User Manual

Bro, 2018-04-23
C887T0007, valid for C-886, C-887

Coordinate Systems for Hexapod Microrobots and Parallel-Kinematic Positioners

Contents

About this Document

When working with a hexapod microrobot ("hexapod") or a parallel-kinematic positioner (e.g., SpaceFAB) it may be necessary for certain applications to define custom coordinate systems and use them instead of the preset coordinate systems. This document describes the fundamental principles for working with user-defined coordinate systems and the commands provided for this purpose.

For better readability, the designations "hexapod microrobot" and "parallel-kinematic positioner" are referred to "positioner" in this document.

Other Applicable Documents

- Manual for the controller used
- Manual for the positioner used
- "Motion of the Hexapod - Position and Orientation in Space, Center of Rotation" (C887T0021 user manual)
- "PI Hexapod Simulation Tool - Determining the Workspace and Permissible of the Hexapod" (A000T0068 user manual).

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. Activate the defined coordinate system so that its properties become effective for working with the positioner
4. Optional: Save the definition and the activation state of the coordinate system so that they are retained when the 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. Exactly one coordinate system or exactly one coordinate system combination is active for each group. The types and coordinate systems that **cannot** be defined by the user appear in **bold** in the table below.

Group	Types	Purpose	Default settings*
Operating coordinate system	**ZERO**, KSD, KSF When applying the work-and-tool concept (p. 4): KSW ("Work Coordinate System"), KST ("Tool Coordinate System") In this case a combination of active work and active tool coordinate system is used; for further details, see p. 4	Adapting the position display, the direction of motion, and the center of rotation for the positioner's motion platform to the application concerned	ZERO type **ZERO** coordinate system is active
Orientational 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)	KSB(PI) type **PI_Base** coordinate system is active
Leveling coordinate system	**KLD(PI)**, **KLF(PI)**, KLD(USER), KLF(USER)	Permanent correction of errors in the positioner's alignment (e.g., due to installation errors)	KLD(PI) type **PI_Levelling** coordinate system is active

*Default settings are defined by PI and can be restored at any time, see "Backing Up and Restoring Settings" (p. 11).

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

Operating coordinate system	Child
Orientational coordinate system	↑
Leveling coordinate system	↑
HEXAPOD coordinate system (based on the configuration file with the geometry data for the positioner)	Parent

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 positioner. The dimensional drawings in the manuals for the H-xxx hexapod microrobots each show the position of the HEXAPOD coordinate system.

The properties of the coordinate system and pivot point described in the manuals for the H-xxx hexapod microrobots, and the behavior during translation and rotation described there correspond to the properties and the behavior when the ZERO operating coordinate system is activated (default setting).

Based on HEXAPOD, orientational and leveling coordinate systems adapt fundamental properties of the active operating coordinate system and in most applications, their custom definition and activation is either 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 active operating coordinate systems ("work coordinate system" and "tool coordinate system"). Generally, the combination consists of one active KST type (tool coordinate system) coordinate system and one of the KSW type (work coordinate system).

The work-and-tool concept is applied when a KST and/or KSW type coordinate system is active. The current position of the positioner's motion platform is then considered as the position of the tool coordinate system in the work coordinate system (get with the POS? command, see the manual for the 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 permanently connected to the positioner's motion platform, 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 positioner's platform 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 KST or KSW type coordinate system is activated. The combinations remain in the volatile memory even if the KST or KSW type coordinate systems contained in them are no longer active, but still exist.

The work-and-tool concept applies even when only a KST or KSW type coordinate system is active but not a coordinate system of the respective other type. A substitute is generated automatically and used instead. 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 with the KLC? command (p. 22), the coordinate systems that are generated and used as an alternative are therefore included in the response under the name "Zero".

2018-04-23
C887T0007, valid for C-886, C-887

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

The user-defined coordinate systems of types KSD (p. 34) and KSF (p. 36) are easier to understand by imagining that a work coordinate system and a tool coordinate system always exist, even if the work-and-tool concept is not applied. **Here as well, the current position of the motion platform of the positioner is to be understood as 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 active, both the work and the tool coordinate systems are congruent to ZERO if the platform of the positioner has the current position X = Y = Z = U = V = W = 0. During motion, the tool coordinate system moves with the platform of the positioner whereas the work coordinate system is spatially fixed and remains congruent to ZERO.

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

- When a KSD type operating coordinate system is defined, the work and the tool coordinate systems are positioned and aligned simultaneously in space according to the offset values specified. After activating a KSD type operating coordinate system, the work and the tool coordinate systems are exactly congruent when the positioner's motion platform is at the X = Y = Z = U = V = W = 0 position. During motion, the tool coordinate system moves with the positioner's platform as usual whereas the work coordinate system remains spatially fixed at the position and alignment defined by KSD.
- When a KSF type operating coordinate system is defined, the work coordinate system is placed and aligned in space so that it is congruent to the tool coordinate system currently placed and aligned in space according to the previous motion of the positioner's platform. After activating a KSF type operating coordinate system, the current position of the positioner's platform is therefore displayed as X = Y = Z = U = V = W = 0, provided that no motion was commanded between the definition of the coordinate system and its activation. During motion, the tool coordinate system moves with the positioner's platform as usual whereas the work coordinate system remains spatially fixed at 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 active operating coordinate system is the ZERO coordinate system or a KSF type coordinate system, the center of rotation can also be moved from the origin of the tool coordinate system in the X and/or Y and/or Z direction with the SPI command (the center of rotation that can be moved with SPI is also referred to as "pivot point"). Changing the coordinates of the center of rotation with SPI is only possible if the current position of the platform is U = V = W = 0. Because the center of rotation is either directly at the origin of the tool coordinate system or is moved away from it by SPI with fixed offset values, the tool coordination system always moves together with the platform.

Defining, Copying and Deleting Coordinate Systems

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

Coordinate System Type	**Command**	**Description, see**
KSD	KSD	p. 34
KSF	KSF	p. 36
KST	KST	p. 38
KSW	KSW	p. 39
KSB(USER)	KSB	p. 33
KLD(USER)	KLD	p. 24
KLF(USER)	KLF	p. 26

A copy of a coordinate system can be generated in the volatile memory with the KCP command (p. 18).

A coordinate system that is not active or is not linked as a parent of the active coordinate system can be deleted from the volatile memory with the KRM command (p. 33). When deleting a coordinate system, its direct parent and its child are linked to one another.

Using the KLS? command (p. 30), the properties of the coordinate systems in the volatile memory can be queried.

Linking Coordinate Systems

Chain Structure and the Usability of KLN

Coordinate systems can be linked to one another with the KLN command (p. 28).

Details on the chain structure and the usability of KLN (see also the scheme on p. 3 for the basic structure of chains):

- The HEXAPOD coordinate system is always based on the configuration file with the geometry data for the positioner and it is the starting point for all coordinate system chains.
- The PI_Levelling leveling coordinate system is always the child of HEXAPOD.
- By default, the child of the PI_Levelling leveling coordinate system is the PI_Base orientational coordinate system.
  PI_Levelling is automatically set as parent when a KLD(USER) or KLF(USER) type leveling coordinate system is defined.
  When a KLD(USER) or KLF(USER) type leveling coordinate system is activated, PI_Base is automatically linked as child of this coordinate system and no longer directly to PI_Levelling.

- By default, the ZERO operating coordinate system is the child of the PI_Base orientational coordinate system.
  When a KSB(USER) type orientational coordinate system is defined, it automatically has the active orientational coordinate system as its parent (PI_Base by default). Multiple KSB(USER) type orientational coordinate systems can be linked to one another with KLN, whereby PI_Base always remains the parent of the respective chain.
  When a KSB(USER) type orientational coordinate system is activated, ZERO is automatically linked to this coordinate system as child.
- When an operating coordinate system of the KSD, KSF, KST or KSW type is defined for the first time, the parent is automatically the ZERO coordinate system. Multiple KSD, KSF, KST or KSW type operating coordinate systems can be linked to one another with KLN, whereby ZERO always remains the parent of the respective chain.
- A coordinate system cannot be linked to itself.
- Although it is possible to form ring connections consisting of at least two coordinate systems, they cannot be activated with KEN (p. 19).

Linking Operating Coordinate Systems (KSD, KSF, KST, KSW types)

KLN can be used to link any number of KSD, KSF, KST, and KSW children to the default ZERO operating coordinate system. For example, this is intended to simplify the representation of individual parts of a serial setup that are mounted on the positioner. You don't have to make any special effort to calculate the resulting operating coordinate system for your application yourself, instead, the controller amalgamates the operating coordinate systems of the active chain into a "replacement coordinate system". The controller uses 4x4 transformation matrices for the calculation; for details, see below.

Remarks:

The replacement coordinate system type is an operating coordinate system activated explicitly by KEN.

If the operating coordinate systems of a chain contain specifications on pivot points, those specifications are not included in the calculations described below. Only any existing pivot point of the operating coordinate system explicitly activated with KEN is considered; this pivot point is then assigned to the replacement coordinate system.

Example:

The tool coordinate system should be on the tip of a three-part setup mounted on the positioner's motion platform. In relation to the origin of the factory-set coordinate system (see dimensional drawing in documentation for the positioner), the setup is shifted as follows:

- Part one "TA": 2 mm in the X direction, 10 mm in the Z direction
- Part two "WA": 1 mm in the X direction, 3 mm in the Z direction
- Part three "TB": 3 mm in the X direction, 4 mm in the Z direction

A KST type coordinate system is defined for each of the three parts in order to represent the corresponding shift in the X and Z direction. The three coordinate systems are then linked to each other, attached to ZERO and the endpoint of the chain is activated. Internally, the controller amalgamates the three KST type coordinate systems to a replacement coordinate system.

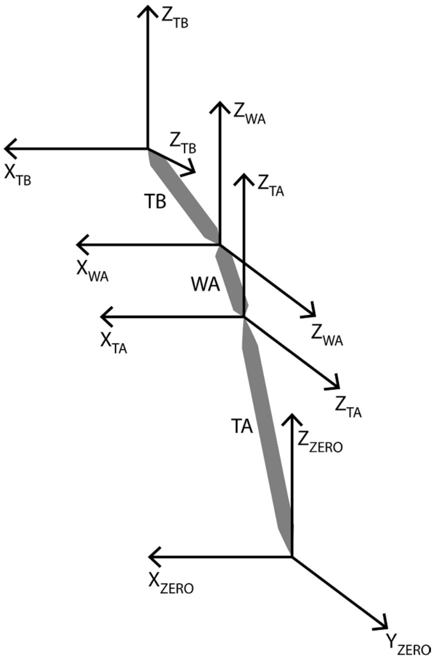


Figure 1: Coordinate systems for the three part setup

Defining the three KST type coordinate systems TA, WA, and TB:

```
kst ta x 2 z 10
kst wa x 1 z 3
kst tb x 3 z 4
```

Linking the coordinate systems:
Note: TB should be the endpoint of the chain; by definition, TA is already attached to ZERO and therefore does not have to be specifically linked to ZERO; the syntax for KLN is "KLN child parent".

```
kln tb wa
kln wa ta
```

Querying the chain for the endpoint TB:

```
kln? tb
TB=WA  TA     ZERO
```

Queries the offset values for the endpoint TB that result from the offset values of the parents linked to this coordinate system:

```
klt? tb
Name=TB  EndCoordinateSystem=ZERO  X=6.000000 Y=0.000000   Z=17.000000  U=0.000000
         V=0.000000      W=0.000000
```

Activating the TB coordinate system:

```
ken tb
```

Querying the coordinate system properties:

```
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="-17.000100" Y="-16.000100" Z="-6.500100" U="-10.000100" V="-10.000100"
        W="-21.000100"/>
       <PLM X="17.000100" Y="16.000100" Z="6.500100" U="10.000100" V="10.000100"
        W="21.000100"/>
       <SSL X="1" Y="1" Z="1" U="1" V="1" W="1"/>
       <SPI R="0.000000" S="0.000000" T="0.000000"/>
       <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>
   <TA Name="TA" Parent="ZERO" Used="True" Type="KST">
       <POS X="2.000000" Y="0.000000" Z="10.000000" U="0.000000" V="0.000000" W="0.000000"/>
   </TA>
   <WA Name="WA" Parent="TA" Used="True" Type="KST">
       <POS X="1.000000" Y="0.000000" Z="3.000000" U="0.000000" V="0.000000" W="0.000000"/>
   </WA>
   <TB Name="TB" Parent="WA" Used="True" Type="KST">
       <POS X="3.000000" Y="0.000000" Z="4.000000" U="0.000000" V="0.000000" W="0.000000"/>
   </TB>
</SingleCoordinateSystem>
```

Position query after the reference move to compare the position values in the case of an active TB coordinate system and active ZERO coordinate system:

```
ken?
TB=KST
PI_LEVELLING=KLD(PI)
PI_BASE=KSB(PI)
pos?
X=6
Y=0
Z=17
U=-0
V=0
W=-0
ken zero
pos?
X=0
Y=0
Z=0
U=-0
V=0
W=-0
```

The replacement coordinate system is calculated on the basis of 4x4 transformation matrices of the individual coordinate systems. The matrix of the respective parent is multiplied FROM THE LEFT with the previously calculated matrix product of the child chain. The factory-set ZERO coordinate system is not included in the calculation of the replacement coordinate system.

4x4 transformation matrices for the example (the chain is TB=WA TA ZERO):

TA has the pose X = 2, Z = 10 (remaining axes = 0)

The associated matrix is:

1 0 0 2
0 1 0 0
0 0 1 10
0 0 0 1

WA has the pose X = 1, Z = 3 (remaining axes = 0)

The associated matrix is:

1 0 0 1
0 1 0 0
0 0 1 3
0 0 0 1

TB has the pose X = 3, Z = 4 (remaining axes = 0)

The associated matrix is:

1 0 0 3
0 1 0 0
0 0 1 4
0 0 0 1

This results in the transformation matrix M of the replacement coordinate system to

$$\mathbf{M=}\quad \begin{matrix}1&0&0&2\\0&1&0&0\\0&0&1&10\\0&0&0&1\end{matrix} \quad * \quad \begin{matrix}1&0&0&1\\0&1&0&0\\0&0&1&3\\0&0&0&1\end{matrix} \quad * \quad \begin{matrix}1&0&0&3\\0&1&0&0\\0&0&1&4\\0&0&0&1\end{matrix} \quad = \quad \begin{matrix}1&0&0&6\\0&1&0&0\\0&0&1&17\\0&0&0&1\end{matrix}$$

Remarks:

Matrix multiplication is NOT commutative; the example above is a special case.

A point in space (x, y, z) is transformed by multiplying its homogenous coordinates (x, y, z, 1) with transformation matrix M as follows: $M*(x, y, z, 1)^T$. The 4x4 transformation matrices are only mentioned here in order to avoid any misunderstanding in the evaluation order.

Activating/Deactivating Coordinate Systems

Exactly one coordinate system or exactly one coordinate system combination is active from the groups described in "Groups and Types of Coordinate Systems" (p. 3).

Only the properties of active coordinate systems are effective for work with the positioner.

Active coordinate systems cannot be changed or deleted. If a coordinate system is linked as parent of the active coordinate system, it is not active itself, however, it cannot be changed or deleted. In this document, active coordinate systems and coordinate systems that are parents of the active coordinate system are also referred to as "coordinate systems used". Active coordinate systems can be queried with the KEN? (p. 20) and the KET? (p. 21) commands. Together with the properties of the coordinate systems, the response to the KLS? command (p. 30) also indicates whether the coordinate systems are currently in use, i.e., whether they are active themselves or are parents of the active coordinate system (value "Used = """).

The chain of parents of the active coordinate system cannot be changed.

Coordinates of the following types can be activated with the KEN command (p. 19):
KSD, KSF, KSW, KST, KSB(USER), KSB(PI), KLF(USER), KLF(PI), KLD(USER), KLD(PI), ZERO

When activating coordinate systems with KEN, motion is not initiated but instead, the display is changed for the current position of the positioner's motion platform.

When a KST or KSW type coordinate system is activated, a combination of work-and-tool coordinate systems is created in the volatile memory. The text below therefore also refers to "activating a coordinate system combination". Using the KLC? command (p. 22), the properties of the coordinate system combinations in the volatile memory can be queried.

Activating a coordinate system for one of the coordinate system groups (operating, orientational, or leveling coordinate systems), simultaneously deactivates the coordinate system or the coordinate system combination that was previously active for this group.

Sending KEN ZERO reactivates the ZERO operating coordinate system that is activated by default. When command level 1 is active (see description of the CCL command in the manual for the controller), KEN ZERO also reactivates the PI_Levelling leveling coordinate system that is active by default, but not the PI_Base orientational coordinate system that is active by default (this can be reactivated by sending KEN PI_Base). When DPA SKS is sent (p. 17), all coordinate systems active by default can be reactivated independent of the currently active command level.

In addition to the offset values for the positions of the X, Y, Z, U, V, W axes and depending on the type of coordinate system, further settings become effective when a coordinate system or a coordinate system combination is activated:

- 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 motion initiated by a manual control unit

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

Backing Up and Restoring Settings

Coordinate systems and coordinate system combinations are always defined, activated, copied, linked, or deleted in the volatile memory of the controller. When the controller is switched off or rebooted, the current settings are lost if they have not been backed up.

To make a backup, the currently valid coordinate system settings can be written to the nonvolatile memory of the controller with the WPA command (p. 44). The settings saved with WPA are loaded from the nonvolatile memory to the volatile memory automatically when the controller is switched on or rebooted.

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

The default settings can be restored with the DPA command (p. 17) or by sending KEN ZERO, see also "Activating/Deactivating Coordinate Systems" (p. 7).

Commanding Motion

Motion of the positioner's platform can always be considered as motion of the tool coordinate system in the work coordinate system, see also "Consideration of KSD and KSF type Coordinate Systems from the "Work" and "Tool" Perspective" (p. 5). The alignment and placement of the work-and-tool coordinate system depends on the active operating coordinate system or the coordinate system combination.

Therefore, absolute motion of the positioner's platform, which is commanded with MOV (for a description, see the manual for the controller), is always regarded as absolute motion of the tool coordinate system in the work coordinate system. Corresponding relative motion can be commanded with MVR (for a description, see the manual for the controller).

Relative motion of the positioner's platform in the tool coordinate system can always be commanded with MRT (p. 41).

Relative motion of the positioner's platform in the work coordinate system can always be commanded with MRW (p. 42).

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

The TRA? command (p. 43) can be used to query the maximum absolute position, if the positioner's platform would move along the specified direction vector.

The current position of the positioner's platform and therefore the position of the tool coordinate system in the work coordinate system, can be queried with the POS? command (for a description, see the manual for the controller).

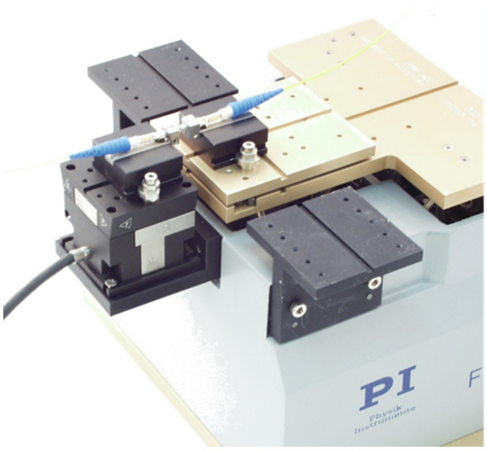
PC Software from PI

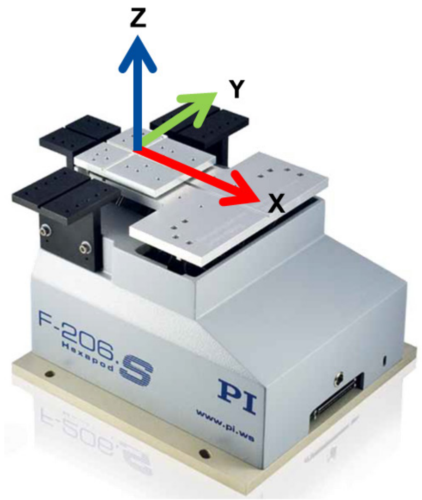
The PC software from PI (includes PIMikroMove®, PI GCS 2 DLL, GCS LabVIEW driver set, ***PI Hexapod Simulation Tool***) was adapted for working with user-defined coordinate systems.

Above all, the modifications in PIMikroMove mainly affect the ***Positioner Platform*** window and the Positioner 3D View card. The ***Show Hexapod Simulation Tool*** menu item for launching the ***PI Hexapod Simulation Tool*** is also available. The settings made in the ***PI Hexapod Simulation Tool*** can be sent to the controller via PIMikroMove.

Examples of User-Defined Coordinate Systems

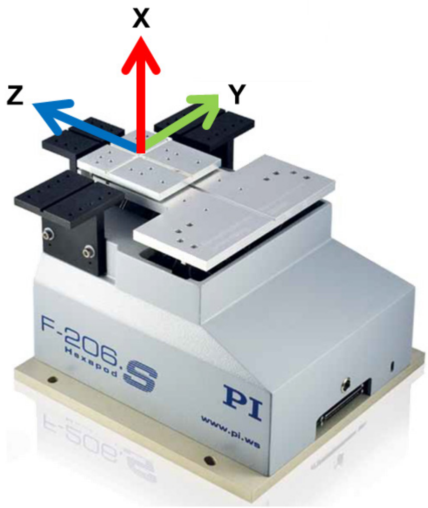
Optics: The Optical Axis is to be the Z Axis



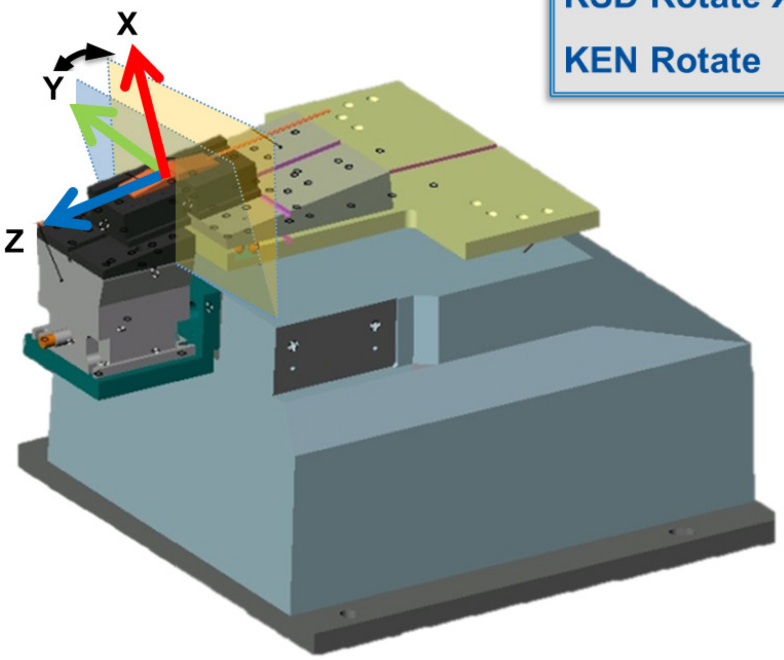


KSB Base V -90

KEN Base



Optics: Scanning in an Inclined Plane

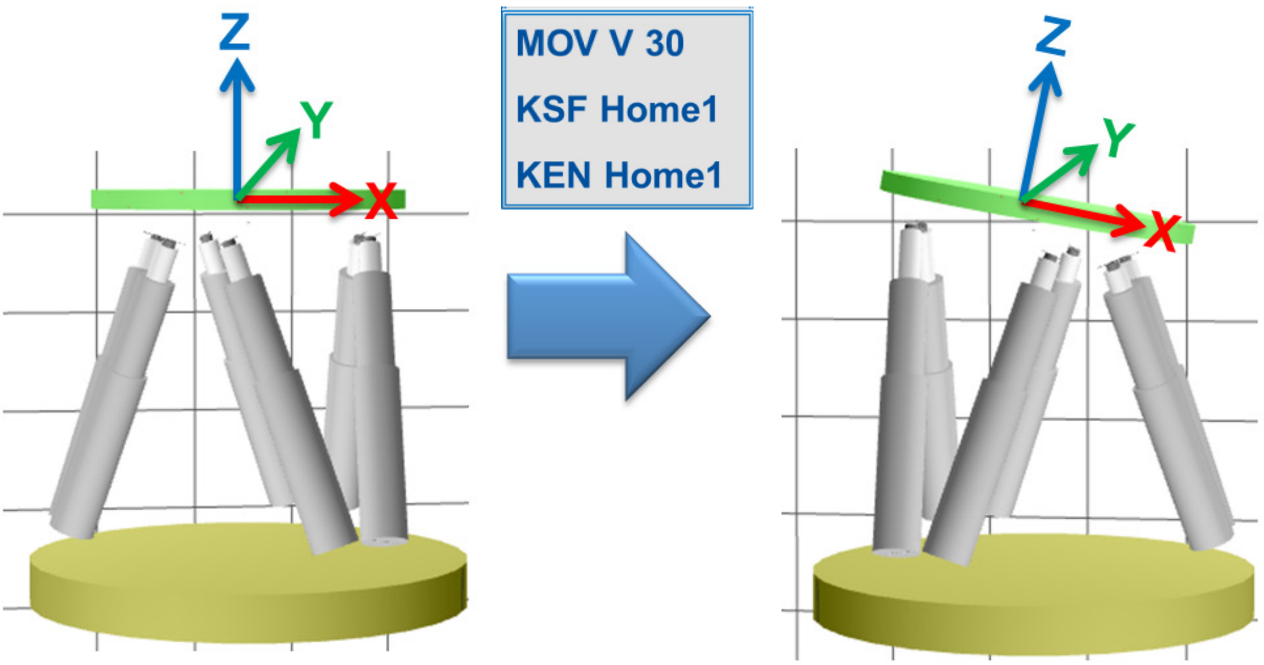


KSD Rotate X -75 Z 20 V -120

KEN Rotate

User-Defined "Home" Position

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

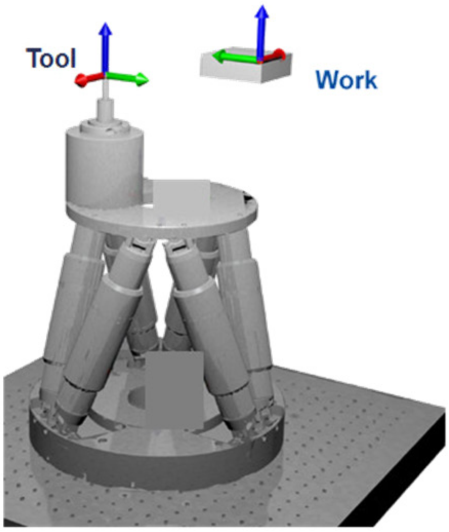


Applying the Work-and-Tool Concept

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

- ➔ Work coordinate system and tool coordinate system are defined and activated.
- ➔ Motion with MRT advances the tool.
- ➔ Motion with MRW is done 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, activate, link and save coordinate systems.

The commands described in this document complement the PI General Command Set (PI GCS), which is described in the manual for the controller. The manual for the controller also contains basic information on the notation used in the command descriptions as well as on the PI GCS syntax.

Command Overview

Command	Format	Description
DPA (p. 17)	DPA <Pswd> [{<ItemID> <PamID>}]	Reset Volatile Memory Settings To Default
KCP (p. 18)	KCP <CSNameSource> <CSNameCopy>	Copy Coordinate System
KEN (p. 19)	KEN <CSName>	Activate Coordinate System
KEN? (p. 20)	KEN? [{<CSName>}]	Get Active Coordinate Systems
KET? (p. 21)	KET? [{<CSType>}]	Get Active Coordinate System Types
KLC? (p. 22)	KLC? [<CSName1>[<CSName2>[<Item1>[<Item2>]]]]	Get Properties Of Work-And-Tool Combinations
KLD (p. 24)	KLD <CSName> [{<AxisID> <Offset>}]	Define Leveling Coordinate System By Specifying Values
KLF (p. 26)	KLF <CSName>	Define Leveling Coordinate System At Current Position
KLN (p. 28)	KLN <ChildCS> <ParentCS>	Link Coordinate Systems
KLN? (p. 29)	KLN? [{<CSName>}]	Get Coordinate System Chains
KLS? (p. 30)	KLS? [<CSName>[<Item1>[<Item2>]]]	Get Coordinate System Properties
KLT? (p. 31)	KLT? [<StartCS> [<EndCS>]]	Get Offsets Resulting From A Chain
KRM (p. 33)	KRM <CSName>	Remove Coordinate System
KSB (p. 33)	KSB <CSName> [{<AxisID> <Angle>}]	Define Orientational Coordinate System
KSD (p. 34)	KSD <CSName> [{<AxisID> <Offset>}]	Define Operating Coordinate System By Specifying Values
KSF (p. 36)	KSF <CSName>	Define Operating Coordinate System At Current Position
KST (p. 38)	KST <CSName> [{<AxisID> <Offset>}]	Define “Tool” Operating Coordinate System
KSW (p. 39)	KSW <CSName> [{<AxisID> <Offset>}]	Define “Work” Operating Coordinate System
MRT (p. 41)	MRT {<AxisID> <Distance>}	Set Target Relative In Tool Coordinate System
MRW (p. 42)	MRW {<AxisID> <Distance>}	Set Target Relative In Work Coordinate System
TRA? (p. 43)	TRA? {<AxisID> <Component>}	Get Maximum Commandable Position For Direction Vector
WPA (p. 44)	WPA <Pswd> [{<ItemID> <PamID>}]	Save Settings To Non-Volatile Memory

Naming Conventions for Coordinate Systems

Permissible characters: 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ_

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. 24), KLF (p. 26), KSB (p. 33), KSD (p. 34), KSF (p. 36), KST (p. 38) and KSW (p. 39).

Command Descriptions

DPA (Reset Volatile Memory Settings To Default)

Description:	Resets parameter values and parameter-independent settings to the default settings.
Format:	DPA <Pswd> [{<ItemID> <PamID>}]
Arguments:	<Pswd> is the password for resetting the volatile memory. See below for details. <ItemID> is the element for which a parameter is to be reset. See below for details. <PamID> is the parameter ID, it can be written in hexadecimal or decimal format. See below for details.
Response:	None
Troubleshooting:	Illegal element identifier, wrong parameter ID, invalid password
Notes:	DPA resets the parameter values and the settings for the coordinate systems stored in the volatile memory of the controller to the default settings. Specification of <ItemID> and <PamID> is unnecessary. The default settings loaded with DPA are independent of the settings in the nonvolatile memory, which can be overwritten with WPA (p. 44). The settings saved with WPA are loaded from the nonvolatile memory to the volatile memory automatically when the controller is switched on or rebooted.

Valid passwords:	100	Resets the values of all parameters and the settings for coordinate systems to the default settings (for details, see password SKS)
	SKS	Default settings restored with DPA SKS: • Orientational coordinate system (KSB() type): PI_BASE is activated • Leveling coordinate system (KLD() or KLF() type): PI_Levelling is activated • ZERO operating coordinate system is activated and based on PI_BASE and PI_Levelling • Pivot point (see SPI), soft limits of axes (see NLM, PLM and SSL), step size for motion initiated by a manual control unit (see SST)

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 link to the parent in a chain of coordinate systems is copied. A link to a child is not copied.

The copy is generated in the volatile memory. The copy can be written to the nonvolatile memory with WPA SKS.

KEN (Activate Coordinate System)

Description: Activates the specified coordinate system. The scope of the settings that are influenced by activating depends on the coordinate system type, see below.

Position values for the positioner's motion platform (get with POS?) refer to the active operating coordinate system.

When applying the work-and-tool concept:

- The work-and-tool concept uses a combination of two active operating coordinate systems. Generally, the combination consists of one active KST type coordinate system and one of the KSW type. If a coordinate system is active for one of the two types, a substitute is automatically used for the other type, for details see p. 4.
- The current position of the positioner's motion platform queried with POS? must be considered as the position of the tool coordinate system in the work coordinate system.

Activating coordinate systems with KEN does not initiate motion.

Format: KEN <CSName>

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

Response: None

Notes: Before activating a coordinate system, KEN checks that the definition of this coordinate system and its linking are correct (for details, see p. 6). If the coordinate system is not correctly defined, it is not activated.

The following coordinate systems can be activated with KEN: KSD, KSF, KSW, KST, KSB(USER), KSB(PI), KLF(USER), KLF(PI), KLD(USER), KLD(PI), ZERO

Exactly one coordinate system or exactly one coordinate system combination is active from the following groups of coordinate systems:

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

Activating a coordinate system for one of the groups simultaneously deactivates the coordinate system or the coordinate system combination that was previously active for this group.

Sending KEN ZERO reactivates the ZERO operating coordinate system that is activated by default. When command level 1 is active (see description of the CCL command in the manual for the controller), KEN ZERO also reactivates the PI_Levelling leveling coordinate system that is active by default, but not the PI_Base orientational coordinate system that is active by default (this can be reactivated by sending KEN PI_Base). When DPA SKS is sent (p. 17), all coordinate systems active by default can be reactivated independent of the currently active command level.

The active 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)
- If supported by the controller:
  SST: Step size for motion initiated by a manual control unit

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

The coordinate systems are activated and deactivated in the volatile memory. The activation state can be written to the nonvolatile memory with WPA SKS.

KEN? (Get Active Coordinate Systems)

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

Format: KEN? [{<CSName>}]

Arguments: <CSName> is the name of an active coordinate system. Illegal: ZERO.

When <CSName> is omitted, all active coordinate systems are listed.

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

whereby

<CSType> specifies the coordinate system type.

Notes: KEN? queries the volatile memory.

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

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

If <CSName> is specified in the query and the corresponding coordinate system is not active, the controller sends an empty response and sets an error (error code query with R?).

KET? (Get Active Coordinate System Types)

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

Format: KET? [{<CSType>}]

Arguments: <CSType> is an active 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 active coordinate system types are listed.

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

whereby

<CSName> specifies the coordinate system name.

Notes: KET? queries the volatile memory.

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

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

If <CSType> is specified in the query and a coordinate system of the corresponding type is not active, the controller sends an empty response and sets an error (error code query with ERR?).

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

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

The work-and-tool concept uses a combination of two active operating coordinate systems. Generally, the combination consists of one active KST type coordinate system and one of the KSW type. If a coordinate system is active 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. 4.

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

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

WPA SKS can used to write the combinations in the volatile memory 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
- If supported by the controller:
  SST: Step size for motion initiated by a manual control unit

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

<Item2> is a controller axis, 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 KLD(USER) type leveling coordinate system for permanent correction of errors in controller alignment (e.g., installation errors).

The leveling coordinate system is defined on the basis of measured values (e.g., using an interferometer) and corrects the linear displacement (X, Y, Z axes) and axis inclination (U, V, W axes) of the positioner's motion platform.

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. The definition can be written to the nonvolatile memory with WPA SKS.

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 specified 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 with 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 activating a KLD(USER) type leveling coordinate system:

1. Perform a reference move (see FRF)
2. Measure the deviation between the position and orientation of the positioner's motion platform and the position and orientation where X = 0, Y = 0, Z = 0, U = 0, V = 0, W = 0 is to be applicable in the future (measured by an external measuring instrument)
3. Switch to command level 1 by sending CCL 1 advanced
4. Define the leveling coordinate system with KLD by specifying the measured discrepancies for the motion platform'S axes (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 axes of the platform can automatically be moved to the zero position after the reference move.
   - Value of parameter 0x07030401 = 0: Axis remains at the reference position after the reference move
   - Value of parameter 0x07030401 = 1: After the reference move, the axis moves to the target position which is specified by parameter 0x07030402
7. Save the settings by sending WPA SKS

The offset values that are displayed in the response to KLS? for the KLD(USER) type coordinate systems are the result of recalculation from all currently active coordinate systems. Therefore, they can change when active coordinate systems are changed. The offset values listed in the response to KLT? however, refer in each case to the specified parent in the chain and are therefore independent of the currently active coordinate systems.

The following applies for leveling coordinate systems defined with KLD:

- The leveling coordinate system is always the direct child to the default PI_Levelling leveling coordinate system (automatic linking).
- KLN **cannot** be used to link leveling coordinate systems to other coordinate systems

DPA SKS reactivates the default active PI_Levelling leveling coordinate system, independent of the currently active command level, for details, see KEN.

KLF (Define Leveling Coordinate System At Current Position)

Description: Defines a KLF(USER) type leveling coordinate system for permanent correction of errors in controller alignment (e.g., installation errors).

To define the leveling coordinate system, the motion 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. The definition can be written to the nonvolatile memory with WPA SKS.

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 with 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.

Definition with KLF is only possible when the positioner is not in motion.

Recommended procedure for defining and activating a KLF(USER) type leveling coordinate system:

1. Perform a reference move (see FRF)
2. Move to the position and orientation of the positioner's motion platform where X = 0, Y = 0, Z = 0, U = 0, V = 0, W = 0 is to be applicable in the future
3. Switch to command level 1 by sending CCL 1 advanced
4. Define the leveling coordinate system with 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 axes of the platform can automatically be moved to the zero position after the reference move.
    - Value of parameter 0x07030401 = 0: Axis remains at the reference position after the reference move
    - Value of parameter 0x07030401 = 1: After the reference move, the axis moves to the target position which is specified by parameter 0x07030402
7. Save the settings by sending WPA SKS

The offset values that are displayed in the response to KLS? for the KLF(USER) type coordinate systems are the result of recalculation from all currently active coordinate systems. Therefore, they can change when active coordinate systems are changed. The offset values listed in the response to KLT? however, refer in each case to the specified parent in the chain and are therefore independent of the currently active coordinate systems.

The following applies for leveling coordinate systems defined with KLF:

- The leveling coordinate system is always the direct child to the default PI_Levelling leveling coordinate system (automatic linking).
- KLN **cannot** be used to link leveling coordinate systems to other coordinate systems

DPA SKS reactivates the default active PI_Levelling leveling coordinate system, independent of the currently active command level, for details, see KEN.

KLN (Link Coordinate Systems)

Description: Links two coordinate systems to create a chain of parent and child.

Format: KLN <ChildCS> <ParentCS>

Arguments: <ChildCS> is the name of the coordinate system that is to be attached to <ParentCS> as child in the chain.

<ParentCS> is the name of the coordinate system that is to be the parent of <ChildCS> 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. 3) and "Linking of Coordinate Systems" (p. 6).

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 positioner, is the "origin" of all chains and the parent of the PI_Levelling leveling coordinate system (fixed link)
- PI_Levelling is the parent of the PI_Base orientational coordinate system
- PI_Base is the parent 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 parents linked to a coordinate system (for details, see KLT?).
- Each coordinate system has exactly one parent and can have one or more than one child.
- When a coordinate system is active, all parents in the chain are also in use and cannot be deleted or overwritten.
- When a coordinate system (unused) is deleted, its parent and child are linked to one another in the chain.

Limitations for creating chains with KLN:

- A coordinate system cannot be linked to itself.
- Although it is possible to form ring connections consisting of at least two coordinate systems, they **cannot** be activated with KEN.
- KLN **cannot** attach a coordinate system in use as child of another coordinate system.
- KLN can be used to attach a coordinate system (not in use) as child of a coordinate system in use.

- Before linking orientational coordinate systems of the KSB(USER) type as child, it is necessary to switch to command level 1 (see CCL).
- KLD(PI), KLF(PI), KLD(USER), KLF(USER) type coordinate systems and the HEXAPOD coordinate system **cannot** be linked with KLN.
- KLN **cannot** be used to attach the PI_Base coordinate system as child of another coordinate system.
- KLN can be used to link KSB(USER) type coordinate systems only to other KSB(USER) coordinate systems or as child of the PI_Base coordinate system.
- KLN **cannot** be used to attach the ZERO coordinate system as child of another coordinate system.

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

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. 6).

Format: KLN? [{<CSName>}]

Arguments: <CSName> is the name of a coordinate system, whose parents are to be listed in the chain.

If the parents of all coordinate systems are to be listed, <CSName> is omitted.

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

whereby

<String> contains the names of the coordinate system parents in the chain. The "origin" of the chain is always at the end of the line

Notes: To provide a clear representation for the KSD, KSF, KSW, and KST type operating coordinate systems, only parents are listed up to the ZERO coordinate system. However, the parents of ZERO can be queried specifically and are also displayed 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 in the volatile memory.

WPA can be used to write the coordinate systems in the volatile memory to the nonvolatile memory.

The properties of the coordinate system combinations for the work-and-tool concept in the volatile memory can be queried with 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
- If supported by the controller:
  SST: Step size for motion 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 specified, <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 parent 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 active itself or it is a parent of the active 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 KLD(USER) and KLF(USER) type leveling coordinate systems are the result of recalculation from all currently active coordinate systems. Therefore, they can change when active coordinate systems are changed.

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 a coordinate system's actual offset values for the position of the axes X, Y, Z, U, V, W that result from the offset values of the parents linked to this coordinate system.

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

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

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

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

Arguments:	<StartCS> is the name of the coordinate system whose offset values that result from the parent are to be queried. <EndCS> is the name of a coordinate system linked as a parent 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 resulting 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 resulting 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 active itself or linked to the active coordinate system as a parent) **cannot** be deleted.

When a coordinate system (unused) is deleted, its parent and its child 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. WPA SKS can be used to transfer the deletion to the nonvolatile memory.

KSB (Define Orientational Coordinate System)

Description: Defines a KSB(USER) type orientational coordinate system for a permanent change in the direction of the X, and/or Y, and/or Z axes.

The direction of the axes is changed by rotating the coordinate system in 90° increments as follows:

- Rotation around X, i.e., angular value specified for U, changes the direction of the Y and Z axes
- Rotation around Y, i.e., angular value specified for V, changes the direction of the X and Z axes
- Rotation around Z, i.e., angular value specified for W, changes the direction of the X and Y axes

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

Format: KSB <CSName> [{<AxisID> <Angle>}]

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

<AxisID> is one axis of the controller. Possible values: U, V, W.

<Angle> is the angle around which the axis is to be rotated. Possible values: 0, 90, 180, 270, -90, -180, -270 (unit: Degree).

For axes not specified in the KSB command, the angle is set to zero.

Response: None

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

Options for defining a coordinate system with KSB:

- <CSName> is a new name. The new orientational coordinate system is created under this name and the active orientational coordinate system becomes its parent automatically
- <CSName> is the name of an existing coordinate system that is not in use. KSB is used to overwrite the coordinate system with the definition and automatically attaches it as child of the active orientational coordinate system

KLN can be used to link KSB(USER) type orientational coordinate systems to other KSB(USER) type coordinate systems or to attach as child of the PI_Base coordinate system.

KSB(USER) type orientational coordinate systems can be activated with KEN. DPA SKS reactivates the PI_Base orientational coordinate system that is activated 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 KSD type coordinate system can be placed and aligned in space arbitrarily. For more details, see the "Work-and-Tool Concept" (p. 4) and in particular the "Consideration of KSD and KSF Type Coordinate Systems from the "Work" and "Tool" Perspective" (p. 5).

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:

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

If a KSD type operating coordinate system was activated with KEN:

- The coordinate system determines the position values for the positioner's motion platform (get the current position with POS?).
- The coordinates of the center of rotation **cannot** be changed with SPI, and the pivot point defined with SPI is **not** used.

If a KSD type coordinate system is not active itself, but is nevertheless linked as a parent of the active operating coordinate system, its offset values are included in the calculation of the offset values for the active operating coordinate system (see KLT?).

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

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 specified in the KSD command, the offset is set to zero.

Response: None

Notes: When a KSD type operating coordinate system is active:

- 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
  - If supported by the controller:
    SST: Step size for motion initiated by a manual control unit
- The current settings can be changed with the corresponding commands.
- The current settings can be queried with KLS?
- WPA SKS is used to save the current settings are saved for the coordinate system.
- For details on initiating motion, see "Commanding Motion" (p. 12).

Options for defining a coordinate system with KSD:

- <CSName> is a new name. The operating coordinate system is recreated under this name and the ZERO operating coordinate system becomes its parent automatically.
- <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 a KSD type, the chain with its parent and child remains unchanged.
- If the overwritten coordinate system was not a KSD type, its type is changed to KSD and is attached to ZERO as the new parent.

KLN can be used to link KSF, KST, KSW type operating coordinate systems to other operating coordinate systems of these types or reattached as child of the ZERO coordinate system.

KEN ZERO and DPA SKS reactivate the ZERO operating coordinate system that is activated by default, for details see KEN.

KSF (Define Operating Coordinate System At Current Position)

Description: Defines a KSF type operating coordinate system at the current position of the positioner's motion platform.

For more details, see the "Work-and-Tool Concept" (p. 4) and in particular the "Consideration of KSD and KSF Type Coordinate Systems from the "Work" and "Tool" Perspective" (p. 5).

If a KSF type operating coordinate system was was activated with KEN, it determines the position values for the positioner's motion platform (get of the current position with POS?), and the pivot point defined with SPI is used as the center of rotation.

If a KSF type coordinate system is not active itself, but is nevertheless linked as a parent of the active operating coordinate system, its offset values are included in the calculation of the offset values for the active operating coordinate system (see KLT?).

Definition with KSF is only possible when the positioner is not in motion.

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

Format: KSF <CSName>

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

Response: None

Notes: If a KSF type coordinate system is defined, its pivot point coordinates (R, S, T, see SPI) are set to the values valid for the currently active operating coordinate system. If the active operating coordinate system does not support the pivot point set by SPI (KSD, KST/KSW types), the pivot point coordinates of the KSF coordinate system are set to R = S = T = 0.

When a KSF type operating coordinate system is active:

- 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
  - If supported by the controller:
    SST: Step size for motion initiated by a manual control unit
- The current settings can be changed with the corresponding commands. The pivot point coordinates can only be changed with SPI if U = V = W = 0 applies for the current position of the platform.
- The current settings can be queried with KLS?
- WPA SKS is used to save the current settings are saved for the coordinate system.
- For details on initiating motion, see "Commanding Motion" (p. 12).

Options for defining a coordinate system with KSF:

- <CSName> is a new name. The operating coordinate system is recreated under this name and the ZERO operating coordinate system becomes its parent automatically.
- <CSName> is the name of an existing coordinate system that is not in use. KSF is used to overwrite the coordinate system with the definition and automatically attached as child of the ZERO operating coordinate system.

KLN can be used to link KSF, KST, KSW type operating coordinate systems to other operating coordinate systems of these types or reattached as child of the ZERO coordinate system.

KEN ZERO and DPA SKS reactivate the ZERO operating coordinate system that is activated 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. 4).

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:

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

When a KST type operating coordinate system is active, the work-and-tool concept is used and the following applies:

- In addition to the tool coordinate system, a work coordinate system must be active. When no KSW type work coordinate system is active, an automatically generated work coordinate system is used as a substitute, for details see p. 4.
- The coordinates of the center of rotation **cannot** be changed with SPI, and the pivot point defined with SPI is **not** used.

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

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

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 specified in the KST command, the offset is set to zero.

Response: None

Notes: When a combination of KSW/KST, or ZERO/KST, or KSW/ZERO type coordinate systems is active:

- 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
  - If supported by the controller:
    SST: Step size for motion initiated by a manual control unit
- The current settings can be changed with the corresponding commands.
- The current settings can be queried with KLC?.
- WPA SKS saves the current settings for the coordinate system combination.
- For details on initiating motion, see "Commanding Motion" (p. 12).

Options for defining a coordinate system with KST:

- <CSName> is a new name. The operating coordinate system is recreated under this name and the ZERO operating coordinate system becomes its parent automatically.
- <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 a KST type, the chain with its parent and child remains unchanged.
  - If the overwritten coordinate system was not a KST type, its type is changed to KST and it is attached to ZERO as the new parent.

KLN can be used to link KSF, KST, KSW type operating coordinate systems to other operating coordinate systems of these types or reattached as child of the ZERO coordinate system.

KEN ZERO and DPA SKS reactivate the ZERO operating coordinate system that is activated 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. 4).

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:

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

and Y axes

When a KSW type operating coordinate system is active, the work-and-tool concept is used and the following applies:

- In addition to the work coordinate system, a tool coordinate system must be active. When no KST type tool coordinate system is active, an automatically generated tool coordinate system is used as a substitute, for details see p. 4.
- The coordinates of the center of rotation **cannot** be changed with SPI, and the pivot point defined with SPI is **not** used.

If a KSW type coordinate system is not active itself, but is nevertheless linked as a parent of the active operating coordinate system, its offset values are included in the calculation of the offset values for the active operating coordinate system (see KLT?).

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

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 specified in the KSW command, the offset is set to zero.

Response: None

Notes: When a combination of KSW/KST, or ZERO/KST, or KSW/ZERO type coordinate systems is active:

- 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
    - If supported by the controller:
      SST: Step size for motion initiated by a manual control unit
- The current settings can be changed with the corresponding commands.
- The current settings can be queried with KLC?.
- WPA SKS saves the current settings for the coordinate system combination.
- For details on initiating motion, see "Commanding Motion" (p. 12).

Options for defining a coordinate system with KSW:

- <CSName> is a new name. The operating coordinate system is recreated under this name and the ZERO operating coordinate system becomes its parent automatically.

- <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 a KSW type, the chain with its parent and child remains unchanged.
  - If the overwritten coordinate system was not a type, its type is changed to KSW and it is attached to ZERO as the new parent.

KLN can be used to link KSF, KST, KSW type operating coordinate systems to other operating coordinate systems of these types or reattached as child of the ZERO coordinate system.

KEN ZERO and DPA SKS reactivate the ZERO operating coordinate system that is activated by default, for details see KEN.

MRT (Set Target Relative In Tool Coordinate System)

Description: Moves the specified axis relative in the tool coordinate system. For more information, see "The Work-and-Tool Concept" (p. 4).

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> specifies 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 workspace.
- The ***Trajectory Source*** parameter (ID 0x19001900) is set to 1 (but must be set to 0 when MRT is used).
- 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, motion is performed with MRT in the tool coordinate system, which exists according to the active operating coordinate system (see "Consideration of KSD and KSF Type Coordinate Systems from the "Work" and "Tool" Perspective", p. 5).

The physical unit for specifying the <Distance> can be queried with 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 specified axis relative in the work coordinate system. For more information, see "The Work-and-Tool Concept" (p. 4).

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> specifies 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 workspace.
- The ***Trajectory Source*** parameter (ID 0x19001900) is set to 1 (but must be set to 0 when MRW is used).
- 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, motion is performed with MRW in the work coordinate system, which exists according to the active operating coordinate system (see "Consideration of KSD and KSF Type Coordinate Systems from the "Work" and "Tool" Perspective", p. 5). The physical unit for specifying the <Distance> can be queried with 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 positioner moves along the direction vector defined by the specified axis components. The maximum commandable position is calculated from the current position and it can be queried only if the platform of the positioner is not moving. Note: "Maximum" refers to the amount of the position value. Therefore, in this description the largest possible displacement in a 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 specified in the query have no component on the direction vector and are not included in the response.
Response	{<AxisID>"="<Position> LF} whereby <Position> is the maximum commandable absolute position for the axis when the positioner's platform moves along the specified 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 active operating coordinate system. The physical unit where the maximum commandable position is specified, can be queried with PUN?.
Example:	The motion platform is to move in the direction of the vector (X, Z) = (2, 4). The PLM command sets the upper soft limit for the X axis to the value 1. Send: TRA? X 2 Z 4 The response indicates the maximum absolute position that can be approached in the direction of the vector (X, Z) = (2, 4) from the current position: X=1.00000 Z=1.99869

WPA (Save Settings To Non-Volatile Memory)

Description:	Writes the current settings from the volatile to the nonvolatile memory. The settings saved with WPA are loaded from the nonvolatile memory to the volatile memory automatically when the controller is switched on or rebooted. **Note: Incorrect settings can cause the system to malfunction. Make sure that the current settings are correct before you execute the WPA command.** Settings in the volatile memory not saved with WPA will be lost when the controller is switched off or rebooted, or when settings are restored.
Format:	WPA <Pswd> [{<ItemID> <PamID>}]
Arguments:	<Pswd> is the password for writing to the nonvolatile memory. See below for details. <ItemID> is the element for which a parameter is to be saved from the volatile to the nonvolatile memory. See below for details. <PamID> is the parameter ID, it can be written in hexadecimal or decimal format. See below for details.
Response:	None
Troubleshooting:	Illegal element identifier, wrong parameter ID, invalid password

Notes:	When parameter settings are to be saved: ▪ Parameter values can be changed in the volatile memory with the SPA command. ▪ An element can be an axis, a drive, an input signal channel (if supported by the controller), or the entire system. The element type depends on the parameter. For further information, see "Adapting Settings" in the controller manual. ▪ You can get a list of all available parameters with HPA?. Valid parameter IDs are also given in "Parameter Overview" in the controller manual. In addition to the parameter settings, WPA can write the settings for coordinate systems to the nonvolatile memory (for details, see the table below). Saving with WPA does not overwrite the default settings, which can be restored with DPA (p. 17). Note: Avoid switching the controller off during the WPA procedure.
Valid passwords for writing to the nonvolatile memory:	100 — Saves the currently valid values of all parameters, and the currently valid settings for coordinate systems (for details, see password SKS). With the C-887 controller, the current assignment of stage types to axes A and B is saved in addition. 101 — Saves the currently valid parameter values. Specification of <ItemID> and <PamID> is optional. SKS — When the SKS password is used, <ItemID> and <PamID> are not needed. Saves the currently valid settings for coordinate systems: • Properties of the coordinate systems and combinations of coordinate systems in the volatile memory; see KLS? and KLC? • Activation state of coordinate systems, see KEN • Linking of coordinate systems, see KLN When ZERO is active: The current values for NLM, PLM, SSL, SPI and SST are not saved. This ensures that KEN ZERO reactivates the default settings fully for the operating coordinate system. A12 — C-887 controller only: Assignment of stage types to axes A and B Valid password can be queried with MAN? WPA.