate system") for the work-and-tool concept. For more information, see "The Work-and-Tool Concept" (p. 3).

The placement of the work coordinate system is defined by giving offset values for the X, Y, Z axes. The alignment of the work coordinate system is defined by giving offset values for the U, V, W axes:

- Rotating around X, i.e., offset given for U, changes the direction of the Y and Z axes
- Rotating around Y, i.e., offset given for V, changes the direction of the X and Z axes
- Rotating around Z, i.e., offset given for W, changes the direction of the X and Y axes

When an operating coordinate system of the KSW type is enabled, the work-and-tool concept is used and the following applies:

- In addition to the work coordinate system, a tool coordinate system must be enabled. When no tool coordinate system of the KST type is enabled, an automatically generated tool coordinate system is used as a substitute, for details see p. 3.
- The coordinates of the center of rotation **cannot** be changed using SPI and the pivot point defined using SPI is **not** used.

If a coordinate system of the KSW type is not enabled itself, but it nonetheless is linked to the enabled operating coordinate system as a predecessor, its offset values are included in the calculation of the offset values for the enabled operating coordinate system (see KLT?).

The coordinate system is defined in the volatile memory. Using WPA, the definition can be written to the nonvolatile memory.

Format: KSW <CSName> [{<AxisID> <Offset>}]

Arguments: <CSName> is the name of the coordinate system to be defined.
 <AxisID> is one axis of the controller. Possible values: X, Y, Z, U, V, W.
 <Offset> is an offset value for placing or aligning the axis; in physical units.
 For axes not given in the KSW command, the offset is set to zero.

Response: None

Notes: When a combination of coordinate systems of the KSW/KST, or ZERO/KST, or KSW/ZERO types is enabled:

- In addition to the offset values for the X, Y, Z, U, V, W axes, the combination specifies values for the following settings:
 - NLM: Limit for the low end of the axis travel range ("soft limit")
 - PLM: Limit for the high end of the axis travel range ("soft limit")
 - SSL: Activation state of the soft limits of the axis
 - SST: Step size for motions initiated by a manual control unit
- The current settings can be changed using the corresponding commands.
- The current settings can be queried using KLC?.
- The current settings are saved using WPA for the coordinate system combination.
- For initiating motions, see "Commanding Motion" (p. 7).

Options for defining a coordinate system using KSW:

- <CSName> is a new name. The operating coordinate system is newly created under this name and automatically has the ZERO operating coordinate system as its predecessor.
- <CSName> is the name of an existing coordinate system that is not in use. The coordinate system is overwritten by the definition created with KSW.
 - If the overwritten coordinate system was also of the KSW type, the chain with its predecessor and successor is retained.
 - If the overwritten coordinate system was not of the KSW type, its type is changed to KSW and it is attached to ZERO as the new predecessor.

Using KLN, operating coordinate systems of the KSD, KSF, KST, KSW types can be linked to other operating coordinate systems of these types or reattached as successor to the ZERO coordinate system.

KEN ZERO and DPA SKS re-enable the ZERO operating coordinate system that is enabled by default, for details see KEN.

MRT (Set Target Relative In Tool Coordinate System)

- Description:** Moves the given axis relative in the tool coordinate system. For more information, see "The Work-and-Tool Concept" (p. 3).
- While the target position to be approached is being determined from the values for <Distance>, the translation is calculated first, then the rotation.
- Servo mode must be switched on for the commanded axis prior to using this command (closed-loop operation).
- Format:** MRT {<AxisID> <Distance>}
- Arguments:** <AxisID> is one axis of the controller. Possible values: X, Y, Z, U, V, W.
 <Distance> gives the distance that the axis is to move; the sum of the distance and the last commanded target position is set as new target position (in physical units).
- Response:** None
- Troubleshooting:**
- Target position outside of the current work space.
 - The **Trajectory Source** parameter (ID 0x19001900) is set to 1 (but must be set to 0 when using MRT).
 - Servo mode is OFF for one of the axes specified.
 - The reference move has not been successfully completed for at least one axis.
- Notes:** When the work-and-tool concept is not applied, motions are executed using MRT in the tool coordinate system which exists according to the enabled operating coordinate system (see "Consideration of Coordinate Systems of the KSD and KSF Type from the "Work" and "Tool" Perspective", p. 4).
- The physical unit in which the <Distance> is to be given can, be queried using PUN?.
- In order to determine whether a motion has been completed, it is recommended to send #5.
- The motion can be aborted by #24, STP and HLT.

MRW (Set Target Relative In Work Coordinate System)

- Description:** Moves the given axis relative in the work coordinate system. For more information, see "The Work-and-Tool Concept" (p. 3).
- While the target position to be approached is being determined from the values for <Distance>, the translation is calculated first, then the rotation.
- Servo mode must be switched on for the commanded axis prior to using this command (closed-loop operation).
- Format:** MRW {<AxisID> <Distance>}
- Arguments:** <AxisID> is one axis of the controller. Possible values: X, Y, Z, U, V, W.
 <Distance> gives the distance that the axis is to move; the sum of the distance and the last commanded target position is set as new target position (in physical units).
- Response:** None
- Troubleshooting:**
- Target position outside of the current work space.
 - The **Trajectory Source** parameter (ID 0x19001900) is set to 1 (but must be set to 0 when using MRW).
 - Servo mode is OFF for one of the axes specified.
 - The reference move has not been successfully completed for at least one axis.
- Notes:** When the work-and-tool concept is not applied, motions are executed using MRW in the work coordinate system which exists according to the enabled operating coordinate system (see "Consideration of Coordinate Systems of the KSD and KSF Type from the "Work" and "Tool" Perspective", p. 4).
- The physical unit in which the <Distance> is to be given can, be queried using PUN?.
- In order to determine whether a motion has been completed, it is recommended to send #5.
- The motion can be aborted by #24, STP and HLT.

TRA? (Get Maximum Commandable Position For Direction Vector)

Description: Queries the maximum absolute position that can be commanded when the platform of the Hexapod moves along the direction vector which is defined by the given axis components.

The maximum commandable position is calculated from the current position and it can be queried only if the platform of the Hexapod is not in motion.

Note: "Maximum" refers to the amount of the position value. Therefore, in this description the largest possible deflection in negative direction is referred to as the "maximum" position as well (and not as the "minimum" position).

Format: TRA? {<AxisID> <Component>}

Arguments: <AxisID> is one axis of the controller. Possible values: X, Y, Z, U, V, W.
 <Component> is the component of the axis on the direction vector. Must be different from zero for at least one queried axis. Can have a negative sign.
 Axes not given in the query have no component on the direction vector and are not contained in the response.

Response {<AxisID>="<Position> LF}

where

<Position> is the maximum commandable absolute position for the axis when the platform of the Hexapod moves along the given direction vector; in physical units.

Notes: Included in the calculation are the current settings for the soft limits (see NLM, PLM, SSL) and for the pivot point defined with SPI (see SPI) if it is used by the enabled operating coordinate system.

The physical unit in which the maximum commandable position is given, can be queried using PUN?.

WPA (Save Settings To Non-Volatile Memory)

Description:	<p>Writes the current settings from the volatile to the nonvolatile memory.</p> <p>The settings saved using WPA are automatically loaded from the nonvolatile memory to the volatile memory when the C-887 is switched on or rebooted.</p> <p>Note: Incorrect settings can cause the system to malfunction. Make sure that the current settings are correct before you execute the WPA command.</p> <p>Settings in the volatile memory not saved using WPA will be lost when the controller is switched off or rebooted, or when settings are restored.</p>
Format:	WPA <Pswd> [{<ItemID> <PamID>}]
Arguments:	<p><Pswd> is the password for writing to the nonvolatile memory. See below for details.</p> <p><ItemID> is the item for which a parameter is to be saved from the volatile to the nonvolatile memory. See below for details.</p> <p><PamID> is the parameter ID, it can be written in hexadecimal or decimal format. See below for details.</p>
Response:	None
Troubleshooting:	Illegal item identifier, wrong parameter ID, invalid password
Notes:	<p>Parameters can be changed in the volatile memory with SPA.</p> <p>In addition to the parameter settings, the following can be written to the nonvolatile memory using WPA:</p> <ul style="list-style-type: none"> ▪ Settings for coordinate systems (for details, see the table below) ▪ Assignment of stage types to the A and B axes (for details, see description of the CST command in the manual of the Hexapod controller) <p>Saving using WPA does not overwrite the default settings for coordinate systems which can be restored using DPA (p. 13).</p> <p>The used password determines what is saved with WPA:</p>

Valid passwords for writing in the nonvolatile memory:

100 Saves the currently valid values of all parameters, the currently valid settings for coordinate systems (for details, see password SKS) and the current assignment of stage types to the A and B axes

101 Saves the currently valid values of all parameters

SKS Saves the currently valid settings for coordinate systems:

- Properties of the coordinate systems and combinations of coordinate systems which are present in the volatile memory, see KLS? and KLC?
- Activation state of coordinate systems, see KEN
- Linking of coordinate systems, see KLN

When ZERO is enabled: The current values for NLM, PLM, SSL, SPI and SST are not saved. This ensures that KEN ZERO fully re-enables the default settings for the operating coordinate system.

A12 Assignment of stage types to the A and B axes

Valid passwords can be queried using MAN? WPA.

Available item IDs and parameter IDs: The item can be an axis, a sensor channel or the overall system; the item type depends on the parameter.

For the identifiers of the items, see "Commandable Items" in the manual of the Hexapod controller.

With HPA? you can obtain a list of all available parameters.