SIEMENS

SIMOTION

Technology Packages System Functions

List Manual

05/2009

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

CAUTION

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Proper use of Siemens products

Note the following:

Warning

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Industry Sector
Postfach 48 48
90026 Nürnberg
GERMANY

Copyright © Siemens AG 2009.
Technical data subject to change.

Preface

Scope and standards

This **document** is part of the **SIMOTION Programming - Reference** documentation package**.**

Scope of validity

- This manual is valid for SIMOTION SCOUT V4.1 SP4:
- SIMOTION SCOUT V4.1 SP4 (engineering system for the SIMOTION product family),
- SIMOTION Kernel from V3.0 to V4.1 SP4.
- SIMOTION technology packages Cam, Cam_ext (Kernel V3.2 and later) and TControl in the version for the respective kernel (including technology packages Gear, Position and BasicMC up to Kernel V3.0).

Sections in this manual

This manual describes the generally applicable System Functions of SIMOTION and technology objects.

- **System Functions - TP Cam_ext** (Chapter 1)
- **System Functions - TP TControl** (Chapter 2)

Reserved Identifiers

Identifiers which must not be used in user programs can be found in the Appendix of the Function Manual Basic Functions under **Reserved Identifiers**.

SIMOTION Documentation

An overview of the SIMOTION documentation can be found in a separate list of references.

This documentation is included as electronic documentation with the supplied SIMOTION SCOUT.

The SIMOTION documentation consists of 9 documentation packages containing approximately 80 SIMOTION documents and documents on related systems (e.g. SINAMICS).

The following documentation packages are available for SIMOTION V4.1 SP4.

- SIMOTION Engineering System
- SIMOTION System and Function Descriptions
- SIMOTION Service and Diagnostics
- SIMOTION Programming
- SIMOTION Programming - References
- SIMOTION C
- SIMOTION P350
- SIMOTION D4xx
- SIMOTION Supplementary Documentation

Hotline and Internet addresses

Siemens Internet address

The latest information about SIMOTION products, product support, and FAQs can be found on the Internet at:

- General information:
  - **http://www.siemens.de/simotion** (German)
  - **http://www.siemens.com/simotion** (international)
- Downloading documentation
  Further links for downloading files from Service & Support.
  http://support.automation.siemens.com/WW/view/en/10805436
- Individually compiling documentation on the basis of Siemens contents with the My Documentation Manager (MDM), refer to
  http://www.siemens.com/mdm
- My Documentation Manager provides you with a range of features for creating your own documentation.
- FAQs
  You can find information on FAQs (frequently asked questions) by clicking **http://support.automation.siemens.com/WW/view/en/10805436/133000**.

Additional support

We also offer introductory courses to help you familiarize yourself with SIMOTION.

For more information, please contact your regional Training Center or the main Training Center in 90027 Nuremberg, Germany.

Information about training courses on offer can be found at:

www.sitrain.com

Technical Support

If you have any questions, please contact our hotline:

	Europe / Africa
Phone	+49 180 5050 222 (subject to charge)
Fax	+49 180 5050 223
0.14/min from German wire-line network, mobile phone prices may differ.	
Internet	http://www.siemens.com/automation/support-request

	Americas
Phone	+1 423 262 2522
Fax	+1 423 262 2200
E-mail	techsupport.sea@siemens.com

	Asia / Pacific
Phone	+86 1064 757575
Fax	+86 1064 747474
E-mail	support.asia.automation@siemens.com

Note

Country-specific telephone numbers for technical support are provided under the following Internet address:

http://www.automation.siemens.com/partner

Questions about this documentation

If you have any questions (suggestions, corrections) regarding this documentation, please fax or e-mail us at:

Fax	+49 9131-98 2176
E-mail	docu.motioncontrol@siemens.com

Contents

■

System Functions - TP Cam_ext

1

1.1 Internal function

1.1.1 _disableExternalEncoderSimulation

This function switches an external encoder out of simulation mode.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place once simulation mode has been switched off.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.1.2 _enableExternalEncoderSimulation

This function switches an external encoder into simulation mode.
It is possible to specify which value the external encoder should output, as well as the velocity and transition condition for adjusting this value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

positionType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	2
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Specifies the type of simulated position value.
With ACTUAL, the current output value remains unchanged.
With DIRECT, the position value is specified directly.

position (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the simulated position value; evaluation depends on the 'positionType' parameter.
Evaluation only occurs with 'positionType:=DIRECT'.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	4
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Type of simulated velocity value specification.
With ACTUAL, the current output value remains unchanged.
With DIRECT, the velocity value is specified directly.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the simulated velocity value; evaluation depends on the 'velocityType' parameter.
Evaluation only occurs with 'velocityType:=DIRECT'.

transitionTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the time for transition to the newly specified simulated values.
This specification is only evaluated with 'positionType:=DIRECT' or 'velocityType:=DIRECT'.
In all other cases, the parameter is irrelevant.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEncoderSimulation
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandEncoderSimulation

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	Transition after switching to simulation mode

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place once simulation mode has been switched on.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.2 PLCopen

1.2.1 MultiAxis

1.2.1.1 _MC_CamIn

The function block _MC_CamIn starts a camming between a master and a slave axis. The cam profile can be scaled and/or positionally offset and optionally also run through once or periodically. The following object of the slave axis must be interconnected with the desired cam and the desired master. The dynamic response of the slave axis during synchronization is determined with the dynamic response parameters 'velocity', 'acceleration', 'deceleration' and 'jerk'.

Parameter:

master

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the reference to the master (name of the technology object).
The function block _MC_CamIn can be used on the following technology objects with respect to the master:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

slave

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	2

Specifies the reference to the slave axis (name of the technology object).
The function block _MC_CamIn can be used on the following technology objects with respect to the slave:

- Synchronous axes (followingAxis)
- Synchronous path axes (_pathAxis)

camTable

Direction:	Input parameters
Data type:	_MC_CAM_REF
Parameter index:	3

Specifies the cam reference (name of the cam).

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Specification of the function block enable.
The slave axis is synchronized with the interconnected master with a rising edge on this input.

masterOffset (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specification of the offset of the master values in the master coordinates.

slaveOffset (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specification of the offset of the slave values in the slave coordinates.

masterScaling (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	1.0

Specification of the scaling for the master values in the master coordinates.

slaveScaling (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	1.0

Specification of the scaling for the slave values in the slave coordinates.

masterAbsolute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	9
System default:	TRUE

Specifies the evaluation of master values.
With TRUE, the master values are applied as absolute values in the domain of the cam.
With FALSE, the master values are evaluated relative to the start value of the cam.

slaveAbsolute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	10
System default:	TRUE

Specifies the interpretation of slave values.
With TRUE, the slave values are applied as absolute values in the range of the cam.
With FALSE, the slave values are applied relative to the start value of the cam. During synchronization, the slave axis also travels the path differential between the start of the cam and the start value of the cam.

cyclicMode (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	11
System default:	TRUE

Specifies the cam processing mode.
With TRUE, the cam repeats after reaching its end point.
With FALSE, the function block is terminated when the cam has been executed once.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	-1.0

Specifies the maximum synchronization velocity.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.velocity' of the interconnected following object is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	-1.0

Specifies the maximum synchronization acceleration.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.positiveaccel' of the interconnected following object is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	-1.0

Specifies the maximum synchronization deceleration.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.negativeAccel' of the interconnected following object is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	-1.0

Specifies the maximum synchronization jerk.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the slave axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the values that were set in the system variables 'userdefault.syncdynamics.positiveaccelstartjerk', 'userdefault.syncdynamics.positiveaccelendjerk', 'userdefault.syncdynamics.negativeaccelstartjerk' and 'userdefault.syncdynamics.negativeaccelendjerk' of the interconnected following object are used. To activate the jerk limiting, the configuration data 'SyncingMotion.smoothAbsoluteSynchronization' of the following object must be switched to 'YES'. Otherwise the parameter specification for 'jerk' will be ignored and a trapezoidal velocity profile will always be used.

mode (optional)

Direction:	Input parameters
Data type:	_MC_CamInMode
Parameter index:	16
System default:	USER_DEFAULT

_MC_CamInMode

USER_DEFAULT (0)	Use default values from axis configuration
IMMEDIATELY_BY_TIME_PROFILE (1)	Immediate synchronizing with time specification

Specifies the type of synchronization parameters.
With USER_DEFAULT, the synchronization setting from the 'userDefault' system variables ('cammingSettings', 'syncDynamics', ...) of the following object are taken over. Excluding the values set at the inputs 'MasterAbsolute' (system variable 'userDefault.cammingSettings.masterMode'), 'SlaveAbsolute' (system variable 'userDefault.cammingSettings.slaveMode') and 'CyclicMode' (system variable 'userDefault.cammingSettings.cammingMode').
With IMMEDIATELY_BY_TIME_PROFILE, synchronization occurs immediately by time, taking into account the dynamic values set on the function block. The synchronization runs with the parameters 'userDefault.syncprofile.syncprofilereference' like 'RELATE_SYNC_PROFILE_TO_TIME', 'userDefault.cammingsettings.synchronizingmode' like 'IMMEDIATELY' and 'userDefault.cammingsettings.synchronizingdirection' like 'SYSTEM_DEFINED'.

inSync

Direction:	Output parameter
Data type:	BOOL
Parameter index:	17
System default:	FALSE

Outputs
Indicates the synchronism of the master and slave axes.
With TRUE, the slave axis is in synchronous operation with the master.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	18
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	19
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is being processed by the command processing, i.e. the function block has active control of the slave axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	20
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	21
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	22
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.1.2 _MC_CamOut

The function block _MC_CamOut terminates a camming and stops the slave axis. The desynchronization conditions can be set in the system variables 'userdefault.cammingSettings' of the interconnected following object.

Parameter:

slave

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Specifies the reference to the slave axis (name of the technology object).
The function block _MC_CamOut can be used on the following technology objects with respect to the slave:

- Synchronous axes (followingAxis)
- Synchronous path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable.
The synchronous operation of the slave axis with the interconnected master is terminated with a rising edge on this input.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the slave axis has been desynchronized from the interconnected master.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is being processed by the command processing, i.e. the function block has active control of the slave axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	8
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.1.3 _MC_GearIn

The function block _MC_GearIn starts a gearing between a master and a slave axis. The gear ratio is specified as a fraction. The following object of the slave axis must be interconnected with the desired master. The dynamic response of the slave axis during synchronization is determined with the dynamic response parameters 'velocity', 'acceleration', 'deceleration' and 'jerk'.

Parameter:

master

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the reference to the master (name of the technology object).
The function block _MC_GearIn can be used on the following technology objects with respect to the master:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

slave

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	2

Specifies the reference to the slave axis (name of the technology object).
The function block _MC_GearIn can be used on the following technology objects with respect to the slave:

- Synchronous axes (followingAxis)
- Synchronous path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Specification of the function block enable.
The slave axis is synchronized with the interconnected master with a rising edge on this input.

ratioNumerator (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	4
System default:	1

Specifies the numerator of the gear ratio.

ratioDenominator (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	5
System default:	1

Specifies the denominator of the gear ratio.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum synchronization velocity.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.velocity' of the interconnected following object is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum synchronization acceleration.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.positiveaccel' of the interconnected following object is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	-1.0

Specifies the maximum synchronization deceleration.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.negativeAccel' of the interconnected following object is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	-1.0

Specifies the maximum synchronization jerk.
The parameter is only taken into account with 'mode' equals IMMEDIATELY_BY_TIME_PROFILE.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the slave axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the values that were set in the system variables 'userdefault.syncdynamics.positiveaccelstartjerk', 'userdefault.syncdynamics.positiveaccelendjerk', 'userdefault.syncdynamics.negativeaccelstartjerk' and 'userdefault.syncdynamics.negativeaccelendjerk' of the interconnected following object are used. To activate the jerk limiting, the configuration data 'SyncingMotion.smoothAbsoluteSynchronization' of the following object must be switched to 'YES'. Otherwise the parameter specification for 'jerk' will be ignored and a trapezoidal velocity profile will always be used.

phaseShift (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the master-value-related phase shift during synchronous operation.

absolute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	11
System default:	TRUE

Specifies type of gearing.
With TRUE, gearing is absolute relative to the axis zero for the relevant axes. A phase shift can be specified in parameter 'phaseShift'.
With FALSE, gearing is relative, in relation to the start position of synchronous position. A phase shift in parameter 'phaseShift' does not result in any additional shift of the slave position relative to the master during the synchronization. However, the phase shift is taken over as absolute shift between master and slave (system variable 'gearingadjustments.master.offset').

mode (optional)

Direction:	Input parameters
Data type:	_MC_GearInMode
Parameter index:	12
System default:	USER_DEFAULT

_MC_GearInMode

USER_DEFAULT (0)	Use default values from axis configuration
IMMEDIATELY_BY_TIME_PROFILE (1)	Immediate synchronizing with time specification

Specifies the type of synchronization parameters.
With USER_DEFAULT, the synchronization setting from the 'userDefault' system variables ('gearingSettings', 'syncDynamics', ...) of the following object are taken over. Excluding the values set at the inputs 'absolute' (system variable 'userDefault.gearingSettings.typeOfGearing'), 'RatioNumerator' (system variables 'userDefault.gearingSettings.defineMode', 'userDefault.gearingSettings.numerator') and 'RatioDenominator' (system variables 'userDefault.gearingSettings.defineMode', 'userDefault.gearingSettings.denominator').
With IMMEDIATELY_BY_TIME_PROFILE, synchronization occurs immediately by time, taking into account the dynamic values set on the function block. The synchronization runs with the parameters 'userDefault.syncprofile.syncprofilereference' like 'RELATE_SYNC_PROFILE_TO_TIME', 'userDefault.cammingsettings.synchronizingmode' like 'IMMEDIATELY' and 'userDefault.cammingsettings.synchronizingdirection' like 'SYSTEM_DEFINED'.

inGear

Direction:	Output parameter
Data type:	BOOL
Parameter index:	13
System default:	FALSE

Outputs
Indicates the synchronism of the master and slave axes.
With TRUE, the slave axis is in synchronous operation with the master.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	14
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	15
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is being processed by the command processing, i.e. the function block has active control of the slave axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	16
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	17
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	18
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.1.4 _MC_GearOut

The function block _MC_GearOut terminates a gearing and stops the slave axis. The desynchronization conditions can be set in the system variables 'userdefault.gearingSettings' of the interconnected following object.

Parameter:

slave

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Specifies the reference to the slave axis (name of the technology object).
The function block _MC_GearOut can be used on the following technology objects with respect to the slave:

- Synchronous axes (followingAxis)
- Synchronous path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable.
The synchronous operation of the slave axis with the interconnected master is terminated with a rising edge on this input.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the slave axis has been desynchronized from the interconnected master.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is being processed by the command processing, i.e. the function block has active control of the slave axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	8
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.1.5 _MC_Phasing

The function block _MC_Phasing offsets the position of the slave axis with respect to the interconnected master during an active camming or gearing. The offset can be absolute or relative to existing offsets. The dynamic response of the motion is determined with the dynamic response parameters 'velocity', 'acceleration', 'deceleration' and 'jerk'.

Parameter:

master

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the reference to the master (name of the technology object).
This parameter presently has no function. The slave axis is always offset with respect to the current master of the synchronous operation.

slave

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	2

Specifies the reference to the slave axis (name of the technology object).
The function block _MC_Phasing can be used on the following technology objects with respect to the slave:

- Synchronous axes (followingAxis)
- Synchronous path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Specification of the function block enable.
With a rising edge on this input, the position of the slave axis is offset with respect to the position of the master.

phaseShift (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the phase shift.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum velocity.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.velocity' of the interconnected following object is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.positiveaccel' of the interconnected following object is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the system variable 'userdefault.syncdynamics.negativeAccel' of the interconnected following object is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the slave axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the values that were set in the system variables 'userdefault.syncdynamics.positiveaccelstartjerk', 'userdefault.syncdynamics.positiveaccelendjerk', 'userdefault.syncdynamics.negativeaccelstartjerk' and 'userdefault.syncdynamics.negativeaccelendjerk' of the interconnected following object are used.

absolute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	9
System default:	TRUE

Indicates the type of phase shift
With TRUE, the phase shift is adopted as an absolute value.
With FALSE, the phase shift is added to an existing offset.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the master and the slave axis have been displaced with respect to one another by the absolute value of the phase shift.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is being processed by the command processing, i.e. the function block has active control of the slave axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	13
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	14
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	15
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2 SingleAxis

1.2.2.1 _MC_Home

The function block _MC_Home establishes a positional relationship between the control and the mechanical system via a measuring system.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the technology object reference (name of the technology object).
The function block _MC_Home can be used on the following technology objects:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable
The homing procedure starts with a rising edge at this input.

position (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specification of the position at the reference point or the position setting value or the position offset value.

homingMode (optional)

Direction:	Input parameters
Data type:	_MC_HomingMode
Parameter index:	4
System default:	ACTIVE_HOMING

_MC_HomingMode

ACTIVE_HOMING (1)	Active homing
PASSIVE_HOMING (2)	On-the-fly homing
DIRECT_HOMING (3)	Direct homing
DIRECT_HOMING_RELATIVE (4)	Relative setting of current position value
ENABLE_OFFSET_OF_ABSOLUTE_ENCODER (5)	Absolute encoder adjustment

Specifies the homing mode.
Only the homing type is specified with the parameter 'homingMode'. The homing procedure itself is performed in accordance with the configuration of the encoder on the axis.
With ACTIVE_HOMING, an active homing operation is started. The homing mode is set during configuration. This mode is only available for axes.
With PASSIVE_HOMING, the next zero mark crossed by the moving axis after start of homing is used as the home position. The homing command is active parallel to the motion. It remains active until the homing procedure is complete.
With DIRECT_HOMING, the home position coordinates are set directly to the current axis coordinate. There is no axis motion.
With DIRECT_HOMING_RELATIVE, the home position coordinates are set relative to the current axis coordinates; the command does not initiate an axis motion.
With ENABLE_OFFSET_OF_ABSOLUTE_ENCODER, the offset is calculated in addition to the existing offset in the system. The total offset is saved in the NVRAM and is available after the controller is disabled. Once a new project has been loaded in the controller, the saved offset is no longer available.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Outputs
Display of the completion of the function block
With TRUE, the homing of the axis has been completed.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command.

The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block.

The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	10
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'.

The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move.

The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.2 _MC_MoveAbsolute

The function block _MC_MoveAbsolute starts a positioning motion of an axis to an absolute position. The dynamic response of the motion is determined with the dynamic response parameters 'velocity', 'acceleration', 'deceleration' and 'jerk'. An active motion command is replaced by the function block. The axis stops after completion of the positioning motion.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the axis reference (name of the technology object).
The function block _MC_MoveAbsolute can be used on the following technology objects:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable.
The positioning operation starts with a rising edge at this input.

position (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specification of the absolute target position of the motion.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	-1.0

Specifies the maximum velocity.
The velocity is reached subject to the travel distance, acceleration, and jerk settings.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'user-defaultdynamics.velocity' is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.positiveaccel' is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

direction (optional)

Direction:	Input parameters
Data type:	_MC_Direction
Parameter index:	8
System default:	USER_DEFAULT

_MC_Direction

USER_DEFAULT (0)	Default value from axis configuration
POSITIVE (1)	Positive direction
SHORTEST_WAY (2)	Shortest path
NEGATIVE (3)	Negative direction
EFFECTIVE (4)	Current direction

Specifies the direction of motion.
With USER_DEFAULT, the direction that was set in axis system variable 'userDefaultDynamics.direction' is used.
With POSITIVE, the axis is traversed in the positive direction relative to the axis coordinate system.
With SHORTEST_WAY, a modulo axis is traversed in the direction of the shortest path.
With NEGATIVE, the axis is traversed in the negative direction relative to the axis coordinate system.
With EFFECTIVE, the axis is traversed in the last programmed direction of rotation/motion.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the programmed target position has been reached.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	13
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	14
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.3 _MC_MoveAdditive

The function block _MC_MoveAdditive positions an axis relative to the target position of the active positioning command. The function block enables a correction of the target position of the previous positioning command by a distance specified at the Distance input. The dynamic response of the motion is determined with the dynamic response parameters 'velocity', 'acceleration', 'deceleration' and 'jerk'. An active motion command is replaced by the function block. The axis stops after completion of the positioning motion.
The function block _MC_MoveAdditive behaves as function block _MC_MoveRelative if the axis is stationary at the start of the job or an active motion command without defined target position is replaced by the function block. The target position then depends on the position of the axis at the time of the replacement and the additional distance to be traversed.

Parameter:
axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the axis reference (name of the technology object).
The function block _MC_MoveAdditive can be used on the following technology objects:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable.
The positioning operation starts with a rising edge at this input.

distance (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the additional distance to be traversed.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	-1.0

Specifies the maximum velocity.
The velocity is reached subject to the travel distance, acceleration, and jerk settings.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'user-defaultdynamics.velocity' is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'user-defaultdynamics.positiveaccel' is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the resulting target position has been reached.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	13
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.4 _MC_MoveRelative

The function block _MC_MoveRelative positions an axis relative to the actual position of the axis. If the axis is already in motion when the job is started, the position that is present in the system at the start of the job processing is used as the start position. It must be taken into account that there is a response time between the processing of the function block and the execution of the motion, which has been programmed by the user task in the function block and which depends on the set interpolation cycle clock.

The dynamic response of the motion is determined with the dynamic response parameters 'velocity', 'acceleration', 'deceleration' and 'jerk'. An active motion command is replaced by the function block. The axis stops after completion of the positioning motion.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the axis reference (name of the technology object).
The function block _MC_MoveRelative can be used on the following technology objects:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable.
The positioning operation starts with a rising edge at this input.

distance (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the distance to be traversed.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	-1.0

Specifies the maximum velocity.
The velocity is reached subject to the travel distance, acceleration, and jerk settings.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.velocity' is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.positiveaccel' is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the resulting target position has been reached.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	13
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.5 _MC_MoveSuperimposed

The function block _MC_MoveSuperImposed starts a positioning motion relative to the active positioning motion of an axis.
This enables a superimposed positioning of an axis, e.g. for the print-mark correction.

The dynamic response of the motion is determined by the dynamic response parameters 'VelocityDiff', 'Acceleration', 'Deceleration' and 'Jerk'. An active motion command (main motion) is not replaced by the function block.

An active superimposed positioning motion is replaced by a restart of the function block _MC_MoveSuperImposed. The remaining distance-to-go of the replaced superimposed positioning motion is lost.
The axis velocity is increased for the superimposed positioning operation. Therefore, the basic motion of the axis should not be performed with the maximum permissible velocity.

Parameter:
axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the axis reference (name of the technology object).
The function block _MC_MoveSuperImposed can be used on the following technology objects:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable.
The positioning operation starts with a rising edge at this input.

distance (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the additional distance to be traversed.

velocityDiff (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	-1.0

Specifies the maximum velocity of the superimposing motion.
The velocity is reached subject to the travel distance, acceleration, and jerk settings.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.velocity' is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.positiveaccel' is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the superimposing motion of the axis has been completed.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	13
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block.

The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move.

The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.6 _MC_MoveVelocity

The technology function _MC_MoveVelocity accelerates or decelerates an axis to a set velocity. The dynamic response of the motion is determined with the dynamic response parameters 'acceleration', 'deceleration' and 'jerk'. If a velocity override is in effect, then the end velocity is calculated under consideration of the override.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the axis reference (name of the technology object).
The function block _MC_MoveVelocity can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
The axis accelerates or decelerates to the programmed set velocity with a rising edge at this input.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	-1.0

Specifies the velocity setpoint.
If a value greater than or equal to zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.velocity' is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.positiveaccel' is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

direction (optional)

Direction:	Input parameters
Data type:	_MC_Direction
Parameter index:	7
System default:	USER_DEFAULT

_MC_Direction

USER_DEFAULT (0)	Default value from axis configuration
POSITIVE (1)	Positive direction
SHORTEST_WAY (2)	Shortest path
NEGATIVE (3)	Negative direction
EFFECTIVE (4)	Current direction

Specifies the direction of motion.
With USER_DEFAULT, the direction that was set in axis system variable 'userDefaultDynamics.direction' is used.
With POSITIVE, the axis is traversed in the positive direction relative to the axis coordinate system.
With NEGATIVE, the axis is traversed in the negative direction relative to the axis coordinate system.
With EFFECTIVE, the last programmed direction setpoint is used.

current (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Type of velocity specification
With TRUE, the actual velocity of the axis is taken over as programmed set velocity.
With FALSE, the set velocity that was programmed at the 'velocity' input is used.

inVelocity

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the axis has reached the programmed setpoint velocity. The output remains unchanged until the function block is aborted, regardless of the subsequent course of the axis velocity.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	13
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	14
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block.

The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move.

The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.7 _MC_PositionProfile

The function block _MC_PositionProfile traverses an axis along a position profile which is specified as an s(t) function.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the axis reference (name of the technology object).
The function block _MC_PositionProfile can be used on the following technology objects:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

timePosition

Direction:	Input parameters
Data type:	_MC_TP_REF
Parameter index:	2

Specification of the reference to the cam (name of the technology object) that describes the position profile.

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Specification of the function block enable.
The positioning operation starts with a rising edge at this input.

absolute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	4
System default:	TRUE

Specification of the traversing method
With TRUE, the cam positions are approached according to the absolute values.
With FALSE, the position profile of the cam is set at the actual position of the axis.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum velocity.
The velocity is reached subject to the travel distance, acceleration, and jerk settings.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'user-defaultdynamics.velocity' is used.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.positiveaccel' is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the specified position profile has been traversed.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	13
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	14
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block.

The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move.

The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.8 _MC_Power

The _MC_Power function block enables or disables an axis technology object or an external encoder technology object. If an active braking is possible before an axis is disabled, it is stopped with the set stop mode.

The parameter 'stopMode' specifies a stop mode for the axis. The stop mode is taken over with a falling edge at the 'enable' input.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the technology object reference (name of the technology object).
The function block _MC_Power can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

enable (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
The axis is enabled with a rising edge at this input. If this is not possible, the attempt to set the enables is repeated as long as 'enable' is set.
The axis is stopped with a falling edge at this input. The axis is disabled after reaching standstill.

mode (optional)

Direction:	Input parameters
Data type:	_MC_EnableMode
Parameter index:	3
System default:	ALL

_MC_EnableMode

ALL (0)	Enable of position controller and all drive enables
DRIVE (3)	Enable of all drive enables: DRIVE and POWER

Specifies the axis enables (does not affect external encoders).
With ALL, the drive and power enables are set, the position controller is enabled, and follow-up mode is deactivated.
With DRIVE, only the drive enable is set. The power enable remains unchanged.

stopMode (optional)

Direction:	Input parameters
Data type:	_MC_StopMode
Parameter index:	4
System default:	WITH_COMMAND_VALUE_ZERO

_MC_StopMode

WITH_COMMAND_VALUE_ZERO (0)	Stop with emergency stop ramp of controller in PLC and switch off the connected drive
WITH_MAXIMAL_DECELERATION (1)	Stop with maximum deceleration in Ipo and switch off the connected drive
IN_DEFINED_TIME (2)	Stop within a defined time (controller or IPO?) and switch off the connected drive
DISABLE_DRIVE_IMMEDIATELY (3)	Cancel all drive enables immediately, motor coasts to a standstill

Specifies stop mode (does not affect external encoders).
With WITH_COMMAND_VALUE_ZERO, the axis is stopped using an emergency stop ramp generated in the controller. The ramp is set during axis configuration.
With WITH_MAXIMAL_DECELERATION, the axis is stopped via interpolation using the maximum dynamic values.
With IN_DEFINED_TIME, the axis is stopped within a specified time irrespective of the start velocity. The default setting for the stop time specified in system variable 'userDefaultDynamics.stopTime' is used.
With DISABLE_DRIVE_IMMEDIATELY, the POWER enable is canceled directly on the axis. The drive coasts to a standstill.

status

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Display of the enable status of the axis.
With TRUE, the enables of the axis are set according to the 'mode' parameter.
With TRUE, the enables of the external encoder are set.
With FALSE, individual or all enables are not set.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	9
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output. The error code contains the number and, if available, the associated reason for the error that occurred in the function block.

The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move.

The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.9 _MC_ReadActualPosition

The function block _MC_ReadActualPosition reads the actual position of an axis or an external encoder.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the technology object reference (name of the technology object).
The function block _MC_ReadActualPosition can be used on the following technology objects:

- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

enable (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable
The value at the 'position' output is updated as long as 'enable' equals TRUE.

valid

Direction:	Output parameter
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Outputs
Display of the validity of the value which can be read at the 'position' output.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Display of a function block error.
With TRUE, either an error has occurred during the initialization of the function block or the actual position of the axis could not be read. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	6
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output.

position

Direction:	Output parameter
Data type:	LREAL
Parameter index:	7
System default:	0.0

Displays the actual position of the axis (positioningState.actualPosition) or the external encoder (motionState.position).

1.2.2.10 _MC_ReadAxisError

The function block _MC_ReadAxisError reads the error status of an axis or an external encoder. The error status is a 32-bit representation of all alarms present on the technology object. These may have been generated by other function blocks. The bit coding of the alarms can be viewed in the form of alarm group association in the alarm configuration for each technology object in the SIMOTION SCOUT software.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the technology object reference (name of the technology object).
The function block _MC_ReadAxisError can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis) (alarms on subordinate following objects are displayed by setting bit 20 of the 'axisErrorId')
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

enable (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
The value at the 'axisErrorId' output is updated as long as 'enable' equals TRUE.

valid

Direction:	Output parameter
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Outputs

Display of the validity of the value which can be read at the 'axisErrorId' output.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Display of an error in the function block.
With TRUE, either an error has occurred during the initialization of the function block or the error status could not be read. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	6
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output.

The error code corresponds to an error from the list of function return values under the _move system function.

axisErrorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	7
System default:	0

Displays the error status of the axis or external encoder.

1.2.2.11 _MC_ReadBoolParameter

The function block _MC_ReadBoolParameter reads values of various parameters of an axis or external encoder. Each parameter is specified by a number.

The return value of the parameter is converted to the data type of the function block (BOOL), i.e. values of the data type LREAL are converted to FALSE when they are exactly 0.0 and to TRUE in all other cases.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the technology object reference (name of the technology object).
The function block _MC_ReadBoolParameter can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

enable (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
The value at the 'value' output is updated as long as 'enable' equals TRUE.

parameterNumber (optional)

Direction:	Input parameters
Data type:	INT
Parameter index:	3
System default:	0

Specifies the number of the parameter to be read out.

For axes:
With 4, the activation status of the software limit switches is read.
With 6, the activation status of the following error monitoring is read.

valid

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Outputs
Display of the validity of the value which can be read at the 'value' output.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Display of an error in the function block.
With TRUE, either an error has occurred during the initialization of the function block or the parameter could not be read. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	7
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output.

The error code corresponds to an error from the list of function return values under the _move system function.

value

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Display of the parameter value.

1.2.2.12 _MC_ReadParameter

The function block _MC_ReadParameter reads values of various parameters of an axis or external encoder. Each parameter is specified by a number.

The return value of the parameter is converted to the data type of the function block (LREAL), i.e. values that are of the data type BOOL are converted to 0.0 for FALSE and to 1.0 for TRUE.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the technology object reference (name of the technology object).
The function block _MC_ReadParameter can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

enable (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable
The value at the 'value' output is updated as long as 'enable' equals TRUE.

parameterNumber (optional)

Direction:	Input parameters
Data type:	INT
Parameter index:	3
System default:	0

Specifies the number of the parameter to be read out.
For axes:
With 1, the position setpoint of the axis is read.
With 2, the position of the positive software limit switch is read. If the limit switch is inactive, the maximum possible LREAL value is output.
With 3, the position of the negative software limit switch is read. If the limit switch is inactive, the minimum possible LREAL value is output.
With 7, the minimum permissible following error is read.
With 8, the maximum permissible axis velocity is read.
With 10, the actual axis velocity is read.
With 11, the velocity setpoint of the axis is read.
With 12, the maximum permissible axis acceleration is read.
With 16, the maximum permissible axis jerk is read.
For external encoders:
With 10, the actual velocity measured by the external encoder is read.

valid

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Outputs
Display of the validity of the value which can be read at the 'value' output.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Display of an error in the function block.
With TRUE, either an error has occurred during the initialization of the function block or the parameter could not be read. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	7
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output. The error code corresponds to an error from the list of function return values under the _move system function.

value

Direction:	Output parameter
Data type:	LREAL
Parameter index:	8
System default:	0.0

Display of the parameter value.

1.2.2.13 _MC_ReadStatus

This function block _MC_ReadStatus reads various statuses of an axis or an external encoder.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the technology object reference (name of the technology object).
The function block _MC_ReadStatus can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

enable (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
The values at the status outputs are updated as long as 'enable' equals TRUE.

valid

Direction:	Output parameter
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Outputs

Display of the validity of the values that can be read at the status outputs.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Display of an error in the function block.
With TRUE, either an error has occurred during the initialization of the function block or the states could not be read. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	6
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output. The error code corresponds to an error from the list of function return values under the _move system function.

errorStop

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

For axes: Display of an axis stop triggered by an error.
For external encoders: Indicates an error on the external encoder.

disabled

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

For axes: Indicates no enable of the axis.
For external encoders: Indicates no enable of the external encoder.

stopping

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

For axes: Indicates an active regular axis stop.

standStill

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

For axes: Indicates the standstill signal of the axis.
For external encoders: Indicates the standstill signal of the external encoder.

discreteMotion

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

For axes: Indicates an active single axis motion with discrete target position.

continuousMotion

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

For axes: Indicates an active single axis motion without a discrete target position.

synchronizedMotion

Direction:	Output parameter
Data type:	BOOL
Parameter index:	13
System default:	FALSE

For axes: Indicates an active synchronous operation on the axis.

homing

Direction:	Output parameter
Data type:	BOOL
Parameter index:	14
System default:	FALSE

For axes: Indicates an active homing procedure on the axis.
For external encoders: Indicates an active homing procedure on the external encoder.

constantVelocity

Direction:	Output parameter
Data type:	BOOL
Parameter index:	15
System default:	FALSE

For axes: Indicates traversing the axis with a constant velocity.
For external encoders: Indicates a constant velocity that is measured/determined by the external encoder.

accelerating

Direction:	Output parameter
Data type:	BOOL
Parameter index:	16
System default:	FALSE

For axes: Indicates axis travel at an increasing velocity.
For external encoders: Indicates an accelerating velocity that is measured/determined by the external encoder.

decelerating

Direction:	Output parameter
Data type:	BOOL
Parameter index:	17
System default:	FALSE

For axes: Indicates traversing the axis with a decelerating velocity.
For external encoders: Indicates a decelerating velocity that is measured/determined by the external encoder.

1.2.2.14 _MC_Reset

The function block _MC_Reset resets all errors on an axis or external encoder that can be acknowledged by means of the software.

Fatal errors can be acknowledged via Power off/on or reloading of the project to the module.

If the 'restart' input is set, the transferred axis is re-initialized via _MC_Reset.

Axes that are operated with incremental encoders return to the "not homed" mode.

The restart depends on the condition set with axis configuration data item 'restartAxisCondition'.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the technology object reference (name of the technology object).
The function block _MC_Reset can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis) (including subordinate following objects)
- Path axes (_pathAxis)
- External encoders (externalEncoderType)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
With a rising edge at this input, the function block acknowledges the errors present on the transferred technology object or optionally triggers a restart.

restart (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Specifies the type of reset.
With TRUE, the transferred technology object is restarted and modified configuration data are accepted.
With FALSE, the alarms pending on the transferred technology object are acknowledged.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Outputs
Indicates termination of the function block.
The alarms have been acknowledged or a restart performed.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block is aborted due to an error in the command processing. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	8
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block.

The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move.

The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.15 _MC_Stop

The function block _MC_Stop terminates all active motion commands on an axis and decelerates it down to standstill. The function block can be replaced or aborted by another motion command, i.e. it is not possible to start a motion command.

The function block is terminated ('done' equals TRUE) when the axis is stationary and the input 'execute' is reset to FALSE. It is then possible again to start a motion command on the axis.

The dynamic response parameters 'deceleration' and 'jerk' define the dynamic response of the stop procedure.

If an illegal numerical value is specified at the input parameter 'deceleration' or 'jerk', _MC_Stop is not executed.

Further information on this function block can be found in the SIMOTION PLCopen Blocks Function Manual.

Note
The _MC_Stop function block has the same behavior as the _stopEmergency system function.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs
Specifies the axis reference (name of the technology object).
The function block _MC_Stop can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable
The axis stops with a rising edge at this input.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used. If the function block _MC_Stop is started with a jerk specification not equal to zero during the acceleration phase of an axis, its velocity can increase to the configured maximum velocity of the drive in an extreme situation. The axis is only decelerated after the reduction of the acceleration by the jerk!
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Outputs
Indicates termination of the function block.
The axis is at standstill and the disable for the motion commands has been removed ('execute' equals FALSE). This output is only set for one cycle.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a stop response with the same or higher priority. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	10
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block. The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move. The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.16 _MC_VelocityProfile

The function block _MC_VelocityProfile traverses an axis along a velocity profile, which is specified as a v(t) function.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the axis reference (name of the technology object).
The function block _MC_VelocityProfile can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

timeVelocity

Direction:	Input parameters
Data type:	_MC_TV_REF
Parameter index:	2

Specification of the reference to the cam (name of the technology object) that describes the velocity profile.

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Specification of the function block enable.
The motion starts with a rising edge at this input.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	-1.0

Specifies the maximum acceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'user-defaultdynamics.positiveaccel' is used.

deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	-1.0

Specifies the maximum deceleration.
If a value greater than zero is specified, this value is used.
With the specification of a value less than zero, the value that was set in the axis system variable 'userdefaultdynamics.negativeaccel' is used.

jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	-1.0

Specifies the maximum jerk.
If a value greater than zero is specified, this value is used.
With the specification of a value equal to zero, the axis travels without jerk limiting along a trapezoidal speed profile.
With the specification of a value less than zero, the default values that were set in the axis system variables 'userdefaultdynamics.profile', 'userdefaultdynamics.positiveaccelstartjerk', 'userdefaultdynamics.positiveaccelendjerk', 'userdefaultdynamics.negativeaccelstartjerk' and 'userdefaultdynamics.negativeaccelendjerk' are used.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the axis has traversed the specified velocity profile.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	8
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	9
System default:	FALSE

Indicates the command activity in the function block.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	10
System default:	FALSE

Indicates aborting of the function block.
With TRUE, the function block has been aborted because of an error in the command processing or by a replacing command. The error description can be read at the 'errorId' output.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Display of an error in the function block.
With TRUE, an error has occurred during initialization of the function block. The function block is terminated. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	12
System default:	0

Indicates a function block error code.
The error code is always output in connection with the outputs 'commandAborted' or 'error'. The error code contains the number and, if available, the associated reason for the error that occurred in the function block.

The error number assigns the lower 16 bits of the error code and corresponds to an error from the list of the function return values under the system function _move.

The reason for the error, which is coded by a number if available, is contained in the upper 16 bits of the error code. The meaning of this number can be found in the list of the possible reasons for the error of alarm 30002 - "Command aborted".

1.2.2.17 _MC_WriteBoolParameter

The function block _MC_WriteBoolParameter writes values of various parameters of an axis. Each axis parameter is specified by a number.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the axis reference (name of the technology object).
The function block _MC_WriteBoolParameter can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
The parameter at the 'value' input is written with a rising edge at this input.

parameterNumber (optional)

Direction:	Input parameters
Data type:	INT
Parameter index:	3
System default:	0

Specifies the number of the parameter to be written.
With 4, the activation status of the software limit switches is written. The limit switches are always activated and deactivated jointly.
With 6, the activation status of the following error monitoring is written. The parameter is modified simultaneously in all configured data sets.

value (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Specification of the parameter value to be written.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the parameter has been written.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Display of an error in the function block.
With TRUE, either an error has occurred during the initialization of the function block or the parameter could not be written. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	8
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output.

The error code corresponds to an error from the list of function return values under the _move system function.

1.2.2.18 _MC_WriteParameter

The function block _MC_WriteParameter writes values of various parameters of an axis. Each axis parameter is specified by a number. The value of the transfer parameter is converted to the data type of the axis parameter, i.e. values that are of the data type BOOL are converted to FALSE for 0.0 and to TRUE for 1.0.

Parameter:

axis

Direction:	Input parameters
Data type:	_AXIS_REF
Parameter index:	1

Inputs

Specifies the axis reference (name of the technology object).
The function block _MC_WriteParameter can be used on the following technology objects:

- Speed-controlled axes (driveAxis)
- Positioning axes (posAxis)
- Synchronous axes (followingAxis)
- Path axes (_pathAxis)

execute (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specification of the function block enable
The parameter at the 'value' input is written with a rising edge at this input.

parameterNumber (optional)

Direction:	Input parameters
Data type:	INT
Parameter index:	3
System default:	0

Specifies the number of the parameter to be written.
With 2, the position of the positive software limit switch is written.
With 3, the position of the negative software limit switch is written.
With 7, the minimum permissible following error is written.
With 8, the maximum permissible axis velocity is written.
With 12, the maximum permissible axis acceleration is written.
With 16, the maximum permissible axis jerk is written.

value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specification of the parameter value to be written.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Outputs
Indicates termination of the function block.
With TRUE, the parameter has been written.

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	6
System default:	FALSE

Indication of the activity of the function block.
With TRUE, the function block has been started.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	7
System default:	FALSE

Display of an error in the function block.
With TRUE, either an error has occurred during the initialization of the function block or the parameter could not be written. The error description can be read at the 'errorId' output.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	8
System default:	0

Indicates a function block error code.
The error code is always output in connection with the 'error' output. The error number corresponds to an error from the list of the function return values under the system function _move.

1.2.3 Advanced functions

1.2.3.1 _MC_Jog

The function block moves the axis with the parameterized dynamic values either continuously or incrementally in a positive or negative direction.
A positive edge at the input parameters Forward or Backward starts the motion. With the negative edge, this is stopped.
When incremental jogging is selected, the axle is stopped after traveling the parameterized distance. If the axle is stopped before reaching the parameterized distance, a restart of the motion completes the residual distance.
A change in direction when incremental jogging is selected is only possible if no residual distance is pending.

Parameter:

Axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifics the positioning axis.

Enable (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	2
System default:	FALSE

Specifies the function block enable.
TRUE Functionality of the function block is processed.
FALSE Functionality of the function block is not processed. An active motion is stopped.

Forward (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	3
System default:	FALSE

Travel in position direction
A positive edge starts the motion. A negative flank stops it.

Backward (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	4
System default:	FALSE

Travel in negative direction
A positive edge starts the motion. A negative flank stops it.

ModeInc (optional)

Direction:	Input parameters
Data type:	BOOL
Parameter index:	5
System default:	FALSE

Specifies continuous or incremental travel
TRUE incremental jogging
FALSE continuous jogging

Increment (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specification of the distance to be traveled when incremental jogging is selected.

Velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	-1.0

Specifies the maximum velocity.
If '-1.0' is specified for the maximum velocity, the value set in system variable 'UserDefaultDynamics.velocity' is used.

Acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	-1.0

Specifies the maximum acceleration.
If '-1.0' is specified for the maximum acceleration, the value set in system variable 'UserDefaultDynamics.positiveaccel' is used.

Deceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	-1.0

Specifies the maximum deceleration.
If '-1.0' is specified for the maximum deceleration, the value set in system variable 'UserDefaultDynamics.negativeAccel' is used.

Jerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	-1.0

Specifies the maximum jerk.
If '-1.0' is specified for the maximum jerk, the values set in system variables 'UserDefaultDynamics.positiveAccelStartJerk' and 'UserDefaultDynamics.positiveAccelEndJerk' are used for acceleration of the axis and 'UserDefaultDynamics.negativeAccelStartJerk' and 'UserDefaultDynamics.negativeAccelEndJerk' are used for deceleration of the axis.

done

Direction:	Output parameter
Data type:	BOOL
Parameter index:	11
System default:	FALSE

Incremental or continuous jogging finished, the axis has been stopped

busy

Direction:	Output parameter
Data type:	BOOL
Parameter index:	12
System default:	FALSE

Indicates the function block activity.
With TRUE, the command is processed by the command processing, i.e., the function block has active control over the axis.
With FALSE, the function block has been completed successfully or was aborted prematurely.

active

Direction:	Output parameter
Data type:	BOOL
Parameter index:	13
System default:	FALSE

Indicates the function block activity.
With TRUE, the command issued in the function block is active.
With FALSE, the command issued in the function block has been completed or was aborted.

commandAborted

Direction:	Output parameter
Data type:	BOOL
Parameter index:	14
System default:	FALSE

Indicates a function block abort.
With TRUE, the function block has been aborted by another motion command or due to an error in the command processing. The error description can be read at output 'errorId'.

error

Direction:	Output parameter
Data type:	BOOL
Parameter index:	15
System default:	FALSE

Indicates an error
An error occurred while the command was being processed. The ErrorID provides information on the cause.

errorId

Direction:	Output parameter
Data type:	DWORD
Parameter index:	16
System default:	0

Indicates the function block error code.
When errors occur during command processing, the error code is composed of the error number (lower 16 bits) and the reason for the error (upper 16 bits). Errors that occur during the initialization phase of the function block are described by an error number only.

1.3 Technology

1.3.1 Addition object - Addition functions

1.3.1.1 _changeEnableModeOfAdditionObjectIn

This function activates/deactivates the selected inputs of the adding object.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

motionIn1 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the first input of the adding object is to be activated.

motionIn2 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the second input of the adding object is to be activated.

motionIn3 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the third input of the adding object is to be activated.

motionIn4 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the fourth input of the adding object is to be activated.

1.3.1.2 _disableAdditionObjectIn

This function deactivates the selected inputs of the adding object.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

motionIn1 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the first input of the adding object is to be deactivated.

motionIn2 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the second input of the adding object is to be deactivated.

motionIn3 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the third input of the adding object is to be deactivated.

motionIn4 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the fourth input of the adding object is to be deactivated.

1.3.1.3 _enableAdditionObjectIn

This function activates the selected inputs of the adding object.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

motionIn1 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the first input of the adding object is to be activated.

motionIn2 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the second input of the adding object is to be activated.

motionIn3 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the third input of the adding object is to be activated.

motionIn4 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the fourth input of the adding object is to be activated.

1.3.2 Addition object - Command tracking

1.3.2.1 _bufferAdditionObjectCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.2.2 _getStateOfAdditionObjectCommand

This function returns the execution state of a command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'commandIdState'
With ACTIVE, the command is being executed.
With NOT_EXISTENT, the 'commandId' is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.
With EXECUTED, execution of the command is complete.
With ABORTED, execution of the command is aborted.

Parameter:
additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.2.3 _removeBufferedAdditionObjectCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.3 Addition object - Object and Alarm Handling

1.3.3.1 _getAdditionObjectErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'errorState'
With NO, the error specified by the 'errorNumber' parameter is not pending.
With YES, the error specified by the 'errorNumber' parameter is pending.

Parameter:
additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.3.2 _getAdditionObjectErrorState

This function provides information on whether TO AdditionObjectType alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'errorState'
With NO, no alarm is pending.
With YES, an alarm is pending.

Return value numberOfErrors:
Total number of pending errors.

Return value error1..8:
In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:
In the return values, the alarm parameters belonging to the error are output.

Parameter:
additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

1.3.3.3 _resetAdditionObject

This function resets an adding object to the initial state.
Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.3.4 _resetAdditionObjectConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

1.3.3.5 _resetAdditionObjectError

This function resets adding object errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

additionObject

Direction:	Input parameters
Data type:	_AdditionObjectType
Parameter index:	1

Specifies the technology object of the 'additionObject' type or a variable of the '_AdditionObjectType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.4 Speed-controlled axis - Command tracking

1.3.4.1 _bufferAxisCommandId

This function enables the commandId and associated command status to be saved for a period after command execution so that it is still available after the command is complete. The maximum number of CommandId and command status pairs that can be saved is set during configuration using typeOfAxis.decodingConfig.numberOfMaxBufferedCommandId.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId' whose command status is saved and thus remains available.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.4.2 _cancelAxisCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its 'CommandId' in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled.

Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.4.3 _getMotionStateOfAxisCommand

The function provides the motion status of an active axis command.

Return value: StructRetMotionCommandState

StructRetMotionCommandState
Return value with error code and motion status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionCommandId-State	Motion status	EnumMotionComman-dIdState	-

EnumMotionCommandIdState

NOT_EXISTENT (94)	'commandId' is not known or command is already completed
BUFFERED (207)	Command is in the command queue
IN_EXECUTION (208)	Command is being executed
IN_ACCELERATION (209)	Motion generated by the command is in the acceleration phase (only for TO axis)
IN_CONSTANT_MOTION (210)	Motion generated by the command is in the constant velocity phase (only for TO axis)
IN_DECELERATION (211)	Motion generated by the command is in the deceleration phase
AXIS_HOMED (212)	Axis is synchronized (only for TO axis)
INTERPOLATION_DONE (213)	Setpoint interpolation of command is complete
SYNCHRONIZING (214)	Synchronizing to synchronous operation (only for TO synchronous operation)
DESYNCHRONIZING (215)	Desynchronizing from synchronous operation (only for TO synchronous operation)
SYNCHRONIZED (216)	Synchronous operation is synchronized (only for TO synchronous operation)
MODIFICATION_ACTIVE (217)	Compensating motion for scaling or offset in synchronous operation is active (only for TO synchronous operation)
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionCommandIdState'
With NOT_EXISTENT, the 'commandId' for this axis instance is unknown or the command has already been completed.
With BUFFERED, the command is in the command queue.
With IN_EXECUTION, the command is being executed but no motion command is active.
With IN_ACCELERATION, motion generated by the command is in the acceleration phase. The value is output for _move and _pos commands.
With IN_CONSTANT_MOTION, motion generated by the command is in the constant velocity phase. The value is output for _move and _pos commands.
With IN_DECELERATION, motion generated by the command is in the deceleration phase. The value is output for _move and _pos commands.
With AXIS_HOMED, the axis has been synchronized. The value is output for homing commands.
With INTERPOLATION_DONE, the setpoint interpolation of the command is complete.

SYNCHRONIZING, DESYNCHRONIZING, SYNCHRONIZED, MODIFICATION_ACTIVE are not indicated on the axis.
With EXECUTED, the command has been successfully executed. The EXECUTED status can only be read if the 'commandId' is stored.
With ABORTED, execution of the command has been aborted. The ABORTED status can only be read if the 'commandId' is stored.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

'commandId' is used to specify the unique system-wide ID of the command for which the status is to be called.

1.3.4.4 _getStateOfAxisCommand

This function returns the execution state of a motion command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'commandIdState'
With ACTIVE, the command is being executed.
With NOT_EXISTENT, the 'commandId' is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for the synchronous start.
With EXECUTED, execution of the command is complete.
With ABORTED, execution of the command has been aborted.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.4.5 _getStateOfMotionBuffer

This function returns the status of the axis command queue.

Return value: StructRetMotionBuffer

StructRetMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumMotionBuffer-State	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':
EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':
Specifies the number of entries in the command queue.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.4.6 _removeBufferedAxisCommandId

This function terminates the saving of the commandId and corresponding command status after command execution. The status for the commandId is no longer available in the system after completion of the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies which commandId and corresponding command status should no longer be available in the system after completion of the command.
With SPECIFIC_ID, a specific commandId and its command status is no longer available once the command has been completed.
With ALL_ID, all commandIds and their command status are no longer available once the command has been completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId'.

1.3.4.7 _resetMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.5 Speed-controlled axis - Motion

1.3.5.1 _continue

This function continues the entire motion or a partial motion of the specified axis, if it was stopped with STOP and STOP_WITHOUT_ABORT in the 'stopMode' parameter.
Continuation of a partial motion is specified using the 'commandId' or the type of motion.
When continuing a motion, the dynamic parameters (e.g., velocity profile, acceleration) of the interrupted command are used.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

continueSpecification (optional)

Direction:	Input parameters
Data type:	EnumContinueSpecification
Parameter index:	2
System default:	ALL_AXIS_MOTION

EnumContinueSpecification

ALL_AXIS_MOTION (9)	Acts on all active motions
BASIC_MOTION (17)	Acts on the basic motion
SPECIFIC_AXIS_MOTION (125)	Acts on the specified motion
SUPERIMPOSED_MOTION (141)	Acts on the superimposed motions

Specifies the motion that is to be continued.
With ALL_AXIS_MOTION, all motions are continued.
With BASIC_MOTION, the basic motion is continued.
With SPECIFIC_AXIS_MOTION, the motion of the command specified in the 'continueId' parameter is continued.
With SUPERIMPOSED_MOTION, the superimposed motion is continued.

continueId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	WHEN_COMMAND_DONE

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.5.2 _disableAxis

This function cancels the axis enables.
The states of the current drive and power enables can be accessed using system variables 'actorMonitoring.driveState' (enable for DRIVE) and 'actorMonitoring.power' (enable for POWER).
The position controller enable can also be cancelled for position-controlled axes. Its current value can be queried via 'servoMonitoring.controlState'.
The position controller enable is ignored for speed-controlled axes.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

disableMode (optional)

Direction:	Input parameters
Data type:	EnumEnableAxisMode
Parameter index:	2
System default:	ALL

EnumEnableAxisMode

ALL (8)	Enable all
DO_NOT_CHANGE (43)	Do not change enables
DRIVE (44)	Enable drive only
POWER (109)	Enable power only
BY_STW_BIT (357)	profiDrive protocol bit mask
STATE_MACHINE_CONTROL_BY_APPLICATION (415)	Evaluation and manipulation of the STW bits by the user program

Specifies resetting of the drive and power enables.
ALL means that the drive and power enables are cancelled.
DO_NOT_CHANGE means that the current enables in effect are not changed.
DRIVE means that the drive enable is cancelled. The power enable is not changed.
POWER means that the power enable is cancelled. The drive enable is not changed.
With BY_STW_BIT, the bits specified in the 'STWBitSet' parameter are canceled in the PROFIdrive protocol.
With STATE_MACHINE_CONTROL_BY_APPLICATION, the user program takes responsibility for evaluating and manipulating the STW bits.

servoControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

servoCommandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	4
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the tracking mode setting.
If ACTIVE, the axis will be set to tracking mode and the setpoints are tracked. With ACTIVE, no motion commands can be executed; this applies to all axis types. Motion monitoring, such as following error, will not be initiated.
In the special mode of the active position controller enable and the active tracking operation, values can be specified in the servo using, for example, 'servoSettings.additionalCommandValue', without the monitoring being initiated and so the position controller can be placed in operation and set independent of the IPO.

If INACTIVE, the tracking operation of the axis will be deactivated. With INACTIVE, motion commands can be executed. For virtual axes, the position controller enable is always set internally and the motion control will be enabled only for inactive tracking operation.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	8
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the setting of the force/pressure controller enable.
This parameter is only relevant for axes with pressure or force control.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

STWBitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	9
System default:	0

Specifies the values of Bit 0 - Bit 6, which are set in the PROFIdrive protocol.
The bits are inverted and ANDed with the existing control word.
The specified bits are reset.

Meaning of the bits in the PROFIdrive protocol:
Bit0 - OFF
Bit1 - Coast stop (OFF2)
Bit2 - Quick stop (OFF3)
Bit3 - Disable operation
Bit4 - Disable ramp generator
Bit5 - Freeze ramp generator
Bit6 - Disable setpoint

The parameter is only used with 'disableMode:=BY_STW_BIT' or 'enableMode:=BY_STW_BIT'.
In all other cases, the parameter is irrelevant.

1.3.5.3 _enableAxis

This function switches the axis enables.
The states of the current drive and power enables can be accessed using system variables 'actorMonitoring.driveState' (enable for DRIVE) and 'actorMonitoring.power' (enable for POWER).
The position controller enable must also be set for position-controlled axes. Its current value can be queried via 'servoMonitoring.controlState'.
The position controller enable is ignored for speed-controlled axes.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

enableMode (optional)

Direction:	Input parameters
Data type:	EnumEnableAxisMode
Parameter index:	2
System default:	ALL

EnumEnableAxisMode

ALL (8)	Enable all
DO_NOT_CHANGE (43)	Do not change enables
DRIVE (44)	Enable drive only
POWER (109)	Enable power only
BY_STW_BIT (357)	profiDrive protocol bit mask
STATE_MACHINE_CONTROL_BY_APPLICATION (415)	Evaluation and manipulation of the STW bits by the user program

Specifies the setting of the drive and power enables.
ALL means that the drive and power enables are set.
DO_NOT_CHANGE means that the current valid enables are not changed.
The axis remains in follow-up mode if all enables are not available.
DRIVE means that the drive enable is set. The power enable is not changed.
POWER means that the power enable is set. The drive enable is not changed.
With BY_STW_BIT, the bits specified in the STWBitSet parameter are set in the PROFIdrive protocol.

servoControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

servoCommandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	4
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the tracking mode setting.
If ACTIVE, the axis will be set to tracking mode and the setpoints are tracked. With ACTIVE, no motion commands can be executed; this applies to all axis types. Motion monitoring, such as following error, will not be initiated.
In the special mode of the active position controller enable and the active tracking operation, values can be specified in the servo using, for example, 'servoSettings.additionalCommandValue', without the monitoring being initiated and so the position controller can be placed in operation and set independent of the IPO.

If INACTIVE, the tracking operation of the axis will be deactivated. With INACTIVE, motion commands can be executed. For virtual axes, the position controller enable is always set internally and the motion control will be enabled only for inactive tracking operation.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	8
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the setting of the force/pressure controller enable.
This parameter is only relevant for axes with pressure or force control.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

STWBitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	9
System default:	0

Specifies the values of Bit 0 - Bit 6, which are set in the PROFIdrive protocol.
The bits are ORed with the existing control word. Thus, the specified bits will be set.

The bits set can be viewed in the SIMOTION SCOUT project navigator under:
Drives-> <Drive name> -> Diagnostics -> Control/status words

Meaning of the bits in the PROFIdrive protocol:
Bit0 - ON
Bit1 - No coast stop (no OFF2)
Bit2 - No quick stop (no OFF3)
Bit3 - Enable operation
Bit4 - Enable ramp generator
Bit5 - Unfreeze ramp generator
Bit6 - Enable setpoint

The parameter is only used with 'enableMode:=BY_STW_BIT' or 'disableMode:=BY_STW_BIT'.
In all other cases, the parameter is irrelevant.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisEnableMovingMode
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumAxisEnableMovingMode

DO_NOT_CHANGE (43)	Retain last setting
POSITION_CONTROLLED (180)	Enable for speed- and position-controlled mode
SPEED_CONTROLLED (181)	Enable for speed-controlled mode

Specifies whether the axis is enabled for position- or speed-controlled motions.
With POSITION_CONTROLLED, the enable is for position- and speed-controlled motions.
With SPEED_CONTROLLED, the enable is for speed-controlled motions. In this mode, the axis can move if the encoder fails and there is no error response.
With DO_NOT_CHANGE, the last mode setting is retained.

1.3.5.4 _move

The axis is accelerated or decelerated to the programmed velocity setpoint.
If a limited constant velocity phase is not specified (parameter 'moveTimeOut-Type:=WITHOUT_TIME_LIMIT'), the velocity is kept constant after the transition to the velocity set-point.
With limitation of the constant velocity phase (parameter 'moveTimeOutType:=WITH_TIME_LIMIT'), the velocity is decelerated to velocity setpoint 0 after the specified time has expired.

Return value: DINT

Description of the return value:

0 -
No error

1 -
Illegal command parameter

2 -
Illegal range specification in command parameters

3 -
Command aborted

4 -
Unknown command

5 -
Command cannot be executed due to current object status

6 -
Command aborted due to termination of user task

7 -
Command rejected due to suspension of command interpretation of the addressed technology object

8 -
Command aborted due to full command buffer

9 -
Insufficient memory

10 -
A connection to a technology object required for this operation does not exist

11 -
No object configuration

12 -
The error cannot be reset due to its configuration

13 -
Axis is not homed

14 -
Measurement job not possible on virtual axis

15 -
Ambiguous 'commandId'

16 -
Command not implemented

17 -
Read access denied

18 -
Write access denied

19 -
Command argument not supported

20 -
The cam has already been interpolated and cannot be manipulated

21 -
The interpolation condition was violated

22 -
The programmed jerk is 0

23 -
The alarm to be deleted is not active

24 -
The command is not possible on a virtual axis

25 -
Synchronized start of this command is not possible

26 -
Higher-level command was aborted because it is not permitted by the active command

27 -
Timeout during communication with the drive

28 -
Actual values are not valid

29 -
This command cannot be executed when velocity control is active

30 -
This command cannot be executed when position control is active

31 -
This command cannot be executed in torque-reduced operation or during travel to fixed limit stop

32 -
This command can only be executed when force or pressure control is active

33 -
This command cannot be executed when force or pressure control is active

34 -
This command can only be executed when pressure limiting is active

35 -
Master values are not valid

36 -
Slave values are not valid

37 -
No slave value can be defined for a master value

38 -
No master value can be defined for a slave value

39 -
This command cannot be executed when synchronous operation is inactive

40 -
This command cannot be executed because of a synchronization error.

41 -
This command cannot be executed when gearing or camming is active

42 -
This command cannot be executed when camming is inactive

43 -

This command can only be used for an interpolated cam

44 -
This command cannot be executed when pressure limiting is active

45 -
There are not enough interpolation points for interpolation of the cam

46 -
Specified path location cannot be reached due to kinematic limitations

47 -
Path axis values are not valid

48 -
Maximum number of active commands exceeded

49 -
Command only possible on a technology object local to the CPU

10000 (greater or equal to) internal error

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumDirection

BY_VALUE (24)	From the sign of the programmed position setpoint / velocity
EFFECTIVE (45)	Last programmed direction setpoint
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the direction of motion.
With BY_VALUE, the direction is derived from the sign of the velocity setpoint ('velocity' parameter).
With EFFECTIVE, the last programmed direction setpoint is used.
With POSITIVE and NEGATIVE, the direction depends on the axis coordinate system.
SHORTEST_WAY is not supported with this command. The command is aborted with alarm 30001 "Illegal command parameter".
With USER_DEFAULT, the default direction defined in system variable 'userDefaultDynamics.direction' is used.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	3
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
With RESULTING, the velocity is that which results when the current axis acceleration or deceleration is terminated with jerk control. The acceleration/deceleration is reduced to zero, taking into account the jerk. The mergeMode 'IMMEDIATELY' must be programmed as transition behavior.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

moveTimeOutType (optional)

Direction:	Input parameters
Data type:	EnumMoveTimeOut
Parameter index:	5
System default:	WITHOUT_TIME_LIMIT

EnumMoveTimeOut

WITHOUT_TIME_LIMIT (169)	Motion without time limit
WITH_TIME_LIMIT (171)	Motion with time limit

Defines the duration from the start of the constant velocity phase to the start of the deceleration phase.
WITHOUT_TIME_LIMIT means that the duration is not limited.
WITH_TIME_LIMIT means that the duration of the constant velocity phase is specified in the 'moveTimeOut' parameter.

moveTimeOut (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the duration of the constant velocity phase; the evaluation depends on the 'moveTimeOutType' parameter.
Only effective if parameter 'moveTimeOutType:=WITH_TIME_LIMIT'. The unit of the duration is specified during configuration of the axis.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.
The conditions AT_DECELERATION_START, WHEN_INTERPOLATION_DONE, and WHEN_MOTION_DONE only result in a transition when 'moveTimeOutType:=WITH_TIME_LIMIT' or if the command is aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	24
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.5.5 _runMotionInPositionLockedVelocityProfile

The axis moves according to a velocity profile that is specified in the cam function Velocity = f (position). Here, the position is an absolute axis position that is specified using the MotionIn interface.
The profile is started at the current MotionIn position.
The dynamic parameters for any required transition motions, e.g., to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the profile or the motion function for the axis motion.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	4
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	8
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positi-veAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStart-Jerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerk-Type' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccel-StartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	16
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	17
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	18
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	19
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	21
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.5.6 _runTimeLockedVelocityProfile

The axis moves according to a velocity profile that is specified as a cam function.
The axis moves along the profile from a selectable start point to the end.
The dynamic parameters for any required transition motions, e.g. to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the profile or the motion function for the axis motion.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the velocity profile is run.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	4
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	8
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccel-StartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAcce-lEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	16
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	17
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	18
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation. It is important to ensure that 'TypeOfAxis.DecodingConfig.behaviourAtTheEndOfProfile=MOVE_WITH_CONSTANT_SPEEd' is set as part of the configuration data, otherwise the function will not be able to transition.

With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	19
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	21
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.5.7 _runVelocityBasedMotionIn

This function activates the motion interface on the axis with the velocity-based motion so that the values can be taken over into the setpoint generation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference (optional)

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2
System default:	0

Specifies the master object.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	3
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	7
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	15
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableMotionIn
Parameter index:	16
System default:	SEQUENTIAL

EnumMergeModeEnableMotionIn

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableMotionIn
Parameter index:	17
System default:	IMMEDIATELY

EnumNextCommandEnableMotionIn

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_COMMAND_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	18
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	20
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	21
System default:	TO_CONNECTION

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the master value is taken from the interface or as value.
With TO_CONNECTION, the MotionIn interface value is used.
With VALUE, the 'motionInType' parameter is evaluated.

motionInType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	22
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the master value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the master value is taken from system variable 'defaultMotionIn'.
With DIRECT, the master value is taken from command parameters 'velocityIn' and 'accelerationIn'.
The parameter is only evaluated when valueReferenceType:=VALUE'.

velocityIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	0.0

Specifies the velocity of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

accelerationIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	0.0

Specifies the acceleration of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

1.3.5.8 _setAxisSTW

The function makes it possible to specify selectable bits in control word 1 and control word 2 of the PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

STW1BitMask (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	2
System default:	0

Selection of the bits to be specified in control word 1 with this command.
The bits to be specified are selected in the form of a bit mask in 'STW1BitSet':
With 1, 'STW1.Bitx' is specified with this command.
With 0, 'STW1.Bitx' is not specified with this command.
The values of the bits are specified in 'STW1BitSet'.

STW1BitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	3
System default:	0

Specifies the values with the bits selected in parameter 'STW1BitMask'.

STW2BitMask (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	4
System default:	0

Selection of the bits to be changed in control word 2.
The selection is performed as a bit mask. The value (0 or 1) of the bit to be changed is specified in parameter 'STW2BitSet'.

STW2BitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	5
System default:	0

Specifies the values with the bits selected in parameter 'STW2BitMask'.

execution (optional)

Direction:	Input parameters
Data type:	EnumToCommandExecution
Parameter index:	6
System default:	TASK_EXECUTION

EnumToCommandExecution

TASK_EXECUTION (413)	Setting the STW bits in the task context of the user program
TO_EXECUTION (414)	Setting the STW bits for synchronizing with TO command processing

Task context for setting the bits in the drive protocol.
With TASK_EXECUTION, the bits are set in the task context of the user program.
With TO_EXECUTION, setting of the bits is synchronized with command processing in the technology object.

1.3.5.9 _stop

This command stops the entire motion or a partial motion of the specified axis using a programmed deceleration ramp. The _stop command affects only single-axis motions and cannot be used for synchronous operation motions.
The motion to be stopped can be interrupted or terminated.
Stopping of a partial motion is specified using either the 'commandId' or the type of motion.
An interrupted motion can be continued with the _continue command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

stopMode (optional)

Direction:	Input parameters
Data type:	EnumStopMode
Parameter index:	2
System default:	STOP_WITHOUT_ABORT

EnumStopMode

STOP_AND_ABORT (135)	Normal stop with abort
STOP_WITHOUT_ABORT (138)	Normal stop without abort

Specifies the stop behavior relative to the current command.
With STOP_AND_ABORT, the active motion command is aborted.
With STOP_WITHOUT_ABORT, the current motion is interrupted and can be continued with the _continue command.

stopSpecification (optional)

Direction:	Input parameters
Data type:	EnumStopSpecification
Parameter index:	3
System default:	ALL_AXIS_MOTION

EnumStopSpecification

ALL_AXIS_MOTION (9)	Acts on all active axis motions
BASIC_MOTION (17)	Acts on the basic axis motion
SPECIFIC_AXIS_MOTION (125)	Acts on a specific axis motion
SUPERIMPOSED_MOTION (141)	Acts on the superimposed axis motions

Specifies the motions for which the stop command is to take effect.
With ALL_AXIS_MOTION, the stop command acts on all active axis motions.
With BASIC_MOTION, only the main motion is stopped.
With SPECIFIC_AXIS_MOTION, the command acts on the command specified in the 'StopId' parameter.
With SUPERIMPOSED_MOTION, the superimposed motion is stopped.

stopId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide ID to stop the command with this 'commandId'. Commands with the standard ID 'stopId.SystemId_low:=0' and 'stopId.SystemId_high:=0' are not stopped.
This parameter is only active with 'stopSpecification:=SPECIFIC_AXIS_MOTION'.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	8
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	18
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeStop
Parameter index:	19
System default:	IMMEDIATELY

EnumMergeModeStop

AT_END_OF_COMMAND (2)	No significance
IMMEDIATELY (60)	Execute command immediately

Specifies when the command takes effect relative to the motion.
AT_END_OF_COMMAND is irrelevant.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	20
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	21
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingModeStopCommand
Parameter index:	23
System default:	CURRENT_MODE

EnumMovingModeStopCommand

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion
CURRENT_MODE (327)	Last set traversing mode

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
CURRENT_MODE means that motion is stopped in the last set traversing mode.
The 'movingMode' parameter has no effect on the speed-controlled axis.

1.3.5.10 _stopEmergency

This function stops the axis with a programmable stop mode.
If a motion command is active, it is aborted and cannot be continued with a _continue command. The axis is not switched to follow-up mode.
The axis is blocked for further motion commands; the status can be revoked with _reset Axis () or _disable Axis ().

Detailed information on this function can be found in the SIMOTION Motion Control, TO Axis, Electric/Hydraulic, External Encoder Function Manual.

Note
The _stopEmergency system function has the same behavior as the _MC_Stop function block.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

stopDriveMode (optional)

Direction:	Input parameters
Data type:	EnumStopDriveMode
Parameter index:	2
System default:	STOP_WITH_MAXIMAL_DECELERATION

EnumStopDriveMode

STOP_IN_DEFINED_TIME (136)	Stop axis using a time-dependent ramp
STOP_WITH_COMMAND_VALUE_ZERO (139)	Stop axis using zero setpoint
STOP_WITH_MAXIMAL_DECELERATION (140)	Stop axis using maximum deceleration
STOP_WITH_DYNAMIC_PARAMETER (326)	Stop axis using programmed dynamic values

Specifies the stop mode of the axis.
With STOP_IN_DEFINED_TIME the axis stops according to the time specified in the 'stopTime' parameter. The specified time is adhered to irrespective of the velocity.
With STOP_WITH_COMMAND_VALUE_ZERO, the axis is stopped using the emergency stop ramp in the controller. This is set during configuration. When the emergency stop ramp is generated (stopDriveMode:=STOP_WITH_COMMAND_VALUE_ZERO), the ramp generator in the servo is based on the 'dynamicData.positionTimeConstant' x velocity setpoint extrapolated actual position.
With STOP_WITH_MAXIMAL_DECELERATION, the axis is stopped via interpolation using the maximum dynamic axis values.
With STOP_WITH_DYNAMIC_PARAMETER, the axis is stopped with the dynamic parameters programmed in the command.

stopTimeType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	3
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of definition for stop time.
This parameter is only effective if 'stopDriveMode:=STOP_IN_DEFINED_TIME'.
With DIRECT, the value set in the 'stopTime' parameter is used as the programmed stop time.
With EFFECTIVE, the last programmed stop time is used.
With USER_DEFAULT, the default setting for the stop time specified in system variable 'userDefault-Dynamics.stopTime' is used.

stopTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the stop time; the evaluation depends on the 'stopDriveMode' parameter.
It is only used when parameter 'stopDriveMode:=STOP_IN_DEFINED_TIME' and 'stopTime-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeStop
Parameter index:	5
System default:	IMMEDIATELY

EnumMergeModeStop

AT_END_OF_COMMAND (2)	No significance
IMMEDIATELY (60)	Execute command immediately

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
AT_END_OF_COMMAND is irrelevant.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingModeStopCommand
Parameter index:	9
System default:	CURRENT_MODE

EnumMovingModeStopCommand

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion
CURRENT_MODE (327)	Last set traversing mode

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
CURRENT_MODE means that motion is stopped in the last set traversing mode.
The movingMode parameter has no effect on the speed-controlled axis.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	10
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	12
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	18
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	20
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	22
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

1.3.6 Speed-controlled axis - Pressure limitation

1.3.6.1 _disableForceLimiting

This command deactivates limitation of the force or pressure.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableForceLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableForceLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.6.2 _enableForceLimitingValue

This command activates limiting of the force or pressure to a fixed value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

limitingValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the force/pressure limiting value.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	6
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.6.3 _enableMotionInPositionLockedForceLimitingProfile

This function activates the position-related force/pressure limitation profile at the interconnected MotionIn position.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	7
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.6.4 _enableTimeLockedForceLimitingProfile

This function activates the force or pressure limiting with a time-related force/pressure limiting profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam at which the profile definition starts.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	7
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.6.5 _getForceControlDataSetParameter

This function reads the force or pressure-related data in the axis data set.

Return value: StructRetGetForceControlDataSetParameter

StructRetGetForceControlDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisForceControlDataSet	-

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceControllerTypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.6.6 _setForceControlDataSetParameter

This function overwrites the force or pressure-related data in the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisForceControlDataSet
Parameter index:	3

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceController-TypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Specifies the data to be used to overwrite the data set.
UNI_DIRECTION means that negative actual values cannot be measured.
BOTH_DIRECTION means that negative actual values can be measured.

1.3.7 Speed-controlled axis - Pressure control

1.3.7.1 _getForceControlDataSetParameter

This function reads the force or pressure-related data in the axis data set.

Return value: StructRetGetForceControlDataSetParameter

StructRetGetForceControlDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisForceControlDataSet	-

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceController-TypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.7.2 _runMotionInPositionLockedForceProfile

The axis executes the predefined function in the cam as a force-path profile. Here, the position is an absolute axis position that is specified using the MotionIn interface.
The profile is started at the current MotionIn position.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the force/pressure profile.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	4
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the rise factor for force/pressure setpoint value transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	6
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.7.3 _runTimeLockedForceProfile

The axis executes the predefined functions in the specified cam as a force/pressure profile.
The profile is executed from a selectable start point to the end.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the force/pressure profile is run.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	4
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the rise factor for force/pressure setpoint transitions. Evaluation depends on the 'derived-CommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	6
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.7.4 _setForceCommandValue

The force or pressure setpoint specified in the command is set on the axis.
The rise factor for force or pressure setpoint transitions for any required transition motions can be programmed in the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the force or pressure setpoint.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise factor for force/pressure setpoint transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	5
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceValue
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandForceValue

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the actual force/pressure value is within the configured force/pressure window.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.7.5 _setForceControlDataSetParameter

This function overwrites the force or pressure-related data in the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisForceControlDataSet
Parameter index:	3

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceController-TypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Specifies the data to be used to overwrite the data set.
UNI_DIRECTION means that negative actual values cannot be measured.
BOTH_DIRECTION means that negative actual values can be measured.

1.3.8 Speed-controlled axis - Hydraulics

1.3.8.1 _disableQFAxis

This function activates the enables and grants the axis access to the valve.
The function returns an error if another axis is already accessing the valve.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

controlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	2
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

commandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the position controller follow-up mode setting.
ACTIVE switches the axis to follow-up mode.
With ACTIVE, no motion commands can be executed; this applies to all axis types.
INACTIVE switches the axis out of follow-up mode.
With INACTIVE, motion commands can be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	7
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the pressure controller enable setting.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

QOutput (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputMode
Parameter index:	8
System default:	DISABLE

EnumDisableQFAxisOutputMode

DISABLE (42)	Enable valve access
DO_NOT_CHANGE (43)	Do not change access

Specifies the reserving and enabling of access to the Q valve.
With DISABLE, access to the Q valve is enabled.
With DO_NOT_CHANGE, no change takes place.

QOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputEnableMode
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
RESET (304)	Cancel enable

Specifies the Q valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With RESET, the enable is canceled.

QOutputValueSetMode (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputSetMode
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumQFAxisOutputSetMode

DO_NOT_CHANGE (43)	Do not change replacement value
SET (178)	Set replacement value

Specifies the setting of a replacement value for the Q valve if a value is not output by an axis.
With DO_NOT_CHANGE, no change takes place.
With SET, the replacement value is set.

QOutputValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the replacement value; the evaluation depends on the 'QOutputValueSetMode' parameter.
It is only used when parameter 'QOutputValueSetMode:=SET'.
In all other cases, the parameter is irrelevant.

QOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	12
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'QOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

QOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the rise limiting value.

FOutput (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputMode
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputMode

DISABLE (42)	Enable valve access
DO_NOT_CHANGE (43)	Do not change access

Specifies the reserving and enabling of access to the pressure limiting valve.
With DO_NOT_CHANGE, no change takes place.
With DISABLE, access to the pressure limiting valve is enabled.

FOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputEnableMode
Parameter index:	15
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
RESET (304)	Cancel enable

Specifies the pressure limiting valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With RESET, the enable is canceled.

FOutputValueSetMode (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputSetMode
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumQFAxisOutputSetMode

DO_NOT_CHANGE (43)	Do not change replacement value
SET (178)	Set replacement value

Specifies the setting of a replacement value for the pressure limiting valve if a value is not output by an axis.
With DO_NOT_CHANGE, no change takes place.
With SET, the replacement value is set.

FOutputValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the replacement value; the evaluation depends on the 'FOutputValueSetMode' parameter.
It is only used when parameter 'FOutputValueSetMode:=SET'.
In all other cases, the parameter is irrelevant.

FOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	18
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'FOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

FOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the rise limiting value.

1.3.8.2 _enableQFAxis

This function activates the enables and grants the axis access to the valve.
The function returns an error if another axis is already accessing the valve.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

controlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	2
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

commandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the position controller follow-up mode setting.
ACTIVE switches the axis to follow-up mode.
With ACTIVE, no motion commands can be executed; this applies to all axis types.
INACTIVE switches the axis out of follow-up mode.
With INACTIVE, motion commands can be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	7
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the pressure controller enable setting.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

QOutput (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputMode
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumQFAxisOutputMode

DO_NOT_CHANGE (43)	Do not change access
ENABLE (48)	Request valve access

Specifies the reserving and enabling of access to the Q valve.
With DISABLE, access to the Q valve is enabled.
With DO_NOT_CHANGE, no change takes place.
With ENABLE, access to the Q valve is reserved.

QOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputEnableMode
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
SET (178)	Set enable

Specifies the Q valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With SET, the enable is set.

QOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	10
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'QOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

QOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the rise limiting value.

FOutput (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputMode
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumQFAxisOutputMode

DO_NOT_CHANGE (43)	Do not change access
ENABLE (48)	Request valve access

Parameter for reserving and enabling access to the pressure limitation valve.
With DO_NOT_CHANGE, no change takes place.
With ENABLE, access to the pressure limitation valve is reserved.

FOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputEnableMode
Parameter index:	13
System default:	DO_NOT_CHANGE

EnumQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
SET (178)	Set enable

Specifies the pressure limiting valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With SET, the enable is set.

FOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	14
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'FOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

FOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the rise limiting value.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisEnableMovingMode
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumAxisEnableMovingMode

DO_NOT_CHANGE (43)	Retain last setting
POSITION_CONTROLLED (180)	Enable for speed- and position-controlled mode
SPEED_CONTROLLED (181)	Enable for speed-controlled mode

Specifies whether the axis is enabled for position- or speed-controlled motions.
With POSITION_CONTROLLED, the enable is for position- and speed-controlled motions.
With SPEED_CONTROLLED, the enable is for speed-controlled motions. In this mode, the axis can move if the encoder fails and there is no error response.
With DO_NOT_CHANGE, the last mode setting is retained.

1.3.8.3 _getQFAxisDataSetParameter

This function reads an axis data set.

Return value: StructRetGetQFAxisDataSetParameter

StructRetGetQFAxisDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructQFAxisDataSet	-

StructQFAxisDataSet
Parameters of an axis data set

Structure	Name	Data type	Unit
dynamicQFData	Dynamic parameters of the control loop	StructAxisDynamicQFData	-
invertQ	Inversion of the Q output value	StructAxisInvertQOutput	-
invertSetPoint	Inversion of the Q output value before the characteristic curve	StructAxisInvertSetPointHydraulicType	-

StructAxisDynamicQFData
Dynamic parameters of the control loop

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant for velocity loop	LREAL	s
QOutputTimeConstant	Equivalent time constant for hydraulic controlled system	LREAL	s

StructAxisInvertQOutput
Inversion of the Q output value

Structure	Name	Data type	Unit
invSetPoint	Activation status	EnumYesNo	-

StructAxisInvertSetPointHydraulicType
Inversion of the Q output value before the characteristic curve

Structure	Name	Data type	Unit
invert	Activation status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.8.4 _setQFAxisDataSetParameter

This function overwrites an axis data set.

Note
The change of the configuration data using the _setQFAxisDataSetParameter command is not displayed in the online view of the expert list.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructQFAxisDataSet
Parameter index:	3

StructQFAxisDataSet
Parameters of an axis data set

Structure	Name	Data type	Unit
dynamicQFData	Dynamic parameters of the control loop	StructAxisDynamicQFData	-
invertQ	Inversion of the Q output value	StructAxisInvertQOutput	-
invertSetPoint	Inversion of the Q output value before the characteristic curve	StructAxisInvertSetPointHydraulicType	-

StructAxisDynamicQFData
Dynamic parameters of the control loop

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant for velocity loop	LREAL	s
QOutputTimeConstant	Equivalent time constant for hydraulic controlled system	LREAL	s

StructAxisInvertQOutput
Inversion of the Q output value

Structure	Name	Data type	Unit
invSetPoint	Activation status	EnumYesNo	-

StructAxisInvertSetPointHydraulicType
Inversion of the Q output value before the characteristic curve

Structure	Name	Data type	Unit
invert	Activation status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the data to be used to overwrite the data set.

1.3.8.5 _setQFAxisFCharacteristics

This system function activates the characteristic curve for a pressure limiting valve. The characteristic curve is defined in a cam that describes the actual pressure value as a function of the valve position. The valve position is specified as a percentage. The value specified in 'maxOutputVoltage' is equivalent to 100 percent.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

characteristics

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the valve characteristic curve.

maxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	3
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value set in the 'maxDerivative' parameter is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

maxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise limiting value; the evaluation depends on the 'maxDerivativeType' parameter.

1.3.8.6 _setQFAxisQCharacteristics

Activates the characteristic curve for a Q valve. The characteristic curve is defined in a cam that describes the axis velocity as a function of the valve position.
The valve position is specified as a percentage. The value specified in 'maxOutputVoltage' is equivalent to 100 percent.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

characteristics

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the valve characteristic curve.

maxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	3
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value set in the 'maxDerivative' parameter is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

maxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise limiting value; the evaluation depends on the 'maxDerivativeType' parameter.

1.3.9 Speed-controlled axis - Compensation and superimposition

1.3.9.1 _disableAxisAdditiveTorque

This function deactivates an additive torque setpoint that was activated with '_enableAxisAdditiveTorque'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.9.2 _disableAxisTorqueLimitNegative

This function deactivates a negative torque limiting that was activated with '_enableAxisTorqueLimitNegativeIn'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.9.3 _disableAxisTorqueLimitPositive

This function deactivates a positive torque limiting that was activated with '_enableAxisTorqueLimitPositiveIn'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.9.4 _disableTorqueLimiting

The "Disable torque limiting" function is used to disable one of the superimposed torque limiting motion commands.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableTorqueLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableTorqueLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command does not take place until the command is completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.9.5 _enableAxisAdditiveTorque

This function activates an additive torque setpoint.
The torque is specified cyclically via the 'AdditiveTorque' interconnection.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the torque value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'additiveTorqueType' parameter is evaluated.

additiveTorqueType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the torque value is be taken from the command or from a system variable.
With DEFAULT_VALUE, the torque value is taken from system variable 'defaultAdditiveTorque'.
The parameter is only evaluated when 'referenceType:=VALUE'.
With DIRECT, the torque value is taken from command parameter 'torqueLimit'.

additiveTorque (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the additive torque.
The parameter is only evaluated when 'referenceType:=VALUE' and 'additiveTorqueType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.9.6 _enableAxisTorqueLimitNegative

This function activates a negative torque limiting which is specified in the extended drive protocol in accordance with the speed controller.
The limiting values are specified cyclically via the 'TorqueLimitNegative' interconnection.

For active B+/B- torque limitation resulting from the _enableAxisTorqueLimitPositive or _enableAxisTorqueLimitNegative command, the following monitoring will be deactivated:

- Following error monitoring
- The velocity error monitoring using reference model
- The time limitations for positioning monitoring and standstill monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the limiting value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'torqueLimitType' parameter is evaluated.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the limit value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the limiting value is taken from system variable 'defaultTorqueLimitNegative'.
The parameter is only evaluated when 'referenceType:=VALUE'.
With DIRECT, the limiting value is taken from command parameter 'torqueLimit'.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the negative limiting value.
The parameter is only evaluated when 'referenceType:=VALUE' and 'torqueLimitType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.9.7 _enableAxisTorqueLimitPositive

This function activates a positive torque limiting, which is specified in the extended drive protocol in accordance with the speed controller.
The limiting values are specified cyclically via the 'TorqueLimitPositive' interconnection.

For active B+/B- torque limitation resulting from the _enableAxisTorqueLimitPositive or _enableAxisTorqueLimitNegative command, the following monitoring will be deactivated:

- Following error monitoring
- The velocity error monitoring using reference model
- The time limitations for positioning monitoring and standstill monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the limiting value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'torqueLimitType' parameter is evaluated.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the limit value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the limiting value is taken from system variable 'defaultTorqueLimitPositive'.
With DIRECT, the limiting value is taken from command parameter 'torqueLimit'.
The parameter is only evaluated when 'referenceType:=VALUE'.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the positive limit value.
The parameter is only evaluated when 'referenceType:=VALUE' and 'torqueLimitType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.9.8 _enableTorqueLimiting

The "Enable torque limiting" function is used to enable torque limiting in parallel to motion. This is effective immediately.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	2
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of the limiting value.
With DIRECT, the value set in the 'torqueLimit' parameter is used as the programmed value.
With EFFECTIVE, the last programmed torque limit value is used.
With USER_DEFAULT, the default torque limit value defined in system variable 'userDefaultTorque-Limiting.torqueLimit' is used. This default can be modified by entering a value in the 'torqueLimit' parameter. The value of the parameter is interpreted as a percentage.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	100.0

Desired axis torque limit value.
This parameter is evaluated independently of the 'torqueLimitType' parameter.
torqueLimitType -> DIRECT: Specifies the value as torque or force; the evaluation depends on the 'torqueLimitUnit' parameter.
torqueLimitType -> USER_DEFAULT: If this parameter is not specified, the default torque limit value defined in system variable 'userDefaultTorqueLimiting.torqueLimit' is used.
If a value is specified, the entry is interpreted as a percentage relative to the torque reduction defined in the 'userDefaultTorqueLimiting.torqueLimit' system variable.
torqueLimitType -> EFFECTIVE: Parameter 'torqueLimit' is not evaluated.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableTorqueLimiting
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableTorqueLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_FUNCTION_DISABLED (232)	When command is finished or aborted
WHEN_TORQUELIMIT_REACHED (233)	As soon as torque is limited
WHEN_TORQUELIMIT_GONE (234)	As soon as torque limiting is deactivated

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_TORQUELIMIT_REACHED, the transition to the next command takes place as soon as the torque is limited (the torque limit setting is reached in drive). This state is derived from the PROFIBUS status word 'MeldW' (PZW 5), bit 1 (M < Mx) of the drive.
With WHEN_TORQUELIMIT_GONE, the transition to the next command takes place as soon as the torque limiting has been reached once and the torque limiting has been exited. This state is derived from the PROFIBUS status word 'MeldW' (PZW 5), bit 1 (M < Mx) of the drive.
With WHEN_FUNCTION_DISABLED, the transition to the next command does not take place until the command is aborted. The command can be aborted by calling the '_disableTorqueLimiting' or '_resetAxis' functions in another task.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

torqueLimitUnit (optional)

Direction:	Input parameters
Data type:	EnumTorqueLimitUnitType
Parameter index:	7
System default:	DEFAULT_UNIT

EnumTorqueLimitUnitType

DEFAULT_UNIT (268)	Unit for the TO
TORQUE (269)	Torque

Reference of limiting values during programming.
With DEFAULT_UNIT, the force or torque relates to the load side. The force or torque relates to the load side. The gear ratio is always taken into account here.
With TORQUE, the limiting value is interpreted as torque relating to the drive side. A gear ratio is not taken into account.

1.3.9.9 _getAxisDataSetParameter

This function reads an axis data set.

Return value: StructRetReadGetAxisDataSet

StructRetReadGetAxisDataSet
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisDataSetReadWrite	-

StructAxisDataSetReadWrite
Parameters of an axis data set

Structure	Name	Data type	Unit
ControllerStruct	Parameters of the controller	StructControllerType	-
DynamicData	Dynamic characteristic values of the cascade control system	StructDynamicData	-
DynamicComp	Parameters for dynamic compensation of the control loop	StructDynamicComp	-
ProcessModel	Parameters of process model	StructProcessModel	-
DynamicFollowing	Parameters of the dynamic following error monitoring	StructDynamicFollowing	-
ControllerDynamic	Parameters of the reference model monitoring	StructControllerDynamic	-
EncoderNumber	Measuring system number	StructEncoderNumber	-
Gear	Ratio of the load gearbox	StructGear	-
ClampingMonitoring	Parameters for setting the torque monitoring	StructClampingMonitoring	-

StructControllerType
Parameters of the controller

Structure	Name	Data type	Unit
conType	Controller type	EnumAxisController-Type	-
PV_Controller	Parameters for the P controller with precontrol	StructPVController	-
PD_Controller	Parameters for the PD controller	StructPDController	-
PID_Controller	Parameters for the PID controller	StructPIDController	-

StructDynamicData
Dynamic characteristic values of the cascade control system

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant of the velocity control loop	LREAL	s
torqueTimeConstant	Equivalent time constant for torque control loop	LREAL	s

StructDynamicComp
Parameters for dynamic compensation of the control loop

Structure	Name	Data type	Unit
enable	Activate dynamic compensation	EnumYesNo	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructProcessModel
Parameters of the process model (PT2)

Structure	Name	Data type	Unit
ks	Transfer ratio of process	LREAL	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructDynamicFollowing
Parameters of the dynamic following error monitoring

Structure	Name	Data type	Unit
enable	Activation of the dynamic following error monitoring	EnumYesNo	-
minVelocity	Velocity value for the start of the rise, proportional to velocity, of the maximum permissible following error	LREAL	m/s
minPositionTolerance	Maximum permissible following error at velocities below the start of the rise, proportional to velocity, of the characteristic curve	LREAL	m
maxPositionTolerance	Maximum permissible following error at maximum velocity	LREAL	m
warningLimit	Warning limit of following error monitoring	LREAL	%

StructControllerDynamic
Parameters of the reference model monitoring

Structure	Name	Data type	Unit
enable	Activation of the reference model monitoring	EnumYesNo	-
maxVeloTolerance	Maximum velocity tolerance	LREAL	%

StructEncoderNumber
Measuring system number

Structure	Name	Data type	Unit
encoderNumber	Measuring system number	DINT	-

StructGear
Load gearing ratio

Structure	Name	Data type	Unit
numFactor	Numerator for load gearing	LREAL	-
denFactor	Denominator for load gearing	LREAL	-

StructClampingMonitoring
Parameters for setting the torque monitoring

Structure	Name	Data type	Unit
recognitionMode	Activation mode for torque monitoring	EnumRecogniti-onMode	-
followingErrorDevia-tion	Specifies the required following error for detection of endstop	LREAL	m
positionTolerance	Specifies the permis-sible deviation of the actual value from the setpoint in the clamped state	LREAL	m

StructPVController
Parameters for the P controller with precontrol

Structure	Name	Data type	Unit
enableDSC	Activation of DSC	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kv	P controller gain	LREAL	rps
kpc	Precontrol weighting	LREAL	%
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

StructPDController
Parameters for the PD controller

Structure	Name	Data type	Unit
kp	P controller gain	LREAL	rps
kd	Gain of DT1 element	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s

StructPIDController
Parameters for the PID controller

Structure	Name	Data type	Unit
preCon	Activate precontrol	EnumYesNo	-
kpc	Precontrol factor	LREAL	%
kp	P component gain	LREAL	rps
ki	I component gain	LREAL	-
kd	D component gain	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s
enableAntiWindup	I component limit	EnumYesNo	-
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

EnumAxisControllerType

NODEF (0)	No significance
DIRECT (1)	Control only
PD (2)	PID controller
PV (3)	PV controller
PID (4)	PID controller
PID_ACTUAL (5)	PID controller with actual value-dependent D-component

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumRecognitionMode

DO_NOT_CLAMP (0)	No detection
CLAMP_BY_FOLLOWING_ERROR_DEVIATION (1)	By following error
CLAMP_WHEN_TORQUE_LIMIT_REACHED (2)	By force moment/torque

EnumBalanceFilterMode

OFF (0)	Balancing filter not active
MODE_1 (1)	Balancing filter active
MODE_2 (2)	Expanded balancing filter active

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.9.10 _setAxisDataSetActive

This function activates the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be activated.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.9.11 _setAxisDataSetParameter

This function overwrites an axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisDataSetReadWrite
Parameter index:	3

StructAxisDataSetReadWrite
Parameters of an axis data set

Structure	Name	Data type	Unit
ControllerStruct	Parameters of the controller	StructControllerType	-
DynamicData	Dynamic characteristic values of the cascade control system	StructDynamicData	-
DynamicComp	Parameters for dynamic compensation of the control loop	StructDynamicComp	-
ProcessModel	Parameters of process model	StructProcessModel	-
DynamicFollowing	Parameters of the dynamic following error monitoring	StructDynamicFollowing	-
ControllerDynamic	Parameters of the reference model monitoring	StructControllerDynamic	-
EncoderNumber	Measuring system number	StructEncoderNumber	-
Gear	Ratio of the load gearbox	StructGear	-
ClampingMonitoring	Parameters for setting the torque monitoring	StructClampingMonitoring	-

StructControllerType
Parameters of the controller

Structure	Name	Data type	Unit
conType	Controller type	EnumAxisControllerType	-
PV_Controller	Parameters for the P controller with precontrol	StructPVController	-
PD_Controller	Parameters for the PD controller	StructPDController	-
PID_Controller	Parameters for the PID controller	StructPIDController	-

StructDynamicData
Dynamic characteristic values of the cascade control system

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant of the velocity control loop	LREAL	s
torqueTimeConstant	Equivalent time constant for torque control loop	LREAL	s

StructDynamicComp
Parameters for dynamic compensation of the control loop

Structure	Name	Data type	Unit
enable	Activate dynamic compensation	EnumYesNo	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructProcessModel
Parameters of the process model (PT2)

Structure	Name	Data type	Unit
ks	Transfer ratio of process	LREAL	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructDynamicFollowing
Parameters of the dynamic following error monitoring

Structure	Name	Data type	Unit
enable	Activation of the dynamic following error monitoring	EnumYesNo	-
minVelocity	Velocity value for the start of the rise, proportional to velocity, of the maximum permissible following error	LREAL	m/s
minPositionTolerance	Maximum permissible following error at velocities below the start of the rise, proportional to velocity, of the characteristic curve	LREAL	m
maxPositionTolerance	Maximum permissible following error at maximum velocity	LREAL	m
warningLimit	Warning limit of following error monitoring	LREAL	%

StructControllerDynamic
Parameters of the reference model monitoring

Structure	Name	Data type	Unit
enable	Activation of the reference model monitoring	EnumYesNo	-
maxVeloTolerance	Maximum velocity tolerance	LREAL	%

StructEncoderNumber
Measuring system number

Structure	Name	Data type	Unit
encoderNumber	Measuring system number	DINT	-

StructGear
Load gearing ratio

Structure	Name	Data type	Unit
numFactor	Numerator for load gearing	LREAL	-
denFactor	Denominator for load gearing	LREAL	-

StructClampingMonitoring
Parameters for setting the torque monitoring

Structure	Name	Data type	Unit
recognitionMode	Activation mode for torque monitoring	EnumRecogniti-onMode	-
followingErrorDevia-tion	Specifies the required following error for detection of endstop	LREAL	m
positionTolerance	Specifies the permis-sible deviation of the actual value from the setpoint in the clamped state	LREAL	m

StructPVController
Parameters for the P controller with precontrol

Structure	Name	Data type	Unit
enableDSC	Activation of DSC	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kv	P controller gain	LREAL	rps
kpc	Precontrol weighting	LREAL	%
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

StructPDController
Parameters for the PD controller

Structure	Name	Data type	Unit
kp	P controller gain	LREAL	rps
kd	Gain of DT1 element	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s

StructPIDController
Parameters for the PID controller

Structure	Name	Data type	Unit
preCon	Activate precontrol	EnumYesNo	-
kpc	Precontrol factor	LREAL	%
kp	P component gain	LREAL	rps
ki	I component gain	LREAL	-
kd	D component gain	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s
enableAntiWindup	I component limit	EnumYesNo	-
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

EnumAxisControllerType

NODEF (0)	No significance
DIRECT (1)	Control only
PD (2)	PID controller
PV (3)	PV controller
PID (4)	PID controller
PID_ACTUAL (5)	PID controller with actual value-dependent D-component

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumRecognitionMode

DO_NOT_CLAMP (0)	No detection
CLAMP_BY_FOLLOWING_ERROR_DEVIATION (1)	By following error
CLAMP_WHEN_TORQUE_LIMIT_REACHED (2)	By force moment/torque

EnumBalanceFilterMode

OFF (0)	Balancing filter not active
MODE_1 (1)	Balancing filter active
MODE_2 (2)	Expanded balancing filter active

Specifies the data to be used to overwrite the data set.

1.3.10 Speed-controlled axis - Object and Alarm Handling

1.3.10.1 _cancelAxisCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its 'CommandId' in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled.

Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBe-Cancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.10.2 _disableAxisInterface

This function deactivates the actuator or encoder interface. When the interface is deactivated, the corresponding drivers are deactivated.

The actuator interface is the interface of the axis to the drive, which in addition to the setpoint also includes the control signals of the axis to the drive and the status information and the actual speed value of the drive to the controller, when connection of the drive is via PROFIdrive message frame.

Correspondingly the encoder interface, in addition to the actual encoder value, also includes the control signals of the axis to the encoder evaluation, e.g. in the drive, and the status information of the encoder evaluation, e.g. in the drive, to the controller, when connection of the encoder is via PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

actor (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	2
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the deactivation of the actuator interface.
With YES, the actuator interface is deactivated.
With NO, the actuator interface status remains unchanged.

sensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	3
System default:	0

Specifies the deactivation of the encoder interface.
If the bit is set for the corresponding encoder interface, the interface is deactivated. If the bit is not set, the encoder interface status remains unchanged.

hwLimitSwitch (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the deactivation of the hardware limit switch monitoring.
With YES, the hardware limit switch monitoring is deactivated.
With NO, the status of the hardware limit switch monitoring remains unchanged.

additionalSensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	0

Specifies the deactivation of the encoder interface of the additional sensor.
If the bit is set for the corresponding additional sensor, the additional sensor is deactivated. If the bit is not set, the additional sensor status remains unchanged.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableAxisInterface
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnableAxisInterface

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.10.3 _disableAxisSimulation

This function switches the axis out of simulation mode.
The actual encoder values displayed in the structure elements of the 'sensorData' system variable are applied as actual values; the status of all other actual values remains unchanged.
Any existing following error is removed immediately.
The current simulation status can be queried in the 'simulation' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.10.4 _enableAxisInterface

This function activates the actuator or encoder interface. When the interface is activated, the corresponding drivers are activated.

The actuator interface is the interface of the axis to the drive, which in addition to the setpoint also includes the control signals of the axis to the drive and the status information and the actual speed value of the drive to the controller, when connection of the drive is via PROFIdrive message frame.

Correspondingly the encoder interface, in addition to the actual encoder value, also includes the control signals of the axis to the encoder evaluation, e.g. in the drive, and the status information of the encoder evaluation, e.g. in the drive, to the controller, when connection of the encoder is via PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

actor (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	2
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the activation of the actuator interface.
With YES, the actuator interface is activated.
With NO, the actuator interface status remains unchanged.

sensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	3
System default:	0

Specifies the activation of the encoder interface.
If the bit is set for the corresponding encoder interface, the interface is activated. If the bit is not set, the encoder interface status remains unchanged.

hwLimitSwitch (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the activation of the hardware limit switch monitoring.
With YES, the hardware limit switch monitoring is activated.
With NO, the status of the hardware limit switch monitoring remains unchanged.

additionalSensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	0

Specifies the activation of the encoder interface of the additional sensor.
If the bit is set for the corresponding additional sensor, the additional sensor is activated. If the bit is not set, the additional sensor status remains unchanged.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableAxisInterface
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnableAxisInterface

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.10.5 _enableAxisSimulation

This function switches the axis to simulation mode.
If the axis is moving, it is stopped by entering a velocity of 0.
The actual values indicated in the system variables are taken from the setpoints in the same way as on a virtual axis.
The current simulation status can be queried in the 'simulation' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.10.6 _enableDistributedMotionDelayValueCalculation

This function can be used for explicit initiation of offset determination for distributed synchronous operation (e.g. after adding an axis to a modular machine).
This function cannot be used to determine the offset for recursive synchronous operation with PROFINET, because in this case the offset is not calculated correctly by the system.

This function can be used as of SIMOTION V4.1, SP4.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

master

Direction:	Input parameters
Data type:	MasterType
Parameter index:	1

This is a positioning axis

mode (optional)

Direction:	Input parameters
Data type:	EnumAxisDelayValueCalculationMode
Parameter index:	2
System default:	STANDARD

EnumAxisDelayValueCalculationMode

ACTUAL (7)	Currently accessible synchronous connections
STANDARD (358)	Configured synchronous connections

Specifies connections for which the offset is to be determined.
With ACTUAL, the offset is determined for all currently accessible synchronous connections.
With STANDARD, the offset is determined for all configured synchronous connections. During startup, the command waits until all configured stations are active.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDelayValueCalculation
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandDelayValueCalculation

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.10.7 _getAxisErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the errorNumber parameter is pending.
NO - The error specified by the errorNumber parameter is not pending.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.10.8 _getAxisErrorState

This function provides information on whether axis alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.10.9 _getStateOfMotionBuffer

This function returns the status of the axis command queue.

Return value: StructRetMotionBuffer

StructRetMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumMotionBuffer-State	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':
EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':
Specifies the number of entries in the command queue.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.10.10 _resetAxis

This function switches the axis to a defined initial state.
All active motions are stopped by entering a setpoint of 0.
Pending commands are deleted, synchronous commands are aborted. The command is executed synchronously.
Pending errors on the axis are deleted. For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.
System variables modified by the program are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.10.11 _resetAxisConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.10.12 _resetAxisError

This function resets axis errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.
The command is asynchronous. When applicable, the error is not reset until the local response activated by the error has been completed.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.10.13 _resetMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.11 External encoder - Command tracking

1.3.11.1 _bufferExternalEncoderCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoderType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.11.2 _cancelExternalEncoderCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its CommandId in the 'commandToBeCancelled' parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled. Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.11.3 _getStateOfExternalEncoderCommand

This function returns the execution state of a command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT
- The 'commandId' is unknown or

command has already been completed

WAITING_FOR_SYNC_START
- Waiting for synchronous start

WAITING
- Command has been decoded, but execution has not

yet started

ACTIVE
- Command is being executed.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.11.4 _removeBufferedExternalEncoderCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.12 External encoder - Actual value manipulation

1.3.12.1 _disableExternalEncoder

This function cancels the external encoder enables.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.12.2 _enableExternalEncoder

This function switches the external encoder enable. The command can be executed synchronously or asynchronously.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.12.3 _redefineExternalEncoderPosition

This function sets the coordinate system of an external encoder. The position value can be defined as an absolute value or as a relative position offset.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

redefineMode (optional)

Direction:	Input parameters
Data type:	EnumRedefineMode
Parameter index:	2
System default:	ABSOLUTE

EnumRedefineMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Specifies the type of assignment for the new position.
With ABSOLUTE, the values are assigned as absolute positions.
With RELATIVE, the values are added to the existing position (offset).

position (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Position specification.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command takes place after the new position has been set.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.12.4 _synchronizeExternalEncoder

This function synchronizes or sets the external encoder reference position.

The execution time of the _synchronizeExternalEncoder command for the absolute encoder adjustment, i.e. for synchronizingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER and synchronizingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION, can take several interpolation cycles.
The status of the command can be queried by querying the CommandID of the homing command using _getstateOfExternalEncoderCommand(), even when the axis is already in the 'homed' state.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoderType' on which the command is to be executed.

synchronizingMode (optional)

Direction:	Input parameters
Data type:	EnumSynchronizingMode
Parameter index:	2
System default:	DIRECT_HOMING

EnumSynchronizingMode

DIRECT_HOMING (41)	Setting of current position value
ENABLE_OFFSET_OF_ABSOLUTE_ENCODER (49)	Absolute encoder offset
PASSIVE_HOMING (104)	Homing with the next zero mark
DIRECT_HOMING_RELATIVE (352)	Relative setting of current position value
SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION (385)	Absolute encoder adjustment with position specification from command

Specifies the synchronization mode for an external encoder.
With DIRECT_HOMING, the current value of the external encoder is set to the 'syncPosition' home position coordinates.

With ENABLE_OFFSET_OF_ABSOLUTE_ENCODER, the offset specified in the 'absHomingEncoder.absshift' configuration data is
- added to the existing value of the external encoder with the setting 'absHomingEncoder.setOffsetOfAbsoluteEncoder=RELATIVE',
- with the setting 'absHomingEncoder.setOffsetOfAbsoluteEncoder=ABSOLUTE' it is added to the existing actual encoder value.

With PASSIVE_HOMING, the next zero mark crossed by the moving sensor is used as the home position.

With DIRECT_HOMING_RELATIVE, the current value of the external encoder is adjusted by the value of the 'syncPosition' home position coordinates.

With the setting SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION, the value of the 'syncPosition' home position coordinates is set as the current value of the external encoder. The resulting absolute encoder offset is calculated from this, displayed in system variable 'absoluteEncoder.totalOffsetValue', and stored in the system as a retain variable.

syncPositionType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	3
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of home position coordinate specification.
With DIRECT, the home position coordinates are specified directly.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default direction defined in system variable 'userDefault.syncPosition' is used.

syncPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the home position coordinates, evaluation depending on parameter 'syncPositionType'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandSyncEncoder
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandSyncEncoder

IMMEDIATELY (60)	Immediate command transition
WHEN_ENCODER_SYNCHRONIZED (161)	Transition after successful homing

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ENCODER_SYNCHRONIZED, the transition to the next command takes place after homing of the encoder.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.13 External encoder - Object and Alarm Handling

1.3.13.1 _cancelExternalEncoderCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its CommandId in the 'commandToBeCancelled' parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoderType' on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled. Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.13.2 _enableDistributedMotionDelayValueCalculation

This function can be used for explicit initiation of offset determination for distributed synchronous operation (e.g. after adding an axis to a modular machine).
This function cannot be used to determine the offset for recursive synchronous operation with PROFINET, because in this case the offset is not calculated correctly by the system.

This function can be used as of SIMOTION V4.1, SP4.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

master

Direction:	Input parameters
Data type:	MasterType
Parameter index:	1

This is a positioning axis

mode (optional)

Direction:	Input parameters
Data type:	EnumAxisDelayValueCalculationMode
Parameter index:	2
System default:	STANDARD

EnumAxisDelayValueCalculationMode

ACTUAL (7)	Currently accessible synchronous connections
STANDARD (358)	Configured synchronous connections

Specifies connections for which the offset is to be determined.
With ACTUAL, the offset is determined for all currently accessible synchronous connections.
With STANDARD, the offset is determined for all configured synchronous connections. During startup, the command waits until all configured stations are active.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDelayValueCalculation
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandDelayValueCalculation

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.13.3 _getExternalEncoderErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.13.4 _getExternalEncoderErrorState

This function provides information on whether external encoder alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'errorState':
With NO, no alarm is pending.
With YES, an alarm is pending.

Return value 'numberOfErrors':
Total number of pending errors.

Return value 'error1..8':
In the return values 'error1 to error8', the error numbers of the first eight alarms are output.

Return value 'error1..8Parameter1..5':
In the return values, the alarm parameters belonging to the error are output.

Parameter:
externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

1.3.13.5 _resetExternalEncoder

This function switches an external encoder to a defined initial state.
Synchronous commands are aborted. The reset command is executed synchronously.
Pending external encoder errors are deleted. For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoder-Type' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.13.6 _resetExternalEncoderConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body.

Configuration data are collected by setting the 'activationModeChangedConfigData' system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT

Description of the return values:
Corresponds to the list of return values under the _move function.

Parameter:
externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoderType' on which the command is to be executed.

1.3.13.7 _resetExternalEncoderError

This function resets the external encoder errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

externalEncoder

Direction:	Input parameters
Data type:	ExternalEncoderType
Parameter index:	1

Specifies the technology object of the 'externalEncoder' type or a variable of the 'ExternalEncoderType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.14 Fixed gear - Command tracking

1.3.14.1 _bufferFixedGearCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.14.2 _getStateOfFixedGearCommand

This function returns the execution state of a synchronous operation command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'commandIdState':
With ACTIVE, the command is being executed.
With NOT_EXISTENT, the 'commandId' is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.
With EXECUTED, execution of the command is complete.
With ABORTED, execution of the command has been aborted.

Parameter:
fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.14.3 _removeBufferedFixedGearCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.15 Fixed gear - Motion

1.3.15.1 _disableFixedGearing

This function deactivates synchronous operation using a gearbox as the transmission function.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

disableMode (optional)

Direction:	Input parameters
Data type:	EnumFixedGearDisableMode
Parameter index:	2
System default:	USER_DEFAULT

EnumFixedGearDisableMode

IMMEDIATELY (60)	Effective immediately
ON_MASTER_POSITION (100)	Specification of the synchronization position of the master axis
USER_DEFAULT (149)	User default

Specifies the desynchronization criterion.
With IMMEDIATELY, desynchronization starts immediately.
With ON_MASTER_POSITION, a master value is used as the desynchronization position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.syncOffMode' is used.

endPositionMasterType (optional)

Direction:	Input parameters
Data type:	EnumFixedGearDirectEffectiveUserDefault
Parameter index:	3
System default:	USER_DEFAULT

EnumFixedGearDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of master position specification for desynchronization, relevant when the ON_MASTER_POSITION synchronization mode was selected.
With DIRECT, the value set in the 'endPositionMaster' parameter is used as the position.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.syncPositionMaster' is used.

endPositionMaster (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the master position for desynchronization.
It is only used when parameter 'disableMode:=ON_MASTER_POSITION' and when parameter 'endPositionMasterType:=DIRECT'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumFixedGearNextCommandDisableGearing
Parameter index:	5
System default:	IMMEDIATELY

EnumFixedGearNextCommandDisableGearing

IMMEDIATELY (60)	Effective immediately
WHEN_MOTION_DONE (163)	Start of synchronous operation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or motion is stopped in some other way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

motionOutBehaviourMode (optional)

Direction:	Input parameters
Data type:	EnumFixedGearMotionOutBehaviorMode
Parameter index:	8
System default:	LAST_VALUE

EnumFixedGearMotionOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0

Behavior of interconnection values on output side.
With DEFAULT_VALUE, the default setting defined in system variable 'motionOutDefault' is used.
With LAST_VALUE, the last interface value is frozen. The derivatives of the interface value are set to zero.
With ZERO_VALUE, a value of zero is specified on the output side.

1.3.15.2 _disableFixedGearMotionIn

This function deactivates the input vector of the fixed gear.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.15.3 _enableFixedGearing

This function activates synchronous operation using a gearbox as the transmission function.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumFixedGearDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumFixedGearDirection

BY_VALUE (24)	Sign
CURRENT (33)	The current direction will be retained
EFFECTIVE (45)	Last programmed direction
NEGATIVE (85)	Opposite direction
POSITIVE (107)	Same direction
USER_DEFAULT (149)	User default

Specifies the motion direction in which the gearing operates.
With BY_VALUE, the gearing direction is determined from the sign of the gear ratio.
With CURRENT, the current direction of the generated values is retained. If the current direction is not available (axis is at a standstill), motion is resumed in the positive direction.
With EFFECTIVE, the last programmed direction setpoint is used.
With NEGATIVE, the synchronous operation direction is opposite.
With POSITIVE, the synchronous operation direction is the same.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.direction' is used.

gearingType (optional)

Direction:	Input parameters
Data type:	EnumFixedGearGearingType
Parameter index:	3
System default:	USER_DEFAULT

EnumFixedGearGearingType

ABSOLUTE (1)	Absolute gearbox
EFFECTIVE (45)	Last programmed type
RELATIVE (115)	Relative gearbox
USER_DEFAULT (149)	User default

Specifies type of gearing.
With ABSOLUTE, gearing is absolute relative to the axis zero for the relevant axes.
With EFFECTIVE, the last programmed type is used.
With RELATIVE, gearing is relative with reference to the start position or synchronization position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.type' is used.

gearingMode (optional)

Direction:	Input parameters
Data type:	EnumFixedGearGearingMode
Parameter index:	4
System default:	USER_DEFAULT

EnumFixedGearGearingMode

EFFECTIVE (45)	Last programmed type
GEARING_WITH_FRACTION (55)	Gear ratio as a fraction
GEARING_WITH_RATIO (56)	Gear ratio as floating-point number
USER_DEFAULT (149)	User default

Type of gear factor specification.
With EFFECTIVE, the last programmed type is used.
With GEARING_WITH_FRACTION, the ratio is specified as a fraction of two integers.
With GEARING_WITH_RATIO, the ratio is specified as a floating-point number.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.defineMode' is used.

gearingRatioType (optional)

Direction:	Input parameters
Data type:	EnumFixedGearDirectEffectiveUserDefault
Parameter index:	5
System default:	USER_DEFAULT

EnumFixedGearDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of of gear ratio specification.
With DIRECT and the gear ratio setting 'gearingMode:=GEARING_WITH_FRACTION', the value set in the 'gearingNumerator' parameter is used as the numerator and the value set in 'gearingDenominator' is used as the denominator.
With DIRECT and the setting 'gearingMode:=GEARING_WITH_RATIO', the value set in the 'gearingRatio' parameter is used as the gear ratio.

With EFFECTIVE, the last programmed values are used in accordance with the 'gearingMode' setting.

With USER_DEFAULT and the setting 'gearingMode:=GEARING_WITH_FRACTION', the default settings defined in system variables 'userDefault.gearingSettings.numerator' and 'userDefault.gearingSettings.denominator' are used.
With USER_DEFAULT and the setting 'gearingMode:=GEARING_WITH_RATIO', the default setting defined in system variable 'userDefault.gearingSettings.ratio' is used.

gearingRatio (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	1.0

Specifies the gear factor.
Only effective with the setting 'gearingMode:=GEARING_WITH_RATIO' and dependent on the parameter 'gearingRatioType'.
With 'gearingRatioType:=DIRECT', the value represents the floating-point gear ratio.
In all other cases, the parameter is irrelevant.

gearingNumerator (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	7
System default:	1

Specifies the numerator of the gear ratio with the setting 'gearingMode:=GEARING_WITH_FRACTION', depending on the parameter "gearingRatioType".
With DIRECT, the value represents the numerator of the gear ratio.
In all other cases, the parameter is irrelevant.

gearingDenominator (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	8
System default:	1

Specifies the denominator of the gear ratio as a fraction, only with the setting 'gearing-Mode:=GEARING_WITH_FRACTION', depending on the parameter "gearingRatioType".
With DIRECT, the value represents the denominator of the gear ratio.
In all other cases, the parameter is irrelevant.

enableMode (optional)

Direction:	Input parameters
Data type:	EnumFixedGearEnableMode
Parameter index:	9
System default:	USER_DEFAULT

EnumFixedGearEnableMode

IMMEDIATELY (60)	Effective immediately
ON_MASTER_POSITION (100)	Specification of the synchronization position of the master axis
USER_DEFAULT (149)	User default

Specifies the synchronization criterion.
With IMMEDIATELY, synchronization takes place immediately.
With ON_MASTER_POSITION, a master position is used as the synchronization criterion.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.synchronizingMode' is used.

startPositionMasterType (optional)

Direction:	Input parameters
Data type:	EnumFixedGearDirectEffectiveUserDefault
Parameter index:	10
System default:	USER_DEFAULT

EnumFixedGearDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of master position specification for synchronization, relevant when the 'synchronizingMode' parameter contains the value ON_MASTER_POSITION.
With DIRECT, the value set in the 'startPositionMaster' parameter is used as the synchronization position.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.startPositionMaster' is used.

startPositionMaster (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the master position for synchronization.
It is only used when parameter 'synchronizingMode:=ON_MASTER_POSITION' and when parameter 'endPositionMasterType:=DIRECT'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumFixedGearNextCommandEnableGearing
Parameter index:	12
System default:	IMMEDIATELY

EnumFixedGearNextCommandEnableGearing

IMMEDIATELY (60)	Effective immediately
WHEN_GEARING_START (309)	Start of synchronous operation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_GEARING_START, the transition takes place after the start of the gearing operation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	13
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.15.4 _enableFixedGearMotionIn

This function activates the input vector of the fixed gear.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.16 Fixed gear - Compensation and superimposition

1.3.16.1 _setFixedGearingOffset

This function activates an offset in the master or slave range.

When active in the current synchronous operation function, the offset remains active until the camming function is replaced or desynchronized.
When permanently set the offset becomes active when subsequent synchronous operation functions are started.

The dynamic response values for the transition can be specified.
The current and permanent offsets in effect can be read in system variable 'cammingAdjustments'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

offsetRange (optional)

Direction:	Input parameters
Data type:	EnumRange
Parameter index:	2
System default:	MASTER_RANGE

EnumRange

MASTER_RANGE (74)	Master range
SLAVE_RANGE (123)	Slave range

Specifies whether the master range or slave range is scaled.
With MASTER_RANGE, the master range is scaled.
With SLAVE_RANGE, the slave range is scaled.

offsetMode (optional)

Direction:	Input parameters
Data type:	EnumAbsoluteRelative
Parameter index:	3
System default:	ABSOLUTE

EnumAbsoluteRelative

ABSOLUTE (1)	Absolute
RELATIVE (115)	Relative

Specifies the type of offset specification.
With ABSOLUTE, the offset is specified as an absolute value.
With RELATIVE, the offset is added to the existing value.

offsetValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the offset.

activationMode (optional)

Direction:	Input parameters
Data type:	EnumGearingActivationMode
Parameter index:	5
System default:	ACTUAL_VALUE

EnumGearingActivationMode

DEFAULT_VALUE (2)	Active at start of all subsequent synchronous operation commands
ACTUAL_VALUE (34)	Active for active synchronous operation command
ACTUAL_AND_DEFAULT_VALUE (237)	Active for active synchronous operation command and at start of all subsequent synchronous operation commands

Specifies how scaling is activated.
With DEFAULT_VALUE, the scaling is active at the start of all subsequent synchronous operation commands.
With ACTUAL_VALUE, the scaling is active in the currently active synchronous operation command.
With ACTUAL_AND_DEFAULT_VALUE, the scaling is active in the active synchronous operation command and in all subsequent synchronous operation commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.16.2 _setFixedGearMaster

This function specifies a master object for the synchronous operation relationship.
The possible master objects for synchronous operation are set in the configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

master

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	3
System default:	WHEN_COMMAND_DONE

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.17 Fixed gear - Object and Alarm Handling

1.3.17.1 _getFixedGearErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'errorState'
With NO, the error specified by the 'errorNumber' parameter is not pending.
With YES, the error specified by the 'errorNumber' parameter is pending.

Parameter:
fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.17.2 _getFixedGearErrorState

This function provides information on whether fixed gear alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'errorState':
With NO, no alarm is pending.
With YES, an alarm is pending.

Return value 'numberOfErrors':
Total number of pending errors.

Return value 'error1..8':
In the return values 'error1 to error8', the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:
In the return values, the alarm parameters belonging to the error are output.

Parameter:
fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

1.3.17.3 _resetFixedGear

This function sets a fixed gear technology object to the initial state.
Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.17.4 _resetFixedGearConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

1.3.17.5 _resetFixedGearError

This function resets fixed gear technology object errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return values:
Corresponds to the list of return values under the _move function.

Parameter:

fixedGear

Direction:	Input parameters
Data type:	_FixedGearType
Parameter index:	1

Specifies the technology object of the 'fixedGear' type or a variable of the '_FixedGearType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.18 Formula object - Command tracking

1.3.18.1 _bufferFormulaObjectCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.

The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.18.2 _getStateOfFormulaObjectCommand

This function returns the execution state of a command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'commandIdState':
With ACTIVE, the command is being executed.
With NOT_EXISTENT, the 'commandId' is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.
With EXECUTED, execution of the command is complete.
With ABORTED, execution of the command has been aborted.

Parameter:
formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.18.3 _removeBufferedFormulaObjectCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.19 Formula object - Formula functions

1.3.19.1 _changeEnableModeOfFormulaObjectIn

This function changes the updating of the specified inputs.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MI1 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

MI2 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

MI3 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI1 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI2 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	6
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI3 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	7
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI4 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI1 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI2 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI3 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	11
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI4 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

1.3.19.2 _changeEnableOfFormula

This function changes the activation of the formula processing for the specified outputs.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MO1s (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO1sBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	3
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO1sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the direct value.

MO1v (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO1vBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	6
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO1vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the direct value.

MO1a (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO1aBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	9
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO1aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the direct value.

MO2s (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	11
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO2sBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	12
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO2sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the direct value.

MO2v (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO2vBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	15
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO2vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	0.0

Specifies the direct value.

MO2a (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	17
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO2aBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	18
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO2aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	0.0

Specifies the direct value.

MO3s (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	20
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO3sBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	21
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO3sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	22
System default:	0.0

Specifies the direct value.

MO3v (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	23
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO3vBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	24
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO3vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Specifies the direct value.

MO3a (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	26
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO3aBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	27
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO3aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	28
System default:	0.0

Specifies the direct value.

LO1 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	29
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO1BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	30
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO1Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	31
System default:	0.0

Specifies the direct value.

LO2 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	32
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO2BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	33
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO2Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	34
System default:	0.0

Specifies the direct value.

LO3 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	35
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO3BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	36
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO3Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	37
System default:	0.0

Specifies the direct value.

LO4 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	38
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO4BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	39
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO4Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	40
System default:	0.0

Specifies the direct value.

DO1 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	41
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO1BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	42
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO1Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	43
System default:	0

Specifies the direct value.

DO2 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	44
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO2BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	45
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO2Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	46
System default:	0

Specifies the direct value.

DO3 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	47
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO3BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	48
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO3Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	49
System default:	0

Specifies the direct value.

DO4 (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	50
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO4BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	51
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO4Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	52
System default:	0

Specifies the direct value.

1.3.19.3 _defineFormula

This function describes a formula function and assigns it to the formula number.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

formulaNumber

Direction:	Input parameters
Data type:	UDINT
Parameter index:	2

Specifies the formula number.

formula

Direction:	Input parameters
Data type:	STRING
Parameter index:	3

Description of the formula.

1.3.19.4 _disableFormula

This function deactivates the formula processing for the specified output.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MO1s (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO1sBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	3
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO1sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the direct value.

MO1v (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO1vBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	6
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO1vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the direct value.

MO1a (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO1aBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	9
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO1aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the direct value.

MO2s (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	11
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO2sBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	12
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO2sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the direct value.

MO2v (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO2vBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	15
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO2vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	0.0

Specifies the direct value.

MO2a (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	17
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO2aBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	18
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO2aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	0.0

Specifies the direct value.

MO3s (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	20
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO3sBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	21
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO3sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	22
System default:	0.0

Specifies the direct value.

MO3v (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	23
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO3vBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	24
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO3vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Specifies the direct value.

MO3a (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	26
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

MO3aBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	27
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

MO3aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	28
System default:	0.0

Specifies the direct value.

LO1 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	29
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO1BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	30
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO1Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	31
System default:	0.0

Specifies the direct value.

LO2 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	32
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO2BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	33
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO2Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	34
System default:	0.0

Specifies the direct value.

LO3 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	35
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO3BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	36
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO3Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	37
System default:	0.0

Specifies the direct value.

LO4 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	38
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

LO4BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	39
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

LO4Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	40
System default:	0.0

Specifies the direct value.

DO1 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	41
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO1BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	42
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO1Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	43
System default:	0

Specifies the direct value.

DO2 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	44
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO2BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	45
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO2Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	46
System default:	0

Specifies the direct value.

DO3 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	47
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO3BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	48
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO3Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	49
System default:	0

Specifies the direct value.

DO4 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	50
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the formula assigned to the output is to be processed.

DO4BehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumFormulaObjectOutBehaviorMode
Parameter index:	51
System default:	LAST_VALUE

EnumFormulaObjectOutBehaviorMode

DEFAULT_VALUE (2)	Default value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	0
DIRECT_VALUE (351)	Specified value

Specifies the value to be transferred to the output interface.

DO4Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	52
System default:	0

Specifies the direct value.

1.3.19.5 _disableFormulaObjectIn

This function activates the updating of the specified inputs.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MI1 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

MI2 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

MI3 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI1 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI2 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	6
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI3 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	7
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

LI4 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI1 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI2 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI3 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	11
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

DI4 (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies whether the input is to be updated.

1.3.19.6 _enableFormula

This function activates the formula processing for the specified output.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MO1s (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO1v (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO1a (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO2s (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO2v (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	6
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO2a (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	7
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO3s (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO3v (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

MO3a (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

LO1 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	11
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

LO2 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

LO3 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	13
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

LO4 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

DO1 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	15
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

DO2 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

DO3 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	17
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

DO4 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	18
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the formula assigned to the output is to be processed.

1.3.19.7 _enableFormulaObjectIn

This function activates the updating of the specified inputs.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MI1 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

MI2 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

MI3 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

LI1 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	5
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

LI2 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	6
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

LI3 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	7
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

LI4 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

DI1 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

DI2 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

DI3 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	11
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

DI4 (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies whether the input is to be updated.

1.3.19.8 _setFormula

This function assigns the result of a formula to an output.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MO1sFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO1sFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	3
System default:	1

Specifies the motion to be assigned.

MO1vFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO1vFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	1

Specifies the motion to be assigned.

MO1aFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	6
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO1aFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	7
System default:	1

Specifies the motion to be assigned.

MO2sFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO2sFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	9
System default:	1

Specifies the motion to be assigned.

MO2vFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO2vFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	11
System default:	1

Specifies the motion to be assigned.

MO2aFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO2aFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	13
System default:	1

Specifies the motion to be assigned.

MO3sFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO3sFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	15
System default:	1

Specifies the motion to be assigned.

MO3vFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO3vFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	17
System default:	1

Specifies the motion to be assigned.

MO3aFormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	18
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

MO3aFormula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	19
System default:	1

Specifies the motion to be assigned.

LO1FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	20
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

LO1Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	21
System default:	1

Specifies the motion to be assigned.

LO2FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	22
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

LO2Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	23
System default:	1

Specifies the motion to be assigned.

LO3FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	24
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

LO3Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	25
System default:	1

Specifies the motion to be assigned.

LO4FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	26
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

LO4Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	27
System default:	1

Specifies the motion to be assigned.

DO1FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	28
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

DO1Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	29
System default:	1

Specifies the motion to be assigned.

DO2FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	30
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

DO2Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	31
System default:	1

Specifies the motion to be assigned.

DO3FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	32
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

DO3Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	33
System default:	1

Specifies the motion to be assigned.

DO4FormulaType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	34
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether a formula is to be assigned to the output.

DO4Formula (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	35
System default:	1

Specifies the motion to be assigned.

1.3.19.9 _setFormulaObjectOutputValue

This function assigns the passed parameter value to an output.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

MO1sValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO1sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the parameter value to be assigned.

MO1vValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO1vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the parameter value to be assigned.

MO1aValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	6
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO1aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the parameter value to be assigned.

MO2sValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO2sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the parameter value to be assigned.

MO2vValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO2vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the parameter value to be assigned.

MO2aValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO2aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the parameter value to be assigned.

MO3sValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO3sValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the parameter value to be assigned.

MO3vValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO3vValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the parameter value to be assigned.

MO3aValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	18
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

MO3aValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	0.0

Specifies the parameter value to be assigned.

LO1ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	20
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

LO1Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	0.0

Specifies the parameter value to be assigned.

LO2ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	22
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

LO2Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	0.0

Specifies the parameter value to be assigned.

LO3ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	24
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

LO3Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Specifies the parameter value to be assigned.

LO4ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	26
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

LO4Value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	27
System default:	0.0

Specifies the parameter value to be assigned.

DO1ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	28
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

DO1Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	29
System default:	0

Specifies the parameter value to be assigned.

DO2ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	30
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

DO2Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	31
System default:	0

Specifies the parameter value to be assigned.

DO3ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	32
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

DO3Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	33
System default:	0

Specifies the parameter value to be assigned.

DO4ValueType (optional)

Direction:	Input parameters
Data type:	EnumDoNotChangeDirect
Parameter index:	34
System default:	DO_NOT_CHANGE

EnumDoNotChangeDirect

DIRECT (40)	Value of the function parameter
DO_NOT_CHANGE (43)	No change

Specifies whether the parameter value is to be assigned.

DO4Value (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	35
System default:	0

Specifies the parameter value to be assigned.

1.3.20 Formula object - Object and Alarm Handling

1.3.20.1 _getFormulaObjectErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'errorState'
With NO, the error specified by the 'errorNumber' parameter is not pending.
With YES, the error specified by the 'errorNumber' parameter is pending.

Parameter:
formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.20.2 _getFormulaObjectErrorState

This function provides information on whether formula object alarms have occurred and how many. In addition, information on the alarms is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'errorState':
With NO, no alarm is pending.
With YES, an alarm is pending.

Return value 'numberOfErrors':
Total number of pending errors.

Return value 'error1..8':
In the return values 'error1 to error8', the error numbers of the first eight alarms are output.

Return value 'error1..8Parameter1..5':
In the return values, the alarm parameters belonging to the error are output.

Parameter:
formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

1.3.20.3 _resetFormulaObject

This function resets a formula object to the initial state.
Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With NO_RESTART_ACTIVATION, the technology object is not restarted.
With ACTIVATE_RESTART, the technology object is restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.20.4 _resetFormulaObjectConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body.

Configuration data are collected by setting the 'activationModeChangedConfigData' system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

1.3.20.5 _resetFormulaObjectError

This function resets formula object errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

formulaObject

Direction:	Input parameters
Data type:	_FormulaObjectType
Parameter index:	1

Specifies the technology object of the 'formulaObject' type or a variable of the '_FormulaObjectType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.21 Synchronous operation - Information and conversion

1.3.21.1 _getMasterValue

This function returns the master value at a programmed slave position.
In cases where several master values can have the same slave value, an approximation value can be specified for the master value.

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'value':
Specifies the master value.

Parameter:
followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

slavePositionType (optional)

Direction:	Input parameters
Data type:	EnumGetValue
Parameter index:	2
System default:	CURRENT

EnumGetValue

CURRENT (33)	Current setpoint
DIRECT (40)	Value entry

Type of specification of the slave position.
With CURRENT, the current slave position is used.
With DIRECT, the value set in the 'slavePosition' parameter is used.

slavePosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the slave value for which the master value is to be determined, only relevant if 'slavePositionType:=DIRECT'.

masterApproachPositionType (optional)

Direction:	Input parameters
Data type:	EnumApproachPositionType
Parameter index:	4
System default:	NONE

EnumApproachPositionType

DIRECT (40)	Value of the function parameter
NONE (93)	No specification (cam start point is used)

Type of specification of the master approximation value.
With DIRECT, the value specified in the 'masterApproachPosition' parameter is used.
The master value to be calculated is the one closest to this approximation value. Only appropriate if several master values exist for a slave value.
With NONE, an approximation value is not used.

masterApproachPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the master approximation value, only relevant if parameter 'masterApproachPositionType:=DIRECT'.

1.3.21.2 _getSlaveValue

This function returns the slave value for the specified master position.

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'value':
Specifies the slave value.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

masterPositionType (optional)

Direction:	Input parameters
Data type:	EnumGetValue
Parameter index:	2
System default:	CURRENT

EnumGetValue

CURRENT (33)	Current setpoint
DIRECT (40)	Value entry

Type of specification of the master position.
With CURRENT, the current master position is used.
With DIRECT, the value specified in the 'masterPosition' parameter is used.

masterPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the master position for which the slave value is to be determined.
Only relevant if 'masterPositionType:=DIRECT'.

1.3.22 Synchronous operation - Command tracking

1.3.22.1 _bufferFollowingObjectCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.22.2 _cancelFollowingObjectCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its CommandId in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled. Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.22.3 _getMotionStateOfFollowingObjectCommand

The function provides the motion status of an active synchronous operation command.

Return value: StructRetMotionCommandState

StructRetMotionCommandState
Return value with error code and motion status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionCommandId-State	Motion status	EnumMotionComman-dIdState	-

EnumMotionCommandIdState

NOT_EXISTENT (94)	'commandId' is not known or command is already completed
BUFFERED (207)	Command is in the command queue
IN_EXECUTION (208)	Command is being executed
IN_ACCELERATION (209)	Motion generated by the command is in the acceleration phase (only for TO axis)
IN_CONSTANT_MOTION (210)	Motion generated by the command is in the constant velocity phase (only for TO axis)
IN_DECELERATION (211)	Motion generated by the command is in the deceleration phase
AXIS_HOMED (212)	Axis is synchronized (only for TO axis)
INTERPOLATION_DONE (213)	Setpoint interpolation of command is complete
SYNCHRONIZING (214)	Synchronizing to synchronous operation (only for TO synchronous operation)
DESYNCHRONIZING (215)	Desynchronizing from synchronous operation (only for TO synchronous operation)
SYNCHRONIZED (216)	Synchronous operation is synchronized (only for TO synchronous operation)
MODIFICATION_ACTIVE (217)	Compensating motion for scaling or offset in synchronous operation is active (only for TO synchronous operation)
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult':
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'motionCommandIdState':
With NOT_EXISTENT, the 'commandId' for this synchronous operation instance is unknown or the command has already been completed.
With BUFFERED, the command is in the command queue.
With IN_EXECUTION, the command is being executed but no synchronous operation command is active.

IN_EXECUTION, IN_CONSTANT_MOTION, IN_DECELERATION, AXIS_HOMED are not displayed for synchronous operation.

With INTERPOLATION_DONE, the setpoint interpolation of the command is complete.
With SYNCHRONIZING, synchronization with a synchronous operation motion is in progress.
With DESYNCHRONIZING, desynchronization away from a synchronous operation motion is in progress.
With SYNCHRONIZED, a synchronous operation motion is synchronized.
With MODIFICATION_ACTIVE, a compensating motion for scaling or offset is active.
With EXECUTED, the command has been successfully executed. The EXECUTED status can only be read if the 'commandId' is stored.
With ABORTED, execution of the command has been aborted. The ABORTED status can only be read if the 'commandId' is stored.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.22.4 _getStateOfFollowingObjectCommand

This function returns the execution state of a synchronous operation command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult':
Description of the return values:
Corresponds to the list of return values under the _move function.

Return value 'commandIdState':
With ACTIVE, the command is being executed.
With NOT_EXISTENT, the 'commandId' is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for a synchronous start.
With EXECUTED, execution of the command is complete.
With ABORTED, execution of the command has been aborted.

Parameter:
followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.22.5 _removeBufferedFollowingObjectCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.23 Synchronous operation - Motion

1.3.23.1 _disableCamming

This function initiates the desynchronization of a master-slave relationship using a cam as the transmission function.
Several desynchronization conditions are supported.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

syncOffMode (optional)

Direction:	Input parameters
Data type:	EnumSyncOffModeCamming
Parameter index:	2
System default:	USER_DEFAULT

EnumSyncOffModeCamming

AT_THE_END_OF_CAM_CYCLE (14)	Cam cycle end
EFFECTIVE (45)	Last programmed setting
IMMEDIATELY (60)	Effective immediately
ON_MASTER_POSITION (100)	Specification of the desynchronization position of the master axis
ON_SLAVE_POSITION (101)	Specification of the desynchronization position of the slave axis
USER_DEFAULT (149)	User default

Specifies the desynchronization criterion.
With AT_THE_END_OF_CAM_CYCLE, desynchronization takes place when the active cam has reached its end point.
With EFFECTIVE, the last programmed setting is used.
With IMMEDIATELY, desynchronization commences immediately.
With ON_MASTER_POSITION, a master position is specified as an absolute desynchronization criterion.
With ON_SLAVE_POSITION, a slave position is specified as an absolute desynchronization criterion.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.cammingSettings.syncOffMode' is used.

syncOffPositionReference (optional)

Direction:	Input parameters
Data type:	EnumSyncOffPositionReference
Parameter index:	3
System default:	USER_DEFAULT

EnumSyncOffPositionReference

AXIS_STOPPED_AT_POSITION (15)	Stop before the desynchronization position
BEGIN_TO_STOP_WHEN_POSITION_REACHED (18)	Stop starting at the desynchronization position
EFFECTIVE (45)	Last programmed setting
STOP_SYMMETRIC_WITH_POSITION (137)	Stop symmetric to the desynchronization position
USER_DEFAULT (149)	User default

Specifies the reference point of the position specified in the 'syncOffMode' parameter.
With AXIS_STOPPED_AT_POSITION, the position refers to the end of the desynchronization operation.
With BEGIN_TO_STOP_WHEN_POSITION_REACHED, the position refers to the start of the desynchronization operation.
With EFFECTIVE, the last programmed mode is used.

With STOP_SYMMETRIC_WITH_POSITION, desynchronization is performed symmetrically around the specified position with the length programmed in parameter 'syncOffLength'.
STOP_SYMMETRIC_WITH_POSITION is only possible when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.

With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncProfile.syncOffPositionReference' is used.

syncProfileReference (optional)

Direction:	Input parameters
Data type:	EnumSyncProfileReference
Parameter index:	4
System default:	USER_DEFAULT

EnumSyncProfileReference

EFFECTIVE (45)	Last programmed setting
RELATE_SYNC_PROFILE_TO_LEADING_VALUE (113)	Master value-related synchronization
RELATE_SYNC_PROFILE_TO_TIME (114)	Time-related synchronization
USER_DEFAULT (149)	User default

Specifies the reference for the desynchronization operation.
With EFFECTIVE, the last programmed mode is used.
With RELATE_SYNC_PROFILE_TO_LEADING_VALUE, the slave is synchronized
relative to the master within the desynchronization length defined in the parameter 'syncOffLength'.
With RELATE_SYNC_PROFILE_TO_TIME, a desynchronization profile defined by the dynamic response data is used.
With USER_DEFAULT, the default mode setting defined in system variable 'userdefault.syncProfile.syncOffProfileReference' is used.

syncOffLengthType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	5
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for the desynchronization distance, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.
With DIRECT, the value set in the 'syncOffLength' parameter is used as the desynchronization length.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncProfile.syncOffLength' is used.

syncOffLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Desynchronization distance, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE' and 'syncOffLengthType:=DIRECT'.

syncOffPositionMasterType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	7
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of master position specification for desynchronization, relevant when the ON_MASTER_POSITION synchronization mode was selected.
With DIRECT, the value set in the 'syncOffPositionMaster' parameter is used as the position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncOffPosition.master' is used.

syncOffPositionMaster (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the master position for desynchronization, evaluation dependent on parameters 'SyncPositionMasterType' and 'syncOffMode'.
It is only used when parameter 'syncOffMode:=ON_MASTER_POSITION' and when parameter 'syncOffPositionMasterType:=DIRECT'.

syncOffPositionSlaveType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	9
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of slave position specification for desynchronization, relevant when the ON_SLAVE_POSITION synchronization mode was selected.
With DIRECT, the value set in the 'syncOffPositionsSlave' parameter is used as the position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncOffPosition.master' is used.

syncOffPositionSlave (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the slave position for desynchronization, evaluation dependent on parameters 'syncPositionSlaveType' and 'syncOffMode'.
It is only used when parameter 'syncOffMode:=ON_SLAVE_POSITION' and when parameter 'syncOffPositionSlaveType:=DIRECT'.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	11
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the transition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	13
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter. It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	15
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDyna-mics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter. It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	19
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	21
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	22
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	23
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	25
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeDisableCamming
Parameter index:	26
System default:	SEQUENTIAL

EnumMergeModeDisableCamming

IMMEDIATELY (60)	Replace immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
For synchronous operation, up to two synchronizing and/or desynchronizing commands can be processed in the interpolator at any one time. All of the commands or just one of the commands can wait for the synchronization event to occur, however only one command can be active at any one time, i.e. undergoing synchronization/desynchronization or in synchronized state. If the synchronization event occurs when a command is waiting, then it interrupts an active command.
In addition to the two commands in effect in the interpolator, there is also a command buffer that works in the same way as the buffer for axis motion commands.

The effectiveness of the synchronization and desynchronization commands with reference to the slave axis motion on other commands and on the buffer control can be set in the 'mergeMode' parameter of these commands.

With 'mergeMode:=IMMEDIATELY', the synchronous operation motion replaces all active slave axis motions. In the event that there are already two commands to be processed, the waiting or last programmed command as well as any command in the command buffer are cancelled.

With 'mergeMode:=SEQUENTIAL / NEXT_MOTION', the synchronous operation motion acts as the main axis motion; superimposing motions are not cancelled. If a main motion is already active on the slave axis, a pending synchronization event is not evaluated and the command continues to wait.
In the event that there are already two commands to be processed, the command stays in the command buffer and is not loaded to the interpolator; it does not cancel a command. Every additional command then waits in the command buffer with the setting 'nextCommand:=WHEN_BUFFER_READY'; with the setting 'nextCommand:=IMMEDIATELY', the command is not executed with an error in the return value.

Example: A waiting command can be deleted with a '_disable' command with 'mergeMode:=IMMEDIATELY' and 'synchronizing criterion:=IMMEDIATELY'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	27
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or motion is stopped in some other way.

The conditions AT_DECELERATION_START, WHEN_INTERPOLATION_DONE, and WHEN_MOTION_DONE only result in a transition when 'moveTimeOutType:=WITH_TIME_LIMIT' or if the command is aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	28
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

synchronizingDirection (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectSynchronizingDirection
Parameter index:	30
System default:	USER_DEFAULT

EnumFollowingObjectSynchronizingDirection

SHORTEST_WAY (121)	Shortest path without specification of the direction
USER_DEFAULT (149)	User default
SYSTEM_DEFINED (316)	Compatibility mode
SAME_DIRECTION (317)	Retain the direction of the slave axis
POSITIVE_DIRECTION (319)	Positive synchronization direction
NEGATIVE_DIRECTION (320)	Negative synchronization direction

Specifies the synchronization direction of the slave.
With SHORTEST_WAY, the shortest path, without specifying the direction, is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.synchronizingDirection' is used.
With SYSTEM_DEFINED (compatibility mode) and a stationary slave axis, the SHORTEST_WAY mode is used. With a moving slave axis, checks are made to see whether the current direction of motion can be retained and, as a result, whether reversing can be prevented.
With SAME_DIRECTION, the direction of the slave axis is retained.
With POSITIVE_DIRECTION, the positive synchronization direction is used.
With NEGATIVE_DIRECTION, the negative synchronization direction is used.

1.3.23.2 _disableGearing

This function deactivates synchronous operation using a gearbox as the transmission function. The desynchronization conditions and the dynamic response parameters can be specified.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

syncOffMode (optional)

Direction:	Input parameters
Data type:	EnumSyncOffModeGearing
Parameter index:	2
System default:	USER_DEFAULT

EnumSyncOffModeGearing

EFFECTIVE (45)	Last programmed setting
IMMEDIATELY (60)	Effective immediately
ON_MASTER_POSITION (100)	Specification of the desynchronization position of the master axis
ON_SLAVE_POSITION (101)	Specification of the desynchronization position of the slave axis
USER_DEFAULT (149)	User default

Specifies the desynchronization criterion.
With EFFECTIVE, the last programmed setting is used.
With IMMEDIATELY, desynchronization starts immediately.
With ON_MASTER_POSITION, a master value is used as the desynchronization position.
With ON_SLAVE_POSITION, a slave value is used as the desynchronization position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.syncOffMode' is used.

syncOffPositionReference (optional)

Direction:	Input parameters
Data type:	EnumSyncOffPositionReference
Parameter index:	3
System default:	USER_DEFAULT

EnumSyncOffPositionReference

AXIS_STOPPED_AT_POSITION (15)	Stop before the desynchronization position
BEGIN_TO_STOP_WHEN_POSITION_REACHED (18)	Stop starting at the desynchronization position
EFFECTIVE (45)	Last programmed setting
STOP_SYMMETRIC_WITH_POSITION (137)	Stop symmetric to the desynchronization position
USER_DEFAULT (149)	User default

Specifies the reference point of the position specified in the 'syncOffMode' parameter.
With AXIS_STOPPED_AT_POSITION, the position refers to the end of the desynchronization operation.
With BEGIN_TO_STOP_WHEN_POSITION_REACHED, the position refers to the start of the desynchronization operation.
With EFFECTIVE, the last programmed setting is used.

With STOP_SYMMETRIC_WITH_POSITION, desynchronization is performed symmetrically around the specified position with the programmed length. STOP_SYMMETRIC_WITH_POSITION is only possible when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.

With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncOffPositionReference' is used.

syncProfileReference (optional)

Direction:	Input parameters
Data type:	EnumSyncProfileReference
Parameter index:	4
System default:	USER_DEFAULT

EnumSyncProfileReference

EFFECTIVE (45)	Last programmed setting
RELATE_SYNC_PROFILE_TO_LEADING_VALUE (113)	Master value-related synchronization
RELATE_SYNC_PROFILE_TO_TIME (114)	Time-related synchronization
USER_DEFAULT (149)	User default

Specifies the desynchronization operation.
With EFFECTIVE, the last programmed mode is used.
With RELATE_SYNC_PROFILE_TO_LEADING_VALUE, the slave is synchronized relative to the motion of the master within the desynchronization length defined in the 'syncLength' parameter.
With RELATE_SYNC_PROFILE_TO_TIME, a synchronization profile specified by the dynamic response data is used.
With USER_DEFAULT, the default mode setting defined in system variable 'userdefault.syncProfile.syncProfileReference' is used.

syncOffLengthType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	5
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for the desynchronization distance, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.
With DIRECT, the value set in the 'syncOffLength' parameter is used as the desynchronization length.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncProfile.syncOffLength' is used.

syncOffLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Desynchronization distance, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE' and 'syncOffLengthType:=DIRECT'.

syncOffPositionMasterType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	7
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of master position specification for desynchronization, relevant when the ON_MASTER_POSITION synchronization mode was selected.
With DIRECT, the value set in the 'syncOffPositionMaster' parameter is used as the position.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.syncPositionMaster' is used.

syncOffPositionMaster (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the master position for desynchronization, evaluation dependent on the parameters 'syncPositionMasterType' and 'synchronizingMode'.
It is only used when parameter 'syncOffMode:=ON_MASTER_POSITION' and when parameter 'syncOffPositionSlaveType:=DIRECT'.

syncOffPositionSlaveType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	9
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of slave position specification for desynchronization, relevant when the ON_SLAVE_POSITION synchronization mode was selected.
With DIRECT, the value set in the 'syncOffPositionsSlave' parameter is used as the position.
With USER_DEFAULT, the default direction defined in system variable 'userdefault.gearingSettings.syncPositionSlave' is used.

syncOffPositionSlave (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the slave position for desynchronization, evaluation dependent on the parameters 'syncPositionSlaveType' and 'syncOffMode'.
It is only used when parameter 'syncOffMode:=ON_SLAVE_POSITION' and when parameter 'syncOffPositionSlaveType:=DIRECT'.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	11
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the transition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is interpreted as the velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	13
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	15
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	19
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	21
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	22
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	23
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	25
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeDisableGearing
Parameter index:	26
System default:	SEQUENTIAL

EnumMergeModeDisableGearing

IMMEDIATELY (60)	Effective immediately
NEXT_MOTION (89)	Attach and discard the pending command
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
For synchronous operation, up to two synchronizing and/or desynchronizing commands can be processed in the interpolator at any one time. All of the commands or just one of the commands can wait for the synchronization event to occur, however only one command can be active at any one time, i.e. undergoing synchronization/desynchronization or in synchronized state. If the synchronization event occurs when a command is waiting, then it interrupts an active command.
In addition to the two commands in effect in the interpolator, there is also a command buffer that works in the same way as the buffer for axis motion commands.

The effectiveness of the synchronization and desynchronization commands with reference to the slave axis motion on other commands and on the buffer control can be set in the 'mergeMode' parameter of these commands.

With 'mergeMode:=IMMEDIATELY', the synchronous operation motion replaces all active slave axis motions. In the event that there are already two commands to be processed, the waiting or last programmed command as well as any command in the command buffer are cancelled.

With 'mergeMode:=SEQUENTIAL / NEXT_MOTION', the synchronous operation motion acts as the main axis motion; superimposing motions are not cancelled. If a main motion is already active on the slave axis, a pending synchronization event is not evaluated and the command continues to wait.
In the event that there are already two commands to be processed, the command stays in the command buffer and is not loaded to the interpolator; it does not cancel a command. Every additional command then waits in the command buffer with the setting 'nextCommand:=WHEN_BUFFER_READY'; with the setting 'nextCommand:=IMMEDIATELY', the command is not executed with an error in the return value.

Example: A waiting command can be deleted with a '_disable' command with 'mergeMode:=IMMEDIATELY' and 'synchronizing criterion:=IMMEDIATELY'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	27
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or motion is stopped in some other way.

The conditions AT_DECELERATION_START, WHEN_INTERPOLATION_DONE, and WHEN_MOTION_DONE only result in a transition when 'moveTimeOutType:=WITH_TIME_LIMIT' or if the command is aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	28
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

synchronizingDirection (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectSynchronizingDirection
Parameter index:	30
System default:	USER_DEFAULT

EnumFollowingObjectSynchronizingDirection

SHORTEST_WAY (121)	Shortest path without specification of the direction
USER_DEFAULT (149)	User default
SYSTEM_DEFINED (316)	Compatibility mode
SAME_DIRECTION (317)	Retain the direction of the slave axis
POSITIVE_DIRECTION (319)	Positive synchronization direction
NEGATIVE_DIRECTION (320)	Negative synchronization direction

Specifies the synchronization direction of the slave.
With SHORTEST_WAY, the shortest path, without specifying the direction, is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.synchronizingDirection' is used.
With SYSTEM_DEFINED (compatibility mode) and a stationary slave axis, the SHORTEST_WAY mode is used. With a moving slave axis, checks are made to see whether the current direction of motion can be retained and, as a result, whether reversing can be prevented.
With SAME_DIRECTION, the direction of the slave axis is retained.
With POSITIVE_DIRECTION, the positive synchronization direction is used.
With NEGATIVE_DIRECTION, the negative synchronization direction is used.

1.3.23.3 _disableVelocityGearing

This function deactivates synchronous velocity operation using a gearbox as the transmission function.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	2
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	4
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	6
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	8
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	14
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeDisableGearing
Parameter index:	15
System default:	SEQUENTIAL

EnumMergeModeDisableGearing

IMMEDIATELY (60)	Effective immediately
NEXT_MOTION (89)	Attach and discard the pending command
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
For synchronous operation, up to two synchronizing and/or desynchronizing commands can be processed in the interpolator at any one time. All of the commands or just one of the commands can wait for the synchronization event to occur, however only one command can be active at any one time, i.e. undergoing synchronization/desynchronization or in synchronized state. If the synchronization event occurs when a command is waiting, then it interrupts an active command.
In addition to the two commands in effect in the interpolator, there is also a command buffer that works in the same way as the buffer for axis motion commands.

The effectiveness of the synchronization and desynchronization commands with reference to the slave axis motion on other commands and on the buffer control can be set in the 'mergeMode' parameter of these commands.

With 'mergeMode:=IMMEDIATELY', the synchronous operation motion replaces all active slave axis motions. In the event that there are already two commands to be processed, the waiting or last programmed command as well as any command in the command buffer are cancelled.

With 'mergeMode:=SEQUENTIAL / NEXT_MOTION', the synchronous operation motion acts as the main axis motion; superimposing motions are not cancelled. If a main motion is already active on the slave axis, a pending synchronization event is not evaluated and the command continues to wait.
In the event that there are already two commands to be processed, the command stays in the command buffer and is not loaded to the interpolator; it does not cancel a command. Every additional command then waits in the command buffer with the setting 'nextCommand:=WHEN_BUFFER_READY'; with the setting 'nextCommand:=IMMEDIATELY', the command is not executed with an error in the return value.

Example: A waiting command can be deleted with a '_disable' command with 'mergeMode:=IMMEDIATELY' and 'synchronizing criterion:=IMMEDIATELY'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	16
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or motion is stopped in some other way.

The conditions AT_DECELERATION_START, WHEN_INTERPOLATION_DONE, and WHEN_MOTION_DONE only result in a transition when 'moveTimeOutType:=WITH_TIME_LIMIT' or if the command is aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	17
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.23.4 _enableCamming

This function initiates the synchronization of a master-slave relationship using a cam as a coupling or synchronous operation function. Several synchronization conditions are supported for absolute and relative cam synchronous operation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumCammingDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumCammingDirection

EFFECTIVE (45)	Last programmed direction
NEGATIVE (85)	Opposite direction
POSITIVE (107)	Same direction
USER_DEFAULT (149)	User default

Specifies the direction in which the cam moves.
With EFFECTIVE, the last programmed direction is used.

With NEGATIVE, the cam moves in the direction opposite to the value change in the domain. When the master moves in the negative direction, the cam is executed in the direction of increasing master values, i.e., to the right.
When the master moves in the positive direction, the cam is executed in the direction of decreasing master values, i.e. 'to the left'. Synchronization conditions specified within the cam refer to the direction specification for camming. Synchronization can be performed in both directions of master motion.

With POSITIVE, the cam moves in the same direction as the value change in the domain. Since the values in the domain of the cam are always arranged in increasing order, motion of the master in the positive direction means that the axis moves along the cam in the direction of increasing master values or 'to the right'.
When the motion of the master is in the negative direction, the axis moves along the cam in the direction of decreasing master values, i.e. 'to the left'. Synchronization can be performed in both directions of master motion.

With USER_DEFAULT, the default setting defined in system variable 'userDefault.cammingSettings.direction' is used.

masterMode (optional)

Direction:	Input parameters
Data type:	EnumMasterMode
Parameter index:	3
System default:	USER_DEFAULT

EnumMasterMode

ABSOLUTE (1)	Absolute master reference
EFFECTIVE (45)	Last programmed master mode
RELATIVE (115)	Relative master reference
USER_DEFAULT (149)	User default

Specifies master value evaluation.
With ABSOLUTE, the master values are applied as absolute values in the domain range of the cam.
With EFFECTIVE, the last programmed master mode is used.
With RELATIVE, the master values are interpreted relative to the curve start point.
With USER_DEFAULT, the default mode setting defined in system variable 'userDefault.cammingSettings.masterMode' is used.

slaveMode (optional)

Direction:	Input parameters
Data type:	EnumSlaveMode
Parameter index:	4
System default:	USER_DEFAULT

EnumSlaveMode

ABSOLUTE (1)	Absolute slave reference
EFFECTIVE (45)	Last programmed slave mode
RELATIVE (115)	Relative slave reference
USER_DEFAULT (149)	User default

Specifies slave value evaluation.
With ABSOLUTE, the slave values are applied as absolute values in the cam's range of values.
With EFFECTIVE, the last programmed slave mode is used.
With RELATIVE, the slave values are applied relative to the start value of the cam. During synchronization, the slave axis also travels the path differential between the start of the cam and the cam start value.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.cammingSettings.slaveMode' is used.

cammingMode (optional)

Direction:	Input parameters
Data type:	EnumCammingMode
Parameter index:	5
System default:	USER_DEFAULT

EnumCammingMode

CYCLIC (35)	Cyclical cam
EFFECTIVE (45)	Last programmed setting
NOCYCLIC (92)	Non-cyclical cam
USER_DEFAULT (149)	User default

Specifies the cam processing mode.
With CYCLIC, the cam is executed cyclically. When the end point of the cam is reached, it continues from the start point.
With EFFECTIVE, the last programmed setting is used.
With NOCYCLIC, the function is terminated when the cam has been executed once.
With USER_DEFAULT, the default mode setting defined in system variable 'userDefault.cammingSettings.cammingMode' is used.

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	6

Specifies the function to be used as a cam.

synchronizingMode (optional)

Direction:	Input parameters
Data type:	EnumSyncModeCamming
Parameter index:	7
System default:	USER_DEFAULT

EnumSyncModeCamming

AT_THE_END_OF_CAM_CYCLE (14)	Transition at the end of the active cam
EFFECTIVE (45)	Last programmed setting
IMMEDIATELY (60)	Effective immediately
NEXT_WITH_REFERENCE (90)	At the next master axis position / not available
ON_MASTER_AND_SLAVE_POSITION (99)	Specification of the synchronization position of the master axis and the slave axis
ON_MASTER_POSITION (100)	Specification of the synchronization position of the master axis
USER_DEFAULT (149)	User default
IMMEDIATELY_AND_SLAVE_POSITION (315)	Effective immediately and synchronous position of the slave axis

Specifies the synchronization criterion.
With AT_THE_END_OF_CAM_CYCLE, the transition to the cam specified in the command takes place when the active cam has reached the endpoint. This function is only active when 'masterMode:=RELATIVE'. With 'masterMode:=ABSOLUTE', an error message is output if the function is started. The connection conditions (master starting point and slave starting point) result from the synchronization conditions of the new cam. The starting point within the cam is defined in the 'camStartPositionMaster' parameter.

With EFFECTIVE, the last programmed setting is used.

With IMMEDIATELY, the current positions of the master axis and the slave axis are used as synchronization positions; otherwise the specifications are the same as with ON_MASTER_AND_SLAVE_POSITION.

With ON_MASTER_AND_SLAVE_POSITION, in addition to the behavior for ON_MASTER_POSITION, a slave start position is specified.
With the setting 'slaveMode:=RELATIVE', the distance specified in the 'syncPositionSlave' parameter is used as an offset for the synchronization length of the slave.
With the setting 'slaveMode:=ABSOLUTE', the slave position determined by adding the slave position resulting from the cam for the master synchronization position and the slave position specified in the 'syncPositionSlave' parameter is used as the synchronization position.

With ON_MASTER_POSITION, a master position is specified as the criterion for synchronization.
With the setting 'masterMode:=RELATIVE', the position specified in the 'syncPositionMaster' parameter is used as the starting point for the cam; the starting point within the cam is specified in the 'camStartPositionMaster' parameter.
With the setting 'masterMode:=ABSOLUTE', the position specified in the 'syncPositionMaster' parameter is used as the synchronization position within the cam; this can be the starting point of the cam.

With USER_DEFAULT, the default setting defined in system variable 'userDefault.cammingSettings.synchronizingMode' is used.

With IMMEDIATELY_AND_SLAVE_POSITION, the current position of the master axis is used for the synchronization positions. The specified slave position works in the same way as with ON_MASTER_AND_SLAVE_POSITION.

syncPositionReference (optional)

Direction:	Input parameters
Data type:	EnumSyncPositionReference
Parameter index:	8
System default:	USER_DEFAULT

EnumSyncPositionReference

BE_SYNCHRONOUS_AT_POSITION (19)	Synchronize before the synchronization position
EFFECTIVE (45)	Last programmed setting
SYNCHRONIZE_SYMMETRIC (143)	Synchronize symmetric to the synchronization position
SYNCHRONIZE_WHEN_POSITION_REACHED (144)	Synchronize starting at synchronization position
USER_DEFAULT (149)	User default

Specifies the location of the master position.
With BE_SYNCHRONOUS_AT_POSITION, the synchronization is complete at the specified synchronization position.
With EFFECTIVE, the last programmed setting is used.

With SYNCHRONIZE_SYMMETRIC, synchronization is symmetric relative to the specified position.
SYNCHRONIZE_SYMMETRIC is only possible when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.

With SYNCHRONIZE_WHEN_POSITION_REACHED, synchronization starts at the specified synchronization position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.syncPositionSlave' of the synchronization profile is used.

syncProfileReference (optional)

Direction:	Input parameters
Data type:	EnumSyncProfileReference
Parameter index:	9
System default:	USER_DEFAULT

EnumSyncProfileReference

EFFECTIVE (45)	Last programmed setting
RELATE_SYNC_PROFILE_TO_LEADING_VALUE (113)	Master value-related synchronization
RELATE_SYNC_PROFILE_TO_TIME (114)	Time-related synchronization
USER_DEFAULT (149)	User default

Specifies the reference for the synchronization operation.
With EFFECTIVE, the last programmed setting is used.
With RELATE_SYNC_PROFILE_TO_LEADING_VALUE, the slave is synchronized relative to the master within the synchronization length defined in parameter 'syncLength'.
With RELATE_SYNC_PROFILE_TO_TIME, a synchronization profile specified by the dynamic response data is used.
With USER_DEFAULT, the default mode setting defined in system variable 'userdefault.syncProfile.syncProfileReference' is used.

syncLengthType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	10
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for the synchronization length, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.
With DIRECT, the value set in the 'syncLength' parameter is used as the synchronization length.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncProfile.syncLength' is used.

syncLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Synchronization length, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE' and 'syncLengthType := DIRECT'.

camStartPositionMasterType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	12
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Specifies a value for the definition of a start point within the cam, only effective if 'masterMode:=RELATIVE'.
With DIRECT, the value set in the 'camStartPositionMaster' parameter is used.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.cammingSettings.camStartPositionMaster' is used.

camStartPositionMaster (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the starting point within the cam.
Only relevant when 'camStartPositionMasterType:=DIRECT' and when 'masterMode:=RELATIVE'.

syncPositionMasterType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	14
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of the master position specification for synchronization,
relevant if ON_MASTER_POSITION or
ON_MASTER_AND_SLAVE_POSITION has been specified.
With 'masterMode:=ABSOLUTE', the start value of the cam is defined by specification of 'syncPositionMaster'.
With DIRECT, the value set in the 'syncPositionMaster' parameter is used as the position.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.cammingSettings.syncPositionMaster' is used.

syncPositionMaster (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the master position for synchronization,
relevant if synchronization mode ON_MASTER_POSITION or
ON_MASTER_AND_SLAVE_POSITION has been specified and 'syncPositionMaster-Type:=DIRECT'.

syncPositionSlaveType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	16
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for the slave position for synchronization; relevant if 'ON_MASTER_AND_SLAVE_POSITION' or 'IMMEDIATELY_AND_SLAVE_POSITION' has been specified as synchronization mode.
With DIRECT, the value set in the 'syncPositionsSlave' parameter is used as the position.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.cammingSettings.syncPositionSlave' is used.

syncPositionSlave (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Offset for the synchronization length of the slave axis, relevant if 'ON_MASTER_AND_SLAVE_POSITION' or 'IMMEDIATELY_AND_SLAVE_POSITION' has been specified as the synchronization mode and 'syncPositionSlaveType:=DIRECT'.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	18
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the transition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	20
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	22
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	24
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerk-Type' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccel-StartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	26
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	27
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	28
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	29
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccel-StartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	30
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' para-meter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	31
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAcce-lEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	32
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableCamming
Parameter index:	33
System default:	SEQUENTIAL

EnumMergeModeEnableCamming

IMMEDIATELY (60)	Effective immediately
NEXT_MOTION (89)	Attach and discard the pending command
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
For synchronous operation, up to two synchronizing and/or desynchronizing commands can be processed in the interpolator at any one time. All of the commands or just one of the commands can wait for the synchronization event to occur, however only one command can be active at any one time, i.e. undergoing synchronization/desynchronization or in synchronized state. If the synchronization event occurs when a command is waiting, then it interrupts an active command.
In addition to the two commands in effect in the interpolator, there is also a command buffer that works in the same way as the buffer for axis motion commands.

The effectiveness of the synchronization and desynchronization commands with reference to the slave axis motion on other commands and on the buffer control can be set in the 'mergeMode' parameter of these commands.
With 'mergeMode:=IMMEDIATELY', the synchronous operation motion replaces all active slave axis motions. In the event that there are already two commands to be processed, the waiting or last programmed command as well as any command in the command buffer are cancelled.

With 'mergeMode:=SEQUENTIAL / NEXT_MOTION', the synchronous operation motion acts as the main axis motion; superimposing motions are not cancelled. If a main motion is already active on the slave axis, a pending synchronization event is not evaluated and the command continues to wait.
In the event that there are already two commands to be processed, the command stays in the command buffer and is not loaded in the interpolator; it does not cancel a command. Every additional command then waits in the command buffer with the setting 'nextCommand:=WHEN_BUFFER_READY'; with the setting 'nextCommand:=IMMEDIATELY', the command is not executed with an error in the return value.

Example: A waiting command can be deleted with a '_disable' command with 'mergeMode:=IMMEDIATELY' and 'synchronizing criterion:=IMMEDIATELY'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCamming
Parameter index:	34
System default:	IMMEDIATELY

EnumNextCommandCamming

AT_MOTION_START (13)	Motion start
IMMEDIATELY (60)	Effective immediately
WHEN_ACCELERATION_DONE (156)	End of acceleration
WHEN_AXIS_SYNCHRONIZED (158)	After synchronization of axis
WHEN_BUFFER_READY (159)	As soon as axis is free

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_AXIS_SYNCHRONIZED, the command is terminated when the axis has been synchronized with the master.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	35
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

synchronizingDirection (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectSynchronizingDirection
Parameter index:	37
System default:	USER_DEFAULT

EnumFollowingObjectSynchronizingDirection

SHORTEST_WAY (121)	Shortest path without specification of the direction
USER_DEFAULT (149)	User default
SYSTEM_DEFINED (316)	Compatibility mode
SAME_DIRECTION (317)	Retain the direction of the slave axis
POSITIVE_DIRECTION (319)	Positive synchronization direction
NEGATIVE_DIRECTION (320)	Negative synchronization direction

Specifies the synchronization direction of the slave.
With SHORTEST_WAY, the shortest path, without specifying the direction, is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.synchronizingDirection' is used.
With SYSTEM_DEFINED (compatibility mode) and a stationary slave axis, the SHORTEST_WAY mode is used. With a moving slave axis, checks are made to see whether the current direction of motion can be retained and, as a result, whether reversing can be prevented.
With SAME_DIRECTION, the direction of the slave axis is retained.
With POSITIVE_DIRECTION, travel takes place in the positive synchronization direction.
With NEGATIVE_DIRECTION, travel takes place in the negative synchronization direction.

1.3.23.5 _enableGearing

This function activates synchronous operation using a gearbox as the transmission function. The synchronous operation modes, synchronizing conditions, and the dynamic response parameters can be specified.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumGearingDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumGearingDirection

BY_VALUE (24)	Sign
CURRENT (33)	Current slave direction
EFFECTIVE (45)	Last programmed direction
NEGATIVE (85)	Opposite direction
POSITIVE (107)	Same direction
REVERSE (116)	Opposite to the current slave direction
USER_DEFAULT (149)	User default

Specifies the motion direction in which the gearing operates.
With BY_VALUE, the gearing direction is determined from the sign of the gear ratio.
With CURRENT, the current direction of the slave values is retained. If the current direction is not available (axis is at a standstill), motion is resumed in the positive direction.
With EFFECTIVE, the last programmed direction setpoint is used.
With NEGATIVE, the synchronous operation direction is opposite.
With POSITIVE, the synchronous operation direction is the same.
With REVERSE, the last programmed direction is inverted; the direction is reversed.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.direction' is used.

gearingType (optional)

Direction:	Input parameters
Data type:	EnumGearingType
Parameter index:	3
System default:	USER_DEFAULT

EnumGearingType

ABSOLUTE (1)	Absolute gearbox
EFFECTIVE (45)	Last programmed type
RELATIVE (115)	Relative gearbox
USER_DEFAULT (149)	User default

Specifies type of gearing.
With ABSOLUTE, gearing is absolute relative to the axis zero for the relevant axes. A phase shift can be specified in parameter 'synchronizingMode:=ON_MASTER_AND_SLAVE_POSITION'.
With EFFECTIVE, the last programmed type is used.
With RELATIVE, gearing is relative with reference to the start position or synchronization position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.type' is used.

gearingMode (optional)

Direction:	Input parameters
Data type:	EnumGearingMode
Parameter index:	4
System default:	EFFECTIVE

EnumGearingMode

EFFECTIVE (45)	Last programmed type
GEARING_WITH_FRACTION (55)	Gear ratio as a fraction
GEARING_WITH_RATIO (56)	Gear ratio as floating-point number
USER_DEFAULT (149)	User default

Type of gear factor specification.
With EFFECTIVE, the last programmed type is used.
With GEARING_WITH_FRACTION, the ratio is specified as a fraction of two integers.
With GEARING_WITH_RATIO, the ratio is specified as a floating-point number.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.defineMode' is used.

gearingRatioType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	5
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of of gear ratio specification.
With DIRECT and the gear ratio setting 'gearingMode:=GEARING_WITH_FRACTION', the value set in the 'gearingNumerator' parameter is used as the numerator and the value set in 'gearingDenominator' is used as the denominator.
With DIRECT and the setting 'gearingMode:=GEARING_WITH_RATIO', the value set in the 'gearingRatio' parameter is used as the gear ratio.

With EFFECTIVE, the last programmed values are used in accordance with the 'gearingMode' setting.

With USER_DEFAULT and the setting 'gearingMode:=GEARING_WITH_FRACTION', the default settings defined in system variables 'userDefault.gearingSettings.numerator' and 'userDefault.gearingSettings.denominator' are used.
With USER_DEFAULT and the setting 'gearingMode:=GEARING_WITH_RATIO', the default setting defined in system variable 'userDefault.gearingSettings.ratio' is used.

gearingRatio (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	1.0

Specifies the gear factor.
Only effective with the setting 'gearingMode:=GEARING_WITH_RATIO' and dependent on the parameter 'gearingRatioType'.
With 'gearingRatioType:=DIRECT', the value represents the floating-point gear ratio.
In all other cases, the parameter is irrelevant.

gearingNumerator (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	7
System default:	1

Specifies the numerator of the gear ratio with the setting 'gearingMode:=GEARING_WITH_FRACTION', depending on the parameter "gearingRatioType".
With DIRECT, the value represents the numerator of the gear ratio.
In all other cases, the parameter is irrelevant.

gearingDenominator (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	8
System default:	1

Specifies the denominator of the gear ratio as a fraction, only with the setting 'gearing-Mode:=GEARING_WITH_FRACTION', depending on the parameter "gearingRatioType".
With DIRECT, the value represents the denominator of the gear ratio.
In all other cases, the parameter is irrelevant.

synchronizingMode (optional)

Direction:	Input parameters
Data type:	EnumSyncModeGearing
Parameter index:	9
System default:	USER_DEFAULT

EnumSyncModeGearing

EFFECTIVE (45)	Last programmed setting
IMMEDIATELY (60)	Effective immediately
NEXT_WITH_REFERENCE (90)	Not available
ON_MASTER_AND_SLAVE_POSITION (99)	Specification of the synchronization position of the master axis and the slave axis
ON_MASTER_POSITION (100)	Specification of the synchronization position of the master axis
ON_SLAVE_POSITION (101)	Specification of the synchronization position of the slave axis
USER_DEFAULT (149)	User default
IMMEDIATELY_AND_SLAVE_POSITION (315)	Synchronize immediately with offset
IMMEDIATELY_AND_BE_SYNCHRONOUS_AT_MASTER_POSITION (370)	Not available

Specifies the synchronization criterion.

With EFFECTIVE, the last programmed setting is used.

With IMMEDIATELY, synchronization takes place immediately.

With ON_MASTER_AND_SLAVE_POSITION, a master value is used as the synchronization criterion and a slave position is also specified at the start point in order to define an offset.

With ON_MASTER_POSITION, a master position is used as the synchronization criterion.

With ON_SLAVE_POSITION, the position of a slave axis is used as the synchronization criterion.

With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.synchronizingMode' is used.

With IMMEDIATELY_AND_SLAVE_POSITION, synchronization is performed immediately. A slave position is also specified for the definition of an offset.

IMMEDIATELY_AND_BE_SYNCHRONOUS_AT_MASTER_POSITION not currently used.

syncPositionReference (optional)

Direction:	Input parameters
Data type:	EnumSyncPositionReference
Parameter index:	10
System default:	USER_DEFAULT

EnumSyncPositionReference

BE_SYNCHRONOUS_AT_POSITION (19)	Synchronize before the synchronization position
EFFECTIVE (45)	Last programmed setting
SYNCHRONIZE_SYMMETRIC (143)	Synchronize symmetric to the synchronization position
SYNCHRONIZE_WHEN_POSITION_REACHED (144)	Synchronize starting at synchronization position
USER_DEFAULT (149)	User default

Specifies the location of the master position for the synchronization profile.
With BE_SYNCHRONOUS_AT_POSITION, the synchronization is complete at the specified synchronization position.
With EFFECTIVE, the last programmed setting is used.

With SYNCHRONIZE_SYMMETRIC, synchronization is symmetric relative to the specified position.
SYNCHRONIZE_SYMMETRIC is only possible when
'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.

With SYNCHRONIZE_WHEN_POSITION_REACHED, synchronization starts at the specified synchronization position.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncProfile.syncPositionReference' is used.

syncProfileReference (optional)

Direction:	Input parameters
Data type:	EnumSyncProfileReference
Parameter index:	11
System default:	USER_DEFAULT

EnumSyncProfileReference

EFFECTIVE (45)	Last programmed setting
RELATE_SYNC_PROFILE_TO_LEADING_VALUE (113)	Master value-related synchronization
RELATE_SYNC_PROFILE_TO_TIME (114)	Time-related synchronization
USER_DEFAULT (149)	User default

Specifies the reference for the synchronization operation.
With EFFECTIVE, the last programmed setting is used.
With RELATE_SYNC_PROFILE_TO_LEADING_VALUE, the slave is synchronized
relative to the master within the synchronization length defined in parameter 'syncLength'.
With RELATE_SYNC_PROFILE_TO_TIME, a synchronization profile specified by the dynamic response data is used.
With USER_DEFAULT, the default mode setting defined in system variable 'userdefault.syncProfile.syncProfileReference' is used.

syncLengthType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	12
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for the synchronization length, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE'.
With DIRECT, the value set in the 'syncLength' parameter is used as the synchronization length.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.syncProfile.syncLength' is used.

syncLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the synchronization length, only relevant when 'syncProfileReference:=RELATE_SYNC_PROFILE_TO_LEADING_VALUE' and 'syncLengthType:=DIRECT'.

syncPositionMasterType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	14
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of master position specification for synchronization, relevant if the 'synchronizingMode' parameter contains the value ON_MASTER_POSITION or ON_MASTER_AND_SLAVE_POSITION.
With DIRECT, the value set in the 'syncPositionMaster' parameter is used as the synchronization position.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.syncPositionMaster' is used.

syncPositionMaster (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the master position for synchronization, evaluation dependent on the parameters 'masterType' and 'synchronizingMode'.
It is only used when parameter 'synchronizingMode:=ON_MASTER_POSITION' or 'synchronizingMode:=ON_MASTER_AND_SLAVE_POSITION' and when parameter 'syncPositionMasterType:=DIRECT'.

syncPositionSlaveType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	16
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of the slave position specification for synchronization, is only used when for parameter 'synchronizingMode' ON_MASTER_AND_SLAVE_POSITION, ON_SLAVE_POSITION or IMMEDIATELY_AND_SLAVE_POSITION has been specified.
With DIRECT, the value set in the 'syncPositionsSlave' parameter is used as the synchronization position.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.gearingSettings.syncPositionSlave' is used.

syncPositionSlave (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the slave position for synchronization, evaluation dependent on parameters 'syncPositionSlaveType' and 'synchronizingMode'.
It is only used when parameter 'synchronizingMode:=ON_MASTER_AND_SLAVE_POSITION', 'ON_SLAVE_POSITION' or 'IMMEDIATELY_AND_SLAVE_POSITION' and when parameter 'syncPositionsSlaveType:=DIRECT'.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	18
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the synchronization operation.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is interpreted as the velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	20
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter. It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	22
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDyna-mics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter. It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	24
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	26
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	27
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	28
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	29
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	30
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	31
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	32
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableGearing
Parameter index:	33
System default:	SEQUENTIAL

EnumMergeModeEnableGearing

IMMEDIATELY (60)	Replace immediately
NEXT_MOTION (89)	Attach and discard the pending command
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
For synchronous operation, up to two synchronizing and/or desynchronizing commands can be processed in the interpolator at any one time. All or only one can wait for the synchronization event to occur, however, only one can be active at any one time, i.e. in the synchronizing/desynchronizing or synchronous condition. If the synchronization event occurs when a command is waiting, then it interrupts an active command.
In addition to the two commands in effect in the interpolator, there is also a command buffer that works in the same way as the buffer for axis motion commands.

The effectiveness of the synchronization and desynchronization commands with reference to the slave axis motion on other commands and on the buffer control can be set in the 'mergeMode' parameter of these commands.

With 'mergeMode:=IMMEDIATELY', the synchronous operation motion replaces all active slave axis motions. In the event that there are already two commands to be processed, the waiting or last programmed command as well as any command in the command buffer are cancelled.

With 'mergeMode:=SEQUENTIAL / NEXT_MOTION', the synchronous operation motion acts as the main axis motion; superimposing motions are not cancelled. If a main motion is already active on the slave axis, a pending synchronization event is not evaluated and the command continues to wait.
In the event that there are already two commands to be processed, the command stays in the command buffer and is not loaded to the interpolator; it does not cancel a command. Every additional command then waits in the command buffer with the setting 'nextCommand:=WHEN_BUFFER_READY'; with the setting 'nextCommand:=IMMEDIATELY', the command is not executed with an error in the return value.

Example:
A waiting command can be deleted with a '_disable' command with 'mergeMode:=IMMEDIATELY' and 'synchronizing criterion:=IMMEDIATELY'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandGearing
Parameter index:	34
System default:	IMMEDIATELY

EnumNextCommandGearing

AT_MOTION_START (13)	Motion start
IMMEDIATELY (60)	Effective immediately
WHEN_ACCELERATION_DONE (156)	End of acceleration
WHEN_AXIS_SYNCHRONIZED (158)	After synchronization of axis
WHEN_BUFFER_READY (159)	As soon as axis is free

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_AXIS_SYNCHRONIZED, the transition takes place after the axis has been synchronized with the master.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	35
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

synchronizingDirection (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectSynchronizingDirection
Parameter index:	37
System default:	USER_DEFAULT

EnumFollowingObjectSynchronizingDirection

SHORTEST_WAY (121)	Shortest path without specification of the direction
USER_DEFAULT (149)	User default
SYSTEM_DEFINED (316)	Compatibility mode
SAME_DIRECTION (317)	Retain the direction of the slave axis
POSITIVE_DIRECTION (319)	Positive synchronization direction
NEGATIVE_DIRECTION (320)	Negative synchronization direction

Specifies the synchronization direction of the slave.
With SHORTEST_WAY, the shortest path, without specifying the direction, is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.synchronizingDirection' is used.
With SYSTEM_DEFINED (compatibility mode) and a stationary slave axis, the SHORTEST_WAY mode is used. With a moving slave axis, checks are made to see whether the current direction of motion can be retained and, as a result, whether reversing can be prevented.
With SAME_DIRECTION, the direction of the slave axis is retained.
With POSITIVE_DIRECTION, the synchronization direction is positive.
With NEGATIVE_DIRECTION, the synchronization direction is negative.

synchronizingWithLookAhead (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectSynchronizeWithLookAhead
Parameter index:	38
System default:	USER_DEFAULT

EnumFollowingObjectSynchronizeWithLookAhead

USER_DEFAULT (149)	User default
STANDARD (358)	Look ahead with s and v
EXTENDED_LOOK_AHEAD (359)	Look ahead with s, v and a

Specifies the use of 'look ahead' master.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.synchronizingWithLookAhead' is used.
With STANDARD, the synchronization motion is calculated taking into account the position and velocity of the master.
With EXTENDED_LOOK_AHEAD, the synchronization motion is calculated taking into account the position, velocity, and acceleration of the master.

1.3.23.6 _enableVelocityGearing

This function activates synchronous velocity operation using a gearbox as the transmission function.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumGearingDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumGearingDirection

BY_VALUE (24)	Sign
CURRENT (33)	Current slave direction
EFFECTIVE (45)	Last programmed direction
NEGATIVE (85)	Opposite direction
POSITIVE (107)	Same direction
REVERSE (116)	Opposite to the current slave direction
USER_DEFAULT (149)	User default

Specifies the motion direction in which the gearing operates.
With BY_VALUE, the gearing direction is determined from the sign of the gear ratio.
With CURRENT, the current direction of the slave values is retained. If the current direction is not available (axis is at a standstill), motion is resumed in the positive direction.
With EFFECTIVE, the last programmed direction setpoint is used.
With NEGATIVE, the synchronous operation direction is opposite.
With POSITIVE, the synchronous operation direction is the same.
With REVERSE, the last programmed direction is inverted; the direction is reversed.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.direction' is used.

gearingRatioType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValueType
Parameter index:	3
System default:	USER_DEFAULT

EnumExtendedValueType

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of of gear ratio specification.
With DIRECT, the value set in the 'gearingRatio' parameter is used as the gear ratio.
With EFFECTIVE, the last programmed gear ratio value is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.gearingSettings.ratio' is used.

gearingRatio (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	1.0

Specifies the gear factor.
Only effective with the setting 'gearingMode:=GEARING_WITH_RATIO' and dependent on the parameter 'gearingRatioType'.
With 'gearingRatioType:=DIRECT', the value represents the floating-point gear ratio.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	17
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableGearing
Parameter index:	18
System default:	SEQUENTIAL

EnumMergeModeEnableGearing

IMMEDIATELY (60)	Replace immediately
NEXT_MOTION (89)	Attach and discard the pending command
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
For synchronous operation, up to two synchronizing and/or desynchronizing commands can be processed in the interpolator at any one time. All or only one can wait for the synchronization event to occur, however, only one can be active at any one time, i.e. in the synchronizing/desynchronizing or synchronous condition. If the synchronization event occurs when a command is waiting, then it interrupts an active command.
In addition to the two commands in effect in the interpolator, there is also a command buffer that works in the same way as the buffer for axis motion commands.

The effectiveness of the synchronization and desynchronization commands with reference to the slave axis motion on other commands and on the buffer control can be set in the 'mergeMode' parameter of these commands.

With 'mergeMode:=IMMEDIATELY', the synchronous operation motion replaces all active slave axis motions. In the event that there are already two commands to be processed, the waiting or last programmed command as well as any command in the command buffer are cancelled.

With 'mergeMode:=SEQUENTIAL / NEXT_MOTION', the synchronous operation motion acts as the main axis motion; superimposing motions are not cancelled. If a main motion is already active on the slave axis, a pending synchronization event is not evaluated and the command continues to wait.
In the event that there are already two commands to be processed, the command stays in the command buffer and is not loaded to the interpolator; it does not cancel a command. Every additional command then waits in the command buffer with the setting 'nextCommand:=WHEN_BUFFER_READY'; with the setting 'nextCommand:=IMMEDIATELY', the command is not executed with an error in the return value.

Example: A waiting command can be deleted with a '_disable' command with 'mergeMode:=IMMEDIATELY' and 'synchronizing criterion:=IMMEDIATELY'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandGearing
Parameter index:	19
System default:	IMMEDIATELY

EnumNextCommandGearing

AT_MOTION_START (13)	Motion start
IMMEDIATELY (60)	Effective immediately
WHEN_ACCELERATION_DONE (156)	End of acceleration
WHEN_AXIS_SYNCHRONIZED (158)	After synchronization of axis
WHEN_BUFFER_READY (159)	As soon as axis is free

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_AXIS_SYNCHRONIZED, the transition takes place after the axis has been synchronized with the master.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	20
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.24 Synchronous operation - Compensation and superimposition

1.3.24.1 _setCammingOffset

This function activates an offset in the master or slave range during camming. When active in the current synchronous operation function, the offset remains active until the camming function is replaced or desynchronized.
When permanently set the offset becomes active when subsequent synchronous operation functions are started.
The dynamic response values for the transition can be specified.
The current and permanent offsets in effect can be read in system variable 'cammingAdjustments'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

offsetRange (optional)

Direction:	Input parameters
Data type:	EnumRange
Parameter index:	2
System default:	MASTER_RANGE

EnumRange

MASTER_RANGE (74)	Master range
SLAVE_RANGE (123)	Slave range

Specifies whether the master range or slave range is scaled.
With MASTER_RANGE, the master range is scaled.
With SLAVE_RANGE, the slave range is scaled.

offsetSpecification (optional)

Direction:	Input parameters
Data type:	EnumScaleSpecification
Parameter index:	3
System default:	IMMEDIATELY

EnumScaleSpecification

IMMEDIATELY (60)	Effective immediately
NEXT_CAM_CYCLE (88)	Next cam cycle

Specifies when the offset takes effect.
With IMMEDIATELY, the offset is effective immediately.
With NEXT_CAM_CYCLE, the offset for a cyclic cam is effective in the next cycle.

offsetMode (optional)

Direction:	Input parameters
Data type:	EnumAbsoluteRelative
Parameter index:	4
System default:	ABSOLUTE

EnumAbsoluteRelative

ABSOLUTE (1)	Absolute
RELATIVE (115)	Relative

Specifies the type of offset specification.
With ABSOLUTE, the offset is specified as an absolute value.
With RELATIVE, the offset is added to the existing value.

offsetValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the offset.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	6
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the transition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	8
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	10
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	18
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	20
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandScaling
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandScaling

AT_DECELERATION_START (12)	Start of deceleration phase
IMMEDIATELY (60)	Asynchronous commands
WHEN_ACCELERATION_DONE (156)	End of acceleration
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

activationMode (optional)

Direction:	Input parameters
Data type:	EnumCammingActivationMode
Parameter index:	24
System default:	ACTUAL_VALUE

EnumCammingActivationMode

DEFAULT_VALUE (2)	Active at start of all subsequent synchronous operation commands
ACTUAL_VALUE (34)	Active for active synchronous operation command
ACTUAL_AND_DEFAULT_VALUE (237)	Active for active synchronous operation command and at start of all subsequent synchronous operation commands

Specifies how scaling is activated.
With DEFAULT_VALUE, the scaling is active at the start of all subsequent synchronous operation commands.
With ACTUAL_VALUE, the scaling is active in the currently active synchronous operation command.
With ACTUAL_AND_DEFAULT_VALUE, the scaling is active in the active synchronous operation command and in all subsequent synchronous operation commands.

dynamicReference (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectDynamicReference
Parameter index:	25
System default:	TOTAL_MOVE

EnumFollowingObjectDynamicReference

TOTAL_MOVE (355)	Dynamic parameters refer to the total motion
OFFSET_MOVE (356)	Dynamic parameters refer to the differential motion

Specifies whether the dynamic response values of the command refer to the resulting overall motion or to the differential motion.
With TOTAL_MOVE, the dynamic response values refer to the total motion.
With OFFSET_MOVE, the dynamic response values refer to the differential motion.

1.3.24.2 _setCammingScale

This function activates scaling of the master or slave range.
When active in the current synchronous operation function, the scaling remains active until the camming function is replaced or desynchronized.
When permanently set, the scaling is active at the start of subsequent synchronous operation functions.
The dynamic response values for the transition can be specified.
The current and permanent scalings in effect can be read in system variable 'cammingAdjustments'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

scalingRange (optional)

Direction:	Input parameters
Data type:	EnumRange
Parameter index:	2
System default:	MASTER_RANGE

EnumRange

MASTER_RANGE (74)	Master range
SLAVE_RANGE (123)	Slave range

Specifies whether the master range or slave range is scaled.
With MASTER_RANGE, the master range is scaled.
With SLAVE_RANGE, the slave range is scaled.

scaleSpecification (optional)

Direction:	Input parameters
Data type:	EnumScaleSpecification
Parameter index:	3
System default:	IMMEDIATELY

EnumScaleSpecification

IMMEDIATELY (60)	Effective immediately
NEXT_CAM_CYCLE (88)	Next cam cycle

Specifies when the scaling is activated.
With IMMEDIATELY, the scaling is effective immediately.
With NEXT_CAM_CYCLE, the scaling for a cyclic cam is effective in the next cam cycle.

scaleValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	1.0

Specifies the scaling factor.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the transition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerk-Type' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccel-StartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccel-StartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' para-meter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAcce-lEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandScaling
Parameter index:	20
System default:	IMMEDIATELY

EnumNextCommandScaling

AT_DECELERATION_START (12)	Start of deceleration phase
IMMEDIATELY (60)	Asynchronous commands
WHEN_ACCELERATION_DONE (156)	End of acceleration
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	21
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

activationMode (optional)

Direction:	Input parameters
Data type:	EnumCammingActivationMode
Parameter index:	23
System default:	ACTUAL_VALUE

EnumCammingActivationMode

DEFAULT_VALUE (2)	Active at start of all subsequent synchronous operation commands
ACTUAL_VALUE (34)	Active for active synchronous operation command
ACTUAL_AND_DEFAULT_VALUE (237)	Active for active synchronous operation command and at start of all subsequent synchronous operation commands

Specifies how scaling is activated.
With DEFAULT_VALUE, the scaling is active at the start of all subsequent synchronous operation commands.
With ACTUAL_VALUE, the scaling is active in the currently active synchronous operation command.
With ACTUAL_AND_DEFAULT_VALUE, the scaling is active in the active synchronous operation command and in all subsequent synchronous operation commands.

dynamicReference (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectDynamicReference
Parameter index:	24
System default:	TOTAL_MOVE

EnumFollowingObjectDynamicReference

TOTAL_MOVE (355)	Dynamic parameters refer to the total motion
OFFSET_MOVE (356)	Dynamic parameters refer to the differential motion

Specifies whether the dynamic response values of the command refer to the resulting overall motion or to the differential motion.
With TOTAL_MOVE, the dynamic response values refer to the total motion.
With OFFSET_MOVE, the dynamic response values refer to the differential motion.

1.3.24.3 _setGearingOffset

With gearing, this function activates an offset in the master or slave range. When active in the current synchronous operation function, the offset remains in use until the gearing function is replaced or desynchronized;
when permanently set the offset becomes active when subsequent synchronous operation functions are started.
The current and permanent offsets in effect can be read in system variables of 'gearingAdjustments'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

offsetRange (optional)

Direction:	Input parameters
Data type:	EnumRange
Parameter index:	2
System default:	MASTER_RANGE

EnumRange

MASTER_RANGE (74)	Master range
SLAVE_RANGE (123)	Slave range

Specifies whether the master range or slave range is scaled.
With MASTER_RANGE, the master range is scaled.
With SLAVE_RANGE, the slave range is scaled.

offsetMode (optional)

Direction:	Input parameters
Data type:	EnumAbsoluteRelative
Parameter index:	3
System default:	ABSOLUTE

EnumAbsoluteRelative

ABSOLUTE (1)	Absolute
RELATIVE (115)	Relative

Specifies the type of offset specification.
With ABSOLUTE, the offset is specified as an absolute value.
With RELATIVE, the offset is added to the existing value.

offsetValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the offset.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the transition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerk-Type' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccel-StartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccel-StartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' para-meter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAcce-lEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandScaling
Parameter index:	20
System default:	IMMEDIATELY

EnumNextCommandScaling

AT_DECELERATION_START (12)	Start of deceleration phase
IMMEDIATELY (60)	Asynchronous commands
WHEN_ACCELERATION_DONE (156)	End of acceleration
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	21
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

activationMode (optional)

Direction:	Input parameters
Data type:	EnumGearingActivationMode
Parameter index:	23
System default:	ACTUAL_VALUE

EnumGearingActivationMode

DEFAULT_VALUE (2)	Active at start of all subsequent synchronous operation commands
ACTUAL_VALUE (34)	Active for active synchronous operation command
ACTUAL_AND_DEFAULT_VALUE (237)	Active for active synchronous operation command and at start of all subsequent synchronous operation commands

Specifies how scaling is activated.
With ACTUAL_VALUE, the scaling is active in the currently active synchronous operation command.
With DEFAULT_VALUE, the scaling is active at the start of all subsequent synchronous operation commands.
With ACTUAL_AND_DEFAULT_VALUE, the scaling is active in the active synchronous operation command and in all subsequent synchronous operation commands.

dynamicReference (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectDynamicReference
Parameter index:	24
System default:	TOTAL_MOVE

EnumFollowingObjectDynamicReference

TOTAL_MOVE (355)	Dynamic parameters refer to the total motion
OFFSET_MOVE (356)	Dynamic parameters refer to the differential motion

Specifies whether the dynamic response values of the command refer to the resulting overall motion or to the differential motion.
With TOTAL_MOVE, the dynamic response values refer to the total motion.
With OFFSET_MOVE, the dynamic response values refer to the differential motion.

1.3.24.4 _setMaster

This function specifies a master object for the synchronous operation relationship.
The possible master objects for synchronous operation are set in the configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

master

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

transientBehavior (optional)

Direction:	Input parameters
Data type:	EnumFollowingObjectDynamicMergeMode
Parameter index:	3
System default:	DIRECT

EnumFollowingObjectDynamicMergeMode

DIRECT (40)	Immediate master changeover
WITH_DYNAMICS (314)	Master changeover with dynamic values
WITH_NEXT_SYNCHRONIZING (384)	Master changeover at next synchronization operation

Type of generated transition to the new master object.
With DIRECT, the transition takes place immediately.
With WITH_DYNAMICS, the transition to the new master takes place with the dynamic parameters specified in the command.
With WITH_NEXT_SYNCHRONIZING, the transition takes place at the next synchronization operation.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	4
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition for the synchronization operation.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is interpreted as the velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefault.syncDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the transition velocity; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.syncDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	8
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.syncDynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases and when a TRAPEZOIDAL velocity profile is used, the parameter is irrelevant.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	18
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.syncDynamics.profile' is used.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandSetMaster
Parameter index:	19
System default:	WHEN_COMMAND_DONE

EnumNextCommandSetMaster

AT_DECELERATION_START (12)	Start of deceleration phase
IMMEDIATELY (60)	Effective immediately
WHEN_ACCELERATION_DONE (156)	End of acceleration
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_COMMAND_DONE, the transition takes place once the axis has been synchronized with the master.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	20
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.25 Synchronous operation - Object and Alarm Handling

1.3.25.1 _cancelFollowingObjectCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its CommandId in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled. Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.25.2 _disableFollowingObjectSimulation

This function switches the synchronous operation relationship out of simulation mode.
The synchronous operation values are output to the slave axis again.
An existing following error is removed immediately with maximum values.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.25.3 _enableFollowingObjectSimulation

This function switches the synchronization operation relationship to simulation mode.
The synchronous operation values are calculated, but are not output to the slave axis.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.25.4 _getFollowingObjectErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.25.5 _getFollowingObjectErrorState

This function provides information on whether synchronous operation relationship alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

1.3.25.6 _resetFollowingObject

This function switches a synchronous operation relationship to an initial state.
Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteSynchronizingCommandsOnly (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	7
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies whether the active commands that are waiting in the technology object are to removed selectively without resetting the entire technology object.
With NO, the entire technology object is reset.
With YES, only the waiting active commands are reset.

1.3.25.7 _resetFollowingObjectConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

1.3.25.8 _resetFollowingObjectError

This function resets errors on the synchronous operation relationship.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

followingObject

Direction:	Input parameters
Data type:	FollowingObjectType
Parameter index:	1

Specifies the technology object of the 'followingObject' type or a variable of the 'FollowingObjectType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.26 Synchronous axis - Information and conversion

1.3.26.1 _getAxisInternalPosition

This function supplies the associated encoder position value for a specified position value in the axis coordinate system.

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

userValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the position value in the axis coordinate system for which the associated encoder position is to be determined.

1.3.26.2 _getAxisUserPosition

This function supplies the position value in the axis coordinate system for a specified encoder position value.

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

internalValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the encoder position value for which the position value is to be determined in the axis coordinate system.

1.3.26.3 _getProgrammedTargetPosition

This function supplies the absolute end position currently programmed on the axis. Any superimposition is taken into account.

Return value: StructRetGetAxisProgrammedTargetPosition

StructRetGetAxisProgrammedTargetPosition
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
targetPositionType	Validity and meaning of targetPosition	EnumAxisProgrammedTargetPosition	-
targetPosition	Programmed target position	LREAL	-

EnumAxisProgrammedTargetPosition

NO_POS_MOTION_ACTIVE (275)	No commands with position reference active
BASIC_POS_MOTION_ACTIVE (276)	Main motion with position reference active
SUPERIMPOSED_POS_MOTION_ACTIVE (277)	Superimposed motion with position reference active
BASIC_AND_SUPERIMPOSED_POS_MOTION_ACTIVE (278)	Main motion and superimposed motion with position reference active
BASIC_POS_AND_SUPERIMPOSED_NON_POS_MOTION_ACTIVE (279)	Main motion with position reference and superimposed motion without position reference active; the targetPosition contains the absolute target position of the main motion in the reference system of the main motion
BASIC_NON_POS_AND_SUPERIMPOSED_POS_MOTION_ACTIVE (280)	Main motion without position reference and superimposed motion with position reference active; targetPosition contains the absolute target position of the superimposed motion is in the reference system of the superimposed motion

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

posCommandSpecification (optional)

Direction:	Input parameters
Data type:	EnumAxisPosCommandSpecification
Parameter index:	2
System default:	ALL_POSITION_RELATED_MOTION

EnumAxisPosCommandSpecification

ALL_POSITION_RELATED_MOTION (273)	All commands with position reference
ONLY_POSITION_COMMAND (274)	_pos commands only

Selection of active commands with position reference in the interpolator that are related to the command result.
With ALL_POSITION_RELATED_MOTION, the output end position relates to all commands with position reference.
With ONLY_POSITION_COMMAND, the output end position relates to the active _pos commands.

1.3.26.4 _setAndGetEncoderValue

This function returns the difference between the specified measuring systems or synchronizes two measuring systems.
During synchronization, it is only possible to change the value of a measuring system if it is not currently being used for position control.

Return value: StructRetEncoderValue

StructRetEncoderValue
Return value with error code and difference

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
encoderDifference	Difference of specified measuring systems	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value encoderDifference:

The difference of the specified measuring systems is indicated in 'encoderDifference'.

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

mode (optional)

Direction:	Input parameters
Data type:	EnumSetNoSet
Parameter index:	2
System default:	SET

EnumSetNoSet

SET (178)	Set value
NO_SET (179)	Do not set value

Specifies whether the actual value of the measuring system that is specified in the 'encoder' parameter should be set.
With SET, the actual value of the measuring system that is specified in the 'encoder' parameter is set.
With NO_SET, the actual value of the measuring system that is specified in the 'encoder' parameter is not set.

encoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	1

Specifies the measuring system to be selected or synchronized.
The encoder number is assigned during configuration.
The valid range is [1, 8]

referenceEncoderType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	4
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Specifies the reference measuring system.
With ACTUAL, the active measuring system is used.
With DIRECT, the measuring system specified in the 'referenceEncoder' parameter is used.

referenceEncoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	5
System default:	1

Specifies the reference measuring system.
The encoder number is assigned during configuration.
The valid range is [1, 8]

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command takes place after changeover to the new measuring system.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.27 Synchronous axis - Command tracking

1.3.27.1 _bufferAxisCommandId

This function enables the commandId and associated command status to be saved for a period after command execution so that it is still available after the command is complete. The maximum number of CommandId and command status pairs that can be saved is set during configuration using typeOfAxis.decodingConfig.numberOfMaxBufferedCommandId.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId' whose command status is saved and thus remains available.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.27.2 _cancelAxisCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its 'CommandId' in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled.

Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.27.3 _getMotionStateOfAxisCommand

The function provides the motion status of an active axis command.

Return value: StructRetMotionCommandState

StructRetMotionCommandState
Return value with error code and motion status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionCommandId-State	Motion status	EnumMotionComman-dIdState	-

EnumMotionCommandIdState

NOT_EXISTENT (94)	'commandId' is not known or command is already completed
BUFFERED (207)	Command is in the command queue
IN_EXECUTION (208)	Command is being executed
IN_ACCELERATION (209)	Motion generated by the command is in the acceleration phase (only for TO axis)
IN_CONSTANT_MOTION (210)	Motion generated by the command is in the constant velocity phase (only for TO axis)
IN_DECELERATION (211)	Motion generated by the command is in the deceleration phase
AXIS_HOMED (212)	Axis is synchronized (only for TO axis)
INTERPOLATION_DONE (213)	Setpoint interpolation of command is complete
SYNCHRONIZING (214)	Synchronizing to synchronous operation (only for TO synchronous operation)
DESYNCHRONIZING (215)	Desynchronizing from synchronous operation (only for TO synchronous operation)
SYNCHRONIZED (216)	Synchronous operation is synchronized (only for TO synchronous operation)
MODIFICATION_ACTIVE (217)	Compensating motion for scaling or offset in synchronous operation is active (only for TO synchronous operation)
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionCommandIdState'
With NOT_EXISTENT, the 'commandId' for this axis instance is unknown or the command has already been completed.
With BUFFERED, the command is in the command queue.
With IN_EXECUTION, the command is being executed but no motion command is active.
With IN_ACCELERATION, motion generated by the command is in the acceleration phase. The value is output for _move and _pos commands.
With IN_CONSTANT_MOTION, motion generated by the command is in the constant velocity phase. The value is output for _move and _pos commands.
With IN_DECELERATION, motion generated by the command is in the deceleration phase. The value is output for _move and _pos commands.
With AXIS_HOMED, the axis has been synchronized. The value is output for homing commands.
With INTERPOLATION_DONE, the setpoint interpolation of the command is complete.

SYNCHRONIZING, DESYNCHRONIZING, SYNCHRONIZED, MODIFICATION_ACTIVE are not indicated on the axis.
With EXECUTED, the command has been successfully executed. The EXECUTED status can only be read if the 'commandId' is stored.
With ABORTED, execution of the command has been aborted. The ABORTED status can only be read if the 'commandId' is stored.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

'commandId' is used to specify the unique system-wide ID of the command for which the status is to be called.

1.3.27.4 _getStateOfAxisCommand

This function returns the execution state of a motion command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'commandIdState'
With ACTIVE, the command is being executed.
With NOT_EXISTENT, the 'commandId' is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for the synchronous start.
With EXECUTED, execution of the command is complete.
With ABORTED, execution of the command has been aborted.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.27.5 _getStateOfMotionBuffer

This function returns the status of the axis command queue.

Return value: StructRetMotionBuffer

StructRetMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumMotionBuffer-State	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':
EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':
Specifies the number of entries in the command queue.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.27.6 _removeBufferedAxisCommandId

This function terminates the saving of the commandId and corresponding command status after command execution. The status for the commandId is no longer available in the system after completion of the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies which commandId and corresponding command status should no longer be available in the system after completion of the command.
With SPECIFIC_ID, a specific commandId and its command status is no longer available once the command has been completed.
With ALL_ID, all commandIds and their command status are no longer available once the command has been completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId'.

1.3.27.7 _resetMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.28 Synchronous axis - Motion

1.3.28.1 _continue

This function continues the entire motion or a partial motion of the specified axis, if it was stopped with STOP and STOP_WITHOUT_ABORT in the 'stopMode' parameter.
Continuation of a partial motion is specified using the 'commandId' or the type of motion.
When continuing a motion, the dynamic parameters (e.g., velocity profile, acceleration) of the interrupted command are used.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

continueSpecification (optional)

Direction:	Input parameters
Data type:	EnumContinueSpecification
Parameter index:	2
System default:	ALL_AXIS_MOTION

EnumContinueSpecification

ALL_AXIS_MOTION (9)	Acts on all active motions
BASIC_MOTION (17)	Acts on the basic motion
SPECIFIC_AXIS_MOTION (125)	Acts on the specified motion
SUPERIMPOSED_MOTION (141)	Acts on the superimposed motions

Specifies the motion that is to be continued.
With ALL_AXIS_MOTION, all motions are continued.
With BASIC_MOTION, the basic motion is continued.
With SPECIFIC_AXIS_MOTION, the motion of the command specified in the 'continueId' parameter is continued.
With SUPERIMPOSED_MOTION, the superimposed motion is continued.

continueId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	WHEN_COMMAND_DONE

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.28.2 _disableAxis

This function cancels the axis enables.
The states of the current drive and power enables can be accessed using system variables 'actorMonitoring.driveState' (enable for DRIVE) and 'actorMonitoring.power' (enable for POWER).
The position controller enable can also be cancelled for position-controlled axes. Its current value can be queried via 'servoMonitoring.controlState'.
The position controller enable is ignored for speed-controlled axes.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

disableMode (optional)

Direction:	Input parameters
Data type:	EnumEnableAxisMode
Parameter index:	2
System default:	ALL

EnumEnableAxisMode

ALL (8)	Enable all
DO_NOT_CHANGE (43)	Do not change enables
DRIVE (44)	Enable drive only
POWER (109)	Enable power only
BY_STW_BIT (357)	profiDrive protocol bit mask
STATE_MACHINE_CONTROL_BY_APPLICATION (415)	Evaluation and manipulation of the STW bits by the user program

Specifies resetting of the drive and power enables.
ALL means that the drive and power enables are cancelled.
DO_NOT_CHANGE means that the current enables in effect are not changed.
DRIVE means that the drive enable is cancelled. The power enable is not changed.
POWER means that the power enable is cancelled. The drive enable is not changed.
With BY_STW_BIT, the bits specified in the 'STWBitSet' parameter are canceled in the PROFIdrive protocol.
With STATE_MACHINE_CONTROL_BY_APPLICATION, the user program takes responsibility for evaluating and manipulating the STW bits.

servoControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

servoCommandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	4
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the tracking mode setting.
If ACTIVE, the axis will be set to tracking mode and the setpoints are tracked. With ACTIVE, no motion commands can be executed; this applies to all axis types. Motion monitoring, such as following error, will not be initiated.
In the special mode of the active position controller enable and the active tracking operation, values can be specified in the servo using, for example, 'servoSettings.additionalCommandValue', without the monitoring being initiated and so the position controller can be placed in operation and set independent of the IPO.

If INACTIVE, the tracking operation of the axis will be deactivated. With INACTIVE, motion commands can be executed. For virtual axes, the position controller enable is always set internally and the motion control will be enabled only for inactive tracking operation.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	8
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the setting of the force/pressure controller enable.
This parameter is only relevant for axes with pressure or force control.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

STWBitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	9
System default:	0

Specifies the values of Bit 0 - Bit 6, which are set in the PROFIdrive protocol.
The bits are inverted and ANDed with the existing control word.
The specified bits are reset.

Meaning of the bits in the PROFIdrive protocol:
Bit0 - OFF
Bit1 - Coast stop (OFF2)
Bit2 - Quick stop (OFF3)
Bit3 - Disable operation
Bit4 - Disable ramp generator
Bit5 - Freeze ramp generator
Bit6 - Disable setpoint

The parameter is only used with 'disableMode:=BY_STW_BIT' or 'enableMode:=BY_STW_BIT'.
In all other cases, the parameter is irrelevant.

1.3.28.3 _enableAxis

This function switches the axis enables.
The states of the current drive and power enables can be accessed using system variables 'actorMonitoring.driveState' (enable for DRIVE) and 'actorMonitoring.power' (enable for POWER).
The position controller enable must also be set for position-controlled axes. Its current value can be queried via 'servoMonitoring.controlState'.
The position controller enable is ignored for speed-controlled axes.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

enableMode (optional)

Direction:	Input parameters
Data type:	EnumEnableAxisMode
Parameter index:	2
System default:	ALL

EnumEnableAxisMode

ALL (8)	Enable all
DO_NOT_CHANGE (43)	Do not change enables
DRIVE (44)	Enable drive only
POWER (109)	Enable power only
BY_STW_BIT (357)	profiDrive protocol bit mask
STATE_MACHINE_CONTROL_BY_APPLICATION (415)	Evaluation and manipulation of the STW bits by the user program

Specifies the setting of the drive and power enables.
ALL means that the drive and power enables are set.
DO_NOT_CHANGE means that the current valid enables are not changed.
The axis remains in follow-up mode if all enables are not available.
DRIVE means that the drive enable is set. The power enable is not changed.
POWER means that the power enable is set. The drive enable is not changed.
With BY_STW_BIT, the bits specified in the STWBitSet parameter are set in the PROFIdrive protocol.

servoControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

servoCommandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	4
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the tracking mode setting.
If ACTIVE, the axis will be set to tracking mode and the setpoints are tracked. With ACTIVE, no motion commands can be executed; this applies to all axis types. Motion monitoring, such as following error, will not be initiated.
In the special mode of the active position controller enable and the active tracking operation, values can be specified in the servo using, for example, 'servoSettings.additionalCommandValue', without the monitoring being initiated and so the position controller can be placed in operation and set independent of the IPO.

If INACTIVE, the tracking operation of the axis will be deactivated. With INACTIVE, motion commands can be executed. For virtual axes, the position controller enable is always set internally and the motion control will be enabled only for inactive tracking operation.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	8
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the setting of the force/pressure controller enable.
This parameter is only relevant for axes with pressure or force control.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

STWBitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	9
System default:	0

Specifies the values of Bit 0 - Bit 6, which are set in the PROFIdrive protocol.
The bits are ORed with the existing control word. Thus, the specified bits will be set.

The bits set can be viewed in the SIMOTION SCOUT project navigator under:
Drives-> <Drive name> -> Diagnostics -> Control/status words

Meaning of the bits in the PROFIdrive protocol:
Bit0 - ON
Bit1 - No coast stop (no OFF2)
Bit2 - No quick stop (no OFF3)
Bit3 - Enable operation
Bit4 - Enable ramp generator
Bit5 - Unfreeze ramp generator
Bit6 - Enable setpoint

The parameter is only used with 'enableMode:=BY_STW_BIT' or 'disableMode:=BY_STW_BIT'.
In all other cases, the parameter is irrelevant.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisEnableMovingMode
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumAxisEnableMovingMode

DO_NOT_CHANGE (43)	Retain last setting
POSITION_CONTROLLED (180)	Enable for speed- and position-controlled mode
SPEED_CONTROLLED (181)	Enable for speed-controlled mode

Specifies whether the axis is enabled for position- or speed-controlled motions.
With POSITION_CONTROLLED, the enable is for position- and speed-controlled motions.
With SPEED_CONTROLLED, the enable is for speed-controlled motions. In this mode, the axis can move if the encoder fails and there is no error response.
With DO_NOT_CHANGE, the last mode setting is retained.

1.3.28.4 _getAxisStoppingData

This function calculates the deceleration distance and the deceleration time when velocity, acceleration, and jerk are specified at the start of the braking operation.

Return value: StructRetGetAxisStoppingData

StructRetGetAxisStoppingData
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
stoppingDistance	Stopping distance	LREAL	-
stoppingTime	Stopping time	LREAL	-

Description of the return value:

0 -
No error

1 -
Illegal command parameter

2 -
Illegal range specification in command parameters

3 -
Command aborted

4 -
Unknown command

5 -
Command cannot be executed due to current object status

6 -
Command aborted due to termination of user task

7 -
Command rejected due to suspension of command interpretation of the addressed technology object

8 -
Command aborted due to full command buffer

9 -
Insufficient memory

10 -
A connection to a technology object required for this operation does not exist

11 -
No object configuration

12 -
The error cannot be reset due to its configuration

13 -
Axis is not homed

14 -
Measurement job not possible on virtual axis

15 -
Ambiguous 'commandId'

16 -
Command not implemented

17 -
Read access denied

18 -
Write access denied

19 -
Command argument not supported

20 -
The cam has already been interpolated and cannot be manipulated

21 -
The interpolation condition was violated

22 -
The programmed jerk is 0

23 -
The alarm to be deleted is not active

24 -
The command is not possible on a virtual axis

25 -
Synchronized start of this command is not possible

26 -
Higher-level command was aborted because it is not permitted by the active command

27 -
Timeout during communication with the drive

28 -
Actual values are not valid

29 -
This command cannot be executed when velocity control is active

30 -
This command cannot be executed when position control is active

31 -
This command cannot be executed in torque-reduced operation or during travel to fixed limit stop

32 -
This command can only be executed when force or pressure control is active

33 -
This command cannot be executed when force or pressure control is active

34 -
This command can only be executed when pressure limiting is active

35 -
Master values are not valid

36 -
Slave values are not valid

37 -
No slave value can be defined for a master value

38 -
No master value can be defined for a slave value

39 -
This command cannot be executed when synchronous operation is inactive

40 -
This command cannot be executed because of a synchronization error.

41 -
This command cannot be executed when gearing or camming is active

42 -
This command cannot be executed when camming is inactive

43 -
This command can only be used for an interpolated cam

44 -
This command cannot be executed when pressure limiting is active

45 -
There are not enough interpolation points for interpolation of the cam

46 -
Specified path location cannot be reached due to kinematic limitations

47 -
Path axis values are not valid

48 -
Maximum number of active commands exceeded

49 -
Command only possible on a technology object local to the CPU

10000 (greater or equal to) internal error

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

initialVelocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the velocity at the start of the braking operation.

initialAcceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the acceleration that occurs at the start of the braking operation. When a positive acceleration value is specified, acceleration occurs; when a negative acceleration value is specified, deceleration occurs. The parameter is only relevant with velocity profiles with jerk limiting 'velocityProfile:=SMOOTH'.

negativeAccel

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the deceleration setpoint.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the acceleration end jerk setpoint.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the deceleration start jerk setpoint.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the deceleration end jerk setpoint.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	8
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

1.3.28.5 _homing

Function for homing an axis.
The dynamic parameters for homing are programmable and refer to all phases of the homing procedure.

The execution time of the _homing command for the absolute encoder adjustment, i.e. for homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER and homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION, can take several interpolation cycles.
The status of the command can be queried by querying the CommandID of the homing command using _getstateOfAxisCommand(), even when the axis is already in the 'homed' state.

Exceptions
The homing approach velocity can be programmed in the command. The homing approach velocity, homing entry velocity and the homing deceleration velocity are set during configuration.
The execution sequence for active homing ('homingMode:=ACTIVE_HOMING') is set during configuration.
The criteria for passive homing ('homingMode:=PASSIVE_HOMING') are set during configuration independently from the active homing.
An axis has the 'referenced' or 'homed' status when the axis coordinate system has been aligned with the homing signal. The status can be read from the 'positioningState.homed' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

homingMode (optional)

Direction:	Input parameters
Data type:	EnumHomingMode
Parameter index:	2
System default:	ACTIVE_HOMING

EnumHomingMode

ACTIVE_HOMING (5)	Homing operation in configured mode
DIRECT_HOMING (41)	Setting of current position value
ENABLE_OFFSET_OF_ABSOLUTE_ENCODER (49)	Absolute encoder offset
PASSIVE_HOMING (104)	Homing with the next zero mark
DIRECT_HOMING_RELATIVE (352)	Relative setting of current position value
SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION (385)	Absolute encoder adjustment with position specification from command

Specifies the type of homing.

With ACTIVE_HOMING, a homing operation is started. The homing mode is set during configuration.

With DIRECT_HOMING, the home position coordinates are set directly to the current axis coordinates; the command does not initiate an axis motion.

With DIRECT_HOMING_RELATIVE, the home position coordinates are set relative to the current axis coordinates; the command does not initiate an axis motion.

With ENABLE_OFFSET_OF_ABSOLUTE_ENCODER, the value of 'absHomingEncoder.abshift' offset is added to the offset already existing in the system with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=RELATIVE;

with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=ABSOLUTE', the value 'absHomingEncoder.absshift' is calculated in as a fixed offset.

The total offset is saved in the NVRAM and is available after the controller is disabled. Once a new project has been loaded in the controller, the saved offset is no longer available.

With 'homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION', the value in the 'homePosition' parameter is set as the current position and the resulting absolute encoder offset is calculated from this. This value is displayed in system variable 'absoluteEncoder.totalOffsetValue', and stored in the system as a retain variable. The value in the configuration data 'absHomingEncoder'-absshift is not changed.

With PASSIVE_HOMING, the next zero mark crossed by the axis when it moves after the '_homing' command is used as the home position. The homing command is active parallel to the motion. It remains active until the homing procedure is complete. It can be cancelled with _resetAxis or _disableAxis. If the homing command is issued, e.g. with a stationary axis, and before the motion command, the homing command should be programmed with 'mergeMode:=IMMEDIATELY' and 'nextCommand:=IMMEDIATELY' and the motion command programmed with 'mergeMode:=IMMEDIATELY' or 'mergeMode:=NEXT'.

If the homing command is issued when a motion command is active, it should be programmed with 'mergeMode:=IMMEDIATELY'.

homePositionType (optional)

Direction:	Input parameters
Data type:	EnumValueSpecification
Parameter index:	3
System default:	USER_DEFAULT

EnumValueSpecification

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the type of home position coordinates.
With DIRECT, the value set in the 'homePosition' parameter is used as the programmed home position coordinate.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultHoming.homePosition' is used.

homePosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Home position coordinates, the evaluation depends on the 'homePositionType' parameter.

With 'homePositionType:=DIRECT', the parameter is used as the new home position coordinate. In all other cases, the parameter is irrelevant.

During the setting of the parameter 'homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER', the parameter has no meaning.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity input for the homing approach velocity.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultHoming.hominggApproachVelocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Homing approach velocity, evaluation dependent on the parameter 'velocityType'.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.
The homing approach velocity is only relevant for ACTIVE_HOMING homingMode CAM_AND_ZM or CAM.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandHoming
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandHoming

AT_MOTION_START (13)	Start of interpolation
IMMEDIATELY (60)	Asynchronous commands
WHEN_AXIS_HOMED (157)	Axis has been homed
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_AXIS_HOMED, the transition takes place when the axis has been synchronized.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or the command is aborted.
WHEN_INTERPOLATION_DONE and WHEN_MOTION_DONE are only effective when homing-Mode := ACTIVE_HOMING.
In all other cases, the command transition takes place in accordance with WHEN_AXIS_HOMED.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.28.6 _move

The axis is accelerated or decelerated to the programmed velocity setpoint.
If a limited constant velocity phase is not specified (parameter 'moveTimeOut-Type:=WITHOUT_TIME_LIMIT'), the velocity is kept constant after the transition to the velocity setpoint.
With limitation of the constant velocity phase (parameter 'moveTimeOutType:=WITH_TIME_LIMIT'), the velocity is decelerated to velocity setpoint 0 after the specified time has expired.

Return value: DINT

Description of the return value:

0 -
No error

1 -
Illegal command parameter

2 -
Illegal range specification in command parameters

3 -
Command aborted

4 -
Unknown command

5 -
Command cannot be executed due to current object status

6 -
Command aborted due to termination of user task

7 -
Command rejected due to suspension of command interpretation of the addressed technology object

8 -
Command aborted due to full command buffer

9 -
Insufficient memory

10 -
A connection to a technology object required for this operation does not exist

11 -
No object configuration

12 -
The error cannot be reset due to its configuration

13 -
Axis is not homed

14 -
Measurement job not possible on virtual axis

15 -
Ambiguous 'commandId'

16 -
Command not implemented

17 -
Read access denied

18 -
Write access denied

19 -
Command argument not supported

20 -
The cam has already been interpolated and cannot be manipulated

21 -
The interpolation condition was violated

22 -
The programmed jerk is 0

23 -
The alarm to be deleted is not active

24 -
The command is not possible on a virtual axis

25 -
Synchronized start of this command is not possible

26 -
Higher-level command was aborted because it is not permitted by the active command

27 -
Timeout during communication with the drive

28 -
Actual values are not valid

29 -
This command cannot be executed when velocity control is active

30 -
This command cannot be executed when position control is active

31 -
This command cannot be executed in torque-reduced operation or during travel to fixed limit stop

32 -
This command can only be executed when force or pressure control is active

33 -
This command cannot be executed when force or pressure control is active

34 -
This command can only be executed when pressure limiting is active

35 -
Master values are not valid

36 -
Slave values are not valid

37 -
No slave value can be defined for a master value

38 -
No master value can be defined for a slave value

39 -
This command cannot be executed when synchronous operation is inactive

40 -
This command cannot be executed because of a synchronization error.

41 -
This command cannot be executed when gearing or camming is active

42 -
This command cannot be executed when camming is inactive

43 -
This command can only be used for an interpolated cam

44 -
This command cannot be executed when pressure limiting is active

45 -
There are not enough interpolation points for interpolation of the cam

46 -
Specified path location cannot be reached due to kinematic limitations

47 -
Path axis values are not valid

48 -
Maximum number of active commands exceeded

49 -
Command only possible on a technology object local to the CPU

10000 (greater or equal to) internal error

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumDirection

BY_VALUE (24)	From the sign of the programmed position setpoint / velocity
EFFECTIVE (45)	Last programmed direction setpoint
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the direction of motion.
With BY_VALUE, the direction is derived from the sign of the velocity setpoint ('velocity' parameter).
With EFFECTIVE, the last programmed direction setpoint is used.
With POSITIVE and NEGATIVE, the direction depends on the axis coordinate system.
SHORTEST_WAY is not supported with this command. The command is aborted with alarm 30001 "Illegal command parameter".
With USER_DEFAULT, the default direction defined in system variable 'userDefaultDynamics.direction' is used.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	3
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
With RESULTING, the velocity is that which results when the current axis acceleration or deceleration is terminated with jerk control. The acceleration/deceleration is reduced to zero, taking into account the jerk. The mergeMode 'IMMEDIATELY' must be programmed as transition behavior.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

moveTimeOutType (optional)

Direction:	Input parameters
Data type:	EnumMoveTimeOut
Parameter index:	5
System default:	WITHOUT_TIME_LIMIT

EnumMoveTimeOut

WITHOUT_TIME_LIMIT (169)	Motion without time limit
WITH_TIME_LIMIT (171)	Motion with time limit

Defines the duration from the start of the constant velocity phase to the start of the deceleration phase.
WITHOUT_TIME_LIMIT means that the duration is not limited.
WITH_TIME_LIMIT means that the duration of the constant velocity phase is specified in the 'moveTimeOut' parameter.

moveTimeOut (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the duration of the constant velocity phase; the evaluation depends on the 'moveTimeOutType' parameter.
Only effective if parameter 'moveTimeOutType:=WITH_TIME_LIMIT'. The unit of the duration is specified during configuration of the axis.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.
The conditions AT_DECELERATION_START, WHEN_INTERPOLATION_DONE, and WHEN_MOTION_DONE only result in a transition when 'moveTimeOutType:=WITH_TIME_LIMIT' or if the command is aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	24
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.28.7 _pos

This function moves the axis to the programmed target position with a definable velocity profile.
The dynamic response parameters are programmable.
The response to the current motion is programmable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumDirection

BY_VALUE (24)	From the sign of the programmed position setpoint / velocity
EFFECTIVE (45)	Last programmed direction setpoint
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the direction of motion.
With an absolute position specification ('positioningMode:=ABSOLUTE'), the direction is directly derived from the target position to be approached for linear and rotary axes.
The following applies to modulo axes as well as to linear and rotary axes with relative positioning ('positioningMode:=RELATIVE'):

With BY_VALUE, the direction is determined from the sign of the position specification.
With EFFECTIVE, the last programmed direction setpoint is used.
With NEGATIVE, motion is in the negative direction relative to the axis coordinate system.
With POSITIVE, motion is in the positive direction relative to the axis coordinate system.

With SHORTEST_WAY, the shortest path is taken on modulo axes.
With SHORTEST_WAY and relative positioning ('positioningMode:=RELATIVE'), the direction is determined from the sign of the position specification for linear and rotary axes.
Further information can be found in the TO Axis, Electric/Hydraulic, External Encoder Function Manual, 'Possible direction specifications for the _pos() command' table

With USER_DEFAULT, the default direction defined in system variable 'userDefaultDynamics.direction' is used.

positioningMode (optional)

Direction:	Input parameters
Data type:	EnumPositioningMode
Parameter index:	3
System default:	ABSOLUTE

EnumPositioningMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified.

position

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the position, the evaluation depends on the 'positioningMode' parameter.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positi-veAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStart-Jerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerk-Type' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccel-StartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

blendingMode (optional)

Direction:	Input parameters
Data type:	EnumBlendingMode
Parameter index:	20
System default:	USER_DEFAULT

EnumBlendingMode

ACTIVE (4)	Blending
EFFECTIVE (45)	The last programmed blending value
INACTIVE (61)	No blending
USER_DEFAULT (149)	User default

Specifies blending mode.
With ACTIVE, the current motion is blended.
With EFFECTIVE, the last programmed blending specification is used.
With INACTIVE, no blending takes place.
With USER_DEFAULT, the current value programmed in the 'userDefaultPositioning.blendingMode' system variable is used.

Blending is a special form of linking the positioning motion specified in the command to the previous positioning motion, whereby the transition takes place at the target point of the current motion and the velocity setpoint specified in the motion commands for the particular motion is not violated at any time. This means:

- The current positioning motion is executed at the command velocity up to the target position.
- Exceptions: - If the velocity of the new motion command has the same sign but a smaller value than the velocity setpoint of the current motion, the axis decelerates to the new velocity before the current target position is reached. - In the event of a direction reversal, the previous motion is completed and the transition to the new motion takes place immediately at the target position. - If the path length of the new motion command is less than the required deceleration distance, the current motion is decelerated accordingly.
- If the velocity of the new command is greater, the velocity is increased after the transition to the new command, i.e. after the previous target position is reached. Active blending requires 'mergeMode' NEXT_MOTION or SEQUENTIAL and an adequate 'leading' program transition, i.e. the new motion to be activated with 'blending' must be known to the interpolator or motion control system at the start of deceleration in the previous command. If the blending command is not detected until later, blending is delayed accordingly.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	21
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	22
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	23
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.28.8 _runMotionInPositionLockedVelocityProfile

The axis moves according to a velocity profile that is specified in the cam function Velocity = f (position). Here, the position is an absolute axis position that is specified using the MotionIn interface.
The profile is started at the current MotionIn position.
The dynamic parameters for any required transition motions, e.g., to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the profile or the motion function for the axis motion.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	4
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	8
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	16
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	17
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	18
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	19
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	21
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.28.9 _runPositionBasedMotionIn

Activation of the motion interface on the axis with the position-based motion so that the values are taken into the setpoint generation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

reference (optional)

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2
System default:	0

Specifies the master object.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	3
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	17
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableMotionIn
Parameter index:	18
System default:	SEQUENTIAL

EnumMergeModeEnableMotionIn

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableMotionIn
Parameter index:	19
System default:	IMMEDIATELY

EnumNextCommandEnableMotionIn

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_COMMAND_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	20
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	22
System default:	TO_CONNECTION

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the master value is taken from the interface or as value.
With TO_CONNECTION, the MotionIn interface value is used.
With VALUE, the 'motionInType' parameter is evaluated.

motionInType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	23
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the master value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the master value is taken from system variable 'defaultMotionIn'.
With DIRECT, the master value is taken from command parameters ''positionIn', velocityIn' and 'accelerationIn'.
The parameter is only evaluated when valueReferenceType:=VALUE'.

positionIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	0.0

Specifies the position of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

velocityIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Specifies the velocity of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

accelerationIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	26
System default:	0.0

Specifies the acceleration of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

1.3.28.10 _runPositionLockedVelocityProfile

The axis moves according to a velocity profile that is specified in the cam function Velocity = f (position). The position in this case is an absolute axis position.
The profile is started at the current axis position.
The dynamic parameters for any required transition motions, e.g., to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the profile or the motion function for the axis motion.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	3
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	7
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	15
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	16
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	17
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	18
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	20
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.28.11 _runTimeLockedPositionProfile

The axis runs the predefined functions in the specified cam as a position profile.
The axis moves along the profile from a selectable start point to the end.
The dynamic parameters for any required transition motions, e.g., to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the profile or the motion function for the axis motion.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the position profile is run.

profileDataMode (optional)

Direction:	Input parameters
Data type:	EnumPositioningMode
Parameter index:	4
System default:	ABSOLUTE

EnumPositioningMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Type of position specification.
With ABSOLUTE, the cam positions are approached according to the absolute values.
With RELATIVE, the cam positions are approached relative to the current position.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	**Name**	**Data type**	**Unit**
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.28.12 _runTimeLockedVelocityProfile

The axis moves according to a velocity profile that is specified as a cam function.
The axis moves along the profile from a selectable start point to the end.
The dynamic parameters for any required transition motions, e.g. to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the profile or the motion function for the axis motion.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the velocity profile is run.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	4
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	8
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	16
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	17
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	18
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation. It is important to ensure that 'TypeOfAxis.DecodingConfig.behaviourAtTheEndOfProfile=MOVE_WITH_CONSTANT_SPEEd' is set as part of the configuration data, otherwise the function will not be able to transition.

With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	19
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	21
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.28.13 _runVelocityBasedMotionIn

This function activates the motion interface on the axis with the velocity-based motion so that the values can be taken over into the setpoint generation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference (optional)

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2
System default:	0

Specifies the master object.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	3
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	7
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	15
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableMotionIn
Parameter index:	16
System default:	SEQUENTIAL

EnumMergeModeEnableMotionIn

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableMotionIn
Parameter index:	17
System default:	IMMEDIATELY

EnumNextCommandEnableMotionIn

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_COMMAND_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	18
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	20
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	21
System default:	TO_CONNECTION

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the master value is taken from the interface or as value.
With TO_CONNECTION, the MotionIn interface value is used.
With VALUE, the 'motionInType' parameter is evaluated.

motionInType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	22
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the master value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the master value is taken from system variable 'defaultMotionIn'.
With DIRECT, the master value is taken from command parameters 'velocityIn' and 'accelerationIn'.
The parameter is only evaluated when valueReferenceType:=VALUE'.

velocityIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	0.0

Specifies the velocity of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

accelerationIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	0.0

Specifies the acceleration of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

1.3.28.14 _setAxisSTW

The function makes it possible to specify selectable bits in control word 1 and control word 2 of the PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

STW1BitMask (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	2
System default:	0

Selection of the bits to be specified in control word 1 with this command.
The bits to be specified are selected in the form of a bit mask in 'STW1BitSet':
With 1, 'STW1.Bitx' is specified with this command.
With 0, 'STW1.Bitx' is not specified with this command.
The values of the bits are specified in 'STW1BitSet'.

STW1BitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	3
System default:	0

Specifies the values with the bits selected in parameter 'STW1BitMask'.

STW2BitMask (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	4
System default:	0

Selection of the bits to be changed in control word 2.
The selection is performed as a bit mask. The value (0 or 1) of the bit to be changed is specified in parameter 'STW2BitSet'.

STW2BitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	5
System default:	0

Specifies the values with the bits selected in parameter 'STW2BitMask'.

execution (optional)

Direction:	Input parameters
Data type:	EnumToCommandExecution
Parameter index:	6
System default:	TASK_EXECUTION

EnumToCommandExecution

TASK_EXECUTION (413)	Setting the STW bits in the task context of the user program
TO_EXECUTION (414)	Setting the STW bits for synchronizing with TO command processing

Task context for setting the bits in the drive protocol.
With TASK_EXECUTION, the bits are set in the task context of the user program.
With TO_EXECUTION, setting of the bits is synchronized with command processing in the technology object.

1.3.28.15 _stop

This command stops the entire motion or a partial motion of the specified axis using a programmed deceleration ramp. The _stop command affects only single-axis motions and cannot be used for synchronous operation motions.
The motion to be stopped can be interrupted or terminated.
Stopping of a partial motion is specified using either the 'commandId' or the type of motion.
An interrupted motion can be continued with the _continue command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

stopMode (optional)

Direction:	Input parameters
Data type:	EnumStopMode
Parameter index:	2
System default:	STOP_WITHOUT_ABORT

EnumStopMode

STOP_AND_ABORT (135)	Normal stop with abort
STOP_WITHOUT_ABORT (138)	Normal stop without abort

Specifies the stop behavior relative to the current command.
With STOP_AND_ABORT, the active motion command is aborted.
With STOP_WITHOUT_ABORT, the current motion is interrupted and can be continued with the _continue command.

stopSpecification (optional)

Direction:	Input parameters
Data type:	EnumStopSpecification
Parameter index:	3
System default:	ALL_AXIS_MOTION

EnumStopSpecification

ALL_AXIS_MOTION (9)	Acts on all active axis motions
BASIC_MOTION (17)	Acts on the basic axis motion
SPECIFIC_AXIS_MOTION (125)	Acts on a specific axis motion
SUPERIMPOSED_MOTION (141)	Acts on the superimposed axis motions

Specifies the motions for which the stop command is to take effect.
With ALL_AXIS_MOTION, the stop command acts on all active axis motions.
With BASIC_MOTION, only the main motion is stopped.
With SPECIFIC_AXIS_MOTION, the command acts on the command specified in the 'StopId' parameter.
With SUPERIMPOSED_MOTION, the superimposed motion is stopped.

stopId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide ID to stop the command with this 'commandId'. Commands with the standard ID 'stopId.SystemId_low:=0' and 'stopId.SystemId_high:=0' are not stopped.
This parameter is only active with 'stopSpecification:=SPECIFIC_AXIS_MOTION'.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	8
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	18
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeStop
Parameter index:	19
System default:	IMMEDIATELY

EnumMergeModeStop

AT_END_OF_COMMAND (2)	No significance
IMMEDIATELY (60)	Execute command immediately

Specifies when the command takes effect relative to the motion.
AT_END_OF_COMMAND is irrelevant.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	20
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	21
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingModeStopCommand
Parameter index:	23
System default:	CURRENT_MODE

EnumMovingModeStopCommand

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion
CURRENT_MODE (327)	Last set traversing mode

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
CURRENT_MODE means that motion is stopped in the last set traversing mode.
The 'movingMode' parameter has no effect on the speed-controlled axis.

1.3.28.16 _stopEmergency

This function stops the axis with a programmable stop mode.
If a motion command is active, it is aborted and cannot be continued with a _continue command. The axis is not switched to follow-up mode.
The axis is blocked for further motion commands; the status can be revoked with _reset Axis () or _disable Axis ().

Detailed information on this function can be found in the SIMOTION Motion Control, TO Axis, Electric/Hydraulic, External Encoder Function Manual.

Note
The _stopEmergency system function has the same behavior as the _MC_Stop function block.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

stopDriveMode (optional)

Direction:	Input parameters
Data type:	EnumStopDriveMode
Parameter index:	2
System default:	STOP_WITH_MAXIMAL_DECELERATION

EnumStopDriveMode

STOP_IN_DEFINED_TIME (136)	Stop axis using a time-dependent ramp
STOP_WITH_COMMAND_VALUE_ZERO (139)	Stop axis using zero setpoint
STOP_WITH_MAXIMAL_DECELERATION (140)	Stop axis using maximum deceleration
STOP_WITH_DYNAMIC_PARAMETER (326)	Stop axis using programmed dynamic values

Specifies the stop mode of the axis.
With STOP_IN_DEFINED_TIME the axis stops according to the time specified in the 'stopTime' parameter. The specified time is adhered to irrespective of the velocity.
With STOP_WITH_COMMAND_VALUE_ZERO, the axis is stopped using the emergency stop ramp in the controller. This is set during configuration. When the emergency stop ramp is generated (stopDriveMode:=STOP_WITH_COMMAND_VALUE_ZERO), the ramp generator in the servo is based on the 'dynamicData.positionTimeConstant' x velocity setpoint extrapolated actual position.
With STOP_WITH_MAXIMAL_DECELERATION, the axis is stopped via interpolation using the maximum dynamic axis values.
With STOP_WITH_DYNAMIC_PARAMETER, the axis is stopped with the dynamic parameters programmed in the command.

stopTimeType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	3
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of definition for stop time.
This parameter is only effective if 'stopDriveMode:=STOP_IN_DEFINED_TIME'.
With DIRECT, the value set in the 'stopTime' parameter is used as the programmed stop time.
With EFFECTIVE, the last programmed stop time is used.
With USER_DEFAULT, the default setting for the stop time specified in system variable 'userDefault-Dynamics.stopTime' is used.

stopTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the stop time; the evaluation depends on the 'stopDriveMode' parameter.
It is only used when parameter 'stopDriveMode:=STOP_IN_DEFINED_TIME' and 'stopTime-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeStop
Parameter index:	5
System default:	IMMEDIATELY

EnumMergeModeStop

AT_END_OF_COMMAND (2)	No significance
IMMEDIATELY (60)	Execute command immediately

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
AT_END_OF_COMMAND is irrelevant.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingModeStopCommand
Parameter index:	9
System default:	CURRENT_MODE

EnumMovingModeStopCommand

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion
CURRENT_MODE (327)	Last set traversing mode

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
CURRENT_MODE means that motion is stopped in the last set traversing mode.
The movingMode parameter has no effect on the speed-controlled axis.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	10
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	12
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	18
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	20
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	22
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

1.3.29 Synchronous axis - Pressure limitation

1.3.29.1 _disableForceLimiting

This command deactivates limitation of the force or pressure.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableForceLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableForceLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.29.2 _enableForceLimitingByCondition

This command causes switchover to force/pressure limiting when the switchover criterion defined in the command is reached.
The switchover criterion is checked in the servo. Switch-over criteria can include a position, force or pressure, a time, or a digital input.
The mode for determining the cam execution method is set via the parameter 'forceProfileMode'. The following possibilities are available:
- User default
- Force-time cam profile
- Force-position cam profile
- Value input by command parameter

The rise factor for transitions when entering and exiting the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

forceLimitingProfile (optional)

Direction:	Input parameters
Data type:	CamType
Parameter index:	2
System default:	0

Specifies the cam that defines the force/pressure profile for force/pressure limiting of the axis.

switchingPositionType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of position specification.

switchingPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position that is compared to the actual position of the axis.
This comparison is carried out depending on the 'switchingPositionType' parameter.
It is only used when 'switchingPositionType := DIRECT'.
In all other cases, the parameter is irrelevant.

positionCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	5
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

switchingForceType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	6
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of comparison pressure specification.

switchingForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the comparison pressure; the evaluation depends on the 'switchingForceType' parameter.
It is only used when 'switchingForceType := DIRECT'.
In all other cases, the parameter is irrelevant.

forceCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	8
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

forceSensorNumberType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	9
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specification type of the force/pressure measurement system to be used for the comparison force/pressure.

forceSensorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	10
System default:	1

Specifies the force/pressure measurement system to be used for the comparison pressure.
The encoder numbers of the force/pressure measurement system are assigned during configuration.
This command parameter refers to encoders configured under TypeOfAxis.NumberOfAdditionalSensors.AdditionalSensor_<forceSensorNumber>.
The encoder number transferred in the command parameter must be used for <forceSensorNumber>.
The number of encoders that can be used is set in the NumberOfAdditionalSensors configuration data.

switchingTimeType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	11
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of time data.

switchingTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the time; the evaluation depends on the 'switchingTimeType' parameter.
It is only used when 'switchingTimeType := DIRECT'.
In all other cases, the parameter is irrelevant.

switchingInputType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	13
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of digital input specification.

switchingInput (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	14
System default:	1

Specifies the digital input to be used for the comparison.
The number of the input is a reference to a list of inputs. This list can be found in the configuration data under TypeOfAxis.ControllerSwitchData. The size of the list is set by means of the NumberOfDigitalInputs configuration data.

switchingInputMode (optional)

Direction:	Input parameters
Data type:	EnumAxisUserDefaultHighLow
Parameter index:	15
System default:	USER_DEFAULT

EnumAxisUserDefaultHighLow

USER_DEFAULT (149)	User default
HIGH (264)	HIGH signal
LOW (265)	LOW signal

Specifies the expected switching status of the digital input.

switchingCondition (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition
Parameter index:	16
System default:	USER_DEFAULT

EnumAxisSwitchingCondition

USER_DEFAULT (149)	User default
CONDITION_1 (242)	Condition 1
CONDITION_2 (243)	Condition 2
CONDITION_1_OR_CONDITION_2 (244)	Condition 1 OR Condition 2
CONDITION_1_AND_CONDITION_2 (245)	Condition 1 AND Condition 2

Specifies the logical configuration of switching conditions 1 and 2.

switchingCondition_1 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_1
Parameter index:	17
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_1

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 1.

switchingCondition_2 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_2
Parameter index:	18
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_2

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 2.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	19
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	0.0

Specifies the rise factor for transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceControl
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandEnableForceControl

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_CONTROL_ENABLED (262)	When the switch-over condition is fulfilled
WHEN_PROFILE_DONE (263)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

forceExtrapolationType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceExtrapolationType
Parameter index:	24
System default:	NONE

EnumAxisForceExtrapolationType

LINEAR (72)	Linear extrapolation
NONE (93)	No extrapolation
POLYNOMIAL (368)	Cubic extrapolation

Type of extrapolation.
With NONE, there is no extrapolation.
With LINEAR, a linear extrapolation takes place between startForce and endForce. The values of the derivatives are not taken into account.
With POLYNOMIAL, a cubic extrapolation takes place between startForce and endForce, taking into account the specified derivatives.

startForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Starting point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

endForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	26
System default:	0.0

End point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

startForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	27
System default:	0.0

Derivative in the starting point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

endForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	28
System default:	0.0

Derivative in the end point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

functionLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	29
System default:	0.0

Specifies the length of the extrapolation curve as time.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

forceProfileMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceProfileModeConditionCommand
Parameter index:	30
System default:	USER_DEFAULT

EnumAxisForceProfileModeConditionCommand

USER_DEFAULT (149)	User default
FORCE_TIME (394)	Force-time cam profile
FORCE_POSITION (395)	Force-position cam profile
FORCE_VALUE (396)	Value input by command parameter

Type of setpoint specification for the force/pressure limiting of the axis.
With FORCE_TIME, the cam profile specified in the 'forceLimitingProfile' parameter is interpreted as a force-time profile.
With FORCE_POSITION, the cam profile specified in the 'forceLimitingProfile' parameter is interpreted as a force-distance profile.
With FORCE_VALUE, the setpoint is specified with the 'forceValueType' and 'forceValue' parameters.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceProfileModeConditionCommand'.

forceValueType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceValueConditionCommand
Parameter index:	31
System default:	USER_DEFAULT

EnumAxisForceValueConditionCommand

ACTUAL_VALUE (34)	Actual value of axis
DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of setpoint specification for the force/pressure limiting of the axis when 'forceProfileMode:=FORCE_VALUE'.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceValueConditionCommand'.
With ACTUAL, the current actual value of the axis is accepted.
With DIRECT, the setpoint is taken from the 'forceValue' parameter.

forceValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	32
System default:	0.0

Type of setpoint specification for the force/pressure limiting of the axis when 'forceValueType:=DIRECT'.

1.3.29.3 _enableForceLimitingValue

This command activates limiting of the force or pressure to a fixed value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

limitingValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the force/pressure limiting value.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	6
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.29.4 _enableMotionInPositionLockedForceLimitingProfile

This function activates the position-related force/pressure limitation profile at the interconnected MotionIn position.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	7
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.29.5 _enablePositionLockedForceLimitingProfile

This function activates force or pressure limiting with a position-related force/pressure limiting profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	6
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.29.6 _enableTimeLockedForceLimitingProfile

This function activates the force or pressure limiting with a time-related force/pressure limiting profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam at which the profile definition starts.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	7
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.29.7 _getForceControlDataSetParameter

This function reads the force or pressure-related data in the axis data set.

Return value: StructRetGetForceControlDataSetParameter

StructRetGetForceControlDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisForceControlDataSet	-

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceControllerTypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.29.8 _setForceControlDataSetParameter

This function overwrites the force or pressure-related data in the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisForceControlDataSet
Parameter index:	3

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceController-TypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Specifies the data to be used to overwrite the data set.
UNI_DIRECTION means that negative actual values cannot be measured.
BOTH_DIRECTION means that negative actual values can be measured.

1.3.30 Synchronous axis - Pressure control

1.3.30.1 _disableVelocityLimiting

This command deactivates limitation of the velocity.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.30.2 _enableForceControlByCondition

This function causes switchover to force/pressure control when the switchover criterion defined in the command is reached.

The switch-over criterion check begins with a query in the servo. Switch-over criteria can include a position, force or pressure, a time, or a digital input.

The mode for determining the cam execution method is set via the parameter 'forceProfileMode'.

The following possibilities are available:

- User default
- Force-time cam profile
- Force-position cam profile
- Value input by command parameter

The rise factor for force setpoint transitions for entering and exiting the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

forceProfile (optional)

Direction:	Input parameters
Data type:	CamType
Parameter index:	2
System default:	0

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

switchingPositionType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of position specification.

switchingPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position that is compared to the actual position of the axis.
This comparison is carried out depending on the 'switchingPositionType' parameter.
It is only used when 'switchingPositionType := DIRECT'.
In all other cases, the parameter is irrelevant.

positionCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	5
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

switchingForceType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	6
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of comparison pressure specification.

switchingForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the comparison pressure; the evaluation depends on the 'switchingForceType' parameter.
It is only used when 'switchingForceType := DIRECT'.
In all other cases, the parameter is irrelevant.

forceCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	8
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

forceSensorNumberType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	9
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specification type of the force/pressure measurement system to be used for the comparison force/pressure.

forceSensorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	10
System default:	1

Specifies the force/pressure measurement system to be used for the comparison pressure.
The encoder numbers of the force/pressure measurement system are assigned during configuration.
This command parameter refers to encoders configured under TypeOfAxis.NumberOfAdditionalSensors.AdditionalSensor_<forceSensorNumber>.
The encoder number transferred in the command parameter must be used for <forceSensorNumber>.
The number of encoders that can be used is set in the NumberOfAdditionalSensors configuration data.

switchingTimeType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	11
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of time data.

switchingTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the time; the evaluation depends on the 'switchingTimeType' parameter.
It is only used when 'switchingTimeType := DIRECT'.
In all other cases, the parameter is irrelevant.

switchingInputType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	13
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of digital input specification.

switchingInput (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	14
System default:	1

Specifies the digital input to be used for the comparison.
The number of the input is a reference to a list of inputs. This list can be found in the configuration data under TypeOfAxis.ControllerSwitchData. The size of the list is set by means of the NumberOfDigitalInputs configuration data.

switchingInputMode (optional)

Direction:	Input parameters
Data type:	EnumAxisUserDefaultHighLow
Parameter index:	15
System default:	USER_DEFAULT

EnumAxisUserDefaultHighLow

USER_DEFAULT (149)	User default
HIGH (264)	HIGH signal
LOW (265)	LOW signal

Specifies the expected switching status of the digital input.

switchingCondition (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition
Parameter index:	16
System default:	USER_DEFAULT

EnumAxisSwitchingCondition

USER_DEFAULT (149)	User default
CONDITION_1 (242)	Condition 1
CONDITION_2 (243)	Condition 2
CONDITION_1_OR_CONDITION_2 (244)	Condition 1 OR Condition 2
CONDITION_1_AND_CONDITION_2 (245)	Condition 1 AND Condition 2

Specifies the logical configuration of switching conditions 1 and 2.

switchingCondition_1 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_1
Parameter index:	17
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_1

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 1.

switchingCondition_2 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_2
Parameter index:	18
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_2

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 2.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	19
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	0.0

Specifies the rise factor for force/pressure setpoint value transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceControl
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandEnableForceControl

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_CONTROL_ENABLED (262)	When the switch-over condition is fulfilled
WHEN_PROFILE_DONE (263)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	24
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Activates a velocity limitation when a transition condition is active.

velocityLimitingValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	25
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of velocity limiting value specification.

velocityLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	26
System default:	100.0

Specifies the velocity limiting value. Evaluation depends on the 'velocityLimitingValueType' parameter.

forceExtrapolationType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceExtrapolationType
Parameter index:	27
System default:	NONE

EnumAxisForceExtrapolationType

LINEAR (72)	Linear extrapolation
NONE (93)	No extrapolation
POLYNOMIAL (368)	Cubic extrapolation

Type of extrapolation.
With NONE, there is no extrapolation.
With LINEAR, a linear extrapolation takes place between startForce and endForce. The values of the derivatives are not taken into account.
With POLYNOMIAL, a cubic extrapolation takes place between startForce and endForce, taking into account the specified derivatives.

startForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	28
System default:	0.0

Starting point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

endForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	29
System default:	0.0

End point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

startForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	30
System default:	0.0

Derivative in the starting point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

endForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	31
System default:	0.0

Derivative in the end point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

functionLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	32
System default:	0.0

Specifies the length of the extrapolation curve as time.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

velocityLimitingValueMode (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingValueMode
Parameter index:	33
System default:	SET_VALUE

EnumAxisVelocityLimitingValueMode

ACTUAL_VALUE (34)	Velocity limiting to current actual value
SET_VALUE (386)	Velocity limiting to programmed value
SETPOINT_VALUE (938)	Velocity limiting to current setpoint

Specifies whether the velocity at the reversing point should be accepted as the velocity limiting value.
With ACTUAL_VALUE, the velocity is limited to the current actual velocity value.
With SET_VALUE, the velocity is limited to the value specified in the parameters 'velocityLimitingValueType' and 'velocityLimitingValue'.
With SETPOINT_VALUE, the velocity is limited to the current velocity setpoint.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	34
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

forceProfileMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceProfileModeConditionCommand
Parameter index:	35
System default:	USER_DEFAULT

EnumAxisForceProfileModeConditionCommand

USER_DEFAULT (149)	User default
FORCE_TIME (394)	Force-time cam profile
FORCE_POSITION (395)	Force-position cam profile
FORCE_VALUE (396)	Value input by command parameter

Type of setpoint specification for the force/pressure control of the axis.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceProfileModeConditionCommand'.
With FORCE_TIME, the cam profile specified in the 'forceProfile' parameter is interpreted as a force-time profile.
With FORCE_POSITION, the cam profile specified in the 'forceProfile' parameter is interpreted as a force-distance profile.
With FORCE_VALUE, the setpoint is specified with the 'forceValueType' and 'forceValue' parameters.

forceValueType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceValueConditionCommand
Parameter index:	36
System default:	USER_DEFAULT

EnumAxisForceValueConditionCommand

ACTUAL_VALUE (34)	Actual value of axis
DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of setpoint specification for the force/pressure control of the axis when 'forceProfileMode:=FORCE_VALUE'.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceValueConditionCommand'.
With ACTUAL, the current actual value of the axis is accepted.
With DIRECT, the setpoint is taken from the 'forceValue' parameter.

forceValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	37
System default:	0.0

Type of setpoint specification for the force/pressure control of the axis when 'forceValueType:=DIRECT'.

1.3.30.3 _enableMotionInPositionLockedVelocityLimitingProfile

This function activates the velocity limiting with a position-related MotionIn limitation profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the velocity profile for axis limiting.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	7
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.30.4 _enablePositionLockedVelocityLimitingProfile

This function activates the velocity limiting with a position-related limitation profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the velocity profile for axis limiting.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	6
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.30.5 _enableTimeLockedVelocityLimitingProfile

This function activates the velocity limiting with a time-related limitation profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the velocity profile for axis limiting.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam at which the profile definition starts.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	7
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.30.6 _enableVelocityLimitingValue

This function activates velocity limiting at a definable value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

limitingValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the velocity limiting value.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingValueMode (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingValueMode
Parameter index:	6
System default:	SET_VALUE

EnumAxisVelocityLimitingValueMode

ACTUAL_VALUE (34)	Velocity limiting to current actual value
SET_VALUE (386)	Velocity limiting to programmed value
SETPOINT_VALUE (938)	Velocity limiting to current setpoint

Specifies whether the velocity at the reversing point should be accepted as the velocity limiting value.
With ACTUAL_VALUE, the velocity is limited to the current actual velocity value.
With SET_VALUE, the velocity is limited to value specified in the 'limitingValue' parameter.
With SETPOINT_VALUE, the velocity is limited to the current velocity setpoint.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	7
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.30.7 _getForceControlDataSetParameter

This function reads the force or pressure-related data in the axis data set.

Return value: StructRetGetForceControlDataSetParameter

StructRetGetForceControlDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisForceControlDataSet	-

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceControllerTypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.30.8 _runMotionInPositionLockedForceProfile

The axis executes the predefined function in the cam as a force-path profile. Here, the position is an absolute axis position that is specified using the MotionIn interface.
The profile is started at the current MotionIn position.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the force/pressure profile.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	4
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the rise factor for force/pressure setpoint value transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	6
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.30.9 _runPositionLockedForceProfile

The axis executes the predefined functions in the cam as a force-time profile. The position in this case is an absolute axis position.
The profile is started at the current axis position.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise factor for force/pressure setpoint value transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	5
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.30.10 _runTimeLockedForceProfile

The axis executes the predefined functions in the specified cam as a force/pressure profile.
The profile is executed from a selectable start point to the end.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the force/pressure profile is run.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	4
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the rise factor for force/pressure setpoint transitions. Evaluation depends on the 'derived-CommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	6
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.30.11 _setForceCommandValue

The force or pressure setpoint specified in the command is set on the axis.
The rise factor for force or pressure setpoint transitions for any required transition motions can be programmed in the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the force or pressure setpoint.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise factor for force/pressure setpoint transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	5
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceValue
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandForceValue

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the actual force/pressure value is within the configured force/pressure window.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.30.12 _setForceControlDataSetParameter

This function overwrites the force or pressure-related data in the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisForceControlDataSet
Parameter index:	3

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceController-TypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Specifies the data to be used to overwrite the data set.
UNI_DIRECTION means that negative actual values cannot be measured.
BOTH_DIRECTION means that negative actual values can be measured.

1.3.31 Synchronous axis - Hydraulics

1.3.31.1 _disableQFAxis

This function activates the enables and grants the axis access to the valve.
The function returns an error if another axis is already accessing the valve.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

controlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	2
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

commandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the position controller follow-up mode setting.
ACTIVE switches the axis to follow-up mode.
With ACTIVE, no motion commands can be executed; this applies to all axis types.
INACTIVE switches the axis out of follow-up mode.
With INACTIVE, motion commands can be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	7
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the pressure controller enable setting.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

QOutput (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputMode
Parameter index:	8
System default:	DISABLE

EnumDisableQFAxisOutputMode

DISABLE (42)	Enable valve access
DO_NOT_CHANGE (43)	Do not change access

Specifies the reserving and enabling of access to the Q valve.
With DISABLE, access to the Q valve is enabled.
With DO_NOT_CHANGE, no change takes place.

QOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputEnableMode
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
RESET (304)	Cancel enable

Specifies the Q valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With RESET, the enable is canceled.

QOutputValueSetMode (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputSetMode
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumQFAxisOutputSetMode

DO_NOT_CHANGE (43)	Do not change replacement value
SET (178)	Set replacement value

Specifies the setting of a replacement value for the Q valve if a value is not output by an axis.
With DO_NOT_CHANGE, no change takes place.
With SET, the replacement value is set.

QOutputValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the replacement value; the evaluation depends on the 'QOutputValueSetMode' parameter.
It is only used when parameter 'QOutputValueSetMode:=SET'.
In all other cases, the parameter is irrelevant.

QOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	12
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'QOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

QOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the rise limiting value.

FOutput (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputMode
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputMode

DISABLE (42)	Enable valve access
DO_NOT_CHANGE (43)	Do not change access

Specifies the reserving and enabling of access to the pressure limiting valve.
With DO_NOT_CHANGE, no change takes place.
With DISABLE, access to the pressure limiting valve is enabled.

FOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputEnableMode
Parameter index:	15
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
RESET (304)	Cancel enable

Specifies the pressure limiting valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With RESET, the enable is canceled.

FOutputValueSetMode (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputSetMode
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumQFAxisOutputSetMode

DO_NOT_CHANGE (43)	Do not change replacement value
SET (178)	Set replacement value

Specifies the setting of a replacement value for the pressure limiting valve if a value is not output by an axis.
With DO_NOT_CHANGE, no change takes place.
With SET, the replacement value is set.

FOutputValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the replacement value; the evaluation depends on the 'FOutputValueSetMode' parameter.
It is only used when parameter 'FOutputValueSetMode:=SET'.
In all other cases, the parameter is irrelevant.

FOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	18
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'FOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

FOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the rise limiting value.

1.3.31.2 _enableQFAxis

This function activates the enables and grants the axis access to the valve.
The function returns an error if another axis is already accessing the valve.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

controlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	2
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

commandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the position controller follow-up mode setting.
ACTIVE switches the axis to follow-up mode.
With ACTIVE, no motion commands can be executed; this applies to all axis types.
INACTIVE switches the axis out of follow-up mode.
With INACTIVE, motion commands can be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	7
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the pressure controller enable setting.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

QOutput (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputMode
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumQFAxisOutputMode

DO_NOT_CHANGE (43)	Do not change access
ENABLE (48)	Request valve access

Specifies the reserving and enabling of access to the Q valve.
With DISABLE, access to the Q valve is enabled.
With DO_NOT_CHANGE, no change takes place.
With ENABLE, access to the Q valve is reserved.

QOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputEnableMode
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
SET (178)	Set enable

Specifies the Q valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With SET, the enable is set.

QOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	10
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'QOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

QOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the rise limiting value.

FOutput (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputMode
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumQFAxisOutputMode

DO_NOT_CHANGE (43)	Do not change access
ENABLE (48)	Request valve access

Parameter for reserving and enabling access to the pressure limitation valve.
With DO_NOT_CHANGE, no change takes place.
With ENABLE, access to the pressure limitation valve is reserved.

FOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputEnableMode
Parameter index:	13
System default:	DO_NOT_CHANGE

EnumQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
SET (178)	Set enable

Specifies the pressure limiting valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With SET, the enable is set.

FOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	14
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'FOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

FOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the rise limiting value.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisEnableMovingMode
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumAxisEnableMovingMode

DO_NOT_CHANGE (43)	Retain last setting
POSITION_CONTROLLED (180)	Enable for speed- and position-controlled mode
SPEED_CONTROLLED (181)	Enable for speed-controlled mode

Specifies whether the axis is enabled for position- or speed-controlled motions.
With POSITION_CONTROLLED, the enable is for position- and speed-controlled motions.
With SPEED_CONTROLLED, the enable is for speed-controlled motions. In this mode, the axis can move if the encoder fails and there is no error response.
With DO_NOT_CHANGE, the last mode setting is retained.

1.3.31.3 _getQFAxisDataSetParameter

This function reads an axis data set.

Return value: StructRetGetQFAxisDataSetParameter

StructRetGetQFAxisDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructQFAxisDataSet	-

StructQFAxisDataSet
Parameters of an axis data set

Structure	Name	Data type	Unit
dynamicQFData	Dynamic parameters of the control loop	StructAxisDynamicQFData	-
invertQ	Inversion of the Q output value	StructAxisInvertQOutput	-
invertSetPoint	Inversion of the Q output value before the characteristic curve	StructAxisInvertSetPointHydraulicType	-

StructAxisDynamicQFData
Dynamic parameters of the control loop

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant for velocity loop	LREAL	s
QOutputTimeConstant	Equivalent time constant for hydraulic controlled system	LREAL	s

StructAxisInvertQOutput
Inversion of the Q output value

Structure	Name	Data type	Unit
invSetPoint	Activation status	EnumYesNo	-

StructAxisInvertSetPointHydraulicType
Inversion of the Q output value before the characteristic curve

Structure	Name	Data type	Unit
invert	Activation status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.31.4 _setQFAxisDataSetParameter

This function overwrites an axis data set.

Note
The change of the configuration data using the _setQFAxisDataSetParameter command is not displayed in the online view of the expert list.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructQFAxisDataSet
Parameter index:	3

StructQFAxisDataSet
Parameters of an axis data set

Structure	Name	Data type	Unit
dynamicQFData	Dynamic parameters of the control loop	StructAxisDynamicQFData	-
invertQ	Inversion of the Q output value	StructAxisInvertQOutput	-
invertSetPoint	Inversion of the Q output value before the characteristic curve	StructAxisInvertSetPointHydraulicType	-

StructAxisDynamicQFData
Dynamic parameters of the control loop

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant for velocity loop	LREAL	s
QOutputTimeConstant	Equivalent time constant for hydraulic controlled system	LREAL	s

StructAxisInvertQOutput
Inversion of the Q output value

Structure	Name	Data type	Unit
invSetPoint	Activation status	EnumYesNo	-

StructAxisInvertSetPointHydraulicType
Inversion of the Q output value before the characteristic curve

Structure	Name	Data type	Unit
invert	Activation status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the data to be used to overwrite the data set.

1.3.31.5 _setQFAxisFCharacteristics

This system function activates the characteristic curve for a pressure limiting valve. The characteristic curve is defined in a cam that describes the actual pressure value as a function of the valve position. The valve position is specified as a percentage. The value specified in 'maxOutputVoltage' is equivalent to 100 percent.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

characteristics

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the valve characteristic curve.

maxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	3
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value set in the 'maxDerivative' parameter is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

maxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise limiting value; the evaluation depends on the 'maxDerivativeType' parameter.

1.3.31.6 _setQFAxisQCharacteristics

Activates the characteristic curve for a Q valve. The characteristic curve is defined in a cam that describes the axis velocity as a function of the valve position.
The valve position is specified as a percentage. The value specified in 'maxOutputVoltage' is equivalent to 100 percent.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

characteristics

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the valve characteristic curve.

maxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	3
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value set in the 'maxDerivative' parameter is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

maxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise limiting value; the evaluation depends on the 'maxDerivativeType' parameter.

1.3.32 Synchronous axis - Compensation and superimposition

1.3.32.1 _disableAxisAdditiveTorque

This function deactivates an additive torque setpoint that was activated with '_enableAxisAdditiveTorque'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.2 _disableAxisTorqueLimitNegative

This function deactivates a negative torque limiting that was activated with '_enableAxisTorqueLimitNegativeIn'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.3 _disableAxisTorqueLimitPositive

This function deactivates a positive torque limiting that was activated with '_enableAxisTorqueLimitPositiveIn'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.4 _disableMonitoringOfEncoderDifference

This function deactivates encoder monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

1.3.32.5 _disableMovingToEndStop

With the system function _disableMovingToEndStop (), the monitoring of the 'travel to a fixed stop' is deactivated at the same time as an axis motion activated by a motion command, the axis is traversed/moved farther in position-regulated mode.

The command _disableMovingToEndStop () does not work with active clamping, i.e. if the clamping torque has already been reached.

In this case, the command _disableMovingToEndStop () is aborted with the alarm 30002. The clamping itself remains active.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableClamping
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableClamping

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command does not take place until the command is completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId' for which the command status should be tracked.

1.3.32.6 _disableTorqueLimiting

The "Disable torque limiting" function is used to disable one of the superimposed torque limiting motion commands.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableTorqueLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableTorqueLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command does not take place until the command is completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.32.7 _enableAxisAdditiveTorque

This function activates an additive torque setpoint.
The torque is specified cyclically via the 'AdditiveTorque' interconnection.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the torque value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'additiveTorqueType' parameter is evaluated.

additiveTorqueType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the torque value is be taken from the command or from a system variable.
With DEFAULT_VALUE, the torque value is taken from system variable 'defaultAdditiveTorque'.
The parameter is only evaluated when 'referenceType:=VALUE'.
With DIRECT, the torque value is taken from command parameter 'torqueLimit'.

additiveTorque (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the additive torque.
The parameter is only evaluated when 'referenceType:=VALUE' and 'additiveTorqueType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.8 _enableAxisTorqueLimitNegative

This function activates a negative torque limiting which is specified in the extended drive protocol in accordance with the speed controller.
The limiting values are specified cyclically via the 'TorqueLimitNegative' interconnection.

For active B+/B- torque limitation resulting from the _enableAxisTorqueLimitPositive or _enableAxisTorqueLimitNegative command, the following monitoring will be deactivated:

- Following error monitoring
- The velocity error monitoring using reference model
- The time limitations for positioning monitoring and standstill monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the limiting value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'torqueLimitType' parameter is evaluated.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the limit value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the limiting value is taken from system variable 'defaultTorqueLimitNegative'.
The parameter is only evaluated when 'referenceType:=VALUE'.
With DIRECT, the limiting value is taken from command parameter 'torqueLimit'.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the negative limiting value.
The parameter is only evaluated when 'referenceType:=VALUE' and 'torqueLimitType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.9 _enableAxisTorqueLimitPositive

This function activates a positive torque limiting, which is specified in the extended drive protocol in accordance with the speed controller.
The limiting values are specified cyclically via the 'TorqueLimitPositive' interconnection.

For active B+/B- torque limitation resulting from the _enableAxisTorqueLimitPositive or _enableAxisTorqueLimitNegative command, the following monitoring will be deactivated:

- Following error monitoring
- The velocity error monitoring using reference model
- The time limitations for positioning monitoring and standstill monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the limiting value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'torqueLimitType' parameter is evaluated.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the limit value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the limiting value is taken from system variable 'defaultTorqueLimitPositive'.
With DIRECT, the limiting value is taken from command parameter 'torqueLimit'.
The parameter is only evaluated when 'referenceType:=VALUE'.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the positive limit value.
The parameter is only evaluated when 'referenceType:=VALUE' and 'torqueLimitType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.10 _enableMonitoringOfEncoderDifference

This function activates the monitoring system for monitoring the maximum permissible difference between the measuring systems specified in the command.
An alarm is generated if the difference is exceeded. Furthermore, the system variable 'sensorMonitoring.slippageTolerance' is set.
In sensorMonitoring.slippageTolerance, LIMIT_EXCEEDED is displayed if the specified actual encoder value difference is exceeded during the command _enableMonitoringOfEncoderDifference.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

referenceEncoderType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	2
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Specifies which measuring system is to be used as the reference system.
With ACTUAL, the measuring system that is currently active is used as the reference system.
With DIRECT, the measuring system specified in the 'referenceEncoder' parameter is used.

referenceEncoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	1

Specifies the difference between two measuring systems.
The parameter is only evaluated if the value DIRECT was transferred in the 'referenceEncoderType' parameter.

encoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	4
System default:	1

Specifies the measuring system for which the difference to the reference encoder is monitored.

maximalEncoderDifferenceType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	5
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Determines how the maximum permissible measuring system difference is specified.
With DIRECT, the value in parameter 'maximalSensorDifference' is used.
With ACTUAL, the value of system variable 'sensorMonitoring.maximalSensorDifference' is used.

maximalEncoderDifference (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the maximum permissible measuring system difference. The parameter is only evaluated if the value DIRECT was transferred in the 'maximalEncoderDifferenceType' parameter.

1.3.32.11 _enableMovingToEndStop

This function is used to activate the monitoring of travel to fixed endstop in parallel to an axis motion activated using a motion command and the clamping torque when the fixed endstop has been reached. This process is known as 'clamping'. Whether a following error or evaluation of the axis torque is used to detect that the fixed endstop is reached is set during configuration.
The _enableMovingToEndStop () function can be used to switch over the clamping torque, even while the clamping is active.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

clampingValueType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	2
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of the clamping value.
With DIRECT, the value set in the 'clampingValue' parameter is used as the programmed clamping value.
With EFFECTIVE, the last programmed clamping value is used.
With USER_DEFAULT, the default clamping value defined in system variable 'userDefaultClamping.clampingValue' is used. This default can be modified by entering a value in the 'clampingValue' parameter. The value of the parameter is interpreted as a percentage.

clampingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	100.0

Clamping value (torque or force), the evaluation depends on the 'clampingValueType' parameter.
It is only used with parameter 'clampingValueType := USER_DEFAULT' or 'clampingValueType := DIRECT'.
In all other cases, the parameter is irrelevant.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableClamping
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableClamping

IMMEDIATELY (60)	Immediate command transition
WHEN_ENDSTOP_REACHED (231)	When clamping value is reached
WHEN_FUNCTION_DISABLED (232)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_ENDSTOP_REACHED, the transition to the next command takes place when the clamping value is reached.
With WHEN_FUNCTION_DISABLED, the transition to the next command does not take place until the command is completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId' for which the command status should be tracked.

torqueLimitUnit (optional)

Direction:	Input parameters
Data type:	EnumTorqueLimitUnitType
Parameter index:	7
System default:	DEFAULT_UNIT

EnumTorqueLimitUnitType

DEFAULT_UNIT (268)	Unit for the TO
TORQUE (269)	Torque

Reference point for limit.
With DEFAULT_UNIT, the force or torque relates to the load side.
With TORQUE, the limiting value is interpreted as torque relating to the drive side.

1.3.32.12 _enableTorqueLimiting

The "Enable torque limiting" function is used to enable torque limiting in parallel to motion. This is effective immediately.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	2
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of the limiting value.
With DIRECT, the value set in the 'torqueLimit' parameter is used as the programmed value.
With EFFECTIVE, the last programmed torque limit value is used.
With USER_DEFAULT, the default torque limit value defined in system variable 'userDefaultTorque-Limiting.torqueLimit' is used. This default can be modified by entering a value in the 'torqueLimit' parameter. The value of the parameter is interpreted as a percentage.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	100.0

Desired axis torque limit value.
This parameter is evaluated independently of the 'torqueLimitType' parameter.
torqueLimitType -> DIRECT: Specifies the value as torque or force; the evaluation depends on the 'torqueLimitUnit' parameter.
torqueLimitType -> USER_DEFAULT: If this parameter is not specified, the default torque limit value defined in system variable 'userDefaultTorqueLimiting.torqueLimit' is used.
If a value is specified, the entry is interpreted as a percentage relative to the torque reduction defined in the 'userDefaultTorqueLimiting.torqueLimit' system variable.
torqueLimitType -> EFFECTIVE: Parameter 'torqueLimit' is not evaluated.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableTorqueLimiting
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableTorqueLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_FUNCTION_DISABLED (232)	When command is finished or aborted
WHEN_TORQUELIMIT_REACHED (233)	As soon as torque is limited
WHEN_TORQUELIMIT_GONE (234)	As soon as torque limiting is deactivated

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_TORQUELIMIT_REACHED, the transition to the next command takes place as soon as the torque is limited (the torque limit setting is reached in drive). This state is derived from the PROFIBUS status word 'MeldW' (PZW 5), bit 1 (M < Mx) of the drive.
With WHEN_TORQUELIMIT_GONE, the transition to the next command takes place as soon as the torque limiting has been reached once and the torque limiting has been exited. This state is derived from the PROFIBUS status word 'MeldW' (PZW 5), bit 1 (M < Mx) of the drive.
With WHEN_FUNCTION_DISABLED, the transition to the next command does not take place until the command is aborted. The command can be aborted by calling the '_disableTorqueLimiting' or '_resetAxis' functions in another task.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

torqueLimitUnit (optional)

Direction:	Input parameters
Data type:	EnumTorqueLimitUnitType
Parameter index:	7
System default:	DEFAULT_UNIT

EnumTorqueLimitUnitType

DEFAULT_UNIT (268)	Unit for the TO
TORQUE (269)	Torque

Reference of limiting values during programming.
With DEFAULT_UNIT, the force or torque relates to the load side. The force or torque relates to the load side. The gear ratio is always taken into account here.
With TORQUE, the limiting value is interpreted as torque relating to the drive side. A gear ratio is not taken into account.

1.3.32.13 _getAxisDataSetParameter

This function reads an axis data set.

Return value: StructRetReadGetAxisDataSet

StructRetReadGetAxisDataSet
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisDataSetReadWrite	-

StructAxisDataSetReadWrite
Parameters of an axis data set

Structure	Name	Data type	Unit
ControllerStruct	Parameters of the controller	StructControllerType	-
DynamicData	Dynamic characteristic values of the cascade control system	StructDynamicData	-
DynamicComp	Parameters for dynamic compensation of the control loop	StructDynamicComp	-
ProcessModel	Parameters of process model	StructProcessModel	-
DynamicFollowing	Parameters of the dynamic following error monitoring	StructDynamicFollowing	-
ControllerDynamic	Parameters of the reference model monitoring	StructControllerDynamic	-
EncoderNumber	Measuring system number	StructEncoderNumber	-
Gear	Ratio of the load gearbox	StructGear	-
ClampingMonitoring	Parameters for setting the torque monitoring	StructClampingMonitoring	-

StructControllerType
Parameters of the controller

Structure	Name	Data type	Unit
conType	Controller type	EnumAxisController-Type	-
PV_Controller	Parameters for the P controller with precont-rol	StructPVController	-
PD_Controller	Parameters for the PD controller	StructPDController	-
PID_Controller	Parameters for the PID controller	StructPIDController	-

StructDynamicData
Dynamic characteristic values of the cascade control system

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time con-stant for position cont-rol loop	LREAL	s
velocityTimeConstant	Equivalent time con-stant of the velocity control loop	LREAL	s
torqueTimeConstant	Equivalent time con-stant for torque control loop	LREAL	s

StructDynamicComp
Parameters for dynamic compensation of the control loop

Structure	Name	Data type	Unit
enable	Activate dynamic com-pensation	EnumYesNo	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructProcessModel
Parameters of the process model (PT2)

Structure	Name	Data type	Unit
ks	Transfer ratio of pro-cess	LREAL	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructDynamicFollowing
Parameters of the dynamic following error monitoring

Structure	Name	Data type	Unit
enable	Activation of the dynamic following error monitoring	EnumYesNo	-
minVelocity	Velocity value for the start of the rise, proportional to velocity, of the maximum permissible following error	LREAL	m/s
minPositionTolerance	Maximum permissible following error at velocities below the start of the rise, proportional to velocity, of the characteristic curve	LREAL	m
maxPositionTolerance	Maximum permissible following error at maximum velocity	LREAL	m
warningLimit	Warning limit of following error monitoring	LREAL	%

StructControllerDynamic
Parameters of the reference model monitoring

Structure	Name	Data type	Unit
enable	Activation of the reference model monitoring	EnumYesNo	-
maxVeloTolerance	Maximum velocity tolerance	LREAL	%

StructEncoderNumber
Measuring system number

Structure	Name	Data type	Unit
encoderNumber	Measuring system number	DINT	-

StructGear
Load gearing ratio

Structure	Name	Data type	Unit
numFactor	Numerator for load gearing	LREAL	-
denFactor	Denominator for load gearing	LREAL	-

StructClampingMonitoring
Parameters for setting the torque monitoring

Structure	Name	Data type	Unit
recognitionMode	Activation mode for torque monitoring	EnumRecogniti-onMode	-
followingErrorDevia-tion	Specifies the required following error for detection of endstop	LREAL	m
positionTolerance	Specifies the permis-sible deviation of the actual value from the setpoint in the clamped state	LREAL	m

StructPVController
Parameters for the P controller with precontrol

Structure	Name	Data type	Unit
enableDSC	Activation of DSC	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kv	P controller gain	LREAL	rps
kpc	Precontrol weighting	LREAL	%
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

StructPDController
Parameters for the PD controller

Structure	Name	Data type	Unit
kp	P controller gain	LREAL	rps
kd	Gain of DT1 element	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s

StructPIDController
Parameters for the PID controller

Structure	Name	Data type	Unit
preCon	Activate precontrol	EnumYesNo	-
kpc	Precontrol factor	LREAL	%
kp	P component gain	LREAL	rps
ki	I component gain	LREAL	-
kd	D component gain	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s
enableAntiWindup	I component limit	EnumYesNo	-
balanceFilterMode	Specifies the balancing filter	EnumBalanceFilterMode	%

EnumAxisControllerType

NODEF (0)	No significance
DIRECT (1)	Control only
PD (2)	PID controller
PV (3)	PV controller
PID (4)	PID controller
PID_ACTUAL (5)	PID controller with actual value-dependent D-component

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumRecognitionMode

DO_NOT_CLAMP (0)	No detection
CLAMP_BY_FOLLOWING_ERROR_DEVIATION (1)	By following error
CLAMP_WHEN_TORQUE_LIMIT_REACHED (2)	By force moment/torque

EnumBalanceFilterMode

OFF (0)	Balancing filter not active
MODE_1 (1)	Balancing filter active
MODE_2 (2)	Expanded balancing filter active

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.32.14 _homing

Function for homing an axis.
The dynamic parameters for homing are programmable and refer to all phases of the homing procedure.

The execution time of the _homing command for the absolute encoder adjustment, i.e. for homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER and homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION, can take several interpolation cycles.
The status of the command can be queried by querying the CommandID of the homing command using _getstateOfAxisCommand(), even when the axis is already in the 'homed' state.

Exceptions
The homing approach velocity can be programmed in the command. The homing approach velocity, homing entry velocity and the homing deceleration velocity are set during configuration.
The execution sequence for active homing ('homingMode:=ACTIVE_HOMING') is set during configuration.
The criteria for passive homing ('homingMode:=PASSIVE_HOMING') are set during configuration independently from the active homing.
An axis has the 'referenced' or 'homed' status when the axis coordinate system has been aligned with the homing signal. The status can be read from the 'positioningState.homed' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

homingMode (optional)

Direction:	Input parameters
Data type:	EnumHomingMode
Parameter index:	2
System default:	ACTIVE_HOMING

EnumHomingMode

ACTIVE_HOMING (5)	Homing operation in configured mode
DIRECT_HOMING (41)	Setting of current position value
ENABLE_OFFSET_OF_ABSOLUTE_ENCODER (49)	Absolute encoder offset
PASSIVE_HOMING (104)	Homing with the next zero mark
DIRECT_HOMING_RELATIVE (352)	Relative setting of current position value
SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION (385)	Absolute encoder adjustment with position specification from command

Specifies the type of homing.

With ACTIVE_HOMING, a homing operation is started. The homing mode is set during configuration.

With DIRECT_HOMING, the home position coordinates are set directly to the current axis coordinates; the command does not initiate an axis motion.

With DIRECT_HOMING_RELATIVE, the home position coordinates are set relative to the current axis coordinates; the command does not initiate an axis motion.

With ENABLE_OFFSET_OF_ABSOLUTE_ENCODER, the value of 'absHomingEncoder.abshift' offset is added to the offset already existing in the system with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=RELATIVE;

with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=ABSOLUTE', the value 'absHomingEncoder.absshift' is calculated in as a fixed offset.

The total offset is saved in the NVRAM and is available after the controller is disabled. Once a new project has been loaded in the controller, the saved offset is no longer available.

With 'homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION', the value in the 'homePosition' parameter is set as the current position and the resulting absolute encoder offset is calculated from this. This value is displayed in system variable 'absoluteEncoder.totalOffsetValue', and stored in the system as a retain variable. The value in the configuration data 'absHomingEncoder'-absshift is not changed.

With PASSIVE_HOMING, the next zero mark crossed by the axis when it moves after the '_homing' command is used as the home position. The homing command is active parallel to the motion. It remains active until the homing procedure is complete. It can be cancelled with _resetAxis or _disableAxis. If the homing command is issued, e.g. with a stationary axis, and before the motion command, the homing command should be programmed with 'mergeMode:=IMMEDIATELY' and 'nextCommand:=IMMEDIATELY' and the motion command programmed with 'mergeMode:=IMMEDIATELY' or 'mergeMode:=NEXT'.

If the homing command is issued when a motion command is active, it should be programmed with 'mergeMode:=IMMEDIATELY'.

homePositionType (optional)

Direction:	Input parameters
Data type:	EnumValueSpecification
Parameter index:	3
System default:	USER_DEFAULT

EnumValueSpecification

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the type of home position coordinates.
With DIRECT, the value set in the 'homePosition' parameter is used as the programmed home position coordinate.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultHoming.homePosition' is used.

homePosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Home position coordinates, the evaluation depends on the 'homePositionType' parameter.

With 'homePositionType:=DIRECT', the parameter is used as the new home position coordinate. In all other cases, the parameter is irrelevant.

During the setting of the parameter 'homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER', the parameter has no meaning.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity input for the homing approach velocity.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultHoming.hominggApproachVelocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Homing approach velocity, evaluation dependent on the parameter 'velocityType'.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.
The homing approach velocity is only relevant for ACTIVE_HOMING homingMode CAM_AND_ZM or CAM.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandHoming
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandHoming

AT_MOTION_START (13)	Start of interpolation
IMMEDIATELY (60)	Asynchronous commands
WHEN_AXIS_HOMED (157)	Axis has been homed
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_AXIS_HOMED, the transition takes place when the axis has been synchronized.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or the command is aborted.
WHEN_INTERPOLATION_DONE and WHEN_MOTION_DONE are only effective when homing-Mode := ACTIVE_HOMING.
In all other cases, the command transition takes place in accordance with WHEN_AXIS_HOMED.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.32.15 _redefinePosition

This function sets the coordinate system of an axis. The position value can be defined as an absolute value or as a relative position offset. The setpoint or the actual value can be set to the specified value. The other value is adjusted, taking into account the current difference and the following error.
The dynamic parameters can also be set for a virtual axis.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

redefineSpecification (optional)

Direction:	Input parameters
Data type:	EnumRedefineSpecification
Parameter index:	2
System default:	COMMAND_VALUE

EnumRedefineSpecification

COMMAND_VALUE (28)	Setpoint
ACTUAL_VALUE (34)	Actual value
VIRTUAL_AXIS (151)	Setpoint for a virtual axis
COMMAND_VALUE_BASIC_MOTION (310)	Setpoint for main motion
COMMAND_VALUE_SUPERIMPOSED_MOTION (311)	Setpoint for superimposed motion

Specifies the assignment destination for the new position.
With COMMAND_VALUE, the value in position is set as the new position setpoint depending on 'redefineMode', and the actual position is adjusted.
With ACTUAL_VALUE, the value in position is set as the new actual position value depending on 'redefineMode', and the position setpoint is adjusted accordingly.
With VIRTUAL_AXIS, the target position of a virtual axis is set in addition to the setpoints for velocity and acceleration.
The values (for s, v, a) are used as start values for the motion command.
The values take effect on the virtual axis when a motion command is started. When the motion starts, the new virtual axis setpoints are calculated according to the settings.

redefineMode (optional)

Direction:	Input parameters
Data type:	EnumRedefineMode
Parameter index:	3
System default:	ABSOLUTE

EnumRedefineMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Specifies the type of assignment for the new position.
With ABSOLUTE, the values are assigned as absolute positions.
With RELATIVE, the values are added to the existing position (offset).
With RELATIVE, the response is the same for 'redefineSpecification := COMMAND_VALUE' and 'redefineSpecification := ACTUAL_VALUE'.

position (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Position specification.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the current velocity when 'redefineSpecification := VIRTUAL_AXIS'.
In all other cases, the parameter is ignored.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the current acceleration when 'redefineSpecification := VIRTUAL_AXIS'.
In all other cases, the parameter is ignored.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command takes place after the new position has been set.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.16 _setAndGetEncoderValue

This function returns the difference between the specified measuring systems or synchronizes two measuring systems.
During synchronization, it is only possible to change the value of a measuring system if it is not currently being used for position control.

Return value: StructRetEncoderValue

StructRetEncoderValue
Return value with error code and difference

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
encoderDifference	Difference of specified measuring systems	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value encoderDifference:

The difference of the specified measuring systems is indicated in 'encoderDifference'.

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

mode (optional)

Direction:	Input parameters
Data type:	EnumSetNoSet
Parameter index:	2
System default:	SET

EnumSetNoSet

SET (178)	Set value
NO_SET (179)	Do not set value

Specifies whether the actual value of the measuring system that is specified in the 'encoder' parameter should be set.
With SET, the actual value of the measuring system that is specified in the 'encoder' parameter is set.
With NO_SET, the actual value of the measuring system that is specified in the 'encoder' parameter is not set.

encoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	1

Specifies the measuring system to be selected or synchronized.
The encoder number is assigned during configuration.
The valid range is [1, 8]

referenceEncoderType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	4
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Specifies the reference measuring system.
With ACTUAL, the active measuring system is used.
With DIRECT, the measuring system specified in the 'referenceEncoder' parameter is used.

referenceEncoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	5
System default:	1

Specifies the reference measuring system.
The encoder number is assigned during configuration.
The valid range is [1, 8]

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command takes place after changeover to the new measuring system.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.17 _setAxisDataSetActive

This function activates the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be activated.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.32.18 _setAxisDataSetParameter

This function overwrites an axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisDataSetReadWrite
Parameter index:	3

StructAxisDataSetReadWrite
Parameters of an axis data set

Structure	Name	Data type	Unit
ControllerStruct	Parameters of the controller	StructControllerType	-
DynamicData	Dynamic characteristic values of the cascade control system	StructDynamicData	-
DynamicComp	Parameters for dynamic compensation of the control loop	StructDynamicComp	-
ProcessModel	Parameters of process model	StructProcessModel	-

Structure	Name	Data type	Unit
DynamicFollowing	Parameters of the dynamic following error monitoring	StructDynamicFollowing	-
ControllerDynamic	Parameters of the reference model monitoring	StructControllerDynamic	-
EncoderNumber	Measuring system number	StructEncoderNumber	-
Gear	Ratio of the load gearbox	StructGear	-
ClampingMonitoring	Parameters for setting the torque monitoring	StructClampingMonitoring	-

StructControllerType
Parameters of the controller

Structure	Name	Data type	Unit
conType	Controller type	EnumAxisControllerType	-
PV_Controller	Parameters for the P controller with precontrol	StructPVController	-
PD_Controller	Parameters for the PD controller	StructPDController	-
PID_Controller	Parameters for the PID controller	StructPIDController	-

StructDynamicData
Dynamic characteristic values of the cascade control system

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant of the velocity control loop	LREAL	s
torqueTimeConstant	Equivalent time constant for torque control loop	LREAL	s

StructDynamicComp
Parameters for dynamic compensation of the control loop

Structure	Name	Data type	Unit
enable	Activate dynamic compensation	EnumYesNo	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructProcessModel
Parameters of the process model (PT2)

Structure	Name	Data type	Unit
ks	Transfer ratio of process	LREAL	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructDynamicFollowing
Parameters of the dynamic following error monitoring

Structure	Name	Data type	Unit
enable	Activation of the dynamic following error monitoring	EnumYesNo	-
minVelocity	Velocity value for the start of the rise, proportional to velocity, of the maximum permissible following error	LREAL	m/s
minPositionTolerance	Maximum permissible following error at velocities below the start of the rise, proportional to velocity, of the characteristic curve	LREAL	m
maxPositionTolerance	Maximum permissible following error at maximum velocity	LREAL	m
warningLimit	Warning limit of following error monitoring	LREAL	%

StructControllerDynamic
Parameters of the reference model monitoring

Structure	Name	Data type	Unit
enable	Activation of the reference model monitoring	EnumYesNo	-
maxVeloTolerance	Maximum velocity tolerance	LREAL	%

StructEncoderNumber
Measuring system number

Structure	Name	Data type	Unit
encoderNumber	Measuring system number	DINT	-

StructGear
Load gearing ratio

Structure	Name	Data type	Unit
numFactor	Numerator for load gearing	LREAL	-
denFactor	Denominator for load gearing	LREAL	-

StructClampingMonitoring
Parameters for setting the torque monitoring

Structure	Name	Data type	Unit
recognitionMode	Activation mode for torque monitoring	EnumRecogniti-onMode	-
followingErrorDevia-tion	Specifies the required following error for detection of endstop	LREAL	m
positionTolerance	Specifies the permis-sible deviation of the actual value from the setpoint in the clamped state	LREAL	m

StructPVController
Parameters for the P controller with precontrol

Structure	Name	Data type	Unit
enableDSC	Activation of DSC	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kv	P controller gain	LREAL	rps
kpc	Precontrol weighting	LREAL	%
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

StructPDController
Parameters for the PD controller

Structure	Name	Data type	Unit
kp	P controller gain	LREAL	rps
kd	Gain of DT1 element	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s

StructPIDController
Parameters for the PID controller

Structure	Name	Data type	Unit
preCon	Activate precontrol	EnumYesNo	-
kpc	Precontrol factor	LREAL	%
kp	P component gain	LREAL	rps
ki	I component gain	LREAL	-
kd	D component gain	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s
enableAntiWindup	I component limit	EnumYesNo	-
balanceFilterMode	Specifies the balancing filter	EnumBalanceFilterMode	%

EnumAxisControllerType

NODEF (0)	No significance
DIRECT (1)	Control only
PD (2)	PID controller
PV (3)	PV controller
PID (4)	PID controller
PID_ACTUAL (5)	PID controller with actual value-dependent D-component

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumRecognitionMode

DO_NOT_CLAMP (0)	No detection
CLAMP_BY_FOLLOWING_ERROR_DEVIATION (1)	By following error
CLAMP_WHEN_TORQUE_LIMIT_REACHED (2)	By force moment/torque

EnumBalanceFilterMode

OFF (0)	Balancing filter not active
MODE_1 (1)	Balancing filter active
MODE_2 (2)	Expanded balancing filter active

Specifies the data to be used to overwrite the data set.

1.3.33 Synchronous axis - Object and Alarm Handling

1.3.33.1 _cancelAxisCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its 'CommandId' in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled.

Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBe-Cancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.33.2 _disableAxisInterface

This function deactivates the actuator or encoder interface. When the interface is deactivated, the corresponding drivers are deactivated.

The actuator interface is the interface of the axis to the drive, which in addition to the setpoint also includes the control signals of the axis to the drive and the status information and the actual speed value of the drive to the controller, when connection of the drive is via PROFIdrive message frame.

Correspondingly the encoder interface, in addition to the actual encoder value, also includes the control signals of the axis to the encoder evaluation, e.g. in the drive, and the status information of the encoder evaluation, e.g. in the drive, to the controller, when connection of the encoder is via PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

actor (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	2
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the deactivation of the actuator interface.
With YES, the actuator interface is deactivated.
With NO, the actuator interface status remains unchanged.

sensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	3
System default:	0

Specifies the deactivation of the encoder interface.
If the bit is set for the corresponding encoder interface, the interface is deactivated. If the bit is not set, the encoder interface status remains unchanged.

hwLimitSwitch (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the deactivation of the hardware limit switch monitoring.
With YES, the hardware limit switch monitoring is deactivated.
With NO, the status of the hardware limit switch monitoring remains unchanged.

additionalSensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	0

Specifies the deactivation of the encoder interface of the additional sensor.
If the bit is set for the corresponding additional sensor, the additional sensor is deactivated. If the bit is not set, the additional sensor status remains unchanged.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableAxisInterface
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnableAxisInterface

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.33.3 _disableAxisSimulation

This function switches the axis out of simulation mode.
The actual encoder values displayed in the structure elements of the 'sensorData' system variable are applied as actual values; the status of all other actual values remains unchanged.
Any existing following error is removed immediately.
The current simulation status can be queried in the 'simulation' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.33.4 _enableAxisInterface

This function activates the actuator or encoder interface. When the interface is activated, the corresponding drivers are activated.

The actuator interface is the interface of the axis to the drive, which in addition to the setpoint also includes the control signals of the axis to the drive and the status information and the actual speed value of the drive to the controller, when connection of the drive is via PROFIdrive message frame.

Correspondingly the encoder interface, in addition to the actual encoder value, also includes the control signals of the axis to the encoder evaluation, e.g. in the drive, and the status information of the encoder evaluation, e.g. in the drive, to the controller, when connection of the encoder is via PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

actor (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	2
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the activation of the actuator interface.
With YES, the actuator interface is activated.
With NO, the actuator interface status remains unchanged.

sensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	3
System default:	0

Specifies the activation of the encoder interface.
If the bit is set for the corresponding encoder interface, the interface is activated. If the bit is not set, the encoder interface status remains unchanged.

hwLimitSwitch (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the activation of the hardware limit switch monitoring.
With YES, the hardware limit switch monitoring is activated.
With NO, the status of the hardware limit switch monitoring remains unchanged.

additionalSensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	0

Specifies the activation of the encoder interface of the additional sensor.
If the bit is set for the corresponding additional sensor, the additional sensor is activated. If the bit is not set, the additional sensor status remains unchanged.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableAxisInterface
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnableAxisInterface

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.33.5 _enableAxisSimulation

This function switches the axis to simulation mode.
If the axis is moving, it is stopped by entering a velocity of 0.
The actual values indicated in the system variables are taken from the setpoints in the same way as on a virtual axis.
The current simulation status can be queried in the 'simulation' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.33.6 _enableDistributedMotionDelayValueCalculation

This function can be used for explicit initiation of offset determination for distributed synchronous operation (e.g. after adding an axis to a modular machine).
This function cannot be used to determine the offset for recursive synchronous operation with PROFINET, because in this case the offset is not calculated correctly by the system.

This function can be used as of SIMOTION V4.1, SP4.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

master

Direction:	Input parameters
Data type:	MasterType
Parameter index:	1

This is a positioning axis

mode (optional)

Direction:	Input parameters
Data type:	EnumAxisDelayValueCalculationMode
Parameter index:	2
System default:	STANDARD

EnumAxisDelayValueCalculationMode

ACTUAL (7)	Currently accessible synchronous connections
STANDARD (358)	Configured synchronous connections

Specifies connections for which the offset is to be determined.
With ACTUAL, the offset is determined for all currently accessible synchronous connections.
With STANDARD, the offset is determined for all configured synchronous connections. During startup, the command waits until all configured stations are active.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDelayValueCalculation
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandDelayValueCalculation

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.33.7 _getAxisErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the errorNumber parameter is pending.
NO - The error specified by the errorNumber parameter is not pending.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.33.8 _getAxisErrorState

This function provides information on whether axis alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.33.9 _getStateOfMotionBuffer

This function returns the status of the axis command queue.

Return value: StructRetMotionBuffer

StructRetMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumMotionBuffer-State	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':
EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':
Specifies the number of entries in the command queue.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.33.10 _resetAxis

This function switches the axis to a defined initial state.
All active motions are stopped by entering a setpoint of 0.
Pending commands are deleted, synchronous commands are aborted. The command is executed synchronously.
Pending errors on the axis are deleted. For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.
System variables modified by the program are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.33.11 _resetAxisConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.33.12 _resetAxisError

This function resets axis errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.
The command is asynchronous. When applicable, the error is not reset until the local response activated by the error has been completed.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.33.13 _resetMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.34 Cam - Message functions

1.3.34.1 _getCamFollowingDerivative

This function returns all cam derivatives up to the nth derivative. If the function does not contain any coefficients, 0.0 is output.

Note
On linear interpolated cam segments, the velocity is constant on a given segment. This results in a constant value on the segment for the first derivative. As acceleration is zero at constant velocity, the value 0.0 is returned on linear interpolated segment sections in the second derivative.

Return value: StructRetGetCamFollowingDerivative

StructRetGetCamFollowingDerivative

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	nth derivative	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:
cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

derivativeOrder (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	2
System default:	1

Specifies the derivative to be read out.
The nth derivative is returned whereby n corresponds to the value transferred in 'derivativeOrder'.

leadingPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	3
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system of the value in the domain (master value).
With ACTUAL, the value is specified in the scaled and offset range.
With BASIC, the value is specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

leadingPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the master value for which the slave value is to be determined.

followingPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	5
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system in which the value is to be returned in the range (slave value).
With ACTUAL, the value is specified in the scaled and offset range.
With BASIC, the value is specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

1.3.34.2 _getCamFollowingValue

This function returns the cam value for a specified value in the domain (master value).

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:
cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

leadingPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	2
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system of the value in the domain (master value).
With ACTUAL, the value is specified in the scaled and offset range.
With BASIC, the value is specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

leadingPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the master value for which the slave value is to be determined.

followingPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	4
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system in which the value is to be returned in the range (slave value).
With ACTUAL, the value is specified in the scaled and offset range.
With BASIC, the value is specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

1.3.34.3 _getCamLeadingValue

This function returns the value in the domain (master value) for a specified value in the range (slave value).
Since this relationship is not always unique, a reference value can be specified.

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Value in the domain of the cam (master value)

Parameter:
cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

followingPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	2
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system in the range.
With ACTUAL, the value is specified in the scaled and offset range.
With BASIC, the value is specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

followingPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the value in the range (slave value).

leadingPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	4
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system of the leading approximation value and the leading value read out.
With ACTUAL, the value is specified in the scaled and offset range.
With BASIC, the value is specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

leadingApproachPositionType (optional)

Direction:	Input parameters
Data type:	EnumApproachPositionType
Parameter index:	5
System default:	NONE

EnumApproachPositionType

DIRECT (40)	Value of the function parameter
NONE (93)	No specification (cam start point is used)

Type of specification of the approximation value for the value in the domain (master value).
With DIRECT, the specification is made in the 'leadingApproachPosition' parameter.
With NONE, there is no explicit specification. In this case, the start point of the cam is used as an approximation value.

leadingApproachPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the approximation value in the domain.
The parameter is only evaluated if 'leadingApproachPositionType:=DIRECT'.

1.3.35 Cam - Command tracking

1.3.35.1 _bufferCamCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.35.2 _getStateOfCamCommand

This function returns the execution state of a command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT
- The 'commandId' is unknown or

command has already been completed

WAITING_FOR_SYNC_START
- Waiting for synchronous start

WAITING
- Command has been decoded, but execution has not

yet started

ACTIVE
- Command is being executed.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.35.3 _removeBufferedCamCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.36 Cam - Geometry

1.3.36.1 _addPointToCam

This function adds an interpolation point 'f' to a cam at position 'p'.
The values in the domain of the cam are always specified as increasing values, i.e. in a positive direction.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

camPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	2
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Reference system for specifying the point coordinates.
With ACTUAL, the values are specified in the scaled and offset range.
With BASIC, the values are specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

leadingRangePosition

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the coordinates of the new point in the domain.

followingRangePosition

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the coordinates of the new point in the range.

1.3.36.2 _addPolynomialSegmentToCam

This function creates segment f = f(t), consisting of a maximum sixth degree polynomial.
The polynomial parameters are input in the real range.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

polynomialCoefficients (optional)

Direction:	Input parameters
Data type:	ARRAY [0..6] OF LREAL
Parameter index:	2
System default:	[0..6] = 0.0

Specifies polynomial coefficients p0 to p6.

segmentBegin (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the start point of the range selected from the normal form.

segmentEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the end point of the range selected from the normal form.

camPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	11
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system for representing the selected range.
With ACTUAL, the values are specified in the scaled and offset range.
With BASIC, the values are specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

1.3.36.3 _addSegmentToCam

This function creates segment f = f(t), consisting of a maximum
6. sixth degree polynomial and a trigonometric component
'sineAmplitude * sin' (sinePeriod * t + sinePhase).
The polynomial parameters, amplitude, period and phase for the sine function are entered in normal form.
The transformation parameters are specified in the basic curve representation (without scaling or offset) or in the actual curve representation (with scaling and offset).

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

polynomialCoefficients (optional)

Direction:	Input parameters
Data type:	ARRAY [0..6] OF LREAL
Parameter index:	2
System default:	[0..6] = 0.0

Specifies polynomial coefficients p0 to p6.

sineAmplitude (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the sine amplitude in the range.
With 0, there is no trigonometric component.

sinePeriod (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the sine period in the domain.

sinePhase (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the sinusoidal phase in the domain.

segmentBegin (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the start point of the range selected from the normal form.

segmentEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the end point of the range selected from the normal form.

camPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	14
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system for representing the selected range.
With ACTUAL, the values are specified in the scaled and offset range.
With BASIC, the values are specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

leadingRangeStartPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the start point for representing the selected area in the domain.
The value specified in 'segmentBegin' is moved to this point.

leadingRangeEndPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	1.0

Specifies the end point for representing the selected area in the domain.
The value specified in 'segmentEnd' is moved to this point.

followingRangeStartPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the start point for representing the selected area in the range.
The minimum in the range of the normal form is moved to this value, within the domain selected by 'segmentBegin' and 'segmentEnd'.

followingRangeEndPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	1.0

Specifies the end point for representing the selected area in the range.
The maximum in the range of the normal form is moved to this value within the domain selected by 'segmentBegin' and 'segmentEnd'.
If the maximum and minimum are the same, the function is mapped onto 'followingRangeStartPoint'.

followingRangeSpecificationMode (optional)

Direction:	Input parameters
Data type:	EnumCamFollowingRangeSpecificationMode
Parameter index:	19
System default:	MIN_MAX

EnumCamFollowingRangeSpecificationMode

MIN_MAX (270)	Minimum and maximum of normal form
STARTPOINT_ENDPOINT (271)	Start and end point of normal form
OFFSET_SCALE (272)	Start point and offset scaling

This function parameter specifies how the followingRangeStartPoint and followingRangeEndPoint parameters are to be interpreted.
MIN_MAX maps the minimum of the normal form onto followingRangeStartPoint and the maximum of the normal form onto followingRangeEndPoint.
STARTPOINT_ENDPOINT maps the start point of the normal form onto followingRangeStartPoint and the end point of the normal form onto followingRangeEndPoint. If the start and end points of the normal form are identical in the range, a shift to the followingRangeStartPoint occurs (followingRange-EndPoint has no significance in this case). The values are interpreted as being identical if the difference between followingRangeStartPoint and followingRangeEndPoint is smaller than the internal numerical accuracy.
OFFSET_SCALE interprets followingRangeStartPoint as an offset of the normal form in the range and followingRangeEndPoint as a scaling in the range. Scaling takes place first, whereby the start point of the normal form is the scaling origin. This is then followed by the offset.

1.3.36.4 _interpolateCam

This function checks a cam for continuity within the cam and at the edges and interpolates the cam with the specified type of interpolation.
The interpolation defines the connections between points, between segments, and between a point and a segment.
The boundary conditions for interpolation can be defined.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

camPositionMode (optional)

Direction:	Input parameters
Data type:	EnumCamPositionMode
Parameter index:	2
System default:	USER_DEFAULT

EnumCamPositionMode

ACTUAL (7)	Representation with scaling and offset
BASIC (16)	Representation without scaling and offset
USER_DEFAULT (149)	User default

Specifies the reference system for specifying the start and end points of the interpolation.
With ACTUAL, the values are specified in the scaled and offset range.
With BASIC, the values are specified in the non-scaled and non-offset range.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.camPositionMode' is used.

leadingRangeStartPointType (optional)

Direction:	Input parameters
Data type:	EnumCamLeadingRangeStartPoint
Parameter index:	3
System default:	LEADING_RANGE_START

EnumCamLeadingRangeStartPoint

LEADING_RANGE_START (69)	Start point of cam
LEADING_RANGE_VALUE (70)	Value of the function parameter

Type of start point specification for interpolation.
With LEADING_RANGE_START, the point is determined internally from the cam description.
With LEADING_RANGE_VALUE, the value specified in 'leadingRangeStartPoint' is used.

leadingRangeStartPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the start point of the range to be interpolated.
The parameter is only evaluated if
the 'leadingRangeStartPointType' parameter has the value LEADING_RANGE_VALUE.

leadingRangeEndPointType (optional)

Direction:	Input parameters
Data type:	EnumLeadingRangeEndPoint
Parameter index:	5
System default:	LEADING_RANGE_END

EnumLeadingRangeEndPoint

LEADING_RANGE_END (68)	End point of cam
LEADING_RANGE_VALUE (70)	Value of the function parameter

Type of end point specification for interpolation.
With LEADING_RANGE_END, the point is determined internally from the cam description.
With LEADING_RANGE_VALUE, the value specified in the 'leadingRangeEndPoint' parameter is used.

leadingRangeEndPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	1.0

Specifies the end point of the range to be interpolated.
The parameter is only evaluated if
'leadingRangeEndPointType' = LEADING_RANGE_VALUE.

camMode (optional)

Direction:	Input parameters
Data type:	EnumCamMode
Parameter index:	7
System default:	NO_CONSTRAINTS

EnumCamMode

CYCLIC_ABSOLUTE (36)	Absolute cyclical cam
CYCLIC_RELATIVE (37)	Relative cyclical cam
NO_CONSTRAINTS (96)	Non-cyclical cam

Specifies the cam characteristic.
With CYCLIC_ABSOLUTE, the start and end points of the cam are set equal (connected with continuity of position).
With CYCLIC_RELATIVE, the first derivative in the start and end points of the cam are set the same (connected with constant velocity).
With NO_CONSTRAINTS, the start and end points of the cam are not set equal; there are no constraints.

interpolationMode (optional)

Direction:	Input parameters
Data type:	EnumCamInterpolationMode
Parameter index:	8
System default:	LINEAR

EnumCamInterpolationMode

B_SPLINE (25)	Approximation using Bezier splines
C_SPLINE (38)	Interpolation using cubic splines
LINEAR (72)	Linear interpolation

Specifies the interpolation type.
With B_SPLINE, an approximation is made using Bezier splines (curve along the interpolation points).
With C_SPLINE, interpolation is made using cubic splines (curve through the interpolation points).
With LINEAR, linear interpolation takes place.

noChangeDifferenceLeadingRange (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the distance between points up to which no addition/interpolation takes place in the domain. The next value is taken.

noChangeDifferenceFollowingRange (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the distance between points up to which no addition/interpolation takes place in the range. The next value is taken.

combineDifferenceLeadingRange (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the distance between points up to which the end point of the previous segment is joined to the start point of the following segment in the domain.
If the gaps between segments/points are greater than the specified distance between points, interpolation takes place.

combineDifferenceFollowingRange (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the distance between points up to which the end point of the previous segment is joined to the start point of the following segment in the range.
If the gaps between segments/points are greater than the specified distance between points, interpolation takes place.

1.3.36.5 _setCamOffset

This function offsets the domain or range of the cam.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

offsetRange (optional)

Direction:	Input parameters
Data type:	EnumCamRange
Parameter index:	2
System default:	LEADING_RANGE

EnumCamRange

FOLLOWING_RANGE (182)	Range
LEADING_RANGE (183)	Domain

Specifies the selection of the coordinate axis to be offset.
With FOLLOWING_RANGE, the range of the function is offset.
With LEADING_RANGE, the domain of the function is offset.

offsetMode (optional)

Direction:	Input parameters
Data type:	EnumOffsetMode
Parameter index:	3
System default:	ABSOLUTE

EnumOffsetMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Specifies the effect of the offset value.
With ABSOLUTE, the offset value applies instead of the previous value.
With RELATIVE, the offset value is added to the current offset.

offset (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Offset value.

1.3.36.6 _setCamScale

This function scales a cam in the range or domain.
Scaling can be performed before or after segments and points are inserted or interpolated.
Scaling is applied when the command is successfully issued.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

scalingRange (optional)

Direction:	Input parameters
Data type:	EnumCamRange
Parameter index:	2
System default:	LEADING_RANGE

EnumCamRange

FOLLOWING_RANGE (182)	Range
LEADING_RANGE (183)	Domain

Specifies the selection of the coordinate axis to be scaled.
With FOLLOWING_RANGE, the range of the function is scaled.
With LEADING_RANGE, the domain of the function is scaled.

scalingSpecification (optional)

Direction:	Input parameters
Data type:	EnumScalingSpecification
Parameter index:	3
System default:	COMPLETE_RANGE

EnumScalingSpecification

COMPLETE_RANGE (29)	Complete scaling
SPECIFIC_PART_OF_RANGE (127)	Scaling in specific range

Specifies the selection of the scaling type.
With COMPLETE_RANGE, the entire cam is scaled.
With SPECIFIC_PART_OF_RANGE, scaling is performed in the range [start point, end point].
The range number is specified in the specificRangeNumber parameter.

specificRangeNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	4
System default:	1

Specifies the number of the range scaling.
The parameter is only evaluated when 'scalingSpecification:=SPECIFIC_PART_OF_RANGE'.

specificRangeStartPointType (optional)

Direction:	Input parameters
Data type:	EnumCamValueType
Parameter index:	5
System default:	EFFECTIVE

EnumCamValueType

DIRECT (40)	Value of the function parameter
EFFECTIVE (45)	Last programmed value

Type of specification of the start point of the range to be scaled.
With DIRECT, the value set in specificRangeStartPoint is used.
With EFFECTIVE, the last programmed value is used.
The parameter is only evaluated when 'scalingSpecification:=SPECIFIC_PART_OF_RANGE'.

specificRangeStartPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the start point of the range to be scaled.
The parameter is only evaluated when 'scalingSpecification:=SPECIFIC_PART_OF_RANGE' and when 'specificRangeStartPointType:=DIRECT'.

specificRangeEndPointType (optional)

Direction:	Input parameters
Data type:	EnumCamValueType
Parameter index:	7
System default:	EFFECTIVE

EnumCamValueType

DIRECT (40)	Value of the function parameter
EFFECTIVE (45)	Last programmed value

Type of specification of the end point of the range to be scaled.
With DIRECT, the value set in 'specificRangeEndPoint' is used.
With EFFECTIVE, the last programmed value is used.
The parameter is only evaluated when 'scalingSpecification:=SPECIFIC_PART_OF_RANGE'.

specificRangeEndPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the end point of the range to be scaled.
The parameter is only evaluated when 'scalingSpecification:=SPECIFIC_PART_OF_RANGE' and when 'specificRangeEndPointType:=DIRECT'.

scaleValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	1.0

Specifies the scaling factor.
Scaling factor 0 can only be specified with the following specification: 'scalin-gRange:=FOLLOWING_RANGE' and 'scalingSpecification:=COMPLETE_RANGE'.

1.3.37 Cam - Object and Alarm Handling

1.3.37.1 _getCamErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:
cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.37.2 _getCamErrorState

This function provides information on whether function alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-

Structure	Name	Data type	Unit
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

1.3.37.3 _resetCam

This function sets a cam to the initial state.
Pending errors are deleted.
The geometry and compensations are deleted.
The system variables are reset depending on the parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

insertMode (optional)

Direction:	Input parameters
Data type:	EnumInsertMode
Parameter index:	2
System default:	BY_START_POSITION

EnumInsertMode

BY_END_POSITION (22)	Determination of segment order according to segment end point
BY_START_POSITION (23)	Determination of segment order according to segment start point
ACCORDING_TO_DEFINITION_ORDER (110)	Determination of segment order according to time of insertion

Specifies the sequence interpretation with respect to subsequent entry of overlapping segments.
With BY_END_POSITION, the segments are arranged according of their end points.
With BY_START_POSITION, the segments act according to their start points.
With ACCORDING_TO_DEFINITION_ORDER, the segments act in the order in which they were inserted. Segments that have been entered later overlap existing segments.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values which were entered during cam configuration, relevant for the variable 'camPositionMode'.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	4
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

camData (optional)

Direction:	Input parameters
Data type:	EnumCamData
Parameter index:	5
System default:	CAM_DATA_RESET

EnumCamData

CAM_DATA_RESET (295)	Delete cam
LOAD_CONFIGURED_DATA (296)	Reset cam to ES data

Specifies whether or not the cam is to be reset to the cam loaded from the engineering system.
With CAM_DATA_RESET, the cam is deleted. In this state, it can be reprogrammed from the user program.
With LOAD_CONFIGURED_DATA, the cam is reset to the cam loaded from the engineering system.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	6
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.37.4 _resetCamConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Collection of configuration data takes place when system variable activationModeChangedConfigData is set to the value 'COLLECT_CHANGED_CONFIG_DATA'.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

1.3.37.5 _resetCamError

This function resets errors on the cam.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

cam

Direction:	Input parameters
Data type:	CamType
Parameter index:	1

Specifies the technology object of the 'cam' type or a variable of the 'CamType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.38 Probe - Command tracking

1.3.38.1 _bufferMeasuringInputCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.38.2 _getStateOfMeasuringInputCommand

This function returns the execution state of a command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT
- The 'commandId' is unknown or

command has already been completed

WAITING_FOR_SYNC_START
- Waiting for synchronous start

WAITING
- Command has been decoded, but execution has not

yet started

ACTIVE
- Command is being executed.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.38.3 _removeBufferedMeasuringInputCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.39 Probe - Measurement

1.3.39.1 _disableMeasuringInput

This function deactivates the measurement job.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

1.3.39.2 _enableMeasuringInput

This function initiates a measurement job. The measuring criteria such as edge and range are specified at the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

measuredEdgeMode (optional)

Direction:	Input parameters
Data type:	EnumMeasuredEdge
Parameter index:	2
System default:	USER_DEFAULT

EnumMeasuredEdge

FALLING_EDGE (86)	Falling edge (high to low)
RISING_EDGE (117)	Rising edge (low to high)
USER_DEFAULT (149)	User default
BOTH_EDGES (224)	Measurement for both edges
BOTH_EDGES_FIRST_RISING (225)	Measurement for both edges, beginning with a rising edge (low to high)
BOTH_EDGES_FIRST_FALLING (226)	Measurement for both edges, beginning with a falling edge (high to low)

Specifies the measuring signal edge to be evaluated.
With FALLING_EDGE, the falling edge is evaluated.
With RISING_EDGE, the rising edge is evaluated.
With USER_DEFAULT, the value set in the 'userDefault.measuredEdge' system variable is used.
With BOTH_EDGES, both edges are evaluated without specification of a specific first edge
With BOTH_EDGES_FIRST_RISING, both edges are evaluated, starting with a rising edge (low to high).
With BOTH_EDGES_FIRST_FALLING, both edges are evaluated, starting with a falling edge (high to low).

measuringRangeMode (optional)

Direction:	Input parameters
Data type:	EnumMeasuringRangeMode
Parameter index:	3
System default:	USER_DEFAULT

EnumMeasuringRangeMode

USER_DEFAULT (149)	User default
WITHOUT_SPECIFIC_AREA (168)	Measurement without specific area
WITH_SPECIFIC_AREA (170)	Measurement with specified area

Specifies whether measurement is to be performed always or within a specific range.
With USER_DEFAULT, the value set in the 'userDefault.measuringRangeMode' system variable is used.
With WITHOUT_SPECIFIC_AREA, no specific measuring range is evaluated.
With WITH_SPECIFIC_AREA, the measurement job is restricted to the measuring range specified in the range parameter.

measuringRangeStartPositionType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	4
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for start point of measuring range.
With DIRECT, the value set in the 'measuringRangeStartPosition' parameter is used as the programmed start position.
With EFFECTIVE, the last programmed start position is used.
With USER_DEFAULT, the default defined in the system variable 'userdefault.measuringRangeStartPosition' is used.

measuringRangeStartPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Start point of the measuring range, only relevant for 'measuringRangeStartPositionType:=DIRECT'.
The start position must be less than the end position for an assigned linear axis.

measuringRangeEndPositionType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	6
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for end point of measuring range.
With DIRECT, the value set in the 'measuringRangeEndPosition' parameter is used as the programmed end position.
With EFFECTIVE, the last programmed end position is used.
With USER_DEFAULT, the default defined in the system variable 'userdefault.measuringRangeEndPosition' is used.

measuringRangeEndPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the end position of the measuring range, only relevant for 'measuringRangeEndPositionType:=DIRECT'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableMeasuring
Parameter index:	8
System default:	IMMEDIATELY

EnumNextCommandEnableMeasuring

IMMEDIATELY (60)	Immediate command transition
WHEN_TRIGGER_OCCURED (165)	Transition when measurement is finished

Specifies condition for transition to next command.
With IMMEDIATELY, the transition to the next command takes place after the measurement job is activated.
With WHEN_TRIGGER_OCCURRED, the transition takes place when the measuring result is received.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	9
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.39.3 _enableMeasuringInputCyclic

This function activates a cyclical measurement job. The results of the last measurement are displayed in the system variables. Up to two measurements can be displayed in one cycle clock.
When both edges are measured, the rising edge values are output in 'measuredValue1' and the falling edge values in 'measuredValue2'.
When only rising edges or only falling edges are measured, the first event in the cycle clock is entered in 'measuredValue1' and the second event in 'measuredValue2'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

mode (optional)

Direction:	Input parameters
Data type:	EnumMeasuringInputCyclicMode
Parameter index:	2
System default:	USER_DEFAULT

EnumMeasuringInputCyclicMode

USER_DEFAULT (149)	User default
ALL_EDGES (360)	All edges
RISING_EDGES_ONLY (361)	Rising edges
FALLING_EDGES_ONLY (362)	Falling edges

Specifies the measuring signal edge to be evaluated.
With USER_DEFAULT, the value set in the 'userDefault.measuringEdgeCyclicMode' system variable is used.
With ALL_EDGES, all edges are measured.
With RISING_EDGES_ONLY, only rising edges are measured.
With FALLING_EDGES_ONLY, only falling edges are measured.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

measuringRangeMode (optional)

Direction:	Input parameters
Data type:	EnumMeasuringRangeMode
Parameter index:	5
System default:	USER_DEFAULT

EnumMeasuringRangeMode

USER_DEFAULT (149)	User default
WITHOUT_SPECIFIC_AREA (168)	Measurement without specific area
WITH_SPECIFIC_AREA (170)	Measurement with specified area

Specifies whether measurement is to be performed always or within a specific range.
With USER_DEFAULT, the value set in the 'userDefault.measuringRangeMode' system variable is used.
With WITHOUT_SPECIFIC_AREA, no specific measuring range is evaluated.
With WITH_SPECIFIC_AREA, the measurement job is restricted to the measuring range specified in the range parameter.

measuringRangeStartPositionType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	6
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for start point of measuring range.
With DIRECT, the value set in the 'measuringRangeStartPosition' parameter is used as the programmed start position.
With EFFECTIVE, the last programmed start position is used.
With USER_DEFAULT, the default defined in the system variable 'userdefault.measuringRangeStartPosition' is used.

measuringRangeStartPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Start point of the measuring range, only relevant for 'measuringRangeStartPositionType:=DIRECT'.
The start position must be less than the end position for an assigned linear axis.

measuringRangeEndPositionType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	8
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for end point of measuring range.
With DIRECT, the value set in the 'measuringRangeEndPosition' parameter is used as the programmed end position.
With EFFECTIVE, the last programmed end position is used.
With USER_DEFAULT, the default defined in the system variable 'userdefault.measuringRangeEndPosition' is used.

measuringRangeEndPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the end position of the measuring range, only relevant for 'measuringRangeEndPositionType:=DIRECT'.

1.3.40 Probe - Object and Alarm Handling

1.3.40.1 _disableMeasuringInputSimulation

This function deactivates measuring input simulation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

1.3.40.2 _enableMeasuringInputSimulation

This function activates the measuring input simulation ('simulation=active'). 'Measurement event arrived' is set and the result is assigned to the programmed measured value.
If the measuring input simulation mode is active, the '_enableMeasuringInput' (trigger) function is used to enter the simulated measured value in 'Measuredvalue1' and to set 'state=trigger_occurred'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

measuredValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2
System default:	0.0

Specifies the value assigned as the measured position.

1.3.40.3 _getMeasuringInputErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.40.4 _getMeasuringInputErrorState

This function provides information on whether measuring input alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

1.3.40.5 _resetMeasuringInput

This function switches a measuring input to an initial state. Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.
System variables modified by the program are reset on request.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.40.6 _resetMeasuringInputConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

1.3.40.7 _resetMeasuringInputError

This function resets measuring input errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

measuringInput

Direction:	Input parameters
Data type:	MeasuringInputType
Parameter index:	1

Specifies the technology object of the 'measuringInput' type or a variable of the 'MeasuringInputType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.41 Output cam - Command tracking

1.3.41.1 _bufferOutputCamCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.41.2 _getStateOfOutputCamCommand

This function returns the execution state of a command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT
- The 'commandId' is unknown or

command has already been completed

WAITING_FOR_SYNC_START
- Waiting for synchronous start

WAITING
- Command has been decoded, but execution has not

yet started

ACTIVE
- Command is being executed.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.41.3 _removeBufferedOutputCamCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.42 Output cam - Object and Alarm Handling

1.3.42.1 _disableOutputCamSimulation

This function resets the output cam from simulation mode. The output of output cam is switched according to the output cam status and the signal inversion.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

1.3.42.2 _enableOutputCamSimulation

This function simulates an output cam by disconnecting the output. The output cam remains active internally. The status is tracked, but the output of the output cam is not switched.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

1.3.42.3 _getOutputCamErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.42.4 _getOutputCamErrorState

This function provides information on whether output cam alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

1.3.42.5 _resetOutputCam

This function switches an output cam to an initial state. Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.42.6 _resetOutputCamConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

1.3.42.7 _resetOutputCamError

This function resets output cam errors. For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.43 Output cam - Switching functions

1.3.43.1 _disableOutputCam

This function deactivates output cam processing. The output cam is switched to NEGATIVE status. The output assigned to the output cam in the configuration remains occupied by the cam.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

1.3.43.2 _enableOutputCam

This function activates output cam processing. The type of output cam is set during configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

switchOnPositionType (optional)

Direction:	Input parameters
Data type:	EnumOutputCamPosition
Parameter index:	2
System default:	USER_DEFAULT

EnumOutputCamPosition

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of specification of the switch-on position of the output cam.
With DIRECT, the value set in the 'switchOnPosition' parameter is used as the programmed switch-on position.
With EFFECTIVE, the last programmed switch-on position is used.
With RELATIVE, the value set in the 'switchOnPosition' parameter is added to the last programmed value.
With USER_DEFAULT, the default defined in system variable 'userDefault.switchOnPosition' is used.

switchOnPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the switch-on position of the output cam.
Only relevant when 'switchOnPositionType:=DIRECT' or when 'switchOnPositionType:=RELATIVE'.

switchOffValueType (optional)

Direction:	Input parameters
Data type:	EnumOutputCamValue
Parameter index:	4
System default:	USER_DEFAULT

EnumOutputCamValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Specifies the end of the output cam.
With DIRECT, the value set in the 'switchOffValue' parameter is used as the programmed position.
With EFFECTIVE, the last programmed position is used.
With RELATIVE, the value set in the 'switchOffValue' parameter is added to the last programmed value.
With USER_DEFAULT, the default setting defined in the 'userDefault.switchOffPosition' system variable is used for a position-based cam or 'userDefault.onDuration' for a time-based cam.

switchOffValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the end of the output cam, only relevant when 'switchOffValueType:=DIRECT' or 'switchOffValueType:=RELATIVE'.
If a position-based cam was configured, then the transferred value corresponds to a position: The unit is set when configuring the technology object.
If a time-based cam was configured, then the transferred value corresponds to a time: The unit is set when configuring the technology object.

forceDirection (optional)

Direction:	Input parameters
Data type:	EnumForceDirection
Parameter index:	6
System default:	USER_DEFAULT

EnumForceDirection

BOTH (20)	Positive and negative effective direction
EFFECTIVE (45)	Last programmed effective direction
NEGATIVE (85)	Negative effective direction
POSITIVE (107)	Positive effective direction
USER_DEFAULT (149)	User default

Specifies the effective direction.
With BOTH, both directions are effective.
With EFFECTIVE, the last programmed direction is used.
With NEGATIVE, the effective direction is defined as the negative axis direction.
With POSITIVE, the effective direction is defined as the positive axis direction.
With USER_DEFAULT, the default direction defined in system variable 'userDefault.forceDirection' is used.

invertOutput (optional)

Direction:	Input parameters
Data type:	EnumOutputCamInvert
Parameter index:	7
System default:	WITHOUT_CHANGE

EnumOutputCamInvert

NO (91)	Issue the output cam signal not inverted
WITHOUT_CHANGE (167)	Retain the current switching state
YES (173)	Issue the output cam signal inverted

Specifies output cam signal inversion.
With NO, the output cam switching signal is output with the non-inverted value.
With WITHOUT_CHANGE, the current signal output setting is retained.
With YES, the output cam switching signal is output inverted.

activationTimeType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	8
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of actuation/delay time for the activation signal edge.
With DIRECT, the value set in the 'activationTime' parameter is used as the programmed value.
With EFFECTIVE, the last programmed actuation/delay time is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.activationTime' is used.

activationTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the actuation/delay time for the activation signal edge.
If the value is greater than 0, the time acts as a switch-on delay; otherwise it acts as an actuation time.
The value is only relevant when 'activationTimeType:=DIRECT'.
The unit is set when configuring the technology object.

deactivationTimeType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	10
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Specifies the actuation/delay time of the deactivation signal edge.
With DIRECT, the value set in the 'deactivationTime' parameter is used as the programmed value.
With EFFECTIVE, the last programmed actuation/delay time is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.deactivationTime' is used.

deactivationTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the actuation/delay time for the deactivation signal edge.
If the value is greater than 0, the time acts as a switch-off delay; otherwise it acts as an actuation time.
The value is only relevant when 'deactivationTimeType:=DIRECT'.
The unit is set when configuring the technology object.

noSwitchingRangeType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	12
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for the hysteresis of the output cam.
With DIRECT, the value set in the 'noSwitchingRange' parameter is used as the programmed value.
With EFFECTIVE, the last programmed hysteresis is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.noSwitchingRange' is used.

noSwitchingRange (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the hysteresis of the output cam switching edges.
Only relevant when 'noSwitchingRangeType:=DIRECT'.

1.3.43.3 _setOutputCamCounter

This function modifies the start count value and/or the current count value of the counter cam.
The current count value decreases by 1 with each switching event of the output cam. If the actual count value is 0, the output of the output cam takes place, and the actual count value is reset to the start count value.
The effective count values can be read from the counterCamData system variable structure.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

actualValueMode (optional)

Direction:	Input parameters
Data type:	EnumSetNoSet
Parameter index:	2
System default:	NO_SET

EnumSetNoSet

SET (178)	Set value
NO_SET (179)	Do not set value

Specifies whether or not the current count value defined in 'actualValue' is to be set.
With SET, the current count value is set.
With NO_SET, the current count value is not set.

actualValue (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	1

Specifies the new current count value.
This parameter is only evaluated when 'actualValueMode:=SET' is specified.

startValueMode (optional)

Direction:	Input parameters
Data type:	EnumSetNoSet
Parameter index:	4
System default:	NO_SET

EnumSetNoSet

SET (178)	Set value
NO_SET (179)	Do not set value

Specifies whether or not the start count value defined in 'startValue' is to be set.
With SET, the start count value is set.
With NO_SET, the start count value is not set.

startValue (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	5
System default:	1

Specifies the new start value.
This parameter is only evaluated when 'startValueMode:=SET' is specified.

1.3.43.4 _setOutputCamState

This function deactivates the output cam function and sets the output cam status to the specified value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

outputCam

Direction:	Input parameters
Data type:	OutputCamType
Parameter index:	1

Specifies the technology object of the 'outputCam' type or a variable of the 'OutputCamType' on which the command is to be executed.

stateType (optional)

Direction:	Input parameters
Data type:	EnumOutputCamState
Parameter index:	2
System default:	WITHOUT_CHANGE

EnumOutputCamState

NEGATIVE (85)	Output cam switched off
POSITIVE (107)	Output cam switched on
WITHOUT_CHANGE (167)	Retain the current switching state

Sets the output to a defined value.
With NEGATIVE, the output is set to FALSE.
With POSITIVE, the output is set to TRUE.
With WITHOUT_CHANGE, the output retains the last programmed switching state.

timeOffset (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies a timed switching edge. The specified time is an offset of the start of the next output cam cycle clock. The time offset must be smaller than the output cam cycle clock. One timed activation signal edge and one timed deactivation signal can be set for each output cam cycle. If multiple activation or deactivation signal edges are issued in a cycle clock, the most recently written values apply. The start of the output cycle clock is shifted with respect to the beginning of the machining cycle clock by the value output in the 'tOutput' system variable.

1.3.44 Cam track (camTrackType) - Command tracking

1.3.44.1 _bufferCamTrackCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.44.2 _getStateOfCamTrackCommand

This function returns the execution state of a command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT
- The 'commandId' is unknown or

command has already been completed

WAITING_FOR_SYNC_START
- Waiting for synchronous start

WAITING
- Command has been decoded, but execution has not

yet started

ACTIVE
- Command is being executed.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.44.3 _removeBufferedCamTrackCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.45 Cam track (camTrackType) - Object and Alarm Handling

1.3.45.1 _disableCamTrackSimulation

This function resets the cam track from simulation mode. The cam track output is switched according to the cam track status and the signal inversion.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

1.3.45.2 _enableCamTrackSimulation

This function simulates a cam track by disconnecting the output. The cam track remains active internally. The status is tracked but the cam track output is not switched.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

1.3.45.3 _getCamTrackErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:
camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.45.4 _getCamTrackErrorState

This function provides information on whether cam track alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

1.3.45.5 _resetCamTrack

This function switches the cam track to an initial state. Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.45.6 _resetCamTrackConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

1.3.45.7 _resetCamTrackError

This function resets output cam track errors. For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.46 Cam track (camTrackType) - Switching functions

1.3.46.1 _disableCamTrack

This function deactivates output cam processing. The output of the output cam is reset. A deactivation mode can be specified.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

stopMode (optional)

Direction:	Input parameters
Data type:	EnumCamTrackStopMode
Parameter index:	2
System default:	USER_DEFAULT

EnumCamTrackStopMode

EFFECTIVE (45)	Last programmed value
IMMEDIATELY (60)	Effective immediately
USER_DEFAULT (149)	User default
BY_CAM_TRACK_END (591)	At output cam track end
IMMEDIATELY_BY_CAM_TRACK_OUTPUT_INACTIVE (596)	Immediate for inactive output cam track output

Specifies the deactivation of output cam track processing.
With IMMEDIATELY, the output cam track deactivation is effective immediately.
With IMMEDIATELY_BY_CAM_TRACK_OUTPUT_INACTIVE, the output cam track is deactivated as soon as the output cam track output becomes inactive (or no single output cams are active).
With BY_CAM_TRACK_END, the output cam track is deactivated at the end of the output cam track.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.stopMode' is used.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.46.2 _enableCamTrack

This function activates the processing of a cam track. The processing of the output cam track can be set for one-time processing or cyclic processing in the command. The activation mode is specified in the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

singleCamSettingsType (optional)

Direction:	Input parameters
Data type:	EnumEffectiveUserDefault
Parameter index:	2
System default:	USER_DEFAULT

EnumEffectiveUserDefault

EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Specifies the single output cam parameters.
With EFFECTIVE, the last programmed single output cam parameters are used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.singleCamSettings' is used.

activationTimeType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	3
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of actuation/delay time for the activation signal edge.
With DIRECT, the value set in the 'activationTime' parameter is used as the programmed value.
With EFFECTIVE, the last programmed actuation/delay time is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.activationTime' is used.

activationTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the actuation/delay time for the activation signal edge.
If the value is greater than 0, the time acts as a switch-on delay; otherwise it acts as an actuation time.
The value is only relevant when 'activationTimeType:=DIRECT'.
The unit is set when configuring the technology object.

deactivationTimeType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	5
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Specifies the actuation/delay time of the deactivation signal edge.
With DIRECT, the value set in the 'deactivationTime' parameter is used as the programmed value.
With EFFECTIVE, the last programmed actuation/delay time is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.deactivationTime' is used.

deactivationTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the actuation/delay time for the deactivation signal edge.
If the value is greater than 0, the time acts as a switch-off delay; otherwise it acts as an actuation time.
The value is only relevant when 'deactivationTimeType:=DIRECT'.
The unit is set when configuring the technology object.

hysteresisRangeType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	7
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification for the hysteresis of the output cam.
With DIRECT, the value set in the 'hysteresisRange' parameter is used as the programmed value.
With EFFECTIVE, the last programmed hysteresis is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.hysteresisRange' is used.

hysteresisRange (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the hysteresis of the output cam switching edges.
Only relevant when 'hysteresisRangeType:=DIRECT'.

axisReferencePositionType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	9
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of the reference position of the output cam track.
With DIRECT, the value set in the 'axisReferencePosition' parameter is used as the programmed value.
With EFFECTIVE, the last programmed reference position is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.axisReferencePosition' is used.

axisReferencePosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the reference position of the output cam track.
Only relevant when 'axisReferencePositionType:=DIRECT'.

camTrackLengthType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	11
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of the output cam track length.
With DIRECT, the value set in the 'camTrackLength' parameter is used as the programmed value.

With EFFECTIVE, the last programmed output cam track length is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.camTrackLength' is used.

camTrackLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the output cam track length.
Only relevant when 'camTrackLengthType:=DIRECT'.

activationMode (optional)

Direction:	Input parameters
Data type:	EnumCamTrackActivationMode
Parameter index:	13
System default:	USER_DEFAULT

EnumCamTrackActivationMode

CYCLIC (35)	Cyclical output cam track activation
EFFECTIVE (45)	Last programmed output cam track activation
USER_DEFAULT (149)	User default
NO_CYCLIC (595)	Non-cyclical output cam track activation

Specifies the output cam track activation.
With CYCLIC, the domain of the output cam track is processed cyclically.
With NO_CYCLIC, the domain of the output cam track is not processed cyclically.
With EFFECTIVE, the last programmed output cam track activation is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.activationMode' is used.

startMode (optional)

Direction:	Input parameters
Data type:	EnumCamTrackStartMode
Parameter index:	14
System default:	USER_DEFAULT

EnumCamTrackStartMode

EFFECTIVE (45)	Last programmed value
IMMEDIATELY (60)	Effective immediately
USER_DEFAULT (149)	User default
IMMEDIATELY_BY_CAM_TRACK_OUTPUT_INACTIVE (596)	Immediate for inactive output cam track output
NEXT_CAM_TRACK_CYCLE (597)	With the next track cycle

Specifies when the output cam track processing is activated.
With IMMEDIATELY, the output cam track processing is activated immediately.
With IMMEDIATELY_BY_CAM_TRACK_OUTPUT_INACTIVE, the output cam track processing is activated as soon as the output cam track output becomes inactive (or no single output cams are active).
With NEXT_CAM_TRACK_CYCLE, the output cam track processing is activated at the next track cycle.
With EFFECTIVE, the last programmed value is used.
With USER_DEFAULT, the default setting defined in system variable 'userdefault.startMode' is used.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumCamTrackNextCommand
Parameter index:	15
System default:	IMMEDIATELY

EnumCamTrackNextCommand

IMMEDIATELY (60)	Immediately
WITH_CAM_TRACK_ACTIVATION (598)	With output cam track activation
WHEN_CAM_TRACK_DONE (599)	At output cam track end

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately as soon as the command is adopted in the interpolator.

In the event of WITH_CAM_TRACK_ACTIVATION, transition takes place when the new cam track is loaded, i.e. when the cam track parameters specified in the command _enableCamTrack are activated. The time when the new cam track is loaded is determined by the 'startMode' command parameter.

In the event of WHEN_CAM_TRACK_DONE, the transition takes place when the end of the first active cam track cycle is reached. The cam track begins with 'default=0' and finishes with the end of the cam track (i.e. cam track end = cam track start + cam track length).

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	16
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.46.3 _setCamTrackState

This function deactivates the cam track function and sets the cam track status to the specified value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

camTrack

Direction:	Input parameters
Data type:	_CamTrackType
Parameter index:	1

Specifies the technology object of the 'camTrack' type or a variable of the '_CamTrackType' on which the command is to be executed.

state (optional)

Direction:	Input parameters
Data type:	EnumCamTrackSetState
Parameter index:	2
System default:	WITHOUT_CHANGE

EnumCamTrackSetState

OFF (97)	Deactivated
ON (98)	Activated
WITHOUT_CHANGE (167)	Last switching state

Sets the output to a defined value.
With OFF, the output is set to FALSE.
With ON, the output is set to TRUE.
With WITHOUT_CHANGE, the output retains the last programmed switching state.

1.3.47 Positioning axis - Information and conversion

1.3.47.1 _getAxisInternalPosition

This function supplies the associated encoder position value for a specified position value in the axis coordinate system.

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

userValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the position value in the axis coordinate system for which the associated encoder position is to be determined.

1.3.47.2 _getAxisUserPosition

This function supplies the position value in the axis coordinate system for a specified encoder position value.

Return value: StructRetGetValue

StructRetGetValue
Return value with error code and LREAL

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
value	Value	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

internalValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the encoder position value for which the position value is to be determined in the axis coordinate system.

1.3.47.3 _getProgrammedTargetPosition

This function supplies the absolute end position currently programmed on the axis. Any superimposition is taken into account.

Return value: StructRetGetAxisProgrammedTargetPosition

StructRetGetAxisProgrammedTargetPosition
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
targetPositionType	Validity and meaning of targetPosition	EnumAxisProgrammedTargetPosition	-
targetPosition	Programmed target position	LREAL	-

EnumAxisProgrammedTargetPosition

NO_POS_MOTION_ACTIVE (275)	No commands with position reference active
BASIC_POS_MOTION_ACTIVE (276)	Main motion with position reference active
SUPERIMPOSED_POS_MOTION_ACTIVE (277)	Superimposed motion with position reference active
BASIC_AND_SUPERIMPOSED_POS_MOTION_ACTIVE (278)	Main motion and superimposed motion with position reference active
BASIC_POS_AND_SUPERIMPOSED_NON_POS_MOTION_ACTIVE (279)	Main motion with position reference and superimposed motion without position reference active; the targetPosition contains the absolute target position of the main motion in the reference system of the main motion
BASIC_NON_POS_AND_SUPERIMPOSED_POS_MOTION_ACTIVE (280)	Main motion without position reference and superimposed motion with position reference active; targetPosition contains the absolute target position of the superimposed motion is in the reference system of the superimposed motion

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value value:

Function value

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

posCommandSpecification (optional)

Direction:	Input parameters
Data type:	EnumAxisPosCommandSpecification
Parameter index:	2
System default:	ALL_POSITION_RELATED_MOTION

EnumAxisPosCommandSpecification

ALL_POSITION_RELATED_MOTION (273)	All commands with position reference
ONLY_POSITION_COMMAND (274)	_pos commands only

Selection of active commands with position reference in the interpolator that are related to the command result.
With ALL_POSITION_RELATED_MOTION, the output end position relates to all commands with position reference.
With ONLY_POSITION_COMMAND, the output end position relates to the active _pos commands.

1.3.47.4 _setAndGetEncoderValue

This function returns the difference between the specified measuring systems or synchronizes two measuring systems.
During synchronization, it is only possible to change the value of a measuring system if it is not currently being used for position control.

Return value: StructRetEncoderValue

StructRetEncoderValue
Return value with error code and difference

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
encoderDifference	Difference of specified measuring systems	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value encoderDifference:

The difference of the specified measuring systems is indicated in 'encoderDifference'.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

mode (optional)

Direction:	Input parameters
Data type:	EnumSetNoSet
Parameter index:	2
System default:	SET

EnumSetNoSet

SET (178)	Set value
NO_SET (179)	Do not set value

Specifies whether the actual value of the measuring system that is specified in the 'encoder' parameter should be set.
With SET, the actual value of the measuring system that is specified in the 'encoder' parameter is set.
With NO_SET, the actual value of the measuring system that is specified in the 'encoder' parameter is not set.

encoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	1

Specifies the measuring system to be selected or synchronized.
The encoder number is assigned during configuration.
The valid range is [1, 8]

referenceEncoderType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	4
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Specifies the reference measuring system.
With ACTUAL, the active measuring system is used.
With DIRECT, the measuring system specified in the 'referenceEncoder' parameter is used.

referenceEncoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	5
System default:	1

Specifies the reference measuring system.
The encoder number is assigned during configuration.
The valid range is [1, 8]

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command takes place after changeover to the new measuring system.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.48 Positioning axis - Command tracking

1.3.48.1 _bufferAxisCommandId

This function enables the commandId and associated command status to be saved for a period after command execution so that it is still available after the command is complete. The maximum number of CommandId and command status pairs that can be saved is set during configuration using typeOfAxis.decodingConfig.numberOfMaxBufferedCommandId.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId' whose command status is saved and thus remains available.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.48.2 _cancelAxisCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its 'CommandId' in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled.

Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.48.3 _getMotionStateOfAxisCommand

The function provides the motion status of an active axis command.

Return value: StructRetMotionCommandState

StructRetMotionCommandState
Return value with error code and motion status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionCommandId-State	Motion status	EnumMotionComman-dIdState	-

EnumMotionCommandIdState

NOT_EXISTENT (94)	'commandId' is not known or command is already completed
BUFFERED (207)	Command is in the command queue
IN_EXECUTION (208)	Command is being executed
IN_ACCELERATION (209)	Motion generated by the command is in the acceleration phase (only for TO axis)
IN_CONSTANT_MOTION (210)	Motion generated by the command is in the constant velocity phase (only for TO axis)
IN_DECELERATION (211)	Motion generated by the command is in the deceleration phase
AXIS_HOMED (212)	Axis is synchronized (only for TO axis)
INTERPOLATION_DONE (213)	Setpoint interpolation of command is complete
SYNCHRONIZING (214)	Synchronizing to synchronous operation (only for TO synchronous operation)
DESYNCHRONIZING (215)	Desynchronizing from synchronous operation (only for TO synchronous operation)
SYNCHRONIZED (216)	Synchronous operation is synchronized (only for TO synchronous operation)
MODIFICATION_ACTIVE (217)	Compensating motion for scaling or offset in synchronous operation is active (only for TO synchronous operation)
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionCommandIdState'
With NOT_EXISTENT, the 'commandId' for this axis instance is unknown or the command has already been completed.
With BUFFERED, the command is in the command queue.
With IN_EXECUTION, the command is being executed but no motion command is active.
With IN_ACCELERATION, motion generated by the command is in the acceleration phase. The value is output for _move and _pos commands.
With IN_CONSTANT_MOTION, motion generated by the command is in the constant velocity phase. The value is output for _move and _pos commands.
With IN_DECELERATION, motion generated by the command is in the deceleration phase. The value is output for _move and _pos commands.
With AXIS_HOMED, the axis has been synchronized. The value is output for homing commands.
With INTERPOLATION_DONE, the setpoint interpolation of the command is complete.

SYNCHRONIZING, DESYNCHRONIZING, SYNCHRONIZED, MODIFICATION_ACTIVE are not indicated on the axis.
With EXECUTED, the command has been successfully executed. The EXECUTED status can only be read if the 'commandId' is stored.
With ABORTED, execution of the command has been aborted. The ABORTED status can only be read if the 'commandId' is stored.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

'commandId' is used to specify the unique system-wide ID of the command for which the status is to be called.

1.3.48.4 _getStateOfAxisCommand

This function returns the execution state of a motion command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'commandIdState'
With ACTIVE, the command is being executed.
With NOT_EXISTENT, the 'commandId' is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for the synchronous start.
With EXECUTED, execution of the command is complete.
With ABORTED, execution of the command has been aborted.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.48.5 _getStateOfMotionBuffer

This function returns the status of the axis command queue.

Return value: StructRetMotionBuffer

StructRetMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumMotionBuffer-State	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':
EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':
Specifies the number of entries in the command queue.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.48.6 _removeBufferedAxisCommandId

This function terminates the saving of the commandId and corresponding command status after command execution. The status for the commandId is no longer available in the system after completion of the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies which commandId and corresponding command status should no longer be available in the system after completion of the command.
With SPECIFIC_ID, a specific commandId and its command status is no longer available once the command has been completed.
With ALL_ID, all commandIds and their command status are no longer available once the command has been completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId'.

1.3.48.7 _resetMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.49 Positioning axis - Motion

1.3.49.1 _continue

This function continues the entire motion or a partial motion of the specified axis, if it was stopped with STOP and STOP_WITHOUT_ABORT in the 'stopMode' parameter.
Continuation of a partial motion is specified using the 'commandId' or the type of motion.
When continuing a motion, the dynamic parameters (e.g., velocity profile, acceleration) of the interrupted command are used.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

continueSpecification (optional)

Direction:	Input parameters
Data type:	EnumContinueSpecification
Parameter index:	2
System default:	ALL_AXIS_MOTION

EnumContinueSpecification

ALL_AXIS_MOTION (9)	Acts on all active motions
BASIC_MOTION (17)	Acts on the basic motion
SPECIFIC_AXIS_MOTION (125)	Acts on the specified motion
SUPERIMPOSED_MOTION (141)	Acts on the superimposed motions

Specifies the motion that is to be continued.
With ALL_AXIS_MOTION, all motions are continued.
With BASIC_MOTION, the basic motion is continued.
With SPECIFIC_AXIS_MOTION, the motion of the command specified in the 'continueId' parameter is continued.
With SUPERIMPOSED_MOTION, the superimposed motion is continued.

continueId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	WHEN_COMMAND_DONE

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.49.2 _disableAxis

This function cancels the axis enables.
The states of the current drive and power enables can be accessed using system variables 'actorMonitoring.driveState' (enable for DRIVE) and 'actorMonitoring.power' (enable for POWER).
The position controller enable can also be cancelled for position-controlled axes. Its current value can be queried via 'servoMonitoring.controlState'.
The position controller enable is ignored for speed-controlled axes.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

disableMode (optional)

Direction:	Input parameters
Data type:	EnumEnableAxisMode
Parameter index:	2
System default:	ALL

EnumEnableAxisMode

ALL (8)	Enable all
DO_NOT_CHANGE (43)	Do not change enables
DRIVE (44)	Enable drive only
POWER (109)	Enable power only
BY_STW_BIT (357)	profiDrive protocol bit mask
STATE_MACHINE_CONTROL_BY_APPLICATION (415)	Evaluation and manipulation of the STW bits by the user program

Specifies resetting of the drive and power enables.
ALL means that the drive and power enables are cancelled.
DO_NOT_CHANGE means that the current enables in effect are not changed.
DRIVE means that the drive enable is cancelled. The power enable is not changed.
POWER means that the power enable is cancelled. The drive enable is not changed.
With BY_STW_BIT, the bits specified in the 'STWBitSet' parameter are canceled in the PROFIdrive protocol.
With STATE_MACHINE_CONTROL_BY_APPLICATION, the user program takes responsibility for evaluating and manipulating the STW bits.

servoControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

servoCommandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	4
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the tracking mode setting.
If ACTIVE, the axis will be set to tracking mode and the setpoints are tracked. With ACTIVE, no motion commands can be executed; this applies to all axis types. Motion monitoring, such as following error, will not be initiated.
In the special mode of the active position controller enable and the active tracking operation, values can be specified in the servo using, for example, 'servoSettings.additionalCommandValue', without the monitoring being initiated and so the position controller can be placed in operation and set independent of the IPO.

If INACTIVE, the tracking operation of the axis will be deactivated. With INACTIVE, motion commands can be executed. For virtual axes, the position controller enable is always set internally and the motion control will be enabled only for inactive tracking operation.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	8
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the setting of the force/pressure controller enable.
This parameter is only relevant for axes with pressure or force control.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

STWBitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	9
System default:	0

Specifies the values of Bit 0 - Bit 6, which are set in the PROFIdrive protocol.
The bits are inverted and ANDed with the existing control word.
The specified bits are reset.

Meaning of the bits in the PROFIdrive protocol:
Bit0 - OFF
Bit1 - Coast stop (OFF2)
Bit2 - Quick stop (OFF3)
Bit3 - Disable operation
Bit4 - Disable ramp generator
Bit5 - Freeze ramp generator
Bit6 - Disable setpoint

The parameter is only used with 'disableMode:=BY_STW_BIT' or 'enableMode:=BY_STW_BIT'.
In all other cases, the parameter is irrelevant.

1.3.49.3 _enableAxis

This function switches the axis enables.
The states of the current drive and power enables can be accessed using system variables 'actorMonitoring.driveState' (enable for DRIVE) and 'actorMonitoring.power' (enable for POWER).
The position controller enable must also be set for position-controlled axes. Its current value can be queried via 'servoMonitoring.controlState'.
The position controller enable is ignored for speed-controlled axes.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

enableMode (optional)

Direction:	Input parameters
Data type:	EnumEnableAxisMode
Parameter index:	2
System default:	ALL

EnumEnableAxisMode

ALL (8)	Enable all
DO_NOT_CHANGE (43)	Do not change enables
DRIVE (44)	Enable drive only
POWER (109)	Enable power only
BY_STW_BIT (357)	profiDrive protocol bit mask
STATE_MACHINE_CONTROL_BY_APPLICATION (415)	Evaluation and manipulation of the STW bits by the user program

Specifies the setting of the drive and power enables.
ALL means that the drive and power enables are set.
DO_NOT_CHANGE means that the current valid enables are not changed.
The axis remains in follow-up mode if all enables are not available.
DRIVE means that the drive enable is set. The power enable is not changed.
POWER means that the power enable is set. The drive enable is not changed.
With BY_STW_BIT, the bits specified in the STWBitSet parameter are set in the PROFIdrive protocol.

servoControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

servoCommandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	4
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the tracking mode setting.
If ACTIVE, the axis will be set to tracking mode and the setpoints are tracked. With ACTIVE, no motion commands can be executed; this applies to all axis types. Motion monitoring, such as following error, will not be initiated.
In the special mode of the active position controller enable and the active tracking operation, values can be specified in the servo using, for example, 'servoSettings.additionalCommandValue', without the monitoring being initiated and so the position controller can be placed in operation and set independent of the IPO.

If INACTIVE, the tracking operation of the axis will be deactivated. With INACTIVE, motion commands can be executed. For virtual axes, the position controller enable is always set internally and the motion control will be enabled only for inactive tracking operation.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	8
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the setting of the force/pressure controller enable.
This parameter is only relevant for axes with pressure or force control.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

STWBitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	9
System default:	0

Specifies the values of Bit 0 - Bit 6, which are set in the PROFIdrive protocol.
The bits are ORed with the existing control word. Thus, the specified bits will be set.

The bits set can be viewed in the SIMOTION SCOUT project navigator under:
Drives-> <Drive name> -> Diagnostics -> Control/status words

Meaning of the bits in the PROFIdrive protocol:
Bit0 - ON
Bit1 - No coast stop (no OFF2)
Bit2 - No quick stop (no OFF3)
Bit3 - Enable operation
Bit4 - Enable ramp generator
Bit5 - Unfreeze ramp generator
Bit6 - Enable setpoint

The parameter is only used with 'enableMode:=BY_STW_BIT' or 'disableMode:=BY_STW_BIT'.
In all other cases, the parameter is irrelevant.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisEnableMovingMode
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumAxisEnableMovingMode

DO_NOT_CHANGE (43)	Retain last setting
POSITION_CONTROLLED (180)	Enable for speed- and position-controlled mode
SPEED_CONTROLLED (181)	Enable for speed-controlled mode

Specifies whether the axis is enabled for position- or speed-controlled motions.
With POSITION_CONTROLLED, the enable is for position- and speed-controlled motions.
With SPEED_CONTROLLED, the enable is for speed-controlled motions. In this mode, the axis can move if the encoder fails and there is no error response.
With DO_NOT_CHANGE, the last mode setting is retained.

1.3.49.4 _getAxisStoppingData

This function calculates the deceleration distance and the deceleration time when velocity, acceleration, and jerk are specified at the start of the braking operation.

Return value: StructRetGetAxisStoppingData

StructRetGetAxisStoppingData
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
stoppingDistance	Stopping distance	LREAL	-
stoppingTime	Stopping time	LREAL	-

Description of the return value:

0 -
No error

1 -
Illegal command parameter

2 -
Illegal range specification in command parameters

3 -
Command aborted

4 -
Unknown command

5 -
Command cannot be executed due to current object status

6 -
Command aborted due to termination of user task

7 -
Command rejected due to suspension of command interpretation of the addressed technology object

8 -
Command aborted due to full command buffer

9 -
Insufficient memory

10 -
A connection to a technology object required for this operation does not exist

11 -
No object configuration

12 -
The error cannot be reset due to its configuration

13 -
Axis is not homed

14 -
Measurement job not possible on virtual axis

15 -
Ambiguous 'commandId'

16 -
Command not implemented

17 -
Read access denied

18 -
Write access denied

19 -
Command argument not supported

20 -
The cam has already been interpolated and cannot be manipulated

21 -
The interpolation condition was violated

22 -
The programmed jerk is 0

23 -
The alarm to be deleted is not active

24 -
The command is not possible on a virtual axis

25 -
Synchronized start of this command is not possible

26 -
Higher-level command was aborted because it is not permitted by the active command

27 -
Timeout during communication with the drive

28 -
Actual values are not valid

29 -
This command cannot be executed when velocity control is active

30 -
This command cannot be executed when position control is active

31 -
This command cannot be executed in torque-reduced operation or during travel to fixed limit stop

32 -
This command can only be executed when force or pressure control is active

33 -
This command cannot be executed when force or pressure control is active

34 -
This command can only be executed when pressure limiting is active

35 -
Master values are not valid

36 -
Slave values are not valid

37 -
No slave value can be defined for a master value

38 -
No master value can be defined for a slave value

39 -
This command cannot be executed when synchronous operation is inactive

40 -
This command cannot be executed because of a synchronization error.

41 -
This command cannot be executed when gearing or camming is active

42 -
This command cannot be executed when camming is inactive

43 -
This command can only be used for an interpolated cam

44 -
This command cannot be executed when pressure limiting is active

45 -
There are not enough interpolation points for interpolation of the cam

46 -
Specified path location cannot be reached due to kinematic limitations

47 -
Path axis values are not valid

48 -
Maximum number of active commands exceeded

49 -
Command only possible on a technology object local to the CPU

10000 (greater or equal to) internal error

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

initialVelocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the velocity at the start of the braking operation.

initialAcceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Specifies the acceleration that occurs at the start of the braking operation. When a positive acceleration value is specified, acceleration occurs; when a negative acceleration value is specified, deceleration occurs. The parameter is only relevant with velocity profiles with jerk limiting 'velocityProfile:=SMOOTH'.

negativeAccel

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the deceleration setpoint.

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the acceleration end jerk setpoint.

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the deceleration start jerk setpoint.

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the deceleration end jerk setpoint.

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	8
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

1.3.49.5 _homing

Function for homing an axis.
The dynamic parameters for homing are programmable and refer to all phases of the homing procedure.

The execution time of the _homing command for the absolute encoder adjustment, i.e. for homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER and homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION, can take several interpolation cycles.
The status of the command can be queried by querying the CommandID of the homing command using _getstateOfAxisCommand(), even when the axis is already in the 'homed' state.

Exceptions
The homing approach velocity can be programmed in the command. The homing approach velocity, homing entry velocity and the homing deceleration velocity are set during configuration.
The execution sequence for active homing ('homingMode:=ACTIVE_HOMING') is set during configuration.
The criteria for passive homing ('homingMode:=PASSIVE_HOMING') are set during configuration independently from the active homing.
An axis has the 'referenced' or 'homed' status when the axis coordinate system has been aligned with the homing signal. The status can be read from the 'positioningState.homed' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

homingMode (optional)

Direction:	Input parameters
Data type:	EnumHomingMode
Parameter index:	2
System default:	ACTIVE_HOMING

EnumHomingMode

ACTIVE_HOMING (5)	Homing operation in configured mode
DIRECT_HOMING (41)	Setting of current position value
ENABLE_OFFSET_OF_ABSOLUTE_ENCODER (49)	Absolute encoder offset
PASSIVE_HOMING (104)	Homing with the next zero mark
DIRECT_HOMING_RELATIVE (352)	Relative setting of current position value
SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION (385)	Absolute encoder adjustment with position specification from command

Specifies the type of homing.

With ACTIVE_HOMING, a homing operation is started. The homing mode is set during configuration.

With DIRECT_HOMING, the home position coordinates are set directly to the current axis coordinates; the command does not initiate an axis motion.

With DIRECT_HOMING_RELATIVE, the home position coordinates are set relative to the current axis coordinates; the command does not initiate an axis motion.

With ENABLE_OFFSET_OF_ABSOLUTE_ENCODER, the value of 'absHomingEncoder.abshift' offset is added to the offset already existing in the system with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=RELATIVE;

with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=ABSOLUTE', the value 'absHomingEncoder.absshift' is calculated in as a fixed offset.

The total offset is saved in the NVRAM and is available after the controller is disabled. Once a new project has been loaded in the controller, the saved offset is no longer available.

With 'homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION', the value in the 'homePosition' parameter is set as the current position and the resulting absolute encoder offset is calculated from this. This value is displayed in system variable 'absoluteEncoder.totalOffsetValue', and stored in the system as a retain variable. The value in the configuration data 'absHomingEncoder'-absshift is not changed.

With PASSIVE_HOMING, the next zero mark crossed by the axis when it moves after the '_homing' command is used as the home position. The homing command is active parallel to the motion. It remains active until the homing procedure is complete. It can be cancelled with _resetAxis or _disableAxis. If the homing command is issued, e.g. with a stationary axis, and before the motion command, the homing command should be programmed with 'mergeMode:=IMMEDIATELY' and 'nextCommand:=IMMEDIATELY' and the motion command programmed with 'mergeMode:=IMMEDIATELY' or 'mergeMode:=NEXT'.

If the homing command is issued when a motion command is active, it should be programmed with 'mergeMode:=IMMEDIATELY'.

homePositionType (optional)

Direction:	Input parameters
Data type:	EnumValueSpecification
Parameter index:	3
System default:	USER_DEFAULT

EnumValueSpecification

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the type of home position coordinates.
With DIRECT, the value set in the 'homePosition' parameter is used as the programmed home position coordinate.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultHoming.homePosition' is used.

homePosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Home position coordinates, the evaluation depends on the 'homePositionType' parameter.

With 'homePositionType:=DIRECT', the parameter is used as the new home position coordinate. In all other cases, the parameter is irrelevant.

During the setting of the parameter 'homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER', the parameter has no meaning.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity input for the homing approach velocity.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultHoming.hominggApproachVelocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Homing approach velocity, evaluation dependent on the parameter 'velocityType'.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.
The homing approach velocity is only relevant for ACTIVE_HOMING homingMode CAM_AND_ZM or CAM.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandHoming
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandHoming

AT_MOTION_START (13)	Start of interpolation
IMMEDIATELY (60)	Asynchronous commands
WHEN_AXIS_HOMED (157)	Axis has been homed
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_AXIS_HOMED, the transition takes place when the axis has been synchronized.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or the command is aborted.
WHEN_INTERPOLATION_DONE and WHEN_MOTION_DONE are only effective when homing-Mode := ACTIVE_HOMING.
In all other cases, the command transition takes place in accordance with WHEN_AXIS_HOMED.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.49.6 _move

The axis is accelerated or decelerated to the programmed velocity setpoint.
If a limited constant velocity phase is not specified (parameter 'moveTimeOut-Type:=WITHOUT_TIME_LIMIT'), the velocity is kept constant after the transition to the velocity set-point.
With limitation of the constant velocity phase (parameter 'moveTimeOutType:=WITH_TIME_LIMIT'), the velocity is decelerated to velocity setpoint 0 after the specified time has expired.

Return value: DINT

Description of the return value:

0 -
No error

1 -
Illegal command parameter

2 -
Illegal range specification in command parameters

3 -
Command aborted

4 -
Unknown command

5 -
Command cannot be executed due to current object status

6 -
Command aborted due to termination of user task

7 -
Command rejected due to suspension of command interpretation of the addressed technology object

8 -
Command aborted due to full command buffer

9 -
Insufficient memory

10 -
A connection to a technology object required for this operation does not exist

11 -
No object configuration

12 -
The error cannot be reset due to its configuration

13 -
Axis is not homed

14 -
Measurement job not possible on virtual axis

15 -
Ambiguous 'commandId'

16 -
Command not implemented

17 -
Read access denied

18 -
Write access denied

19 -
Command argument not supported

20 -
The cam has already been interpolated and cannot be manipulated

21 -
The interpolation condition was violated

22 -
The programmed jerk is 0

23 -
The alarm to be deleted is not active

24 -
The command is not possible on a virtual axis

25 -
Synchronized start of this command is not possible

26 -
Higher-level command was aborted because it is not permitted by the active command

27 -
Timeout during communication with the drive

28 -
Actual values are not valid

29 -
This command cannot be executed when velocity control is active

30 -
This command cannot be executed when position control is active

31 -
This command cannot be executed in torque-reduced operation or during travel to fixed limit stop

32 -
This command can only be executed when force or pressure control is active

33 -
This command cannot be executed when force or pressure control is active

34 -
This command can only be executed when pressure limiting is active

35 -
Master values are not valid

36 -
Slave values are not valid

37 -
No slave value can be defined for a master value

38 -
No master value can be defined for a slave value

39 -
This command cannot be executed when synchronous operation is inactive

40 -
This command cannot be executed because of a synchronization error.

41 -
This command cannot be executed when gearing or camming is active

42 -
This command cannot be executed when camming is inactive

43 -
This command can only be used for an interpolated cam

44 -
This command cannot be executed when pressure limiting is active

45 -
There are not enough interpolation points for interpolation of the cam

46 -
Specified path location cannot be reached due to kinematic limitations

47 -
Path axis values are not valid

48 -
Maximum number of active commands exceeded

49 -
Command only possible on a technology object local to the CPU

10000 (greater or equal to) internal error

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumDirection

BY_VALUE (24)	From the sign of the programmed position setpoint / velocity
EFFECTIVE (45)	Last programmed direction setpoint
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the direction of motion.
With BY_VALUE, the direction is derived from the sign of the velocity setpoint ('velocity' parameter).
With EFFECTIVE, the last programmed direction setpoint is used.
With POSITIVE and NEGATIVE, the direction depends on the axis coordinate system.
SHORTEST_WAY is not supported with this command. The command is aborted with alarm 30001 "Illegal command parameter".
With USER_DEFAULT, the default direction defined in system variable 'userDefaultDynamics.direction' is used.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	3
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
With RESULTING, the velocity is that which results when the current axis acceleration or deceleration is terminated with jerk control. The acceleration/deceleration is reduced to zero, taking into account the jerk. The mergeMode 'IMMEDIATELY' must be programmed as transition behavior.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

moveTimeOutType (optional)

Direction:	Input parameters
Data type:	EnumMoveTimeOut
Parameter index:	5
System default:	WITHOUT_TIME_LIMIT

EnumMoveTimeOut

WITHOUT_TIME_LIMIT (169)	Motion without time limit
WITH_TIME_LIMIT (171)	Motion with time limit

Defines the duration from the start of the constant velocity phase to the start of the deceleration phase.
WITHOUT_TIME_LIMIT means that the duration is not limited.
WITH_TIME_LIMIT means that the duration of the constant velocity phase is specified in the 'moveTimeOut' parameter.

moveTimeOut (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the duration of the constant velocity phase; the evaluation depends on the 'moveTimeOutType' parameter.
Only effective if parameter 'moveTimeOutType:=WITH_TIME_LIMIT'. The unit of the duration is specified during configuration of the axis.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.
The conditions AT_DECELERATION_START, WHEN_INTERPOLATION_DONE, and WHEN_MOTION_DONE only result in a transition when 'moveTimeOutType:=WITH_TIME_LIMIT' or if the command is aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	24
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.49.7 _pos

This function moves the axis to the programmed target position with a definable velocity profile.
The dynamic response parameters are programmable.
The response to the current motion is programmable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

direction (optional)

Direction:	Input parameters
Data type:	EnumDirection
Parameter index:	2
System default:	USER_DEFAULT

EnumDirection

BY_VALUE (24)	From the sign of the programmed position setpoint / velocity
EFFECTIVE (45)	Last programmed direction setpoint
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the direction of motion.
With an absolute position specification ('positioningMode:=ABSOLUTE'), the direction is directly derived from the target position to be approached for linear and rotary axes.
The following applies to modulo axes as well as to linear and rotary axes with relative positioning ('positioningMode:=RELATIVE'):

With BY_VALUE, the direction is determined from the sign of the position specification.
With EFFECTIVE, the last programmed direction setpoint is used.
With NEGATIVE, motion is in the negative direction relative to the axis coordinate system.
With POSITIVE, motion is in the positive direction relative to the axis coordinate system.

With SHORTEST_WAY, the shortest path is taken on modulo axes.
With SHORTEST_WAY and relative positioning ('positioningMode:=RELATIVE'), the direction is determined from the sign of the position specification for linear and rotary axes.
Further information can be found in the TO Axis, Electric/Hydraulic, External Encoder Function Manual, 'Possible direction specifications for the _pos() command' table

With USER_DEFAULT, the default direction defined in system variable 'userDefaultDynamics.direction' is used.

positioningMode (optional)

Direction:	Input parameters
Data type:	EnumPositioningMode
Parameter index:	3
System default:	ABSOLUTE

EnumPositioningMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified.

position

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the position, the evaluation depends on the 'positioningMode' parameter.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positi-veAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStart-Jerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerk-Type' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccel-StartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

blendingMode (optional)

Direction:	Input parameters
Data type:	EnumBlendingMode
Parameter index:	20
System default:	USER_DEFAULT

EnumBlendingMode

ACTIVE (4)	Blending
EFFECTIVE (45)	The last programmed blending value
INACTIVE (61)	No blending
USER_DEFAULT (149)	User default

Specifies blending mode.
With ACTIVE, the current motion is blended.
With EFFECTIVE, the last programmed blending specification is used.
With INACTIVE, no blending takes place.
With USER_DEFAULT, the current value programmed in the 'userDefaultPositioning.blendingMode' system variable is used.

Blending is a special form of linking the positioning motion specified in the command to the previous positioning motion, whereby the transition takes place at the target point of the current motion and the velocity setpoint specified in the motion commands for the particular motion is not violated at any time. This means:

- The current positioning motion is executed at the command velocity up to the target position.
- Exceptions: - If the velocity of the new motion command has the same sign but a smaller value than the velocity setpoint of the current motion, the axis decelerates to the new velocity before the current target position is reached. - In the event of a direction reversal, the previous motion is completed and the transition to the new motion takes place immediately at the target position. - If the path length of the new motion command is less than the required deceleration distance, the current motion is decelerated accordingly.
- If the velocity of the new command is greater, the velocity is increased after the transition to the new command, i.e. after the previous target position is reached. Active blending requires 'mergeMode' NEXT_MOTION or SEQUENTIAL and an adequate 'leading' program transition, i.e. the new motion to be activated with 'blending' must be known to the interpolator or motion control system at the start of deceleration in the previous command. If the blending command is not detected until later, blending is delayed accordingly.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	21
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	22
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	23
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.49.8 _runMotionInPositionLockedVelocityProfile

The axis moves according to a velocity profile that is specified in the cam function Velocity = f (position). Here, the position is an absolute axis position that is specified using the MotionIn interface.
The profile is started at the current MotionIn position.
The dynamic parameters for any required transition motions, e.g., to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the profile or the motion function for the axis motion.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	4
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	8
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	16
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	17
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	18
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	19
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	21
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.49.9 _runPositionBasedMotionIn

Activation of the motion interface on the axis with the position-based motion so that the values are taken into the setpoint generation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

reference (optional)

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2
System default:	0

Specifies the master object.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	3
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	17
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableMotionIn
Parameter index:	18
System default:	SEQUENTIAL

EnumMergeModeEnableMotionIn

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableMotionIn
Parameter index:	19
System default:	IMMEDIATELY

EnumNextCommandEnableMotionIn

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_COMMAND_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	20
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	22
System default:	TO_CONNECTION

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the master value is taken from the interface or as value.
With TO_CONNECTION, the MotionIn interface value is used.
With VALUE, the 'motionInType' parameter is evaluated.

motionInType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	23
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the master value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the master value is taken from system variable 'defaultMotionIn'.
With DIRECT, the master value is taken from command parameters ''positionIn', velocityIn' and 'accelerationIn'.
The parameter is only evaluated when valueReferenceType:=VALUE'.

positionIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	0.0

Specifies the position of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

velocityIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Specifies the velocity of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

accelerationIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	26
System default:	0.0

Specifies the acceleration of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

1.3.49.10 _runPositionLockedVelocityProfile

The axis moves according to a velocity profile that is specified in the cam function Velocity = f (position). The position in this case is an absolute axis position.
The profile is started at the current axis position.
The dynamic parameters for any required transition motions, e.g., to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the profile or the motion function for the axis motion.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	3
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	7
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	15
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	16
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	17
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	18
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	**Name**	**Data type**	**Unit**
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	20
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.49.11 _runTimeLockedPositionProfile

The axis runs the predefined functions in the specified cam as a position profile.
The axis moves along the profile from a selectable start point to the end.
The dynamic parameters for any required transition motions, e.g., to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the profile or the motion function for the axis motion.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the position profile is run.

profileDataMode (optional)

Direction:	Input parameters
Data type:	EnumPositioningMode
Parameter index:	4
System default:	ABSOLUTE

EnumPositioningMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Type of position specification.
With ABSOLUTE, the cam positions are approached according to the absolute values.
With RELATIVE, the cam positions are approached relative to the current position.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	**Name**	**Data type**	**Unit**
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.49.12 _runTimeLockedVelocityProfile

The axis moves according to a velocity profile that is specified as a cam function.
The axis moves along the profile from a selectable start point to the end.
The dynamic parameters for any required transition motions, e.g. to move towards the profile and to move away from the profile, can be programmed in the command.
The motion behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the profile or the motion function for the axis motion.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the velocity profile is run.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	4
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	8
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	16
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	17
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandProfile
Parameter index:	18
System default:	IMMEDIATELY

EnumNextCommandProfile

IMMEDIATELY (60)	Asynchronous commands
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached
AT_PROFILE_START (221)	Start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation. It is important to ensure that 'TypeOfAxis.DecodingConfig.behaviourAtTheEndOfProfile=MOVE_WITH_CONSTANT_SPEEd' is set as part of the configuration data, otherwise the function will not be able to transition.

With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	19
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	21
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

1.3.49.13 _runVelocityBasedMotionIn

This function activates the motion interface on the axis with the velocity-based motion so that the values can be taken over into the setpoint generation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference (optional)

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2
System default:	0

Specifies the master object.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	3
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	5
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	7
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	15
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeEnableMotionIn
Parameter index:	16
System default:	SEQUENTIAL

EnumMergeModeEnableMotionIn

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion. If a superimposed motion is already active, it is replaced.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableMotionIn
Parameter index:	17
System default:	IMMEDIATELY

EnumNextCommandEnableMotionIn

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_COMMAND_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	18
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingMode
Parameter index:	20
System default:	POSITION_CONTROLLED

EnumMovingMode

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
The movingMode parameter has no effect on the speed-controlled axis.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	21
System default:	TO_CONNECTION

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the master value is taken from the interface or as value.
With TO_CONNECTION, the MotionIn interface value is used.
With VALUE, the 'motionInType' parameter is evaluated.

motionInType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	22
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the master value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the master value is taken from system variable 'defaultMotionIn'.
With DIRECT, the master value is taken from command parameters 'velocityIn' and 'accelerationIn'.
The parameter is only evaluated when valueReferenceType:=VALUE'.

velocityIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	0.0

Specifies the velocity of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

accelerationIn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	0.0

Specifies the acceleration of the master value.
The parameter is only evaluated when 'valueReferenceType:=VALUE' and 'motionInType:=DIRECT'.

1.3.49.14 _setAxisSTW

The function makes it possible to specify selectable bits in control word 1 and control word 2 of the PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

STW1BitMask (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	2
System default:	0

Selection of the bits to be specified in control word 1 with this command.
The bits to be specified are selected in the form of a bit mask in 'STW1BitSet':
With 1, 'STW1.Bitx' is specified with this command.
With 0, 'STW1.Bitx' is not specified with this command.
The values of the bits are specified in 'STW1BitSet'.

STW1BitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	3
System default:	0

Specifies the values with the bits selected in parameter 'STW1BitMask'.

STW2BitMask (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	4
System default:	0

Selection of the bits to be changed in control word 2.
The selection is performed as a bit mask. The value (0 or 1) of the bit to be changed is specified in parameter 'STW2BitSet'.

STW2BitSet (optional)

Direction:	Input parameters
Data type:	UINT
Parameter index:	5
System default:	0

Specifies the values with the bits selected in parameter 'STW2BitMask'.

execution (optional)

Direction:	Input parameters
Data type:	EnumToCommandExecution
Parameter index:	6
System default:	TASK_EXECUTION

EnumToCommandExecution

TASK_EXECUTION (413)	Setting the STW bits in the task context of the user program
TO_EXECUTION (414)	Setting the STW bits for synchronizing with TO command processing

Task context for setting the bits in the drive protocol.
With TASK_EXECUTION, the bits are set in the task context of the user program.
With TO_EXECUTION, setting of the bits is synchronized with command processing in the technology object.

1.3.49.15 _stop

This command stops the entire motion or a partial motion of the specified axis using a programmed deceleration ramp. The _stop command affects only single-axis motions and cannot be used for synchronous operation motions.
The motion to be stopped can be interrupted or terminated.
Stopping of a partial motion is specified using either the 'commandId' or the type of motion.
An interrupted motion can be continued with the _continue command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

stopMode (optional)

Direction:	Input parameters
Data type:	EnumStopMode
Parameter index:	2
System default:	STOP_WITHOUT_ABORT

EnumStopMode

STOP_AND_ABORT (135)	Normal stop with abort
STOP_WITHOUT_ABORT (138)	Normal stop without abort

Specifies the stop behavior relative to the current command.
With STOP_AND_ABORT, the active motion command is aborted.
With STOP_WITHOUT_ABORT, the current motion is interrupted and can be continued with the _continue command.

stopSpecification (optional)

Direction:	Input parameters
Data type:	EnumStopSpecification
Parameter index:	3
System default:	ALL_AXIS_MOTION

EnumStopSpecification

ALL_AXIS_MOTION (9)	Acts on all active axis motions
BASIC_MOTION (17)	Acts on the basic axis motion
SPECIFIC_AXIS_MOTION (125)	Acts on a specific axis motion
SUPERIMPOSED_MOTION (141)	Acts on the superimposed axis motions

Specifies the motions for which the stop command is to take effect.
With ALL_AXIS_MOTION, the stop command acts on all active axis motions.
With BASIC_MOTION, only the main motion is stopped.
With SPECIFIC_AXIS_MOTION, the command acts on the command specified in the 'StopId' parameter.
With SUPERIMPOSED_MOTION, the superimposed motion is stopped.

stopId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide ID to stop the command with this 'commandId'. Commands with the standard ID 'stopId.SystemId_low:=0' and 'stopId.SystemId_high:=0' are not stopped.
This parameter is only active with 'stopSpecification:=SPECIFIC_AXIS_MOTION'.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	6
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	8
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	10
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	12
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	18
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeStop
Parameter index:	19
System default:	IMMEDIATELY

EnumMergeModeStop

AT_END_OF_COMMAND (2)	No significance
IMMEDIATELY (60)	Execute command immediately

Specifies when the command takes effect relative to the motion.
AT_END_OF_COMMAND is irrelevant.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	20
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	21
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingModeStopCommand
Parameter index:	23
System default:	CURRENT_MODE

EnumMovingModeStopCommand

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion
CURRENT_MODE (327)	Last set traversing mode

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
CURRENT_MODE means that motion is stopped in the last set traversing mode.
The 'movingMode' parameter has no effect on the speed-controlled axis.

1.3.49.16 _stopEmergency

This function stops the axis with a programmable stop mode.
If a motion command is active, it is aborted and cannot be continued with a _continue command. The axis is not switched to follow-up mode.
The axis is blocked for further motion commands; the status can be revoked with _reset Axis () or _disable Axis ().

Detailed information on this function can be found in the SIMOTION Motion Control, TO Axis, Electric/Hydraulic, External Encoder Function Manual.

Note
The _stopEmergency system function has the same behavior as the _MC_Stop function block.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

stopDriveMode (optional)

Direction:	Input parameters
Data type:	EnumStopDriveMode
Parameter index:	2
System default:	STOP_WITH_MAXIMAL_DECELERATION

EnumStopDriveMode

STOP_IN_DEFINED_TIME (136)	Stop axis using a time-dependent ramp
STOP_WITH_COMMAND_VALUE_ZERO (139)	Stop axis using zero setpoint
STOP_WITH_MAXIMAL_DECELERATION (140)	Stop axis using maximum deceleration
STOP_WITH_DYNAMIC_PARAMETER (326)	Stop axis using programmed dynamic values

Specifies the stop mode of the axis.
With STOP_IN_DEFINED_TIME the axis stops according to the time specified in the 'stopTime' parameter. The specified time is adhered to irrespective of the velocity.
With STOP_WITH_COMMAND_VALUE_ZERO, the axis is stopped using the emergency stop ramp in the controller. This is set during configuration. When the emergency stop ramp is generated (stopDriveMode:=STOP_WITH_COMMAND_VALUE_ZERO), the ramp generator in the servo is based on the 'dynamicData.positionTimeConstant' x velocity setpoint extrapolated actual position.
With STOP_WITH_MAXIMAL_DECELERATION, the axis is stopped via interpolation using the maximum dynamic axis values.
With STOP_WITH_DYNAMIC_PARAMETER, the axis is stopped with the dynamic parameters programmed in the command.

stopTimeType (optional)

Direction:	Input parameters
Data type:	EnumExtendedValue
Parameter index:	3
System default:	USER_DEFAULT

EnumExtendedValue

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of definition for stop time.
This parameter is only effective if 'stopDriveMode:=STOP_IN_DEFINED_TIME'.
With DIRECT, the value set in the 'stopTime' parameter is used as the programmed stop time.
With EFFECTIVE, the last programmed stop time is used.
With USER_DEFAULT, the default setting for the stop time specified in system variable 'userDefaultDynamics.stopTime' is used.

stopTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the stop time; the evaluation depends on the 'stopDriveMode' parameter.
It is only used when parameter 'stopDriveMode:=STOP_IN_DEFINED_TIME' and 'stopTimeType:=DIRECT'.
In all other cases, the parameter is irrelevant.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeStop
Parameter index:	5
System default:	IMMEDIATELY

EnumMergeModeStop

AT_END_OF_COMMAND (2)	No significance
IMMEDIATELY (60)	Execute command immediately

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
AT_END_OF_COMMAND is irrelevant.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommand
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommand

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

movingMode (optional)

Direction:	Input parameters
Data type:	EnumMovingModeStopCommand
Parameter index:	9
System default:	CURRENT_MODE

EnumMovingModeStopCommand

POSITION_CONTROLLED (180)	Position-controlled motion
SPEED_CONTROLLED (181)	Speed-controlled motion
CURRENT_MODE (327)	Last set traversing mode

Specifies whether motion is position- or speed-controlled.
POSITION_CONTROLLED means that motion is position-controlled.
SPEED_CONTROLLED means that motion is speed-controlled.
CURRENT_MODE means that motion is stopped in the last set traversing mode.
The movingMode parameter has no effect on the speed-controlled axis.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	10
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	12
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	18
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	20
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	22
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

1.3.50 Positioning axis - Pressure limitation

1.3.50.1 _disableForceLimiting

This command deactivates limitation of the force or pressure.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableForceLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableForceLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.50.2 _enableForceLimitingByCondition

This command causes switchover to force/pressure limiting when the switchover criterion defined in the command is reached.
The switchover criterion is checked in the servo. Switch-over criteria can include a position, force or pressure, a time, or a digital input.
The mode for determining the cam execution method is set via the parameter 'forceProfileMode'. The following possibilities are available:
- User default
- Force-time cam profile
- Force-position cam profile
- Value input by command parameter

The rise factor for transitions when entering and exiting the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

forceLimitingProfile (optional)

Direction:	Input parameters
Data type:	CamType
Parameter index:	2
System default:	0

Specifies the cam that defines the force/pressure profile for force/pressure limiting of the axis.

switchingPositionType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of position specification.

switchingPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position that is compared to the actual position of the axis.
This comparison is carried out depending on the 'switchingPositionType' parameter.
It is only used when 'switchingPositionType := DIRECT'.
In all other cases, the parameter is irrelevant.

positionCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	5
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

switchingForceType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	6
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of comparison pressure specification.

switchingForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the comparison pressure; the evaluation depends on the 'switchingForceType' parameter.
It is only used when 'switchingForceType := DIRECT'.
In all other cases, the parameter is irrelevant.

forceCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	8
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

forceSensorNumberType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	9
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specification type of the force/pressure measurement system to be used for the comparison force/pressure.

forceSensorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	10
System default:	1

Specifies the force/pressure measurement system to be used for the comparison pressure.
The encoder numbers of the force/pressure measurement system are assigned during configuration.
This command parameter refers to encoders configured under TypeOfAxis.NumberOfAdditionalSensors.AdditionalSensor_<forceSensorNumber>.
The encoder number transferred in the command parameter must be used for <forceSensorNumber>.
The number of encoders that can be used is set in the NumberOfAdditionalSensors configuration data.

switchingTimeType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	11
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of time data.

switchingTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the time; the evaluation depends on the 'switchingTimeType' parameter.
It is only used when 'switchingTimeType := DIRECT'.
In all other cases, the parameter is irrelevant.

switchingInputType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	13
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of digital input specification.

switchingInput (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	14
System default:	1

Specifies the digital input to be used for the comparison.
The number of the input is a reference to a list of inputs. This list can be found in the configuration data under TypeOfAxis.ControllerSwitchData. The size of the list is set by means of the NumberOfDigitalInputs configuration data.

switchingInputMode (optional)

Direction:	Input parameters
Data type:	EnumAxisUserDefaultHighLow
Parameter index:	15
System default:	USER_DEFAULT

EnumAxisUserDefaultHighLow

USER_DEFAULT (149)	User default
HIGH (264)	HIGH signal
LOW (265)	LOW signal

Specifies the expected switching status of the digital input.

switchingCondition (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition
Parameter index:	16
System default:	USER_DEFAULT

EnumAxisSwitchingCondition

USER_DEFAULT (149)	User default
CONDITION_1 (242)	Condition 1
CONDITION_2 (243)	Condition 2
CONDITION_1_OR_CONDITION_2 (244)	Condition 1 OR Condition 2
CONDITION_1_AND_CONDITION_2 (245)	Condition 1 AND Condition 2

Specifies the logical configuration of switching conditions 1 and 2.

switchingCondition_1 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_1
Parameter index:	17
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_1

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 1.

switchingCondition_2 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_2
Parameter index:	18
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_2

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 2.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	19
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	0.0

Specifies the rise factor for transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceControl
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandEnableForceControl

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_CONTROL_ENABLED (262)	When the switch-over condition is fulfilled
WHEN_PROFILE_DONE (263)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

forceExtrapolationType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceExtrapolationType
Parameter index:	24
System default:	NONE

EnumAxisForceExtrapolationType

LINEAR (72)	Linear extrapolation
NONE (93)	No extrapolation
POLYNOMIAL (368)	Cubic extrapolation

Type of extrapolation.
With NONE, there is no extrapolation.
With LINEAR, a linear extrapolation takes place between startForce and endForce. The values of the derivatives are not taken into account.
With POLYNOMIAL, a cubic extrapolation takes place between startForce and endForce, taking into account the specified derivatives.

startForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Starting point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

endForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	26
System default:	0.0

End point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

startForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	27
System default:	0.0

Derivative in the starting point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

endForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	28
System default:	0.0

Derivative in the end point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

functionLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	29
System default:	0.0

Specifies the length of the extrapolation curve as time.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

forceProfileMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceProfileModeConditionCommand
Parameter index:	30
System default:	USER_DEFAULT

EnumAxisForceProfileModeConditionCommand

USER_DEFAULT (149)	User default
FORCE_TIME (394)	Force-time cam profile
FORCE_POSITION (395)	Force-position cam profile
FORCE_VALUE (396)	Value input by command parameter

Type of setpoint specification for the force/pressure limiting of the axis.
With FORCE_TIME, the cam profile specified in the 'forceLimitingProfile' parameter is interpreted as a force-time profile.
With FORCE_POSITION, the cam profile specified in the 'forceLimitingProfile' parameter is interpreted as a force-distance profile.
With FORCE_VALUE, the setpoint is specified with the 'forceValueType' and 'forceValue' parameters.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceProfileModeConditionCommand'.

forceValueType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceValueConditionCommand
Parameter index:	31
System default:	USER_DEFAULT

EnumAxisForceValueConditionCommand

ACTUAL_VALUE (34)	Actual value of axis
DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of setpoint specification for the force/pressure limiting of the axis when 'forceProfileMode:=FORCE_VALUE'.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceValueConditionCommand'.
With ACTUAL, the current actual value of the axis is accepted.
With DIRECT, the setpoint is taken from the 'forceValue' parameter.

forceValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	32
System default:	0.0

Type of setpoint specification for the force/pressure limiting of the axis when 'forceValueType:=DIRECT'.

1.3.50.3 _enableForceLimitingValue

This command activates limiting of the force or pressure to a fixed value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

limitingValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the force/pressure limiting value.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	6
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.50.4 _enableMotionInPositionLockedForceLimitingProfile

This function activates the position-related force/pressure limitation profile at the interconnected MotionIn position.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	7
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.50.5 _enablePositionLockedForceLimitingProfile

This function activates force or pressure limiting with a position-related force/pressure limiting profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableForceLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_LIMITING_ACTIVATED (266)	When force/pressure limiting is activated
WHEN_FORCE_LIMIT_REACHED (267)	As soon as force/pressure is limited the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	6
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.50.6 _enableTimeLockedForceLimitingProfile

This function activates the force or pressure limiting with a time-related force/pressure limiting profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam at which the profile definition starts.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

derivativeLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceDerivativeLimitingMode
Parameter index:	7
System default:	USER_DEFAULT

EnumAxisForceDerivativeLimitingMode

USER_DEFAULT (149)	User default
WITHOUT_LIMITING (340)	No rise limiting
LIMITING_BY_USER_DEFAULT_VALUE (341)	Rise limiting with user default
LIMITING_BY_DIRECT_VALUE (342)	Rise limiting with function parameter

Type of rise factor specification.

derivativeLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the rise factor for force/pressure limiting value transitions. Evaluation depends on the 'derivativeLimitingMode' parameter.

1.3.50.7 _getForceControlDataSetParameter

This function reads the force or pressure-related data in the axis data set.

Return value: StructRetGetForceControlDataSetParameter

StructRetGetForceControlDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisForceControlDataSet	-

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceControllerTypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.50.8 _setForceControlDataSetParameter

This function overwrites the force or pressure-related data in the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisForceControlDataSet
Parameter index:	3

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceController-TypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Specifies the data to be used to overwrite the data set.
UNI_DIRECTION means that negative actual values cannot be measured.
BOTH_DIRECTION means that negative actual values can be measured.

1.3.51 Positioning axis - Pressure control

1.3.51.1 _disableVelocityLimiting

This command deactivates limitation of the velocity.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.51.2 _enableForceControlByCondition

This function causes switchover to force/pressure control when the switchover criterion defined in the command is reached.

The switch-over criterion check begins with a query in the servo. Switch-over criteria can include a position, force or pressure, a time, or a digital input.

The mode for determining the cam execution method is set via the parameter 'forceProfileMode'.

The following possibilities are available:

- User default
- Force-time cam profile
- Force-position cam profile
- Value input by command parameter

The rise factor for force setpoint transitions for entering and exiting the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

forceProfile (optional)

Direction:	Input parameters
Data type:	CamType
Parameter index:	2
System default:	0

Specifies the cam that defines the force/pressure profile for force/pressure control of the axis.

switchingPositionType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of position specification.

switchingPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position that is compared to the actual position of the axis.
This comparison is carried out depending on the 'switchingPositionType' parameter.
It is only used when 'switchingPositionType := DIRECT'.
In all other cases, the parameter is irrelevant.

positionCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	5
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

switchingForceType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	6
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of comparison pressure specification.

switchingForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the comparison pressure; the evaluation depends on the 'switchingForceType' parameter.
It is only used when 'switchingForceType := DIRECT'.
In all other cases, the parameter is irrelevant.

forceCompareMode (optional)

Direction:	Input parameters
Data type:	EnumAxisCompareMode
Parameter index:	8
System default:	USER_DEFAULT

EnumAxisCompareMode

USER_DEFAULT (149)	User default
GREATER_EQUAL (240)	Greater than or equal
LESS (241)	Less than

Type of compare operation.

forceSensorNumberType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	9
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specification type of the force/pressure measurement system to be used for the comparison force/pressure.

forceSensorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	10
System default:	1

Specifies the force/pressure measurement system to be used for the comparison pressure.
The encoder numbers of the force/pressure measurement system are assigned during configuration.
This command parameter refers to encoders configured under TypeOfAxis.NumberOfAdditionalSensors.AdditionalSensor_<forceSensorNumber>.
The encoder number transferred in the command parameter must be used for <forceSensorNumber>.
The number of encoders that can be used is set in the NumberOfAdditionalSensors configuration data.

switchingTimeType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	11
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of time data.

switchingTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the time; the evaluation depends on the 'switchingTimeType' parameter.
It is only used when 'switchingTimeType := DIRECT'.
In all other cases, the parameter is irrelevant.

switchingInputType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	13
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of digital input specification.

switchingInput (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	14
System default:	1

Specifies the digital input to be used for the comparison.
The number of the input is a reference to a list of inputs. This list can be found in the configuration data under TypeOfAxis.ControllerSwitchData. The size of the list is set by means of the NumberOfDigitalInputs configuration data.

switchingInputMode (optional)

Direction:	Input parameters
Data type:	EnumAxisUserDefaultHighLow
Parameter index:	15
System default:	USER_DEFAULT

EnumAxisUserDefaultHighLow

USER_DEFAULT (149)	User default
HIGH (264)	HIGH signal
LOW (265)	LOW signal

Specifies the expected switching status of the digital input.

switchingCondition (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition
Parameter index:	16
System default:	USER_DEFAULT

EnumAxisSwitchingCondition

USER_DEFAULT (149)	User default
CONDITION_1 (242)	Condition 1
CONDITION_2 (243)	Condition 2
CONDITION_1_OR_CONDITION_2 (244)	Condition 1 OR Condition 2
CONDITION_1_AND_CONDITION_2 (245)	Condition 1 AND Condition 2

Specifies the logical configuration of switching conditions 1 and 2.

switchingCondition_1 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_1
Parameter index:	17
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_1

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 1.

switchingCondition_2 (optional)

Direction:	Input parameters
Data type:	EnumAxisSwitchingCondition_2
Parameter index:	18
System default:	USER_DEFAULT

EnumAxisSwitchingCondition_2

USER_DEFAULT (149)	User default
FORCE_CONDITION (246)	Force/pressure
POSITION_CONDITION (247)	Position
TIME_CONDITION (248)	Time
INPUT_CONDITION (249)	Input
FORCE_AND_POSITION (250)	Force/pressure and position
FORCE_AND_TIME (251)	Force/pressure and time
FORCE_AND_INPUT (252)	Force/pressure and input
FORCE_OR_POSITION (253)	Force/pressure or position
FORCE_OR_TIME (254)	Force/pressure or time
FORCE_OR_INPUT (255)	Force/pressure or input
POSITION_AND_TIME (256)	Position and time
POSITION_AND_INPUT (257)	Position and input
POSITION_OR_TIME (258)	Position or time
POSITION_OR_INPUT (259)	Position or input
TIME_AND_INPUT (260)	Time and input
TIME_OR_INPUT (261)	Time or input

Definition of switching condition 2.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	19
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	0.0

Specifies the rise factor for force/pressure setpoint value transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableForceControl
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandEnableForceControl

IMMEDIATELY (60)	Immediate command transition
WHEN_FORCE_CONTROL_ENABLED (262)	When the switch-over condition is fulfilled
WHEN_PROFILE_DONE (263)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingMode (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	24
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Activates a velocity limitation when a transition condition is active.

velocityLimitingValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	25
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of velocity limiting value specification.

velocityLimitingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	26
System default:	100.0

Specifies the velocity limiting value. Evaluation depends on the 'velocityLimitingValueType' parameter.

forceExtrapolationType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceExtrapolationType
Parameter index:	27
System default:	NONE

EnumAxisForceExtrapolationType

LINEAR (72)	Linear extrapolation
NONE (93)	No extrapolation
POLYNOMIAL (368)	Cubic extrapolation

Type of extrapolation.
With NONE, there is no extrapolation.
With LINEAR, a linear extrapolation takes place between startForce and endForce. The values of the derivatives are not taken into account.
With POLYNOMIAL, a cubic extrapolation takes place between startForce and endForce, taking into account the specified derivatives.

startForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	28
System default:	0.0

Starting point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

endForce (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	29
System default:	0.0

End point for determining the extrapolation parameters.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

startForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	30
System default:	0.0

Derivative in the starting point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

endForceDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	31
System default:	0.0

Derivative in the end point for determining the extrapolation parameters.
The parameter is used only when 'forceExtrapolationType := POLYNOMIAL'.

functionLength (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	32
System default:	0.0

Specifies the length of the extrapolation curve as time.
The parameter is only used when 'forceExtrapolationType := LINEAR' or 'forceExtrapolationType := POLYNOMIAL'.

velocityLimitingValueMode (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingValueMode
Parameter index:	33
System default:	SET_VALUE

EnumAxisVelocityLimitingValueMode

ACTUAL_VALUE (34)	Velocity limiting to current actual value
SET_VALUE (386)	Velocity limiting to programmed value
SETPOINT_VALUE (938)	Velocity limiting to current setpoint

Specifies whether the velocity at the reversing point should be accepted as the velocity limiting value.
With ACTUAL_VALUE, the velocity is limited to the current actual velocity value.
With SET_VALUE, the velocity is limited to the value specified in the parameters 'velocityLimitingValueType' and 'velocityLimitingValue'.
With SETPOINT_VALUE, the velocity is limited to the current velocity setpoint.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	34
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

forceProfileMode (optional)

Direction:	Input parameters
Data type:	EnumAxisForceProfileModeConditionCommand
Parameter index:	35
System default:	USER_DEFAULT

EnumAxisForceProfileModeConditionCommand

USER_DEFAULT (149)	User default
FORCE_TIME (394)	Force-time cam profile
FORCE_POSITION (395)	Force-position cam profile
FORCE_VALUE (396)	Value input by command parameter

Type of setpoint specification for the force/pressure control of the axis.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceProfileModeConditionCommand'.
With FORCE_TIME, the cam profile specified in the 'forceProfile' parameter is interpreted as a force-time profile.
With FORCE_POSITION, the cam profile specified in the 'forceProfile' parameter is interpreted as a force-distance profile.
With FORCE_VALUE, the setpoint is specified with the 'forceValueType' and 'forceValue' parameters.

forceValueType (optional)

Direction:	Input parameters
Data type:	EnumAxisForceValueConditionCommand
Parameter index:	36
System default:	USER_DEFAULT

EnumAxisForceValueConditionCommand

ACTUAL_VALUE (34)	Actual value of axis
DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of setpoint specification for the force/pressure control of the axis when 'forceProfileMode:=FORCE_VALUE'.
With USER_DEFAULT, the setting is taken from the system variable 'userDefaultForceControl.forceValueConditionCommand'.
With ACTUAL, the current actual value of the axis is accepted.
With DIRECT, the setpoint is taken from the 'forceValue' parameter.

forceValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	37
System default:	0.0

Type of setpoint specification for the force/pressure control of the axis when 'forceValueType:=DIRECT'.

1.3.51.3 _enableMotionInPositionLockedVelocityLimitingProfile

This function activates the velocity limiting with a position-related MotionIn limitation profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the velocity profile for axis limiting.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	7
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.51.4 _enablePositionLockedVelocityLimitingProfile

This function activates the velocity limiting with a position-related limitation profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the velocity profile for axis limiting.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	6
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.51.5 _enableTimeLockedVelocityLimitingProfile

This function activates the velocity limiting with a time-related limitation profile.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the velocity profile for axis limiting.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam at which the profile definition starts.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	7
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.51.6 _enableVelocityLimitingValue

This function activates velocity limiting at a definable value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

limitingValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the velocity limiting value.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableLimitingValue
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnableLimitingValue

IMMEDIATELY (60)	Immediate command transition
WHEN_LIMITING_COMMAND_ACTIVATED (302)	When limiting is activated
WHEN_LIMIT_REACHED (303)	As soon as limiting occurs the first time

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

velocityLimitingValueMode (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingValueMode
Parameter index:	6
System default:	SET_VALUE

EnumAxisVelocityLimitingValueMode

ACTUAL_VALUE (34)	Velocity limiting to current actual value
SET_VALUE (386)	Velocity limiting to programmed value
SETPOINT_VALUE (938)	Velocity limiting to current setpoint

Specifies whether the velocity at the reversing point should be accepted as the velocity limiting value.
With ACTUAL_VALUE, the velocity is limited to the current actual velocity value.
With SET_VALUE, the velocity is limited to value specified in the 'limitingValue' parameter.
With SETPOINT_VALUE, the velocity is limited to the current velocity setpoint.

velocityLimitingDirection (optional)

Direction:	Input parameters
Data type:	EnumAxisVelocityLimitingDirection
Parameter index:	7
System default:	BOTH

EnumAxisVelocityLimitingDirection

ACTUAL (7)	Velocity limiting is effective in the direction of the current actual velocity
BOTH (20)	Velocity limiting is effective in both directions
NEGATIVE (85)	Velocity limiting is effective in the negative direction
POSITIVE (107)	Velocity limiting is effective in the positive direction

Specifies the direction for the effectiveness of velocity limiting.
With ACTUAL, the velocity limiting is active in the direction of the current actual velocity.
With BOTH, the velocity limiting is active in both directions.
With NEGATIVE, the velocity limiting is active in the negative direction.
With POSITIVE, the velocity limiting is active in the positive direction.

1.3.51.7 _getForceControlDataSetParameter

This function reads the force or pressure-related data in the axis data set.

Return value: StructRetGetForceControlDataSetParameter

StructRetGetForceControlDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisForceControlDataSet	-

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceControllerTypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.51.8 _runMotionInPositionLockedForceProfile

The axis executes the predefined function in the cam as a force-path profile. Here, the position is an absolute axis position that is specified using the MotionIn interface.
The profile is started at the current MotionIn position.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

reference

Direction:	Input parameters
Data type:	MasterType
Parameter index:	2

Specifies the master object.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	3

Specifies the cam that defines the force/pressure profile.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	4
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the rise factor for force/pressure setpoint value transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	6
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.51.9 _runPositionLockedForceProfile

The axis executes the predefined functions in the cam as a force-time profile. The position in this case is an absolute axis position.
The profile is started at the current axis position.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise factor for force/pressure setpoint value transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	5
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.51.10 _runTimeLockedForceProfile

The axis executes the predefined functions in the specified cam as a force/pressure profile.
The profile is executed from a selectable start point to the end.
The rise factor for force/pressure setpoint transitions for any required transition motions to move towards the profile and to move away from the profile can be programmed in the command.
The behavior at the end of the profile is set during axis configuration.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

profile

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the force/pressure profile.

startTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	0.0

Start time (= domain) in the cam from which the force/pressure profile is run.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	4
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	100.0

Specifies the rise factor for force/pressure setpoint transitions. Evaluation depends on the 'derived-CommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	6
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceProfile
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandForceProfile

IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted
AT_PROFILE_START (221)	At start of interpolation
BY_PROFILE_END (222)	End of profile generation

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed or stopped in another way.
With AT_PROFILE_START, the transition takes place at the start of interpolation.
With BY_PROFILE_END, the transition takes place at the end of profile generation.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.51.11 _setForceCommandValue

The force or pressure setpoint specified in the command is set on the axis.
The rise factor for force or pressure setpoint transitions for any required transition motions can be programmed in the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the force or pressure setpoint.

derivedCommandValueType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	3
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of rise factor specification.

derivedCommandValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise factor for force/pressure setpoint transitions. Evaluation depends on the 'derivedCommandValueType' parameter.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeForceTimeProfile
Parameter index:	5
System default:	SEQUENTIAL

EnumMergeModeForceTimeProfile

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandForceValue
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandForceValue

AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	When command is finished or aborted

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the actual force/pressure value is within the configured force/pressure window.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.51.12 _setForceControlDataSetParameter

This function overwrites the force or pressure-related data in the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisForceControlDataSet
Parameter index:	3

StructAxisForceControlDataSet
Force/pressure-related axis parameters

Structure	Name	Data type	Unit
forceControllerDifference	Monitoring of the system deviation of the force controller	StructForceControllerDifference	-
forceControllerData	Data of force controller	StructForceControllerData2	-
AdditionalSensorNumber	Measuring system number	StructAdditionalSensorNumber	-

StructForceControllerDifference
Monitoring of the system deviation of the force control loop

Structure	Name	Data type	Unit
enable	Activation of the function	EnumYesNo	-
maxForceTolerance	Maximum permissible system deviation	LREAL	-

StructForceControllerData2
Setting of force controller

Structure	Name	Data type	Unit
outputLimits	Controlled variable limiting of force controller	StructOutputLimits	-
PID_Controller	Setting of PID controller	StructPID_Controller	-
controllerType	Controller type	EnumForceController	-
typeOfSensorData	Effective direction of manipulated variable	EnumForceController-TypeOfSensorData	-
invertControlValue	Inversion of manipulated variable of force controller	EnumYesNo	-
filterForceControl	PT2 filter at the pressure control output	StructFilterForceControl	-

StructAdditionalSensorNumber
Measuring system number

Structure	Name	Data type	Unit
additionalSensorNumber	Measuring system number	DINT	-

StructOutputLimits
Limits of manipulated variable of force controller

Structure	Name	Data type	Unit
min	Minimum manipulated variable of force controller	LREAL	-
max	Maximum manipulated variable of force controller	LREAL	-

StructPID_Controller
Setting of PID controller

Structure	Name	Data type	Unit
kp	Proportional component	LREAL	-
ki	Integral component	LREAL	-
kd	Derivative component	LREAL	-
decayTime	Delay time for the derivative component	LREAL	-
enableAntiWindup	Activation of anti-windup	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kpc	Precontrol weighting	LREAL	%

StructFilterForceControl
PT2 filter at the pressure control output

Structure	Name	Data type	Unit
enable	Activation of filter	EnumYesNo	-
filterType	Type of filter	EnumForceController-FilterType	-
t1	Time constant T1 of PT2 filter	LREAL	-
t2	Time constant T1 of PT2 filter	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumForceController

PID (1)	PID controller

EnumForceControllerTypeOfSensorData

UNI_DIRECTION (1)	Measured values in one direction only
BOTH_DIRECTION (2)	Measured values in both directions

EnumForceControllerFilterType

PT2 (909)	PT2

Specifies the data to be used to overwrite the data set.
UNI_DIRECTION means that negative actual values cannot be measured.
BOTH_DIRECTION means that negative actual values can be measured.

1.3.52 Positioning axis - Hydraulics

1.3.52.1 _disableQFAxis

This function activates the enables and grants the axis access to the valve.
The function returns an error if another axis is already accessing the valve.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

controlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	2
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

commandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the position controller follow-up mode setting.
ACTIVE switches the axis to follow-up mode.
With ACTIVE, no motion commands can be executed; this applies to all axis types.
INACTIVE switches the axis out of follow-up mode.
With INACTIVE, motion commands can be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	7
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the pressure controller enable setting.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

QOutput (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputMode
Parameter index:	8
System default:	DISABLE

EnumDisableQFAxisOutputMode

DISABLE (42)	Enable valve access
DO_NOT_CHANGE (43)	Do not change access

Specifies the reserving and enabling of access to the Q valve.
With DISABLE, access to the Q valve is enabled.
With DO_NOT_CHANGE, no change takes place.

QOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputEnableMode
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
RESET (304)	Cancel enable

Specifies the Q valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With RESET, the enable is canceled.

QOutputValueSetMode (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputSetMode
Parameter index:	10
System default:	DO_NOT_CHANGE

EnumQFAxisOutputSetMode

DO_NOT_CHANGE (43)	Do not change replacement value
SET (178)	Set replacement value

Specifies the setting of a replacement value for the Q valve if a value is not output by an axis.
With DO_NOT_CHANGE, no change takes place.
With SET, the replacement value is set.

QOutputValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the replacement value; the evaluation depends on the 'QOutputValueSetMode' parameter.
It is only used when parameter 'QOutputValueSetMode:=SET'.
In all other cases, the parameter is irrelevant.

QOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	12
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'QOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

QOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the rise limiting value.

FOutput (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputMode
Parameter index:	14
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputMode

DISABLE (42)	Enable valve access
DO_NOT_CHANGE (43)	Do not change access

Specifies the reserving and enabling of access to the pressure limiting valve.
With DO_NOT_CHANGE, no change takes place.
With DISABLE, access to the pressure limiting valve is enabled.

FOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumDisableQFAxisOutputEnableMode
Parameter index:	15
System default:	DO_NOT_CHANGE

EnumDisableQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
RESET (304)	Cancel enable

Specifies the pressure limiting valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With RESET, the enable is canceled.

FOutputValueSetMode (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputSetMode
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumQFAxisOutputSetMode

DO_NOT_CHANGE (43)	Do not change replacement value
SET (178)	Set replacement value

Specifies the setting of a replacement value for the pressure limiting valve if a value is not output by an axis.
With DO_NOT_CHANGE, no change takes place.
With SET, the replacement value is set.

FOutputValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the replacement value; the evaluation depends on the 'FOutputValueSetMode' parameter.
It is only used when parameter 'FOutputValueSetMode:=SET'.
In all other cases, the parameter is irrelevant.

FOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	18
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'FOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

FOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the rise limiting value.

1.3.52.2 _enableQFAxis

This function activates the enables and grants the axis access to the valve.
The function returns an error if another axis is already accessing the valve.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

controlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	2
System default:	ACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the activation of the setpoint path in the servo.
With ACTIVE, the setpoint path is activated.
With INACTIVE, the setpoint path is not activated. If this is active, it will be deactivated.
The setpoint path should also be activated when switching to the Speed specification mode.

commandToActualMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	3
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the position controller follow-up mode setting.
ACTIVE switches the axis to follow-up mode.
With ACTIVE, no motion commands can be executed; this applies to all axis types.
INACTIVE switches the axis out of follow-up mode.
With INACTIVE, motion commands can be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

forceControlMode (optional)

Direction:	Input parameters
Data type:	EnumActiveInactive
Parameter index:	7
System default:	INACTIVE

EnumActiveInactive

ACTIVE (4)	Active
INACTIVE (61)	Inactive

Specifies the pressure controller enable setting.
ACTIVE means that the enable is set.
INACTIVE means that the enable is not set. If it is active, it is reset.

QOutput (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputMode
Parameter index:	8
System default:	DO_NOT_CHANGE

EnumQFAxisOutputMode

DO_NOT_CHANGE (43)	Do not change access
ENABLE (48)	Request valve access

Specifies the reserving and enabling of access to the Q valve.
With DISABLE, access to the Q valve is enabled.
With DO_NOT_CHANGE, no change takes place.
With ENABLE, access to the Q valve is reserved.

QOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputEnableMode
Parameter index:	9
System default:	DO_NOT_CHANGE

EnumQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
SET (178)	Set enable

Specifies the Q valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With SET, the enable is set.

QOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	10
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'QOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

QOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the rise limiting value.

FOutput (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputMode
Parameter index:	12
System default:	DO_NOT_CHANGE

EnumQFAxisOutputMode

DO_NOT_CHANGE (43)	Do not change access
ENABLE (48)	Request valve access

Parameter for reserving and enabling access to the pressure limitation valve.
With DO_NOT_CHANGE, no change takes place.
With ENABLE, access to the pressure limitation valve is reserved.

FOutputEnable (optional)

Direction:	Input parameters
Data type:	EnumQFAxisOutputEnableMode
Parameter index:	13
System default:	DO_NOT_CHANGE

EnumQFAxisOutputEnableMode

DO_NOT_CHANGE (43)	Do not change enables
SET (178)	Set enable

Specifies the pressure limiting valve enable setting.
With DO_NOT_CHANGE, no change takes place.
With SET, the enable is set.

FOutputMaxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	14
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value in parameter 'FOutputMaxDerivative' is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

FOutputMaxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the rise limiting value.

movingMode (optional)

Direction:	Input parameters
Data type:	EnumAxisEnableMovingMode
Parameter index:	16
System default:	DO_NOT_CHANGE

EnumAxisEnableMovingMode

DO_NOT_CHANGE (43)	Retain last setting
POSITION_CONTROLLED (180)	Enable for speed- and position-controlled mode
SPEED_CONTROLLED (181)	Enable for speed-controlled mode

Specifies whether the axis is enabled for position- or speed-controlled motions.
With POSITION_CONTROLLED, the enable is for position- and speed-controlled motions.
With SPEED_CONTROLLED, the enable is for speed-controlled motions. In this mode, the axis can move if the encoder fails and there is no error response.
With DO_NOT_CHANGE, the last mode setting is retained.

1.3.52.3 _getQFAxisDataSetParameter

This function reads an axis data set.

Return value: StructRetGetQFAxisDataSetParameter

StructRetGetQFAxisDataSetParameter
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructQFAxisDataSet	-

StructQFAxisDataSet
Parameters of an axis data set

Structure	Name	Data type	Unit
dynamicQFData	Dynamic parameters of the control loop	StructAxisDynamicQFData	-
invertQ	Inversion of the Q output value	StructAxisInvertQOutput	-
invertSetPoint	Inversion of the Q output value before the characteristic curve	StructAxisInvertSetPointHydraulicType	-

StructAxisDynamicQFData
Dynamic parameters of the control loop

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant for velocity loop	LREAL	s
QOutputTimeConstant	Equivalent time constant for hydraulic controlled system	LREAL	s

StructAxisInvertQOutput
Inversion of the Q output value

Structure	Name	Data type	Unit
invSetPoint	Activation status	EnumYesNo	-

StructAxisInvertSetPointHydraulicType
Inversion of the Q output value before the characteristic curve

Structure	Name	Data type	Unit
invert	Activation status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.52.4 _setQFAxisDataSetParameter

This function overwrites an axis data set.

Note
The change of the configuration data using the _setQFAxisDataSetParameter command is not displayed in the online view of the expert list.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructQFAxisDataSet
Parameter index:	3

StructQFAxisDataSet
Parameters of an axis data set

Structure	Name	Data type	Unit
dynamicQFData	Dynamic parameters of the control loop	StructAxisDynamicQFData	-
invertQ	Inversion of the Q output value	StructAxisInvertQOutput	-
invertSetPoint	Inversion of the Q output value before the characteristic curve	StructAxisInvertSetPointHydraulicType	-

StructAxisDynamicQFData
Dynamic parameters of the control loop

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant for velocity loop	LREAL	s
QOutputTimeConstant	Equivalent time constant for hydraulic controlled system	LREAL	s

StructAxisInvertQOutput
Inversion of the Q output value

Structure	Name	Data type	Unit
invSetPoint	Activation status	EnumYesNo	-

StructAxisInvertSetPointHydraulicType
Inversion of the Q output value before the characteristic curve

Structure	Name	Data type	Unit
invert	Activation status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the data to be used to overwrite the data set.

1.3.52.5 _setQFAxisFCharacteristics

This system function activates the characteristic curve for a pressure limiting valve. The characteristic curve is defined in a cam that describes the actual pressure value as a function of the valve position. The valve position is specified as a percentage. The value specified in 'maxOutputVoltage' is equivalent to 100 percent.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

characteristics

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the valve characteristic curve.

maxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	3
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value set in the 'maxDerivative' parameter is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.FOutput' is used.

maxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise limiting value; the evaluation depends on the 'maxDerivativeType' parameter.

1.3.52.6 _setQFAxisQCharacteristics

Activates the characteristic curve for a Q valve. The characteristic curve is defined in a cam that describes the axis velocity as a function of the valve position.
The valve position is specified as a percentage. The value specified in 'maxOutputVoltage' is equivalent to 100 percent.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

characteristics

Direction:	Input parameters
Data type:	CamType
Parameter index:	2

Specifies the cam that defines the valve characteristic curve.

maxDerivativeType (optional)

Direction:	Input parameters
Data type:	EnumValueType
Parameter index:	3
System default:	USER_DEFAULT

EnumValueType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the setting of a rise limiting value.
With DIRECT, the value set in the 'maxDerivative' parameter is used as the limiting value.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultQFAxis.maxDerivative.QOutput' is used.

maxDerivative (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the rise limiting value; the evaluation depends on the 'maxDerivativeType' parameter.

1.3.53 Positioning axis - Compensation and superimposition

1.3.53.1 _disableAxisAdditiveTorque

This function deactivates an additive torque setpoint that was activated with '_enableAxisAdditiveTorque'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.2 _disableAxisTorqueLimitNegative

This function deactivates a negative torque limiting that was activated with '_enableAxisTorqueLimitNegativeIn'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.3 _disableAxisTorqueLimitPositive

This function deactivates a positive torque limiting that was activated with '_enableAxisTorqueLimitPositiveIn'.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.4 _disableMonitoringOfEncoderDifference

This function deactivates encoder monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

1.3.53.5 _disableMovingToEndStop

With the system function _disableMovingToEndStop (), the monitoring of the 'travel to a fixed stop' is deactivated at the same time as an axis motion activated by a motion command, the axis is traversed/moved farther in position-regulated mode.

The command _disableMovingToEndStop () does not work with active clamping, i.e. if the clamping torque has already been reached.

In this case, the command _disableMovingToEndStop () is aborted with the alarm 30002. The clamping itself remains active.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableClamping
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableClamping

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command does not take place until the command is completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	**Name**	**Data type**	**Unit**
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId' for which the command status should be tracked.

1.3.53.6 _disableTorqueLimiting

The "Disable torque limiting" function is used to disable one of the superimposed torque limiting motion commands.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDisableTorqueLimiting
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandDisableTorqueLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command does not take place until the command is completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.53.7 _enableAxisAdditiveTorque

This function activates an additive torque setpoint.
The torque is specified cyclically via the 'AdditiveTorque' interconnection.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the torque value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'additiveTorqueType' parameter is evaluated.

additiveTorqueType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the torque value is be taken from the command or from a system variable.
With DEFAULT_VALUE, the torque value is taken from system variable 'defaultAdditiveTorque'.
The parameter is only evaluated when 'referenceType:=VALUE'.
With DIRECT, the torque value is taken from command parameter 'torqueLimit'.

additiveTorque (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the additive torque.
The parameter is only evaluated when 'referenceType:=VALUE' and 'additiveTorqueType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.8 _enableAxisTorqueLimitNegative

This function activates a negative torque limiting which is specified in the extended drive protocol in accordance with the speed controller.
The limiting values are specified cyclically via the 'TorqueLimitNegative' interconnection.

For active B+/B- torque limitation resulting from the _enableAxisTorqueLimitPositive or _enableAxisTorqueLimitNegative command, the following monitoring will be deactivated:

- Following error monitoring
- The velocity error monitoring using reference model
- The time limitations for positioning monitoring and standstill monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the limiting value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'torqueLimitType' parameter is evaluated.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the limit value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the limiting value is taken from system variable 'defaultTorqueLimitNegative'.
The parameter is only evaluated when 'referenceType:=VALUE'.
With DIRECT, the limiting value is taken from command parameter 'torqueLimit'.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the negative limiting value.
The parameter is only evaluated when 'referenceType:=VALUE' and 'torqueLimitType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.9 _enableAxisTorqueLimitPositive

This function activates a positive torque limiting, which is specified in the extended drive protocol in accordance with the speed controller.
The limiting values are specified cyclically via the 'TorqueLimitPositive' interconnection.

For active B+/B- torque limitation resulting from the _enableAxisTorqueLimitPositive or _enableAxisTorqueLimitNegative command, the following monitoring will be deactivated:

- Following error monitoring
- The velocity error monitoring using reference model
- The time limitations for positioning monitoring and standstill monitoring.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

valueReferenceType (optional)

Direction:	Input parameters
Data type:	EnumAxisValueReferenceType
Parameter index:	2
System default:	VALUE

EnumAxisValueReferenceType

TO_CONNECTION (324)	Interface value
VALUE (325)	Programmed value

Setting indicating whether the limiting value is taken from the interface or as value.
With TO_CONNECTION, the interface value is used.
With VALUE, the 'torqueLimitType' parameter is evaluated.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDefaultValueDirect
Parameter index:	3
System default:	DEFAULT_VALUE

EnumDefaultValueDirect

DEFAULT_VALUE (2)	Default value from system variable
DIRECT (40)	Command parameter

Setting indicating whether the limit value is to be taken from the command or from a system variable.
With DEFAULT_VALUE, the limiting value is taken from system variable 'defaultTorqueLimitPositive'.
With DIRECT, the limiting value is taken from command parameter 'torqueLimit'.
The parameter is only evaluated when 'referenceType:=VALUE'.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the positive limit value.
The parameter is only evaluated when 'referenceType:=VALUE' and 'torqueLimitType:=DIRECT'.

behaviorInvalidInterface (optional)

Direction:	Input parameters
Data type:	EnumLastValidInterfaceValueDefaultValue
Parameter index:	5
System default:	LAST_VALID_INTERFACE_VALUE

EnumLastValidInterfaceValueDefaultValue

LAST_VALID_INTERFACE_VALUE (1)	Last valid value
DEFAULT_VALUE (2)	Replacement value

Setting, whether with active but invalid interconnection value, the replacement value or the last valid value should be taken.
With LAST_VALID_INTERFACE_VALUE, the last valid value from the interface is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.10 _enableMonitoringOfEncoderDifference

This function activates the monitoring system for monitoring the maximum permissible difference between the measuring systems specified in the command.
An alarm is generated if the difference is exceeded. Furthermore, the system variable 'sensorMonitoring.slippageTolerance' is set.
In sensorMonitoring.slippageTolerance, LIMIT_EXCEEDED is displayed if the specified actual encoder value difference is exceeded during the command _enableMonitoringOfEncoderDifference.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

referenceEncoderType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	2
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Specifies which measuring system is to be used as the reference system.
With ACTUAL, the measuring system that is currently active is used as the reference system.
With DIRECT, the measuring system specified in the 'referenceEncoder' parameter is used.

referenceEncoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	1

Specifies the difference between two measuring systems.
The parameter is only evaluated if the value DIRECT was transferred in the 'referenceEncoderType' parameter.

encoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	4
System default:	1

Specifies the measuring system for which the difference to the reference encoder is monitored.

maximalEncoderDifferenceType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	5
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Determines how the maximum permissible measuring system difference is specified.
With DIRECT, the value in parameter 'maximalSensorDifference' is used.
With ACTUAL, the value of system variable 'sensorMonitoring.maximalSensorDifference' is used.

maximalEncoderDifference (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the maximum permissible measuring system difference. The parameter is only evaluated if the value DIRECT was transferred in the 'maximalEncoderDifferenceType' parameter.

1.3.53.11 _enableMovingToEndStop

This function is used to activate the monitoring of travel to fixed endstop in parallel to an axis motion activated using a motion command and the clamping torque when the fixed endstop has been reached. This process is known as 'clamping'. Whether a following error or evaluation of the axis torque is used to detect that the fixed endstop is reached is set during configuration.
The _enableMovingToEndStop () function can be used to switch over the clamping torque, even while the clamping is active.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

clampingValueType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	2
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of the clamping value.
With DIRECT, the value set in the 'clampingValue' parameter is used as the programmed clamping value.
With EFFECTIVE, the last programmed clamping value is used.
With USER_DEFAULT, the default clamping value defined in system variable 'userDefaultClamping.clampingValue' is used. This default can be modified by entering a value in the 'clampingValue' parameter. The value of the parameter is interpreted as a percentage.

clampingValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	100.0

Clamping value (torque or force), the evaluation depends on the 'clampingValueType' parameter.
It is only used with parameter 'clampingValueType := USER_DEFAULT' or 'clampingValueType := DIRECT'.
In all other cases, the parameter is irrelevant.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableClamping
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableClamping

IMMEDIATELY (60)	Immediate command transition
WHEN_ENDSTOP_REACHED (231)	When clamping value is reached
WHEN_FUNCTION_DISABLED (232)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_ENDSTOP_REACHED, the transition to the next command takes place when the clamping value is reached.
With WHEN_FUNCTION_DISABLED, the transition to the next command does not take place until the command is completed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique system-wide 'commandId' for which the command status should be tracked.

torqueLimitUnit (optional)

Direction:	Input parameters
Data type:	EnumTorqueLimitUnitType
Parameter index:	7
System default:	DEFAULT_UNIT

EnumTorqueLimitUnitType

DEFAULT_UNIT (268)	Unit for the TO
TORQUE (269)	Torque

Reference point for limit.
With DEFAULT_UNIT, the force or torque relates to the load side.
With TORQUE, the limiting value is interpreted as torque relating to the drive side.

1.3.53.12 _enableTorqueLimiting

The "Enable torque limiting" function is used to enable torque limiting in parallel to motion. This is effective immediately.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

torqueLimitType (optional)

Direction:	Input parameters
Data type:	EnumDirectEffectiveUserDefault
Parameter index:	2
System default:	USER_DEFAULT

EnumDirectEffectiveUserDefault

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of specification of the limiting value.
With DIRECT, the value set in the 'torqueLimit' parameter is used as the programmed value.
With EFFECTIVE, the last programmed torque limit value is used.
With USER_DEFAULT, the default torque limit value defined in system variable 'userDefaultTorque-Limiting.torqueLimit' is used. This default can be modified by entering a value in the 'torqueLimit' parameter. The value of the parameter is interpreted as a percentage.

torqueLimit (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	100.0

Desired axis torque limit value.
This parameter is evaluated independently of the 'torqueLimitType' parameter.
torqueLimitType -> DIRECT: Specifies the value as torque or force; the evaluation depends on the 'torqueLimitUnit' parameter.
torqueLimitType -> USER_DEFAULT: If this parameter is not specified, the default torque limit value defined in system variable 'userDefaultTorqueLimiting.torqueLimit' is used.
If a value is specified, the entry is interpreted as a percentage relative to the torque reduction defined in the 'userDefaultTorqueLimiting.torqueLimit' system variable.
torqueLimitType -> EFFECTIVE: Parameter 'torqueLimit' is not evaluated.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableTorqueLimiting
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandEnableTorqueLimiting

IMMEDIATELY (60)	Immediate command transition
WHEN_FUNCTION_DISABLED (232)	When command is finished or aborted
WHEN_TORQUELIMIT_REACHED (233)	As soon as torque is limited
WHEN_TORQUELIMIT_GONE (234)	As soon as torque limiting is deactivated

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_TORQUELIMIT_REACHED, the transition to the next command takes place as soon as the torque is limited (the torque limit setting is reached in drive). This state is derived from the PROFIBUS status word 'MeldW' (PZW 5), bit 1 (M < Mx) of the drive.
With WHEN_TORQUELIMIT_GONE, the transition to the next command takes place as soon as the torque limiting has been reached once and the torque limiting has been exited. This state is derived from the PROFIBUS status word 'MeldW' (PZW 5), bit 1 (M < Mx) of the drive.
With WHEN_FUNCTION_DISABLED, the transition to the next command does not take place until the command is aborted. The command can be aborted by calling the '_disableTorqueLimiting' or '_resetAxis' functions in another task.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

torqueLimitUnit (optional)

Direction:	Input parameters
Data type:	EnumTorqueLimitUnitType
Parameter index:	7
System default:	DEFAULT_UNIT

EnumTorqueLimitUnitType

DEFAULT_UNIT (268)	Unit for the TO
TORQUE (269)	Torque

Reference of limiting values during programming.
With DEFAULT_UNIT, the force or torque relates to the load side. The force or torque relates to the load side. The gear ratio is always taken into account here.
With TORQUE, the limiting value is interpreted as torque relating to the drive side. A gear ratio is not taken into account.

1.3.53.13 _getAxisDataSetParameter

This function reads an axis data set.

Return value: StructRetReadGetAxisDataSet

StructRetReadGetAxisDataSet
Return value with error code and data set

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
dataSetData	Data of data set	StructAxisDataSetReadWrite	-

StructAxisDataSetReadWrite
Parameters of an axis data set

Structure	Name	Data type	Unit
ControllerStruct	Parameters of the controller	StructControllerType	-
DynamicData	Dynamic characteristic values of the cascade control system	StructDynamicData	-
DynamicComp	Parameters for dynamic compensation of the control loop	StructDynamicComp	-
ProcessModel	Parameters of process model	StructProcessModel	-
DynamicFollowing	Parameters of the dynamic following error monitoring	StructDynamicFollowing	-
ControllerDynamic	Parameters of the reference model monitoring	StructControllerDynamic	-
EncoderNumber	Measuring system number	StructEncoderNumber	-
Gear	Ratio of the load gearbox	StructGear	-
ClampingMonitoring	Parameters for setting the torque monitoring	StructClampingMonitoring	-

StructControllerType
Parameters of the controller

Structure	Name	Data type	Unit
conType	Controller type	EnumAxisController-Type	-
PV_Controller	Parameters for the P controller with precontrol	StructPVController	-
PD_Controller	Parameters for the PD controller	StructPDController	-
PID_Controller	Parameters for the PID controller	StructPIDController	-

StructDynamicData
Dynamic characteristic values of the cascade control system

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant of the velocity control loop	LREAL	s
torqueTimeConstant	Equivalent time constant for torque control loop	LREAL	s

StructDynamicComp
Parameters for dynamic compensation of the control loop

Structure	Name	Data type	Unit
enable	Activate dynamic compensation	EnumYesNo	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructProcessModel
Parameters of the process model (PT2)

Structure	Name	Data type	Unit
ks	Transfer ratio of process	LREAL	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructDynamicFollowing
Parameters of the dynamic following error monitoring

Structure	Name	Data type	Unit
enable	Activation of the dynamic following error monitoring	EnumYesNo	-
minVelocity	Velocity value for the start of the rise, proportional to velocity, of the maximum permissible following error	LREAL	m/s
minPositionTolerance	Maximum permissible following error at velocities below the start of the rise, proportional to velocity, of the characteristic curve	LREAL	m
maxPositionTolerance	Maximum permissible following error at maximum velocity	LREAL	m
warningLimit	Warning limit of following error monitoring	LREAL	%

StructControllerDynamic
Parameters of the reference model monitoring

Structure	Name	Data type	Unit
enable	Activation of the reference model monitoring	EnumYesNo	-
maxVeloTolerance	Maximum velocity tolerance	LREAL	%

StructEncoderNumber
Measuring system number

Structure	Name	Data type	Unit
encoderNumber	Measuring system number	DINT	-

StructGear
Load gearing ratio

Structure	Name	Data type	Unit
numFactor	Numerator for load gearing	LREAL	-
denFactor	Denominator for load gearing	LREAL	-

StructClampingMonitoring
Parameters for setting the torque monitoring

Structure	Name	Data type	Unit
recognitionMode	Activation mode for torque monitoring	EnumRecogniti-onMode	-
followingErrorDevia-tion	Specifies the required following error for detection of endstop	LREAL	m
positionTolerance	Specifies the permis-sible deviation of the actual value from the setpoint in the clamped state	LREAL	m

StructPVController
Parameters for the P controller with precontrol

Structure	Name	Data type	Unit
enableDSC	Activation of DSC	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kv	P controller gain	LREAL	rps
kpc	Precontrol weighting	LREAL	%
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

StructPDController
Parameters for the PD controller

Structure	Name	Data type	Unit
kp	P controller gain	LREAL	rps
kd	Gain of DT1 element	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s

StructPIDController
Parameters for the PID controller

Structure	Name	Data type	Unit
preCon	Activate precontrol	EnumYesNo	-
kpc	Precontrol factor	LREAL	%
kp	P component gain	LREAL	rps
ki	I component gain	LREAL	-
kd	D component gain	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s
enableAntiWindup	I component limit	EnumYesNo	-
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

EnumAxisControllerType

NODEF (0)	No significance
DIRECT (1)	Control only
PD (2)	PID controller
PV (3)	PV controller
PID (4)	PID controller
PID_ACTUAL (5)	PID controller with actual value-dependent D-component

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumRecognitionMode

DO_NOT_CLAMP (0)	No detection
CLAMP_BY_FOLLOWING_ERROR_DEVIATION (1)	By following error
CLAMP_WHEN_TORQUE_LIMIT_REACHED (2)	By force moment/torque

EnumBalanceFilterMode

OFF (0)	Balancing filter not active
MODE_1 (1)	Balancing filter active
MODE_2 (2)	Expanded balancing filter active

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value dataSet:

Data read from data set.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies data set to be read.

1.3.53.14 _homing

Function for homing an axis.
The dynamic parameters for homing are programmable and refer to all phases of the homing procedure.

The execution time of the _homing command for the absolute encoder adjustment, i.e. for homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER and homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION, can take several interpolation cycles.
The status of the command can be queried by querying the CommandID of the homing command using _getstateOfAxisCommand(), even when the axis is already in the 'homed' state.

Exceptions
The homing approach velocity can be programmed in the command. The homing approach velocity, homing entry velocity and the homing deceleration velocity are set during configuration.
The execution sequence for active homing ('homingMode:=ACTIVE_HOMING') is set during configuration.
The criteria for passive homing ('homingMode:=PASSIVE_HOMING') are set during configuration independently from the active homing.
An axis has the 'referenced' or 'homed' status when the axis coordinate system has been aligned with the homing signal. The status can be read from the 'positioningState.homed' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

homingMode (optional)

Direction:	Input parameters
Data type:	EnumHomingMode
Parameter index:	2
System default:	ACTIVE_HOMING

EnumHomingMode

ACTIVE_HOMING (5)	Homing operation in configured mode
DIRECT_HOMING (41)	Setting of current position value
ENABLE_OFFSET_OF_ABSOLUTE_ENCODER (49)	Absolute encoder offset
PASSIVE_HOMING (104)	Homing with the next zero mark
DIRECT_HOMING_RELATIVE (352)	Relative setting of current position value
SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION (385)	Absolute encoder adjustment with position specification from command

Specifies the type of homing.

With ACTIVE_HOMING, a homing operation is started. The homing mode is set during configuration.

With DIRECT_HOMING, the home position coordinates are set directly to the current axis coordinates; the command does not initiate an axis motion.

With DIRECT_HOMING_RELATIVE, the home position coordinates are set relative to the current axis coordinates; the command does not initiate an axis motion.

With ENABLE_OFFSET_OF_ABSOLUTE_ENCODER, the value of 'absHomingEncoder.abshift' offset is added to the offset already existing in the system with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=RELATIVE;

with the setting 'absHomingEncoder.setOffsetOffAbsoluteEncoder:=ABSOLUTE', the value 'absHomingEncoder.absshift' is calculated in as a fixed offset.

The total offset is saved in the NVRAM and is available after the controller is disabled. Once a new project has been loaded in the controller, the saved offset is no longer available.

With 'homingMode:=SET_OFFSET_OF_ABSOLUTE_ENCODER_BY_POSITION', the value in the 'homePosition' parameter is set as the current position and the resulting absolute encoder offset is calculated from this. This value is displayed in system variable 'absoluteEncoder.totalOffsetValue', and stored in the system as a retain variable. The value in the configuration data 'absHomingEncoder'-absshift is not changed.

With PASSIVE_HOMING, the next zero mark crossed by the axis when it moves after the '_homing' command is used as the home position. The homing command is active parallel to the motion. It remains active until the homing procedure is complete. It can be cancelled with _resetAxis or _disableAxis. If the homing command is issued, e.g. with a stationary axis, and before the motion command, the homing command should be programmed with 'mergeMode:=IMMEDIATELY' and 'nextCommand:=IMMEDIATELY' and the motion command programmed with 'mergeMode:=IMMEDIATELY' or 'mergeMode:=NEXT'.

If the homing command is issued when a motion command is active, it should be programmed with 'mergeMode:=IMMEDIATELY'.

homePositionType (optional)

Direction:	Input parameters
Data type:	EnumValueSpecification
Parameter index:	3
System default:	USER_DEFAULT

EnumValueSpecification

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Specifies the type of home position coordinates.
With DIRECT, the value set in the 'homePosition' parameter is used as the programmed home position coordinate.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultHoming.homePosition' is used.

homePosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Home position coordinates, the evaluation depends on the 'homePositionType' parameter.

With 'homePositionType:=DIRECT', the parameter is used as the new home position coordinate. In all other cases, the parameter is irrelevant.

During the setting of the parameter 'homingMode:=ENABLE_OFFSET_OF_ABSOLUTE_ENCODER', the parameter has no meaning.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumVelocity
Parameter index:	5
System default:	USER_DEFAULT

EnumVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default
RESULTING (312)	Resulting velocity at end of acceleration

Type of velocity input for the homing approach velocity.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultHoming.hominggApproachVelocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.
RESULTING has no effect here and is interpreted the same as 'USER_DEFAULT'.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Homing approach velocity, evaluation dependent on the parameter 'velocityType'.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.
The homing approach velocity is only relevant for ACTIVE_HOMING homingMode CAM_AND_ZM or CAM.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	7
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefaultDynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	9
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	11
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	13
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	15
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	17
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefaultDynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumProfile
Parameter index:	19
System default:	USER_DEFAULT

EnumProfile

EFFECTIVE (45)	Last programmed velocity profile
PARABOLIC (103)	Parabolic velocity profile / not available
SINUSOIDAL (122)	Sinusoidal velocity profile / not available
SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
With EFFECTIVE, the last programmed velocity profile is used.
Parameters TRAPEZOIDAL, PARABOLIC, SINUSOIDAL, and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefaultDynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeMode
Parameter index:	20
System default:	SEQUENTIAL

EnumMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach
SUPERIMPOSED_MOTION_MERGE (142)	Superimpose

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.
With SUPERIMPOSED_MOTION_MERGE, the command is superimposed immediately on the current motion.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandHoming
Parameter index:	21
System default:	IMMEDIATELY

EnumNextCommandHoming

AT_MOTION_START (13)	Start of interpolation
IMMEDIATELY (60)	Asynchronous commands
WHEN_AXIS_HOMED (157)	Axis has been homed
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	End of setpoint interpolation
WHEN_MOTION_DONE (163)	Position reached

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_AXIS_HOMED, the transition takes place when the axis has been synchronized.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the positioning window is reached or the command is aborted.
WHEN_INTERPOLATION_DONE and WHEN_MOTION_DONE are only effective when homing-Mode := ACTIVE_HOMING.
In all other cases, the command transition takes place in accordance with WHEN_AXIS_HOMED.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	22
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.53.15 _redefinePosition

This function sets the coordinate system of an axis. The position value can be defined as an absolute value or as a relative position offset. The setpoint or the actual value can be set to the specified value. The other value is adjusted, taking into account the current difference and the following error.
The dynamic parameters can also be set for a virtual axis.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

redefineSpecification (optional)

Direction:	Input parameters
Data type:	EnumRedefineSpecification
Parameter index:	2
System default:	COMMAND_VALUE

EnumRedefineSpecification

COMMAND_VALUE (28)	Setpoint
ACTUAL_VALUE (34)	Actual value
VIRTUAL_AXIS (151)	Setpoint for a virtual axis
COMMAND_VALUE_BASIC_MOTION (310)	Setpoint for main motion
COMMAND_VALUE_SUPERIMPOSED_MOTION (311)	Setpoint for superimposed motion

Specifies the assignment destination for the new position.
With COMMAND_VALUE, the value in position is set as the new position setpoint depending on 'redefineMode', and the actual position is adjusted.
With ACTUAL_VALUE, the value in position is set as the new actual position value depending on 'redefineMode', and the position setpoint is adjusted accordingly.
With VIRTUAL_AXIS, the target position of a virtual axis is set in addition to the setpoints for velocity and acceleration.
The values (for s, v, a) are used as start values for the motion command.
The values take effect on the virtual axis when a motion command is started. When the motion starts, the new virtual axis setpoints are calculated according to the settings.

redefineMode (optional)

Direction:	Input parameters
Data type:	EnumRedefineMode
Parameter index:	3
System default:	ABSOLUTE

EnumRedefineMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Specifies the type of assignment for the new position.
With ABSOLUTE, the values are assigned as absolute positions.
With RELATIVE, the values are added to the existing position (offset).
With RELATIVE, the response is the same for 'redefineSpecification := COMMAND_VALUE' and 'redefineSpecification := ACTUAL_VALUE'.

position (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Position specification.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the current velocity when 'redefineSpecification := VIRTUAL_AXIS'.
In all other cases, the parameter is ignored.

acceleration (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the current acceleration when 'redefineSpecification := VIRTUAL_AXIS'.
In all other cases, the parameter is ignored.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	7
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command takes place after the new position has been set.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.16 _setAndGetEncoderValue

This function returns the difference between the specified measuring systems or synchronizes two measuring systems.
During synchronization, it is only possible to change the value of a measuring system if it is not currently being used for position control.

Return value: StructRetEncoderValue

StructRetEncoderValue
Return value with error code and difference

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
encoderDifference	Difference of specified measuring systems	LREAL	-

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value encoderDifference:

The difference of the specified measuring systems is indicated in 'encoderDifference'.

Parameter:

axis

Direction:	Input parameters
Data type:	PosAxis
Parameter index:	1

Specifies the technology object of the 'posAxis' type or a variable of the 'PosAxis' type on which the command is to be executed.

mode (optional)

Direction:	Input parameters
Data type:	EnumSetNoSet
Parameter index:	2
System default:	SET

EnumSetNoSet

SET (178)	Set value
NO_SET (179)	Do not set value

Specifies whether the actual value of the measuring system that is specified in the 'encoder' parameter should be set.
With SET, the actual value of the measuring system that is specified in the 'encoder' parameter is set.
With NO_SET, the actual value of the measuring system that is specified in the 'encoder' parameter is not set.

encoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	1

Specifies the measuring system to be selected or synchronized.
The encoder number is assigned during configuration.
The valid range is [1, 8]

referenceEncoderType (optional)

Direction:	Input parameters
Data type:	EnumActualDirect
Parameter index:	4
System default:	ACTUAL

EnumActualDirect

ACTUAL (7)	Actual value
DIRECT (40)	Value entry

Specifies the reference measuring system.
With ACTUAL, the active measuring system is used.
With DIRECT, the measuring system specified in the 'referenceEncoder' parameter is used.

referenceEncoder (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	5
System default:	1

Specifies the reference measuring system.
The encoder number is assigned during configuration.
The valid range is [1, 8]

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Transition condition to the next command of the program execution.
With IMMEDIATELY, the transition to the next command takes place immediately.
With WHEN_COMMAND_DONE, the transition to the next command takes place after changeover to the new measuring system.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.17 _setAxisDataSetActive

This function activates the axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be activated.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.53.18 _setAxisDataSetParameter

This function overwrites an axis data set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

dataSetNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	2
System default:	1

Specifies the data set to be overwritten.

dataSetData

Direction:	Input parameters
Data type:	StructAxisDataSetReadWrite
Parameter index:	3

StructAxisDataSetReadWrite
Parameters of an axis data set

Structure	Name	Data type	Unit
ControllerStruct	Parameters of the controller	StructControllerType	-
DynamicData	Dynamic characteristic values of the cascade control system	StructDynamicData	-
DynamicComp	Parameters for dynamic compensation of the control loop	StructDynamicComp	-
ProcessModel	Parameters of process model	StructProcessModel	-

Structure	Name	Data type	Unit
DynamicFollowing	Parameters of the dynamic following error monitoring	StructDynamicFollowing	-
ControllerDynamic	Parameters of the reference model monitoring	StructControllerDynamic	-
EncoderNumber	Measuring system number	StructEncoderNumber	-
Gear	Ratio of the load gearbox	StructGear	-
ClampingMonitoring	Parameters for setting the torque monitoring	StructClampingMonitoring	-

StructControllerType
Parameters of the controller

Structure	Name	Data type	Unit
conType	Controller type	EnumAxisControllerType	-
PV_Controller	Parameters for the P controller with precontrol	StructPVController	-
PD_Controller	Parameters for the PD controller	StructPDController	-
PID_Controller	Parameters for the PID controller	StructPIDController	-

StructDynamicData
Dynamic characteristic values of the cascade control system

Structure	Name	Data type	Unit
positionTimeConstant	Equivalent time constant for position control loop	LREAL	s
velocityTimeConstant	Equivalent time constant of the velocity control loop	LREAL	s
torqueTimeConstant	Equivalent time constant for torque control loop	LREAL	s

StructDynamicComp
Parameters for dynamic compensation of the control loop

Structure	Name	Data type	Unit
enable	Activate dynamic compensation	EnumYesNo	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructProcessModel
Parameters of the process model (PT2)

Structure	Name	Data type	Unit
ks	Transfer ratio of process	LREAL	-
T1	First time constant	LREAL	s
T2	Second time constant	LREAL	s

StructDynamicFollowing
Parameters of the dynamic following error monitoring

Structure	Name	Data type	Unit
enable	Activation of the dynamic following error monitoring	EnumYesNo	-
minVelocity	Velocity value for the start of the rise, proportional to velocity, of the maximum permissible following error	LREAL	m/s
minPositionTolerance	Maximum permissible following error at velocities below the start of the rise, proportional to velocity, of the characteristic curve	LREAL	m
maxPositionTolerance	Maximum permissible following error at maximum velocity	LREAL	m
warningLimit	Warning limit of following error monitoring	LREAL	%

StructControllerDynamic
Parameters of the reference model monitoring

Structure	Name	Data type	Unit
enable	Activation of the reference model monitoring	EnumYesNo	-
maxVeloTolerance	Maximum velocity tolerance	LREAL	%

StructEncoderNumber
Measuring system number

Structure	Name	Data type	Unit
encoderNumber	Measuring system number	DINT	-

StructGear
Load gearing ratio

Structure	Name	Data type	Unit
numFactor	Numerator for load gearing	LREAL	-
denFactor	Denominator for load gearing	LREAL	-

StructClampingMonitoring
Parameters for setting the torque monitoring

Structure	Name	Data type	Unit
recognitionMode	Activation mode for torque monitoring	EnumRecogniti-onMode	-
followingErrorDevia-tion	Specifies the required following error for detection of endstop	LREAL	m
positionTolerance	Specifies the permis-sible deviation of the actual value from the setpoint in the clamped state	LREAL	m

StructPVController
Parameters for the P controller with precontrol

Structure	Name	Data type	Unit
enableDSC	Activation of DSC	EnumYesNo	-
preCon	Activation of precontrol	EnumYesNo	-
kv	P controller gain	LREAL	rps
kpc	Precontrol weighting	LREAL	%
balanceFilterMode	Specifies the balan-cing filter	EnumBalanceFilter-Mode	%

StructPDController
Parameters for the PD controller

Structure	Name	Data type	Unit
kp	P controller gain	LREAL	rps
kd	Gain of DT1 element	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s

StructPIDController
Parameters for the PID controller

Structure	Name	Data type	Unit
preCon	Activate precontrol	EnumYesNo	-
kpc	Precontrol factor	LREAL	%
kp	P component gain	LREAL	rps
ki	I component gain	LREAL	-
kd	D component gain	LREAL	-
decayTime	Time constant of DT1 element	LREAL	s
enableAntiWindup	I component limit	EnumYesNo	-
balanceFilterMode	Specifies the balancing filter	EnumBalanceFilterMode	%

EnumAxisControllerType

NODEF (0)	No significance
DIRECT (1)	Control only
PD (2)	PID controller
PV (3)	PV controller
PID (4)	PID controller
PID_ACTUAL (5)	PID controller with actual value-dependent D-component

EnumYesNo

NO (91)	No
YES (173)	Yes

EnumRecognitionMode

DO_NOT_CLAMP (0)	No detection
CLAMP_BY_FOLLOWING_ERROR_DEVIATION (1)	By following error
CLAMP_WHEN_TORQUE_LIMIT_REACHED (2)	By force moment/torque

EnumBalanceFilterMode

OFF (0)	Balancing filter not active
MODE_1 (1)	Balancing filter active
MODE_2 (2)	Expanded balancing filter active

Specifies the data to be used to overwrite the data set.

1.3.54 Positioning axis - Object and Alarm Handling

1.3.54.1 _cancelAxisCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its 'CommandId' in the 'commandToBeCancelled' parameter.
When an active command is canceled and there is no further command in the command buffer for the axis concerned, the axis is stopped by the axis dynamic parameters.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled.

Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBe-Cancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.54.2 _disableAxisInterface

This function deactivates the actuator or encoder interface. When the interface is deactivated, the corresponding drivers are deactivated.

The actuator interface is the interface of the axis to the drive, which in addition to the setpoint also includes the control signals of the axis to the drive and the status information and the actual speed value of the drive to the controller, when connection of the drive is via PROFIdrive message frame.

Correspondingly the encoder interface, in addition to the actual encoder value, also includes the control signals of the axis to the encoder evaluation, e.g. in the drive, and the status information of the encoder evaluation, e.g. in the drive, to the controller, when connection of the encoder is via PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

actor (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	2
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the deactivation of the actuator interface.
With YES, the actuator interface is deactivated.
With NO, the actuator interface status remains unchanged.

sensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	3
System default:	0

Specifies the deactivation of the encoder interface.
If the bit is set for the corresponding encoder interface, the interface is deactivated. If the bit is not set, the encoder interface status remains unchanged.

hwLimitSwitch (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the deactivation of the hardware limit switch monitoring.
With YES, the hardware limit switch monitoring is deactivated.
With NO, the status of the hardware limit switch monitoring remains unchanged.

additionalSensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	0

Specifies the deactivation of the encoder interface of the additional sensor.
If the bit is set for the corresponding additional sensor, the additional sensor is deactivated. If the bit is not set, the additional sensor status remains unchanged.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableAxisInterface
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnableAxisInterface

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.54.3 _disableAxisSimulation

This function switches the axis out of simulation mode.
The actual encoder values displayed in the structure elements of the 'sensorData' system variable are applied as actual values; the status of all other actual values remains unchanged.
Any existing following error is removed immediately.
The current simulation status can be queried in the 'simulation' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.54.4 _enableAxisInterface

This function activates the actuator or encoder interface. When the interface is activated, the corresponding drivers are activated.

The actuator interface is the interface of the axis to the drive, which in addition to the setpoint also includes the control signals of the axis to the drive and the status information and the actual speed value of the drive to the controller, when connection of the drive is via PROFIdrive message frame.

Correspondingly the encoder interface, in addition to the actual encoder value, also includes the control signals of the axis to the encoder evaluation, e.g. in the drive, and the status information of the encoder evaluation, e.g. in the drive, to the controller, when connection of the encoder is via PROFIdrive message frame.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

actor (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	2
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the activation of the actuator interface.
With YES, the actuator interface is activated.
With NO, the actuator interface status remains unchanged.

sensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	3
System default:	0

Specifies the activation of the encoder interface.
If the bit is set for the corresponding encoder interface, the interface is activated. If the bit is not set, the encoder interface status remains unchanged.

hwLimitSwitch (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the activation of the hardware limit switch monitoring.
With YES, the hardware limit switch monitoring is activated.
With NO, the status of the hardware limit switch monitoring remains unchanged.

additionalSensor (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	0

Specifies the activation of the encoder interface of the additional sensor.
If the bit is set for the corresponding additional sensor, the additional sensor is activated. If the bit is not set, the additional sensor status remains unchanged.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnableAxisInterface
Parameter index:	6
System default:	IMMEDIATELY

EnumNextCommandEnableAxisInterface

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.54.5 _enableAxisSimulation

This function switches the axis to simulation mode.
If the axis is moving, it is stopped by entering a velocity of 0.
The actual values indicated in the system variables are taken from the setpoints in the same way as on a virtual axis.
The current simulation status can be queried in the 'simulation' system variable.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.54.6 _enableDistributedMotionDelayValueCalculation

This function can be used for explicit initiation of offset determination for distributed synchronous operation (e.g. after adding an axis to a modular machine).
This function cannot be used to determine the offset for recursive synchronous operation with PROFINET, because in this case the offset is not calculated correctly by the system.

This function can be used as of SIMOTION V4.1, SP4.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

master

Direction:	Input parameters
Data type:	MasterType
Parameter index:	1

This is a positioning axis

mode (optional)

Direction:	Input parameters
Data type:	EnumAxisDelayValueCalculationMode
Parameter index:	2
System default:	STANDARD

EnumAxisDelayValueCalculationMode

ACTUAL (7)	Currently accessible synchronous connections
STANDARD (358)	Configured synchronous connections

Specifies connections for which the offset is to be determined.
With ACTUAL, the offset is determined for all currently accessible synchronous connections.
With STANDARD, the offset is determined for all configured synchronous connections. During startup, the command waits until all configured stations are active.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandDelayValueCalculation
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandDelayValueCalculation

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.54.7 _getAxisErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the errorNumber parameter is pending.
NO - The error specified by the errorNumber parameter is not pending.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.54.8 _getAxisErrorState

This function provides information on whether axis alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.54.9 _getStateOfMotionBuffer

This function returns the status of the axis command queue.

Return value: StructRetMotionBuffer

StructRetMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumMotionBuffer-State	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':
EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':
Specifies the number of entries in the command queue.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.54.10 _resetAxis

This function switches the axis to a defined initial state.
All active motions are stopped by entering a setpoint of 0.
Pending commands are deleted, synchronous commands are aborted. The command is executed synchronously.
Pending errors on the axis are deleted. For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.
System variables modified by the program are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.54.11 _resetAxisConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.54.12 _resetAxisError

This function resets axis errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.
The command is asynchronous. When applicable, the error is not reset until the local response activated by the error has been completed.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.54.13 _resetMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
axis

Direction:	Input parameters
Data type:	DriveAxis
Parameter index:	1

Specifies the technology object of the 'driveAxis', 'followingAxis' or 'posAxis' type or a variable of the 'DriveAxis' type on which the command is to be executed.

1.3.55 Controller object - Command tracking

1.3.55.1 _bufferControllerObjectCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedControllerObjectCommandId is called or controller object goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.55.2 _getStateOfControllerObjectCommand

This function returns the execution state of a command.

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT- Unknown 'commandId' or
command has already been completed
WAITING_FOR_SYNC_START- Waiting for synchronous start
WAITING - Command has been decoded, but it is still waiting to be executed.
ACTIVE - Command is being executed.

Parameter:
controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.55.3 _removeBufferedControllerObjectCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.56 Controller object - Object and Alarm Handling

1.3.56.1 _getControllerObjectErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:
controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.56.2 _getControllerObjectErrorState

This function provides information on whether controller object alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

1.3.56.3 _resetControllerObject

This function resets a controller object to the initial state.
Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.56.4 _resetControllerObjectConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

1.3.56.5 _resetControllerObjectError

This function resets controller object errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.57 Controller object - Controller functions

1.3.57.1 _changeEnableModeOfControllerObjectIn

This function activates/deactivates the input interconnections of the object.
The interconnections for the actual variable, the setpoint variable, and the disturbance variable are available for selection.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

setpointIn (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies the activation of the setpoint variable.
With ACTIVE, interconnection is activated.
With INACTIVE, interconnection is deactivated.
With DO_NOT_CHANGE, no change takes place.

actualValueIn (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies the activation of the actual variable.
With ACTIVE, interconnection is activated.
With INACTIVE, interconnection is deactivated.
With DO_NOT_CHANGE, no change takes place.

precontrolValueIn (optional)

Direction:	Input parameters
Data type:	EnumActiveInactiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumActiveInactiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies the activation of the disturbance variable.
With ACTIVE, interconnection is activated.
With INACTIVE, interconnection is deactivated.
With DO_NOT_CHANGE, no change takes place.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	**Name**	**Data type**	**Unit**
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.57.2 _disableControllerObject

This function deactivates the controller. The command can be told whether the last value is to be retained or a substitute value will be output.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

valueBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumControllerObjectValueBehaviorMode
Parameter index:	2
System default:	LAST_VALUE

EnumControllerObjectValueBehaviorMode

DEFAULT_VALUE (2)	Replacement value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	Substitute value zero

Specifies whether the replacement value or the last valid value is to be taken.
With LAST_VALUE, the last valid value from the interconnection is taken.
With ZERO_VALUE, the replacement value of zero is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.57.3 _disableControllerObjectIn

This function deactivates the input interconnections of the object.
The interconnections for the actual variable, the setpoint variable, and the disturbance variable are available for selection.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

setpointIn (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies the activation of the setpoint variable.
With INACTIVE, interconnection is deactivated.
With DO_NOT_CHANGE, no change takes place.

actualValueIn (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies the activation of the actual variable.
With INACTIVE, interconnection is deactivated.
With DO_NOT_CHANGE, no change takes place.

precontrolValueIn (optional)

Direction:	Input parameters
Data type:	EnumInactiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumInactiveNoChange

DO_NOT_CHANGE (43)	No change
INACTIVE (61)	Deactivated

Specifies the activation of the disturbance variable.
With INACTIVE, interconnection is deactivated.
With DO_NOT_CHANGE, no change takes place.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.57.4 _enableControllerObject

This function activates the controller.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.57.5 _enableControllerObjectIn

This function activates the input interconnections of the object.
The interconnections for the actual variable, the setpoint variable, and the disturbance variable are available for selection.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

setpointIn (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies the activation of the setpoint variable.
With ACTIVE, interconnection is activated.
With DO_NOT_CHANGE, no change takes place.

actualValueIn (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	3
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies the activation of the actual variable.
With ACTIVE, interconnection is activated.
With DO_NOT_CHANGE, no change takes place.

precontrolValueIn (optional)

Direction:	Input parameters
Data type:	EnumActiveNoChange
Parameter index:	4
System default:	DO_NOT_CHANGE

EnumActiveNoChange

ACTIVE (4)	Activated
DO_NOT_CHANGE (43)	No change

Specifies the activation of the disturbance variable.
With ACTIVE, interconnection is activated.
With DO_NOT_CHANGE, no change takes place.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.57.6 _setControllerObjectPIDControl

This function switches to the PID controller structure and the controller parameters specified in the command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

controller

Direction:	Input parameters
Data type:	_ControllerObjectType
Parameter index:	1

Specifies the technology object of the 'controllerObject' type or a variable of the '_ControllerObjectType' on which the command is to be executed.

krType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	2
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of specification of the P component gain.
With DIRECT, the value specified in the 'kr' parameter is used.
With USER_DEFAULT, the default setting defined in system variable 'pidControllerDefault.kp' is used.

kr (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3
System default:	1.0

Specifies the P component gain; the evaluation depends on the 'krType' parameter.
It is only used when parameter 'krType := DIRECT'.
In all other cases, the parameter is irrelevant.

pMode (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultYesNo
Parameter index:	4
System default:	USER_DEFAULT

EnumUserDefaultYesNo

NO (91)	No
USER_DEFAULT (149)	User default
YES (173)	Yes

Closed-loop controller with proportional component.
YES means that the proportional component is activated.
YES means that the proportional component is deactivated.
With USER_DEFAULT, the default setting defined in system variable 'pidControllerDefault.pMode' is used.

tnType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	5
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of specification of the integration time of the I component.
With DIRECT, the value set in the 'tn' parameter is used.
With USER_DEFAULT, the default setting defined in system variable 'pidControllerDefault.tn' is used.

tn (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the integration time of the I component; the evaluation depends on the 'tnType' parameter.
It is only used when parameter 'tnType := DIRECT'.
In all other cases, the parameter is irrelevant.

iHoldMode (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	7
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Freezing of I component.
YES means that the I component is frozen.
NO means that the I component is calculated.

iSetMode (optional)

Direction:	Input parameters
Data type:	EnumControllerObjectISetMode
Parameter index:	8
System default:	NO

EnumControllerObjectISetMode

DIRECT (40)	Direct setting
NO (91)	No setting
RESET (304)	Reset to zero
SYSTEM (354)	Set to continuous output value

Type of specification of the I component.
With NO, the I component is not set.
With RESET, the I component is reset.
With DIRECT, the value set in the 'iSetValue' parameter is used.
With SYSTEM, I component is set to continuous output signal.

iSetValue (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the I component.
It is only used when parameter 'iSetMode:= DIRECT'.
In all other cases, the parameter is irrelevant.

antiwindup (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultYesNo
Parameter index:	10
System default:	USER_DEFAULT

EnumUserDefaultYesNo

NO (91)	No
USER_DEFAULT (149)	User default
YES (173)	Yes

Antiwindup of I component.
With YES, antiwindup is activated.
With NO, antiwindup is deactivated.
With USER_DEFAULT, the default setting defined in system variable 'pidControllerDefault.antiwindup' is used.

tvType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	11
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of specification of the actuation time of the D component.
With DIRECT, the value specified in the 'tv' parameter is used.
With USER_DEFAULT, the default setting defined in system variable 'pidControllerDefault.tv' is used.

tv (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the actuation time of the D component; the evaluation depends on the 'tvType' parameter.
It is only used when parameter 'tvType := DIRECT'.
In all other cases, the parameter is irrelevant.

decayTimeType (optional)

Direction:	Input parameters
Data type:	EnumUserDefaultDirect
Parameter index:	13
System default:	USER_DEFAULT

EnumUserDefaultDirect

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of specification of the decay time of the D component.
With DIRECT, the value specified in the 'decayTime' parameter is used.
With USER_DEFAULT, the default setting defined in system variable 'pidControllerDefault.decayTime' is used.

decayTime (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	0.0

Specifies the decay time of the D component; the evaluation depends on the 'decayTimeType' parameter.
It is only used when parameter 'decayTimeType:= DIRECT'.
In all other cases, the parameter is irrelevant.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	15
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	16
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.58 Sensor - Command tracking

1.3.58.1 _bufferSensorCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedSensorCommandId is called or sensor goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.58.2 _getStateOfSensorCommand

This function returns the execution state of a command.

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT- Unknown 'commandId' or
command has already been completed
WAITING_FOR_SYNC_START- Waiting for synchronous start
WAITING - Command has been decoded, but it is still waiting to be executed.
ACTIVE - Command is being executed.

Parameter:
sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.58.3 _removeBufferedSensorCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.59 Sensor - Actual value manipulation

1.3.59.1 _disableSensor

This function deactivates the peripheral value of the sensor and deactivates the default value of a user substitute value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

valueBehaviorMode (optional)

Direction:	Input parameters
Data type:	EnumSensorValueBehaviorMode
Parameter index:	2
System default:	LAST_VALUE

EnumSensorValueBehaviorMode

DEFAULT_VALUE (2)	Replacement value
LAST_VALUE (338)	Last value
ZERO_VALUE (339)	Substitute value zero

Specifies whether the replacement value or the last valid value is to be taken.
With LAST_VALUE, the last valid value from the interconnection is taken.
With ZERO_VALUE, the replacement value of zero is taken.
With DEFAULT_VALUE, the default value is taken.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandSensor
Parameter index:	3
System default:	IMMEDIATELY

EnumNextCommandSensor

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.59.2 _enableSensor

This function activates the peripheral value of the object and deactivates the default value of the user substitute value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandSensor
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandSensor

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

1.3.60 Sensor - Object and Alarm Handling

1.3.60.1 _getSensorErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:

sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.60.2 _getSensorErrorState

This function provides information on whether sensor alarms have occurred and how many. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - An alarm is on.
NO - No alarm is on.

Return value numberOfErrors:

Total number of pending errors.

Return value error1..8:

In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:

In the return values, the alarm parameters belonging to the error are output.

Parameter:
sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

1.3.60.3 _resetSensor

This function resets a sensor to the initial state.
Pending errors are deleted.
Modified system variables are reset to the configured values on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.60.4 _resetSensorConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

1.3.60.5 _resetSensorError

This function resets sensor errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

sensor

Direction:	Input parameters
Data type:	_SensorType
Parameter index:	1

Specifies the technology object of the 'sensor' type or a variable of the '_SensorType on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.61 Path interpolation - Message functions

1.3.61.1 _getCircularPathData

This command is used to calculate path data of circular interpolation via the system without starting the path motion.

Return value: StructRetGetCircularPathData

StructRetGetCircularPathData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
pathLength	Path length	LREAL	-

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

circularType (optional)

Direction:	Input parameters
Data type:	EnumPathCircularType
Parameter index:	3
System default:	USER_DEFAULT

EnumPathCircularType

USER_DEFAULT (149)	User default
WITH_RADIUS_AND_ENDPOSITION (372)	Circular interpolation with radius, end point, and orientation
BY_CENTER_AND_ARC (373)	Circular interpolation with midpoint and angle
OVER_POSITION_TO_ENDPOSITION (374)	Circular interpolation with intermediate and end points

Specifies the circle type.
With WITH_RADIUS_AND_ENDPOSITION, the circular interpolation is performed in a main plane with radius, end point, and orientation. A 2D end point is approached on a circular path starting from the current position. The end point is specified in parameters x, y, and z, the radius of the circle is specified in 'radius', and the orientation is specified in 'circleDirection'. The radius specification and orientation are not yet explicit. Whether or not the arc to be traveled is less than or greater than 180 degrees must still be specified. The enum 'circleDirection' is used for this purpose. With the 'POSITIVE' or 'NEGATIVE' setting, travel follows an angle of less than or equal to 180 degrees. With the 'LONG_RUN_POSITIVE' or 'LONG_RUN_NEGATIVE' setting, travel follows an angle of greater than or equal to 180 degrees.
With BY_CENTER_AND_ARC, the circular interpolation is performed in a main plane with a midpoint and angle. A 2D end point is approached on a circular path starting from the current position. The midpoint of the circle is specified in parameters i, j, and k, the angle to be traveled is specified in 'arc', and the orientation is specified in 'circleDirection'.
With OVER_POSITION_TO_ENDPOSITION, the circular interpolation is performed in a main plane or in the spatial plane with intermediate and end points. A 2D or 3D end point is approached on a circular path starting from the current position. The intermediate point is specified in parameters i, j, and k and the endpoint in parameters x, y, and z. With 3D circular interpolation, the current position, the intermediate point, and the end point define the spatial plane in which travel is to take place.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.circularType' is used.

circleDirection (optional)

Direction:	Input parameters
Data type:	EnumPathCircularDirection
Parameter index:	4
System default:	POSITIVE

EnumPathCircularDirection

NEGATIVE (85)	Negative
POSITIVE (107)	Positive
LONG_RUN_POSITIVE (377)	Positive, long arc
LONG_RUN_NEGATIVE (378)	Negative, long arc

Specifies the orientation of the arc.
With POSITIVE, direction of rotation in the plane is positive.
With POSITIVE, direction of rotation in the plane is negative.
With LONG_RUN_POSITIVE, direction of rotation in the plane is positive. When multiple solutions exist, travel follows an angle of greater than or equal to 180 degrees. When only one solution exists, the setting corresponds to the enum 'POSITIVE'.
With LONG_RUN_NEGATIVE, direction of rotation in the plane is negative. When multiple solutions exist, travel follows an angle of greater than or equal to 180 degrees. When only one solution exists, the setting corresponds to the enum 'NEGATIVE'.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=BY_CENTER_AND_ARC'.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	5
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Specifies the type of end position coordinates.
With ABSOLUTE, the end position coordinates are specified as an absolute value.
With RELATIVE, the path to the end position coordinates is specified starting from the current position.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

xEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the end position coordinates in the x-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

yEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the end position coordinates in the y-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

zEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the end position coordinates in the Z-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

ijkMode (optional)

Direction:	Input parameters
Data type:	EnumPathIjkMode
Parameter index:	9
System default:	USER_DEFAULT

EnumPathIjkMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default
TARGET_POSITION_MODE (376)	Value specification same as target position

Type of specification of the midpoint/intermediate coordinates.
With ABSOLUTE, the midpoint/intermediate point coordinates are specified as an absolute value.
With RELATIVE, the path to the midpoint/intermediate coordinates is specified starting from the current position.
With TARGET_POSITION_MODE, the specification is taken from the 'pathMode' parameter.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.ijkMode' is used.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

i (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the x-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

j (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the y-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

k (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the z-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

arc (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Only relevant when 'circularType:=BY_CENTER_AND_ARC'.

radius (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	0.0

Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION'.

xStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the start position in the x-direction.

yStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	0.0

Specifies the start position in the y-direction.

zStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the start position in the z-direction.

1.3.61.2 _getCircularPathGeometricData

This command is used to calculate path data of circular interpolation via the system without starting the path motion.

Return value: StructRetGetCircularPathGeometricData

StructRetGetCircularPathGeometricData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
position	Position	StructPathVector	-
firstGeometricDerivative	First derivative	StructPathVector	-
secondGeometricDerivative	Second derivative	StructPathVector	-

StructPathVector
Trace data

Structure	Name	Data type	Unit
x	x coordinate	LREAL	-
y	y coordinate	LREAL	-
z	z coordinate	LREAL	-

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

circularType (optional)

Direction:	Input parameters
Data type:	EnumPathCircularType
Parameter index:	3
System default:	USER_DEFAULT

EnumPathCircularType

USER_DEFAULT (149)	User default
WITH_RADIUS_AND_ENDPOSITION (372)	Circular interpolation with radius, end point, and orientation
BY_CENTER_AND_ARC (373)	Circular interpolation with midpoint and angle
OVER_POSITION_TO_ENDPOSITION (374)	Circular interpolation with intermediate and end points

Specifies the circle type.
With WITH_RADIUS_AND_ENDPOSITION, the circular interpolation is performed in a main plane with radius, end point, and orientation. A 2D end point is approached on a circular path starting from the current position. The end point is specified in parameters x, y, and z, the radius of the circle is specified in 'radius', and the orientation is specified in 'circleDirection'. The radius specification and orientation are not yet explicit. Whether or not the arc to be traveled is less than or greater than 180 degrees must still be specified. The enum 'circleDirection' is used for this purpose. With the 'POSITIVE' or 'NEGATIVE' setting, travel follows an angle of less than or equal to 180 degrees. With the 'LONG_RUN_POSITIVE' or 'LONG_RUN_NEGATIVE' setting, travel follows an angle of greater than or equal to 180 degrees.
With BY_CENTER_AND_ARC, the circular interpolation is performed in a main plane with a midpoint and angle. A 2D end point is approached on a circular path starting from the current position. The midpoint of the circle is specified in parameters i, j, and k, the angle to be traveled is specified in 'arc', and the orientation is specified in 'circleDirection'.
With OVER_POSITION_TO_ENDPOSITION, the circular interpolation is performed in a main plane or in the spatial plane with intermediate and end points. A 2D or 3D end point is approached on a circular path starting from the current position. The intermediate point is specified in parameters i, j, and k and the endpoint in parameters x, y, and z. With 3D circular interpolation, the current position, the intermediate point, and the end point define the spatial plane in which travel is to take place.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.circularType' is used.

circleDirection (optional)

Direction:	Input parameters
Data type:	EnumPathCircularDirection
Parameter index:	4
System default:	POSITIVE

EnumPathCircularDirection

NEGATIVE (85)	Negative
POSITIVE (107)	Positive
LONG_RUN_POSITIVE (377)	Positive, long arc
LONG_RUN_NEGATIVE (378)	Negative, long arc

Specifies the orientation of the arc.
With POSITIVE, direction of rotation in the plane is positive.
With POSITIVE, direction of rotation in the plane is negative.
With LONG_RUN_POSITIVE, direction of rotation in the plane is positive. When multiple solutions exist, travel follows an angle of greater than or equal to 180 degrees. When only one solution exists, the setting corresponds to the enum 'POSITIVE'.
With LONG_RUN_NEGATIVE, direction of rotation in the plane is negative. When multiple solutions exist, travel follows an angle of greater than or equal to 180 degrees. When only one solution exists, the setting corresponds to the enum 'NEGATIVE'.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=BY_CENTER_AND_ARC'.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	5
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Specifies the type of end position coordinates.
With ABSOLUTE, the end position coordinates are specified as an absolute value.
With RELATIVE, the path to the end position coordinates is specified starting from the current position.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

xEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the end position coordinates in the x-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

yEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the end position coordinates in the y-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

zEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the end position coordinates in the Z-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

ijkMode (optional)

Direction:	Input parameters
Data type:	EnumPathIjkMode
Parameter index:	9
System default:	USER_DEFAULT

EnumPathIjkMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default
TARGET_POSITION_MODE (376)	Value specification same as target position

Type of specification of the midpoint/intermediate coordinates.
With ABSOLUTE, the midpoint/intermediate point coordinates are specified as an absolute value.
With RELATIVE, the path to the midpoint/intermediate coordinates is specified starting from the current position.
With TARGET_POSITION_MODE, the specification is taken from the 'pathMode' parameter.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.ijkMode' is used.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

i (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the x-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circularType:=OVER_POSITION_TO_ENDPOSITION'.

j (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the y-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

k (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the z-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

arc (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Only relevant when 'circularType:=BY_CENTER_AND_ARC'.

radius (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	0.0

Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION'.

xStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the start position in the x-direction.

yStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	0.0

Specifies the start position in the y-direction.

zStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the start position in the z-direction.

pathPointType (optional)

Direction:	Input parameters
Data type:	EnumPathPointType
Parameter index:	18
System default:	END_POINT

EnumPathPointType

START_POINT (401)	Start position of path
END_POINT (402)	End position of path
SPECIFIC_POINT (403)	Specific path point

Specifies the type of the path point for which the path data are to be determined.
With START_POINT, the path data are determined at the start position of the command.
With END_POINT, the path data are determined at the end position of the command.
With SPECIFIC_POINT, a point on the path is specified in the 'specificPathPoint' parameter.

specificPathPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	0.0

Specifies the path length distance to the start position of the path. The path data are determined for the specified path point. If the specified path length is outside the range of the calculated path length, an error is output.
This is only effective with the parameter 'pathPointType:=SPECIFIC_POINT'.

1.3.61.3 _getLinearPathData

This command is used to calculate path data of linear interpolation via the system without starting the path motion.

Return value: StructRetGetLinearPathData

StructRetGetLinearPathData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
pathLength	Path length	LREAL	-

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	3
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified up to the destination.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.

xEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position in the x-direction; the evaluation depends on the 'pathMode' parameter.

yEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the position in the y-direction; the evaluation depends on the 'pathMode' parameter.

zEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the position in the z-direction; the evaluation depends on the 'pathMode' parameter.

xStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the start position in the x-direction.

yStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the start position in the y-direction.

zStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the start position in the z-direction.

1.3.61.4 _getLinearPathGeometricData

This command is used to calculate path data of linear interpolation via the system without starting the path motion.

Return value: StructRetGetLinearPathGeometricData

StructRetGetLinearPathGeometricData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
position	Position	StructPathVector	-
firstGeometricDerivative	First derivative	StructPathVector	-
secondGeometricDerivative	Second derivative	StructPathVector	-

StructPathVector
Trace data

Structure	Name	Data type	Unit
x	x coordinate	LREAL	-
y	y coordinate	LREAL	-
z	z coordinate	LREAL	-

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	3
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified up to the destination.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.

xEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position in the x-direction; the evaluation depends on the 'pathMode' parameter.

yEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the position in the y-direction; the evaluation depends on the 'pathMode' parameter.

zEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the position in the z-direction; the evaluation depends on the 'pathMode' parameter.

xStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the start position in the x-direction.

yStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the start position in the y-direction.

zStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the start position in the z-direction.

pathPointType (optional)

Direction:	Input parameters
Data type:	EnumPathPointType
Parameter index:	10
System default:	END_POINT

EnumPathPointType

START_POINT (401)	Start position of path
END_POINT (402)	End position of path
SPECIFIC_POINT (403)	Specific path point

Specifies the type of the path point for which the path data are to be determined.
With START_POINT, the path data are determined at the start position of the command.
With END_POINT, the path data are determined at the end position of the command.
With SPECIFIC_POINT, a point on the path is specified in the 'specificPathPoint' parameter.

specificPathPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the path length distance to the start position of the path. The path data are determined for the specified path point. If the specified path length is outside the range of the calculated path length, an error is output.
This is only effective with the parameter 'pathPointType:=SPECIFIC_POINT'.

1.3.61.5 _getPolynomialPathData

This command is used to calculate path data of polynomial interpolation via the system without starting the path motion.

Return value: StructRetGetPolynomialPathData

StructRetGetPolynomialPathData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
pathLength	Path length	LREAL	-

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	3
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified up to the destination.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.

polynomialMode (optional)

Direction:	Input parameters
Data type:	EnumPathPolynomialMode
Parameter index:	4
System default:	USER_DEFAULT

EnumPathPolynomialMode

USER_DEFAULT (149)	User default
SPECIFIC_START_DATA (399)	Explicit start values
ATTACHED_STEADILY (400)	Attach steadily
SETTING_OF_COEFFICIENTS (410)	Specification of the polynomial coefficients

Specifies the type of derivatives at the start position.

With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.polynomialMode' is used.

With SPECIFIC_START_DATA, the first and second derivatives at the start position are specified by the 'vector1' and 'vector2' command parameters. The derivatives at the end position are specified by the 'vector3' and 'vector4' command parameters.

With ATTACHED_STEADILY, the derivatives at the start position are taken from the previous geometry.
If the previous geometry cannot be determined, error 50002, reason 3 are output.
If the previous geometry cannot be determined, the direction vector is taken from the start position to the end position for the first derivative, and the zero vector is taken for the second derivative. The derivatives at the end position are specified by the 'vector1' and 'vector2' command parameters.

With SETTING_OF_COEFFICIENTS, the polynomial coefficients are specified directly.
In the formula a0+a1*p+a2*p^2+a3*p^3+a4*p^4+a5*p^5 where 0<=p<=1, a0 is the starting point of the polynomial,
a2='vector1', a3='vector2', a4='vector3' and a5='vector4'.
a1=end point-starting point-a2-a3-a4-a5.

xEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the position in the x-direction; the evaluation depends on the 'pathMode' parameter.

yEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the position in the y-direction; the evaluation depends on the 'pathMode' parameter.

zEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the position in the z-direction; the evaluation depends on the 'pathMode' parameter.

vector1x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the first derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector1y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the first derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector1z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the first derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the second derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the second derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the second derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	0.0

Specifies the first derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the first derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	0.0

Specifies the first derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the second derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	0.0

Specifies the second derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	0.0

Specifies the second derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

xStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	0.0

Specifies the start position in the x-direction.

yStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	0.0

Specifies the start position in the y-direction.

zStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	22
System default:	0.0

Specifies the start position in the z-direction.

1.3.61.6 _getPolynomialPathGeometricData

This command is used to calculate path data of polynomial interpolation via the system without starting the path motion.

Return value: StructRetGetPolynomialPathGeometricData

StructRetGetPolynomialPathGeometricData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
position	Position	StructPathVector	-
firstGeometricDerivative	First derivative	StructPathVector	-
secondGeometricDerivative	Second derivative	StructPathVector	-

StructPathVector
Trace data

Structure	Name	Data type	Unit
x	x coordinate	LREAL	-
y	y coordinate	LREAL	-
z	z coordinate	LREAL	-

Return value 'functionResult'
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	3
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified up to the destination.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.

polynomialMode (optional)

Direction:	Input parameters
Data type:	EnumPathPolynomialMode
Parameter index:	4
System default:	USER_DEFAULT

EnumPathPolynomialMode

USER_DEFAULT (149)	User default
SPECIFIC_START_DATA (399)	Explicit start values
ATTACHED_STEADILY (400)	Attach steadily
SETTING_OF_COEFFICIENTS (410)	Specification of the polynomial coefficients

Specifies the type of derivatives at the start position.
With SPECIFIC_START_DATA, the first and second derivatives at the start position are specified by the 'vector1' and 'vector2' command parameters. The derivatives at the end position are specified by the 'vector3' and 'vector4' command parameters.
With ATTACHED_STEADILY, the derivatives at the start position are taken from the previous geometry. If the previous geometry cannot be determined, the direction vector is taken from the start position to the end position for the first derivative, and the zero vector is taken for the second derivative. The derivatives at the end position are specified by the 'vector1' and 'vector2' command parameters.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.polynomialMode' is used.

xEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the position in the x-direction; the evaluation depends on the 'pathMode' parameter.

yEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the position in the y-direction; the evaluation depends on the 'pathMode' parameter.

zEnd (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the position in the z-direction; the evaluation depends on the 'pathMode' parameter.

vector1x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the first derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector1y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the first derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector1z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the first derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the second derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the second derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the second derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	0.0

Specifies the first derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the first derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	0.0

Specifies the first derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the second derivative in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	0.0

Specifies the second derivative in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	0.0

Specifies the second derivative in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

xStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	0.0

Specifies the start position in the x-direction.

yStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	0.0

Specifies the start position in the y-direction.

zStart (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	22
System default:	0.0

Specifies the start position in the z-direction.

pathPointType (optional)

Direction:	Input parameters
Data type:	EnumPathPointType
Parameter index:	23
System default:	END_POINT

EnumPathPointType

START_POINT (401)	Start position of path
END_POINT (402)	End position of path
SPECIFIC_POINT (403)	Specific path point

Specifies the type of the path point for which the path data are to be determined.
With START_POINT, the path data are determined at the start position of the command.
With END_POINT, the path data are determined at the end position of the command.
With SPECIFIC_POINT, a point on the path is specified in the 'specificPathPoint' parameter.

specificPathPoint (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	24
System default:	0.0

Specifies the path length distance to the start position of the path. The path data are determined for the specified path point. If the specified path length is outside the range of the calculated path length, an error is output.
This is only effective with the parameter 'pathPointType:=SPECIFIC_POINT'.

1.3.62 Path interpolation - Object and Alarm Handling

1.3.62.1 _cancelPathObjectCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its CommandId in the 'commandToBeCancelled' parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled. Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.62.2 _disablePathObjectSimulation

This function switches the path interpolation out of simulation mode. The path values are output to the path axes again. An existing following error is removed immediately with maximum values.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.62.3 _enablePathObjectSimulation

This function switches the path interpolation to simulation mode. The path values are calculated, but are not output to the path axes.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.62.4 _getPathObjectErrorNumberState

This function returns the status (pending, not pending) of an error specified by its error number.

Return value: StructRetGetErrorNumberState

StructRetGetErrorNumberState
Return value with error code and alarm status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value errorState:

YES - The error specified by the 'errorNumber' parameter is pending.
NO - The error specified by the 'errorNumber' parameter is not pending.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

errorNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the error to be read out.

1.3.62.5 _getPathObjectErrorState

This function provides information on whether and how many alarms are pending at path object. In addition, information on the errors is output. The additional alarm information is returned in the command as LREAL.

Return value: StructRetGetToError

StructRetGetToError
Return value with error code and alarm description

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
errorState	Alarm status	EnumYesNo	-
numberOfErrors	Number of pending errors	DINT	-
error1	Error number 1	DINT	-
error1Parameter1	Additional parameter 1 for error 1	LREAL	-
error1Parameter2	Additional parameter 2 for error 1	LREAL	-
error1Parameter3	Additional parameter 3 for error 1	LREAL	-
error1Parameter4	Additional parameter 4 for error 1	LREAL	-
error1Parameter5	Additional parameter 5 for error 1	LREAL	-
error2	Error number 2	DINT	-
error2Parameter1	Additional parameter 1 for error 2	LREAL	-
error2Parameter2	Additional parameter 2 for error 2	LREAL	-
error2Parameter3	Additional parameter 3 for error 2	LREAL	-
error2Parameter4	Additional parameter 4 for error 2	LREAL	-
error2Parameter5	Additional parameter 5 for error 2	LREAL	-
error3	Error number 3	DINT	-
error3Parameter1	Additional parameter 1 for error 3	LREAL	-
error3Parameter2	Additional parameter 2 for error 3	LREAL	-
error3Parameter3	Additional parameter 3 for error 3	LREAL	-
error3Parameter4	Additional parameter 4 for error 3	LREAL	-
error3Parameter5	Additional parameter 5 for error 3	LREAL	-
error4	Error number 4	DINT	-

Structure	Name	Data type	Unit
error4Parameter1	Additional parameter 1 for error 4	LREAL	-
error4Parameter2	Additional parameter 2 for error 4	LREAL	-
error4Parameter3	Additional parameter 3 for error 4	LREAL	-
error4Parameter4	Additional parameter 4 for error 4	LREAL	-
error4Parameter5	Additional parameter 5 for error 4	LREAL	-
error5	Error number 5	DINT	-
error5Parameter1	Additional parameter 1 for error 5	LREAL	-
error5Parameter2	Additional parameter 2 for error 5	LREAL	-
error5Parameter3	Additional parameter 3 for error 5	LREAL	-
error5Parameter4	Additional parameter 4 for error 5	LREAL	-
error5Parameter5	Additional parameter 5 for error 5	LREAL	-
error6	Error number 6	DINT	-
error6Parameter1	Additional parameter 1 for error 6	LREAL	-
error6Parameter2	Additional parameter 2 for error 6	LREAL	-
error6Parameter3	Additional parameter 3 for error 6	LREAL	-
error6Parameter4	Additional parameter 4 for error 6	LREAL	-
error6Parameter5	Additional parameter 5 for error 6	LREAL	-
error7	Error number 7	DINT	-
error7Parameter1	Additional parameter 1 for error 7	LREAL	-
error7Parameter2	Additional parameter 2 for error 7	LREAL	-
error7Parameter3	Additional parameter 3 for error 7	LREAL	-
error7Parameter4	Additional parameter 4 for error 7	LREAL	-
error7Parameter5	Additional parameter 5 for error 7	LREAL	-
error8	Error number 8	DINT	-

Structure	Name	Data type	Unit
error8Parameter1	Additional parameter 1 for error 8	LREAL	-
error8Parameter2	Additional parameter 2 for error 8	LREAL	-
error8Parameter3	Additional parameter 3 for error 8	LREAL	-
error8Parameter4	Additional parameter 4 for error 8	LREAL	-
error8Parameter5	Additional parameter 5 for error 8	LREAL	-

EnumYesNo

NO (91)	No
YES (173)	Yes

Return value functionResult:
Corresponds to the list of return values under the _move function.

Return value errorState:
With YES, an alarm is pending.
With NO, no alarm is pending.

Return value numberOfErrors:
Total number of pending errors.

Return value error1..8:
In the return values error1 to error8, the error numbers of the first eight alarms are output.

Return value error1..8Parameter1..5:
In the return values, the alarm parameters belonging to the error are output.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

1.3.62.6 _getStateOfPathObjectMotionBuffer

This function returns the status of the command queue of the path object.

Return value: StructRetGetPathMotionBuffer

StructRetGetPathMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumPathMotionBuf-ferState	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumPathMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult':

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':

EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':

Specifies the number of entries in the command queue.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

1.3.62.7 _resetPathObject

This function resets a path object to its initial state.
Pending errors are deleted.
Modified system variables are reset on request.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

userDefaultData (optional)

Direction:	Input parameters
Data type:	EnumDataDefault
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumDataDefault

ACTIVATE_CONFIGURATION_DATA (3)	Set user default values to the configured values
DO_NOT_CHANGE (43)	Do not reset the user default values

Specifies whether the system variables are reset to the configured values.
With ACTIVATE_CONFIGURATION_DATA, the values are reset to the values entered during axis configuration.
With DO_NOT_CHANGE, the values are not reset.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumToRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumToRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With ACTIVATE_RESTART, the technology object is restarted.
With NO_RESTART_ACTIVATION, the technology object is not restarted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	WHEN_COMMAND_DONE

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.62.8 _resetPathObjectConfigDataBuffer

When configuring in RUN mode, modified configuration data can be collected in a buffer and activated in a body. Configuration data are collected by setting the activationModeChangedConfigData system variable to the value COLLECT_CHANGED_CONFIG_DATA.
This function clears the configuration data collected in the buffer since the last activation without activating them.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

1.3.62.9 _resetPathObjectError

This function resets path object errors.
For errors that cannot be acknowledged at the time, the command is terminated with a negative acknowledgment.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumErrorReset
Parameter index:	2
System default:	ALL_ERRORS

EnumErrorReset

ALL_ERRORS (10)	All errors
SPECIFIC_ERROR (126)	Selected errors

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, the error specified in 'errorNumber' is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	3
System default:	0

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandReset
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandReset

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.
The command status cannot be tracked when 'nextCommand:=IMMEDIATELY' and 'commandId:=(0, 0)'.

1.3.62.10 _resetPathObjectMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT
Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

1.3.63 Path interpolation - Motion

1.3.63.1 _continuePath

This function continues the path motion, if it was stopped with STOP and STOP_WITHOUT_ABORT in the 'stopMode' parameter.
When continuing a motion, the dynamic parameters (e.g., velocity profile, acceleration) of the interrupted command are used.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	2
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the transition takes place when the command is finished or aborted.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.63.2 _movePathCircular

In the case of circular interpolation, an end point is approached on a circular path starting from the current position. The arc is described by different circle types, which are selected by a command parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

circularType (optional)

Direction:	Input parameters
Data type:	EnumPathCircularType
Parameter index:	3
System default:	USER_DEFAULT

EnumPathCircularType

USER_DEFAULT (149)	User default
WITH_RADIUS_AND_ENDPOSITION (372)	Circular interpolation with radius, end point, and orientation
BY_CENTER_AND_ARC (373)	Circular interpolation with midpoint and angle
OVER_POSITION_TO_ENDPOSITION (374)	Circular interpolation with intermediate and end points

Specifies the circle type.
With WITH_RADIUS_AND_ENDPOSITION, the circular interpolation is performed in a main plane with radius, end point, and orientation. A 2D end point is approached on a circular path starting from the current position. The end point is specified in parameters x, y, and z, the radius of the circle is specified in 'radius', and the orientation is specified in 'circleDirection'. The radius specification and orientation are not yet explicit. Whether or not the arc to be traveled is less than or greater than 180 degrees must still be specified. The enum 'circleDirection' is used for this purpose. With the 'POSITIVE' or 'NEGATIVE' setting, travel follows an angle of less than or equal to 180 degrees. With the 'LONG_RUN_POSITIVE' or 'LONG_RUN_NEGATIVE' setting, travel follows an angle of greater than or equal to 180 degrees.
With BY_CENTER_AND_ARC, the circular interpolation is performed in a main plane with a midpoint and angle. A 2D end point is approached on a circular path starting from the current position. The midpoint of the circle is specified in parameters i, j, and k, the angle to be traveled is specified in 'arc', and the orientation is specified in 'circleDirection'.
With OVER_POSITION_TO_ENDPOSITION, the circular interpolation is performed in a main plane or in space with intermediate and end points. A 2D or 3D end point is approached on a circular path starting from the current position. The intermediate point is specified in parameters i, j, and k and the endpoint in parameters x, y, and z. With 3D circular interpolation, the current position, the intermediate point, and the end point define the spatial plane in which travel is to take place.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.circularType' is used.

circleDirection (optional)

Direction:	Input parameters
Data type:	EnumPathCircularDirection
Parameter index:	4
System default:	POSITIVE

EnumPathCircularDirection

NEGATIVE (85)	Negative
POSITIVE (107)	Positive
LONG_RUN_POSITIVE (377)	Positive, long arc
LONG_RUN_NEGATIVE (378)	Negative, long arc

Specifies the orientation of the arc.
With POSITIVE, direction of rotation in the plane is positive.
With POSITIVE, direction of rotation in the plane is negative.
With LONG_RUN_POSITIVE, direction of rotation in the plane is positive. When multiple solutions exist, travel follows an angle of greater than or equal to 180 degrees. When only one solution exists, the setting corresponds to the enum 'POSITIVE'.
With LONG_RUN_NEGATIVE, direction of rotation in the plane is negative. When multiple solutions exist, travel follows an angle of greater than or equal to 180 degrees. When only one solution exists, the setting corresponds to the enum 'NEGATIVE'.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=BY_CENTER_AND_ARC'.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	5
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of specification of the end point coordinates.
With ABSOLUTE, the end position coordinates are specified as an absolute value.
With RELATIVE, the path to the end position coordinates is specified starting from the current position.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the end position coordinates in the x-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the end position coordinates in the y-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the end position coordinates in the Z-direction; the evaluation depends on the 'pathMode' parameter.
Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

ijkMode (optional)

Direction:	Input parameters
Data type:	EnumPathIjkMode
Parameter index:	9
System default:	USER_DEFAULT

EnumPathIjkMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default
TARGET_POSITION_MODE (376)	Value specification same as target position

Type of specification of the midpoint/intermediate coordinates.
With ABSOLUTE, the midpoint/intermediate point coordinates are specified as an absolute value.
With RELATIVE, the path to the midpoint/intermediate coordinates is specified starting from the current position.
With TARGET_POSITION_MODE, the specification is taken from the 'pathMode' parameter.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.ijkMode' is used.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

i (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the x-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

j (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the y-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

k (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the midpoint/intermediate point coordinates in the z-direction; the evaluation depends on the 'ijkMode' parameter.
Only relevant when 'circularType:=BY_CENTER_AND_ARC' and 'circular-Type:=OVER_POSITION_TO_ENDPOSITION'.

arc (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Only relevant when 'circularType:=BY_CENTER_AND_ARC'.

radius (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	0.0

Only relevant when 'circularType:=WITH_RADIUS_AND_ENDPOSITION'.

dynamicAdaption (optional)

Direction:	Input parameters
Data type:	EnumPathDynamicAdaption
Parameter index:	15
System default:	USER_DEFAULT

EnumPathDynamicAdaption

INACTIVE (61)	Maximum axial dynamic response values are not taken into consideration
USER_DEFAULT (149)	User default
ACTIVE_WITH_CONSTANT_LIMITS (397)	Maximum axial dynamic response values are taken into consideration
ACTIVE_WITH_VARIABLE_LIMITS (398)	Maximum axial dynamic response values are taken into account with path segmentation

Adjustment of path dynamic response to axis dynamic response.
With INACTIVE, the maximum axial dynamic response values are not taken into account.
With ACTIVE_WITH_CONSTANT_LIMITS, the maximum path dynamic response values are determined according to the maximum axial dynamic response values via the path.
With ACTIVE_WITH_VARIABLE_LIMITS, the maximum path dynamic response values are determined in the same manner as with the ACTIVE_WITH_CONSTANT_LIMITS setting. However, with this setting, the path dynamic response is not constant over the entire path area. Instead, the path is segmented with different maximum path dynamic response values.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.dynamicAdaption' is used.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumPathVelocity
Parameter index:	16
System default:	USER_DEFAULT

EnumPathVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

specificVelocityProfile (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	18
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the velocity profile with a cam.
With NO, the velocity is specified with the 'velocityType' and 'velocity' parameters.
With YES, the velocity is specifies with the cam in the 'profileReference' parameter.

profileReference (optional)

Direction:	Input parameters
Data type:	CamType
Parameter index:	19
System default:	0

Specifies the velocity profile with a cam. With the 'profileStartPosition' and 'profileEndPosition' parameters, a range within the domain of the cam is selected and mapped to the path length of the command. The resulting values in the domain of the cam are interpreted in the velocity unit of the path object.
This is only effective with parameter 'specificVelocityProfile:=YES'.

profileStartPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	20
System default:	0.0

Specifies the start position for the velocity profile of the cam transferred in the 'profileReference' parameters.
This is only effective with parameter 'specificVelocityProfile:=YES'.

profileEndPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	0.0

Specifies the end position for the velocity profile of the cam transferred in the 'profileReference' parameters.
This is only effective with parameter 'specificVelocityProfile:=YES'.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	22
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.dynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	24
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	26
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	27
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	28
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	29
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	30
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	31
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	32
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelEndJerk' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	33
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumPathVelocityProfile
Parameter index:	34
System default:	USER_DEFAULT

EnumPathVelocityProfile

SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
Parameters TRAPEZOIDAL and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.dynamics.profile' is used.

w (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	35
System default:	0.0

Specifies the target position of the synchronous axis; the evaluation depends on the 'wMode' parameter.

wMode (optional)

Direction:	Input parameters
Data type:	EnumPathWMode
Parameter index:	36
System default:	USER_DEFAULT

EnumPathWMode

ABSOLUTE (1)	Absolute target position
RELATIVE (115)	Relative target position
USER_DEFAULT (149)	User default
OUTPUT_PATH_LENGTH (379)	Absolute path length
OUTPUT_PATH_LENGTH_ADDITIVE (380)	Additive path length

Type of target position specification of the synchronous axis.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified.
With OUTPUT_PATH_LENGTH, the current traveled path length is output beginning from zero.
With OUTPUT_PATH_LENGTH_ADDITIVE, the current traveled path length is added to the existing value.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.w.mode' is used.

wDirection (optional)

Direction:	Input parameters
Data type:	EnumPathWDirection
Parameter index:	37
System default:	USER_DEFAULT

EnumPathWDirection

BY_VALUE (24)	From sign of target position
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the motion direction of the synchronous axis.
With an absolute position specification ('positioningMode:=ABSOLUTE'), the direction is directly derived from the target position to be approached for linear and rotary axes.
The following applies to modulo axes as well as to linear and rotary axes with relative positioning ('positioningMode:=RELATIVE'):

With BY_VALUE, the direction is determined from the sign of the position specification.
With NEGATIVE, motion is in the negative direction relative to the axis coordinate system.
With POSITIVE, motion is in the positive direction relative to the axis coordinate system.
With SHORTEST_WAY, the shortest path is taken on modulo axes.
With SHORTEST_WAY and relative positioning ('positioningMode:=RELATIVE'), the direction is determined from the sign of the position specification for linear and rotary axes.
With USER_DEFAULT, the default direction defined in system variable 'userDefault.w.direction' is used.

blendingMode (optional)

Direction:	Input parameters
Data type:	EnumPathBlendingMode
Parameter index:	38
System default:	USER_DEFAULT

EnumPathBlendingMode

INACTIVE (61)	No blending
USER_DEFAULT (149)	User default
ACTIVE_WITHOUT_DYNAMIC_ADAPTION (383)	Blending without consideration of the dynamic axis response
ACTIVE_WITH_DYNAMIC_ADAPTION (411)	Blending with consideration of the dynamic axis response

Specifies blending mode.
With INACTIVE, no blending takes place.
With ACTIVE_WITH_DYNAMIC_ADAPTION, the current motion is blended with consideration of the dynamic axis response.
With ACTIVE_WITHOUT_DYNAMIC_ADAPTION, blending is carried out without consideration of the axis dynamic response.
With USER_DEFAULT, the current value programmed in system variable 'userDefault.blending.mode' is used.

Blending is a special form of linking the path motion specified in the command to the previous path motion, whereby the transition takes place at the target point of the current motion and the path velocity in the motion commands for the particular motion is not violated at any time.
This means:

- The current path motion is executed at the command velocity up to the target position.
- Exceptions: - If the velocity of the new path command has the same value as the velocity setpoint of the current path motion, the axis decelerates to the new velocity before the current target position is reached. - If the path length of the new motion command is less than the required deceleration distance, the current motion is decelerated accordingly. - If for ACTIVE_WITH_DYNAMIC_ADAPTION, the dynamic limitations of the individual axes are exceeded, the path velocity in the blending point will be reduced appropriately.
- If the velocity of the path command is greater, the velocity is increased after the transition to the new command, i.e. after the previous target position is reached. Active blending requires 'mergeMode' SEQUENTIAL and an adequate 'leading' program transition, i.e. the new motion to be activated with 'blending' must be known to the interpolator or motion control system at the start of deceleration in the previous command.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumPathMergeMode
Parameter index:	39
System default:	SEQUENTIAL

EnumPathMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandPathMove
Parameter index:	40
System default:	IMMEDIATELY

EnumNextCommandPathMove

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	41
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

csType (optional)

Direction:	Input parameters
Data type:	EnumPathObjectCsType
Parameter index:	43
System default:	BCS

EnumPathObjectCsType

BCS (416)	Path coordinate system
OCS (417)	Object coordinate system

Specifies the type of coordinate system on which the specified path motion is to be based.
With BCS, motion occurs in the path coordinate system.
With OCS, motion occurs in the object coordinate system.

csNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	44
System default:	1

Specifies the number for the type of coordinate system set in parameter 'csType'.
Only effective with parameter 'csType:=OCS'.

1.3.63.3 _movePathLinear

In the case of linear interpolation, an end point is approached on a line starting from the current position. The main plane in which motion takes place or the interpolation in the spatial plane is selected using a command parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	3
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified up to the destination.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.

x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position in the x-direction; the evaluation depends on the 'pathMode' parameter.

y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the position in the y-direction; the evaluation depends on the 'pathMode' parameter.

z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the position in the z-direction; the evaluation depends on the 'pathMode' parameter.

dynamicAdaption (optional)

Direction:	Input parameters
Data type:	EnumPathDynamicAdaption
Parameter index:	7
System default:	USER_DEFAULT

EnumPathDynamicAdaption

INACTIVE (61)	Maximum axial dynamic response values are not taken into consideration
USER_DEFAULT (149)	User default
ACTIVE_WITH_CONSTANT_LIMITS (397)	Maximum axial dynamic response values are taken into consideration
ACTIVE_WITH_VARIABLE_LIMITS (398)	Maximum axial dynamic response values are taken into account with path segmentation

Adjustment of path dynamic response to axis dynamic response.

With INACTIVE, the maximum axial dynamic response values are not taken into account.

With ACTIVE_WITH_CONSTANT_LIMITS, the maximum path dynamic response values are determined according to the maximum axial dynamic response values via the path.

With ACTIVE_WITH_VARIABLE_LIMITS, the maximum path dynamic response values are determined in the same manner as with the ACTIVE_WITH_CONSTANT_LIMITS setting. However, with this setting, the path dynamic response is not constant over the entire path area. Instead, the path is segmented with different maximum path dynamic response values.

With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.dynamicAdaption' is used.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumPathVelocity
Parameter index:	8
System default:	USER_DEFAULT

EnumPathVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

specificVelocityProfile (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	10
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the velocity profile with a cam.
With NO, the velocity is specified with the 'velocityType' and 'velocity' parameters.
With YES, the velocity is specifies with the cam in the 'profileReference' parameter.

profileReference (optional)

Direction:	Input parameters
Data type:	CamType
Parameter index:	11
System default:	0

Specifies the velocity profile with a cam. With the 'profileStartPosition' and 'profileEndPosition' parameters, a range within the domain of the cam is selected and mapped to the path length of the command. The resulting values in the domain of the cam are interpreted in the velocity unit of the path object.
This is only effective with parameter 'specificVelocityProfile:=YES'.

profileStartPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the start position for the velocity profile of the cam transferred in the 'profileReference' parameters.
This is only effective with parameter 'specificVelocityProfile:=YES'.

profileEndPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the end position for the velocity profile of the cam transferred in the 'profileReference' parameters.
This is only effective with parameter 'specificVelocityProfile:=YES'.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	14
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.dynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	16
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	18
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	20
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	22
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	24
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelEndJerk' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumPathVelocityProfile
Parameter index:	26
System default:	USER_DEFAULT

EnumPathVelocityProfile

SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
Parameters TRAPEZOIDAL and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.dynamics.profile' is used.

w (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	27
System default:	0.0

Specifies the target position of the synchronous axis; the evaluation depends on the 'wMode' parameter.

wMode (optional)

Direction:	Input parameters
Data type:	EnumPathWMode
Parameter index:	28
System default:	USER_DEFAULT

EnumPathWMode

ABSOLUTE (1)	Absolute target position
RELATIVE (115)	Relative target position
USER_DEFAULT (149)	User default
OUTPUT_PATH_LENGTH (379)	Absolute path length
OUTPUT_PATH_LENGTH_ADDITIVE (380)	Additive path length

Type of target position specification of the synchronous axis.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified.
With OUTPUT_PATH_LENGTH, the current traveled path length is output beginning from zero.
With OUTPUT_PATH_LENGTH_ADDITIVE, the current traveled path length is added to the existing value.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.w.mode' is used.

wDirection (optional)

Direction:	Input parameters
Data type:	EnumPathWDirection
Parameter index:	29
System default:	USER_DEFAULT

EnumPathWDirection

BY_VALUE (24)	From sign of target position
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the motion direction of the synchronous axis.
With an absolute position specification ('positioningMode:=ABSOLUTE'), the direction is directly derived from the target position to be approached for linear and rotary axes.
The following applies to modulo axes as well as to linear and rotary axes with relative positioning ('positioningMode:=RELATIVE'):

With BY_VALUE, the direction is determined from the sign of the position specification.
With POSITIVE, motion is in the positive direction relative to the axis coordinate system.
With NEGATIVE, motion is in the negative direction relative to the axis coordinate system.
With SHORTEST_WAY, the shortest path is taken on modulo axes.
With SHORTEST_WAY and relative positioning ('positioningMode:=RELATIVE'), the direction is determined from the sign of the position specification for linear and rotary axes.
With USER_DEFAULT, the default direction defined in system variable 'userDefault.w.direction' is used.

blendingMode (optional)

Direction:	Input parameters
Data type:	EnumPathBlendingMode
Parameter index:	30
System default:	USER_DEFAULT

EnumPathBlendingMode

INACTIVE (61)	No blending
USER_DEFAULT (149)	User default
ACTIVE_WITHOUT_DYNAMIC_ADAPTION (383)	Blending without consideration of the dynamic axis response
ACTIVE_WITH_DYNAMIC_ADAPTION (411)	Blending with consideration of the dynamic axis response

Specifies blending mode.
With INACTIVE, no blending takes place.
With ACTIVE_WITH_DYNAMIC_ADAPTION, the current motion is blended with consideration of the dynamic axis response.
With ACTIVE_WITHOUT_DYNAMIC_ADAPTION, blending is carried out without consideration of the axis dynamic response.
With USER_DEFAULT, the current value programmed in system variable 'userDefault.blending.mode' is used.

Blending is a special form of linking the path motion specified in the command to the previous path motion, whereby the transition takes place at the target point of the current motion and the path velocity in the motion commands for the particular motion is not violated at any time.
This means:

- The current path motion is executed at the command velocity up to the target position.
- Exceptions: - If the velocity of the new path command has the same value as the velocity setpoint of the current path motion, the axis decelerates to the new velocity before the current target position is reached. - If the path length of the new motion command is less than the required deceleration distance, the current motion is decelerated accordingly. - If for ACTIVE_WITH_DYNAMIC_ADAPTION, the dynamic limitations of the individual axes are exceeded, the path velocity in the blending point will be reduced appropriately.
- If the velocity of the path command is greater, the velocity is increased after the transition to the new command, i.e. after the previous target position is reached. Active blending requires 'mergeMode' SEQUENTIAL and an adequate 'leading' program transition, i.e. the new motion to be activated with 'blending' must be known to the interpolator or motion control system at the start of deceleration in the previous command.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumPathMergeMode
Parameter index:	31
System default:	SEQUENTIAL

EnumPathMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandPathMove
Parameter index:	32
System default:	IMMEDIATELY

EnumNextCommandPathMove

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	33
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

csType (optional)

Direction:	Input parameters
Data type:	EnumPathObjectCsType
Parameter index:	35
System default:	BCS

EnumPathObjectCsType

BCS (416)	Path coordinate system
OCS (417)	Object coordinate system

Specifies the type of coordinate system on which the specified path motion is to be based.
With BCS, motion occurs in the path coordinate system.
With OCS, motion occurs in the object coordinate system.

csNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	36
System default:	1

Specifies the number for the type of coordinate system set in parameter 'csType'.
Only effective with parameter 'csType:=OCS'.

1.3.63.4 _movePathPolynomial

With polynomial interpretation, an end point is approached starting from the current position and taking into consideration the derivates specified in the starting and end points. It is also possible to specify the polynomial coefficients directly.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

pathPlane (optional)

Direction:	Input parameters
Data type:	EnumPathPlane
Parameter index:	2
System default:	USER_DEFAULT

EnumPathPlane

USER_DEFAULT (149)	User default
X_Y (297)	xy plane
Y_Z (298)	yz plane
Z_X (299)	zx plane
X_Y_Z (371)	3D interpolation

Interpolation in the spatial plane or specification of the main plane.

pathMode (optional)

Direction:	Input parameters
Data type:	EnumPathMode
Parameter index:	3
System default:	USER_DEFAULT

EnumPathMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value
USER_DEFAULT (149)	User default

Type of position specification.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified up to the destination.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.mode' is used.

polynomialMode (optional)

Direction:	Input parameters
Data type:	EnumPathPolynomialMode
Parameter index:	4
System default:	USER_DEFAULT

EnumPathPolynomialMode

USER_DEFAULT (149)	User default
SPECIFIC_START_DATA (399)	Explicit start values
ATTACHED_STEADILY (400)	Attach steadily
SETTING_OF_COEFFICIENTS (410)	Specification of the polynomial coefficients

Type of specification of the polynomial.

With SETTING_OF_COEFFICIENTS, the polynomial coefficients are specified directly.
In the formula a0+a1*p+a2*p^2+a3*p^3+a4*p^4+a5*p^5 where 0<=p<=1, a0 is the starting point of the polynomial,
a2='vector1', a3='vector2', a4='vector3' and a5='vector4'.
a1=end point-starting point-a2-a3-a4-a5.

With SPECIFIC_START_DATA, the first and second derivatives at the start position are specified by the 'vector1' and 'vector2' command parameters. The derivatives at the end position are specified by the 'vector3' and 'vector4' command parameters.

With ATTACHED_STEADILY, the derivatives at the start position are taken from the previous geometry.
If the previous geometry cannot be determined, error 50002, reason 3 are output.
If the previous geometry cannot be determined, the direction vector is taken from the start position to the end position for the first derivative, and the zero vector is taken for the second derivative. The derivatives at the end position are specified by the 'vector1' and 'vector2' command parameters.

With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.polynomialMode' is used.

x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the position in the x-direction; the evaluation depends on the 'pathMode' parameter.

y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the position in the y-direction; the evaluation depends on the 'pathMode' parameter.

z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the position in the z-direction; the evaluation depends on the 'pathMode' parameter.

vector1x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the first derivative/polynomial coefficients in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector1y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the first derivative/polynomial coefficients in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector1z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the first derivative/polynomial coefficients in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	0.0

Specifies the second derivative/polynomial coefficients in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	12
System default:	0.0

Specifies the second derivative/polynomial coefficients in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector2z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	0.0

Specifies the second derivative/polynomial coefficients in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	14
System default:	0.0

Specifies the first derivative/polynomial coefficients in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	0.0

Specifies the first derivative/polynomial coefficients in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector3z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	16
System default:	0.0

Specifies the first derivative/polynomial coefficients in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	0.0

Specifies the second derivative/polynomial coefficients in the x-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	18
System default:	0.0

Specifies the second derivative/polynomial coefficients in the y-direction; the evaluation depends on the 'polynomialMode' parameter.

vector4z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	0.0

Specifies the second derivative/polynomial coefficients in the z-direction; the evaluation depends on the 'polynomialMode' parameter.

dynamicAdaption (optional)

Direction:	Input parameters
Data type:	EnumPathDynamicAdaption
Parameter index:	20
System default:	USER_DEFAULT

EnumPathDynamicAdaption

INACTIVE (61)	Maximum axial dynamic response values are not taken into consideration
USER_DEFAULT (149)	User default
ACTIVE_WITH_CONSTANT_LIMITS (397)	Maximum axial dynamic response values are taken into consideration
ACTIVE_WITH_VARIABLE_LIMITS (398)	Maximum axial dynamic response values are taken into account with path segmentation

Adjustment of path dynamic response to axis dynamic response.
With INACTIVE, the maximum axial dynamic response values are not taken into account.
With ACTIVE_WITH_CONSTANT_LIMITS, the maximum path dynamic response values are determined according to the maximum axial dynamic response values via the path.
With ACTIVE_WITH_VARIABLE_LIMITS, the maximum path dynamic response values are determined in the same manner as with the ACTIVE_WITH_CONSTANT_LIMITS setting. However, with this setting, the path dynamic response is not constant over the entire path area. Instead, the path is segmented with different maximum path dynamic response values.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.path.dynamicAdaption' is used.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumPathVelocity
Parameter index:	21
System default:	USER_DEFAULT

EnumPathVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	22
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

specificVelocityProfile (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	23
System default:	NO

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the velocity profile with a cam.
With NO, the velocity is specified with the 'velocityType' and 'velocity' parameters.
With YES, the velocity is specifies with the cam in the 'profileReference' parameter.

profileReference (optional)

Direction:	Input parameters
Data type:	CamType
Parameter index:	24
System default:	0

Specifies the velocity profile with a cam. With the 'profileStartPosition' and 'profileEndPosition' parameters, a range within the domain of the cam is selected and mapped to the path length of the command. The resulting values in the domain of the cam are interpreted in the velocity unit of the path object.
This is only effective with parameter 'specificVelocityProfile:=YES'.

profileStartPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	25
System default:	0.0

Specifies the start position for the velocity profile of the cam transferred in the 'profileReference' parameters.
This is only effective with parameter 'specificVelocityProfile:=YES'.

profileEndPosition (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	26
System default:	0.0

Specifies the end position for the velocity profile of the cam transferred in the 'profileReference' parameters.
This is only effective with parameter 'specificVelocityProfile:=YES'.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	27
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.dynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	28
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	29
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negati-veAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	30
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccel-Type:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	31
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	32
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	33
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	34
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	35
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	36
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	37
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelEndJerk' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	38
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumPathVelocityProfile
Parameter index:	39
System default:	USER_DEFAULT

EnumPathVelocityProfile

SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
Parameters TRAPEZOIDAL and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.dynamics.profile' is used.

w (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	40
System default:	0.0

Specifies the target position of the synchronous axis; the evaluation depends on the 'wMode' parameter.

wMode (optional)

Direction:	Input parameters
Data type:	EnumPathWMode
Parameter index:	41
System default:	USER_DEFAULT

EnumPathWMode

ABSOLUTE (1)	Absolute target position
RELATIVE (115)	Relative target position
USER_DEFAULT (149)	User default
OUTPUT_PATH_LENGTH (379)	Absolute path length
OUTPUT_PATH_LENGTH_ADDITIVE (380)	Additive path length

Type of target position specification of the synchronous axis.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified.
With OUTPUT_PATH_LENGTH, the current traveled path length is output beginning from zero.
With OUTPUT_PATH_LENGTH_ADDITIVE, the current traveled path length is added to the existing value.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.w.mode' is used.

wDirection (optional)

Direction:	Input parameters
Data type:	EnumPathWDirection
Parameter index:	42
System default:	USER_DEFAULT

EnumPathWDirection

BY_VALUE (24)	From sign of target position
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the motion direction of the synchronous axis.
With an absolute position specification ('positioningMode:=ABSOLUTE'), the direction is directly derived from the target position to be approached for linear and rotary axes.
The following applies to modulo axes as well as to linear and rotary axes with relative positioning ('positioningMode:=RELATIVE'):

With BY_VALUE, the direction is determined from the sign of the position specification.
With POSITIVE, motion is in the positive direction relative to the axis coordinate system.
With NEGATIVE, motion is in the negative direction relative to the axis coordinate system.
With SHORTEST_WAY, the shortest path is taken on modulo axes.
With SHORTEST_WAY and relative positioning ('positioningMode:=RELATIVE'), the direction is determined from the sign of the position specification for linear and rotary axes.
With USER_DEFAULT, the default direction defined in system variable 'userDefault.w.direction' is used.

blendingMode (optional)

Direction:	Input parameters
Data type:	EnumPathBlendingMode
Parameter index:	43
System default:	USER_DEFAULT

EnumPathBlendingMode

INACTIVE (61)	No blending
USER_DEFAULT (149)	User default
ACTIVE_WITHOUT_DYNAMIC_ADAPTION (383)	Blending without consideration of the dynamic axis response
ACTIVE_WITH_DYNAMIC_ADAPTION (411)	Blending with consideration of the dynamic axis response

Specifies blending mode.
With INACTIVE, no blending takes place.
With ACTIVE_WITH_DYNAMIC_ADAPTION, the current motion is blended with consideration of the dynamic axis response.
With ACTIVE_WITHOUT_DYNAMIC_ADAPTION, blending is carried out without consideration of the axis dynamic response.
With USER_DEFAULT, the current value programmed in system variable 'userDefault.blending.mode' is used.

Blending is a special form of linking the path motion specified in the command to the previous path motion, whereby the transition takes place at the target point of the current motion and the path velocity in the motion commands for the particular motion is not violated at any time.
This means:

- The current path motion is executed at the command velocity up to the target position.
- Exceptions: - If the velocity of the new path command has the same value as the velocity setpoint of the current path motion, the axis decelerates to the new velocity before the current target position is reached. - If the path length of the new motion command is less than the required deceleration distance, the current motion is decelerated accordingly. - If for ACTIVE_WITH_DYNAMIC_ADAPTION, the dynamic limitations of the individual axes are exceeded, the path velocity in the blending point will be reduced appropriately.
- If the velocity of the path command is greater, the velocity is increased after the transition to the new command, i.e. after the previous target position is reached. Active blending requires 'mergeMode' SEQUENTIAL and an adequate 'leading' program transition, i.e. the new motion to be activated with 'blending' must be known to the interpolator or motion control system at the start of deceleration in the previous command.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumPathMergeMode
Parameter index:	44
System default:	SEQUENTIAL

EnumPathMergeMode

IMMEDIATELY (60)	Replace current motion immediately
NEXT_MOTION (89)	Attach and clear buffer
SEQUENTIAL (119)	Attach

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.
With NEXT_MOTION, the command is attached to the current motion, the current motion is completed, and the commands queued in the buffer are deleted immediately.
With SEQUENTIAL, the command is attached to the existing motion commands.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandPathMove
Parameter index:	45
System default:	IMMEDIATELY

EnumNextCommandPathMove

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	46
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

csType (optional)

Direction:	Input parameters
Data type:	EnumPathObjectCsType
Parameter index:	48
System default:	BCS

EnumPathObjectCsType

BCS (416)	Path coordinate system
OCS (417)	Object coordinate system

Specifies the type of coordinate system on which the specified path motion is to be based.
With BCS, motion occurs in the path coordinate system.
With OCS, motion occurs in the object coordinate system.

csNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	49
System default:	1

Specifies the number for the type of coordinate system set in parameter 'csType'.
Only effective with parameter 'csType:=OCS'.

1.3.63.5 _stopPath

This command stops the path motion on the basis of a programmed braking ramp.
The motion to be stopped can be interrupted or terminated.
An interrupted motion can be continued with the _continuePath command.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

stopMode (optional)

Direction:	Input parameters
Data type:	EnumPathStopMode
Parameter index:	2
System default:	STOP_WITHOUT_ABORT

EnumPathStopMode

STOP_AND_ABORT (135)	Normal stop with abort
STOP_WITHOUT_ABORT (138)	Normal stop without abort
STOP_AND_ABORT_AND_HOLD (412)	Normal stop with abort and block to prevent additional motion commands

Specifies the stop behavior relative to the current command.
With STOP_AND_ABORT, the active motion command is aborted.
With STOP_AND_ABORT_AND_HOLD, the active motion command is aborted. The path object is prevented from receiving additional motion commands. The state can be revoked with _resetPathObject(), _continuePath() or _cancelPathObjectCommand(commandToBeCancelled:=stopCommandId).
With STOP_WITHOUT_ABORT, the current motion is interrupted and can be continued with the _continuePath command.

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	3
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	5
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	7
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	9
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelEndJerk' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumPathVelocityProfile
Parameter index:	11
System default:	USER_DEFAULT

EnumPathVelocityProfile

SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
Parameters TRAPEZOIDAL and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.dynamics.profile' is used.

mergeMode (optional)

Direction:	Input parameters
Data type:	EnumMergeModeStop
Parameter index:	12
System default:	IMMEDIATELY

EnumMergeModeStop

AT_END_OF_COMMAND (2)	No significance
IMMEDIATELY (60)	Execute command immediately

Specifies when the command takes effect relative to the motion.
With IMMEDIATELY, the current motion is replaced immediately and any motion commands still in the buffer are deleted.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandPathMove
Parameter index:	13
System default:	IMMEDIATELY

EnumNextCommandPathMove

AT_DECELERATION_START (12)	At start of deceleration phase
AT_MOTION_START (13)	At start of interpolation
IMMEDIATELY (60)	Immediate command transition
WHEN_ACCELERATION_DONE (156)	At end of acceleration phase
WHEN_BUFFER_READY (159)	After entry in the command queue
WHEN_INTERPOLATION_DONE (162)	At end of setpoint interpolation
WHEN_MOTION_DONE (163)	When motion is finished

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_BUFFER_READY, the transition takes place after the command is entered in the command queue.
With AT_MOTION_START, the transition takes place at the start of interpolation.
With WHEN_ACCELERATION_DONE, the transition takes place at the end of the acceleration phase.
With AT_DECELERATION_START, the transition takes place at the start of the deceleration phase.
With WHEN_INTERPOLATION_DONE, the transition takes place at the end of setpoint interpolation.
With WHEN_MOTION_DONE, the transition takes place when the motion has been completed is stopped in another way.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	14
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

csType (optional)

Direction:	Input parameters
Data type:	EnumPathObjectCsType
Parameter index:	16
System default:	BCS

EnumPathObjectCsType

BCS (416)	Path coordinate system
OCS (417)	Object coordinate system

Specifies the coordinate system.
With BCS, motion is stopped in the path coordinate system.
With OCS, motion is stopped in the coordinate system currently programmed.

1.3.64 Path interpolation - Command tracking

1.3.64.1 _bufferPathObjectCommandId

This function enables the 'commandId' and corresponding command status to be stored for a period of time following the command execution.
The command whose status is to be saved is specified in the 'commandId' parameter. The maximum number of command states that can be saved is specified by the 'decodingConfig.numberOfMaxBufferedCommandId' configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

deleteCommandIdWithReset (optional)

Direction:	Input parameters
Data type:	EnumYesNo
Parameter index:	4
System default:	YES

EnumYesNo

NO (91)	No
YES (173)	Yes

Specifies the criteria for ending continuous saving of command statuses.
With NO, saving of command statuses ends when _removeBufferedAxisCommandId is called or axis goes to STOP mode.
With YES, saving of command statuses ends when the criteria for NO are satisfied as well as when a reset or restart occurs.

1.3.64.2 _cancelPathObjectCommand

This command cancels a command that is waiting or active in the Ipo. The command to be cancelled is specified by the indication of its CommandId in the 'commandToBeCancelled' parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

commandToBeCancelled

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of the command to be cancelled. Commands with the standard ID 'commandToBeCancelled.SystemId_low:=0' and 'commandToBeCancelled.SystemId_high:=0' are not cancelled.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandCancelCommand
Parameter index:	4
System default:	IMMEDIATELY

EnumNextCommandCancelCommand

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When commands with the specified 'commandId' are cancelled

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, the advance takes place when the command with the specified 'commandId' is cancelled.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Specifies the 'commandId' of this command.

1.3.64.3 _getMotionStateOfPathObjectCommand

This function returns the current state of a command motion.

Return value: StructRetPathMotionCommandState

StructRetPathMotionCommandState

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionCommandId-State	Motion status	EnumPathMoti-onCommandIdState	-

EnumPathMotionCommandIdState

NOT_EXISTENT (94)	commandId is not known or command is already completed
BUFFERED (207)	Command is in the command queue
IN_EXECUTION (208)	Command is being executed
IN_ACCELERATION (209)	Motion generated by the command is in the acceleration phase
IN_CONSTANT_MOTION (210)	Motion generated by the command is in the constant velocity phase
IN_DECELERATION (211)	Motion generated by the command is in the deceleration phase
INTERPOLATION_DONE (213)	Setpoint interpolation of command is complete
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

Return value functionResult
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With BUFFERED_READY, the command is in the command queue.
With IN_EXECUTION, the command is being executed.
With IN_ACCELERATION, motion generated by the command is in the acceleration phase.
With IN_CONSTANT_MOTION, motion generated by the command is in the constant velocity phase.
With IN_DECELERATION, motion generated by the command is in the deceleration phase.
With INTERPOLATION_DONE, the setpoint interpolation of the command is complete.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.64.4 _getStateOfPathObjectCommand

This function returns the execution state of an interpolation command.

Return value: StructRetCommandState

StructRetCommandState
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
commandIdState	Command status	EnumCommandId-State	-
abortId	Reason for command abort	DINT	-

EnumCommandIdState

ACTIVE (4)	Command is active
NOT_EXISTENT (94)	commandId is not known or command is already completed
WAITING (154)	Command is decoded, but execution has not yet begun
WAITING_FOR_SYNC_START (155)	Command is decoded, waiting for a synchronous start
EXECUTED (227)	Command executed
ABORTED (228)	Command execution aborted

With ACTIVE, the command is being executed.
With NOT_EXISTENT, the commandId is unknown or the command is already completed.
With WAITING, the command has been decoded but execution has not yet begun.
With WAITING_FOR_SYNC_START, the command has been decoded and is waiting for its synchronous start.

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value commandIdState:

NOT_EXISTENT
- The 'commandId' is unknown or

command has already been completed

WAITING_FOR_SYNC_START
- Waiting for synchronous start

WAITING
- Command has been decoded, but execution has not

yet started

ACTIVE
- Command is being executed.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	2

CommandIdType

System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.64.5 _getStateOfPathObjectMotionBuffer

This function returns the status of the command queue of the path object.

Return value: StructRetGetPathMotionBuffer

StructRetGetPathMotionBuffer
Return value with error code and command status

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
motionBufferState	Status	EnumPathMotionBuf-ferState	-
numberOfExisten-tEntries	Number of entries	DINT	-

EnumPathMotionBufferState

EMPTY (47)	Command queue is empty
FULL (52)	Command queue is full
WRITEABLE (172)	Command queue can be written

Return value 'functionResult':

Description of the return value:
Corresponds to the list of return values under the _move function.

Return value 'motionBufferState':

EMPTY means that the command queue is empty.
FULL means that an entry cannot be written to the command queue, the command queue is full. The command is executed synchronously.
WRITABLE means that an entry can be made in the command queue.

Return value 'numberOfExistentEntries':

Specifies the number of entries in the command queue.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

1.3.64.6 _removeBufferedPathObjectCommandId

This function stops saving of the 'commandId' and corresponding command status after command execution.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

removeMode (optional)

Direction:	Input parameters
Data type:	EnumRemoveMode
Parameter index:	2
System default:	SPECIFIC_ID

EnumRemoveMode

SPECIFIC_ID (229)	Remove special commandId
ALL_ID (230)	Remove all stored commandIds

Specifies the 'commandId' for which permanent saving is to be terminated.
With SPECIFIC_ID, the command status specified with 'commandId' is removed.
With ALL_ID, all saved 'commandId's are removed.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.64.7 _resetPathObjectMotionBuffer

This function clears all commands from the command queue.
Alarm '030002 Command aborted' is issued for each of the deleted commands.
Synchronously issued commands are returned with return value 3 'Command aborted'.
The command for clearing the command queue is executed synchronously.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

1.3.65 Path interpolation - Kinematics transformation

1.3.65.1 _getPathAxesData

This function transforms the specified path coordinates to the associated axis values. An articulated arm position is transferred for purposes of resolving possible kinematic-related ambiguities.

Return value: StructRetPathAxesData

StructRetPathAxesData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
a1	Axis 1	StructPathMotionVector	-
a2	Axis 2	StructPathMotionVector	-
a3	Axis 3	StructPathMotionVector	-

StructPathMotionVector
Trace data

Structure	Name	Data type	Unit
position	Position	LREAL	Unit set by the user
velocity	Velocity	LREAL	Unit set by the user
acceleration	Acceleration	LREAL	Unit set by the user

Return value functionResult
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value data:

The 'data' structure contains the transformed axis coordinates.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

xPosition

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the x-coordinate of the path.

xVelocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the y-coordinate of the path velocity.

xAcceleration

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the x-coordinate of the path acceleration.

yPosition

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5

Specifies the y-coordinate of the path.

yVelocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6

Specifies the y-coordinate of the path velocity.

yAcceleration

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7

Specifies the y-coordinate of the path acceleration.

zPosition

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8

Specifies the z-coordinate of the path.

zVelocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9

Specifies the z-coordinate of the path velocity.

zAcceleration

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10

Specifies the z-coordinate of the path acceleration.

linkConstellation (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	11
System default:	1

Specifies the link constellation.

1.3.65.2 _getPathAxesPosition

This function transforms the specified path coordinates to the associated axis values. An articulated arm position is transferred for purposes of resolving possible kinematic-related ambiguities.

Return value: StructRetPathAxesPosition

StructRetPathAxesPosition

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
a1	Position axis 1	LREAL	-
a2	Position axis 2	LREAL	-
a3	Position axis 3	LREAL	-

Return value functionResult
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value data:

The 'data' structure contains the transformed axis coordinates.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

x

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the x-coordinate of the path.

y

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the y-coordinate of the path.

z

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the z-coordinate of the path.

linkConstellation (optional)

Direction:	Input parameters
Data type:	UDINT
Parameter index:	5
System default:	1

Specifies the link constellation.

1.3.65.3 _getPathCartesianData

This function transforms the specified axis values to the coordinate system of the path.

Return value: StructRetPathCartesianData

StructRetPathCartesianData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
x	x coordinate	StructPathMotionVector	-
y	y coordinate	StructPathMotionVector	-
z	z coordinate	StructPathMotionVector	-
linkConstellation	Link constellation	UDINT	-

StructPathMotionVector
Trace data

Structure	Name	Data type	Unit
position	Position	LREAL	Unit set by the user
velocity	Velocity	LREAL	Unit set by the user
acceleration	Acceleration	LREAL	Unit set by the user

Return value functionResult
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value data:

The 'data' structure contains the transformed path coordinates.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

a1Position

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the axis value of the first axis.

a1Velocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the velocity of the first axis.

a1Acceleration

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the acceleration of the first axis.

a2Position

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5

Specifies the axis value of the second axis.

a2Velocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6

Specifies the velocity of the second axis.

a2Acceleration

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7

Specifies the acceleration of the second axis.

a3Position

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8

Specifies the axis value of the third axis.

a3Velocity

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9

Specifies the velocity of the third axis.

a3Acceleration

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10

Specifies the acceleration of the third axis.

1.3.65.4 _getPathCartesianPosition

This function transforms the specified axis values to the coordinate system of the path.

Return value: StructRetPathCartesianPosition

StructRetPathCartesianPosition

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
position	Position	StructPathVector	-
linkConstellation	Link constellation	UDINT	-

StructPathVector
Trace data

Structure	Name	Data type	Unit
x	x coordinate	LREAL	-
y	y coordinate	LREAL	-
z	z coordinate	LREAL	-

Return value functionResult
Description of the return value:
Corresponds to the list of return values under the _move function.

Return value data:

The 'data' structure contains the transformed path coordinates.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

a1

Direction:	Input parameters
Data type:	LREAL
Parameter index:	2

Specifies the axis value of the first axis.

a2

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the axis value of the second axis.

a3

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

Specifies the axis value of the third axis.

1.3.66 Path interpolation - Object coordinates

1.3.66.1 _enablePathObjectTracking

This function synchronizes the motion calculated at the path object with an object coordinate system tracked in accordance with a motional sequence reference value.
The synchronization conditions and the dynamic response parameters can be specified.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

ocsNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the object coordinate system used.

synchronizingMode (optional)

Direction:	Input parameters
Data type:	EnumPathObjectTrackingSynchronizingMode
Parameter index:	3
System default:	IMMEDIATELY

EnumPathObjectTrackingSynchronizingMode

IMMEDIATELY (60)	Immediate synchronization
ON_POSITION (420)	Synchronization to belt position

Specifies the synchronization criterion.
With IMMEDIATELY, synchronization with the object coordination system occurs immediately, and the position specified in parameters 'x', 'y' and 'z' in the object coordinate system is approached.
With ON_POSITION, preceding synchronization with the specified tracking position ('trackingPosition') of the object coordination system occurs, and the position specified in parameters 'x', 'y' and 'z' in the object coordinate system is approached.

x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the x-direction target position in the object coordinate system which is to be approached with the synchronization command.

y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the y-direction target position in the object coordinate system which is to be approached with the synchronization command.

z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the z-direction target position in the object coordinate system which is to be approached with the synchronization command.

position (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the tracking position ('trackingPosition') of the object coordinate system on which synchronization is to be based with 'synchronizingMode:=ON_POSITION'.

velocityType (optional)

Direction:	Input parameters
Data type:	EnumPathVelocity
Parameter index:	8
System default:	USER_DEFAULT

EnumPathVelocity

CURRENT (33)	Interpolated velocity setpoint
DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed velocity setpoint
USER_DEFAULT (149)	User default

Type of velocity definition.
With CURRENT, the current interpolated velocity setpoint is valid.
With DIRECT, the value set in the 'velocity' parameter is used as the programmed velocity setpoint.
With EFFECTIVE, the last programmed velocity setpoint is used.
With USER_DEFAULT, the default velocity defined in system variable 'userDefaultDynamics.velocity' is used. This default can be modified by entering a value in the 'velocity' parameter. The value of the parameter is interpreted as a percentage.

velocity (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	100.0

Specifies the velocity setpoint; the evaluation depends on the 'velocityType' parameter.
It is only used when parameter 'velocityType := USER_DEFAULT' or 'velocityType := DIRECT'.
In all other cases, the parameter is irrelevant.

positiveAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	10
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of acceleration definition.
With DIRECT, the value set in the 'positiveAccel' parameter is used as the programmed acceleration setpoint.
With EFFECTIVE, the last programmed acceleration setpoint is used.
With USER_DEFAULT, the default acceleration defined in system variable 'userDefault.dynamics.positiveAccel' is used. This default can be modified by entering a value in the 'positiveAccel' parameter. The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	11
System default:	100.0

Specifies the acceleration setpoint; the evaluation depends on the 'positiveAccelType' parameter.
It is only used when parameter 'positiveAccelType:=USER_DEFAULT' or 'positiveAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelType (optional)

Direction:	Input parameters
Data type:	EnumAcceleration
Parameter index:	12
System default:	USER_DEFAULT

EnumAcceleration

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed value
USER_DEFAULT (149)	User default

Type of deceleration definition.
With DIRECT, the value set in the 'negativeAccel' parameter is used as the programmed deceleration setpoint.
With EFFECTIVE, the last programmed deceleration setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccel' is used. This default can be modified by entering a value in the 'negativeAccel' parameter.
The value of the parameter is interpreted as a percentage.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccel (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	13
System default:	100.0

Specifies the deceleration setpoint; the evaluation depends on the 'negativeAccelType' parameter.
It is only used when parameter 'negativeAccelType:=USER_DEFAULT' or 'negativeAccelType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	14
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration start jerk.
With DIRECT, the value set in the 'positiveAccelStartJerk' parameter is used as the programmed acceleration start jerk setpoint.
With EFFECTIVE, the last programmed acceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelStartJerk' is used. This default can be modified by entering a value in the 'positiveAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	15
System default:	100.0

Specifies the acceleration start jerk setpoint; the evaluation depends on the 'positiveAccelStartJerkType' parameter.
It is only used when parameter 'positiveAccelStartJerkType:=USER_DEFAULT' or 'positiveAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	16
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for acceleration end jerk.
With DIRECT, the value set in the 'positiveAccelEndJerk' parameter is used as the acceleration end jerk setpoint.
With EFFECTIVE, the last programmed acceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.positiveAccelEndJerk' is used. This default can be modified by entering a value in the 'positiveAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

positiveAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	17
System default:	100.0

Specifies the acceleration end jerk setpoint; the evaluation depends on the 'positiveAccelEndJerkType' parameter.
It is only used when parameter 'positiveAccelEndJerkType:=USER_DEFAULT' or 'positiveAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	18
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration start jerk.
With DIRECT, the value set in the 'negativeAccelStartJerk' parameter is used as the deceleration start jerk setpoint.
With EFFECTIVE, the last programmed deceleration start jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelStartJerk' is used. This default can be modified by entering a value in the 'negativeAccelStartJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelStartJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	19
System default:	100.0

Specifies the deceleration start jerk setpoint; the evaluation depends on the 'negativeAccelStartJerkType' parameter.
It is only used when parameter 'negativeAccelStartJerkType:=USER_DEFAULT' or 'negativeAccelStartJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerkType (optional)

Direction:	Input parameters
Data type:	EnumJerk
Parameter index:	20
System default:	USER_DEFAULT

EnumJerk

DIRECT (40)	Value entry
EFFECTIVE (45)	Last programmed jerk
USER_DEFAULT (149)	User default

Type of definition for deceleration end jerk.
With DIRECT, the value set in the 'negativeAccelEndJerk' parameter is used as the deceleration end jerk setpoint.
With EFFECTIVE, the last programmed deceleration deceleration end jerk setpoint is used.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.dynamics.negativeAccelEndJerk' is used. This default can be modified by entering a value in the 'negativeAccelEndJerk' parameter. The value of the parameter is interpreted as a percentage.
With the velocity profile setting TRAPEZOIDAL (parameter 'velocityProfile'), the jerk entry is ignored.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

negativeAccelEndJerk (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	21
System default:	100.0

Specifies the deceleration end jerk setpoint; the evaluation depends on the 'negativeAccelEndJerkType' parameter.
It is only used when parameter 'negativeAccelEndJerkType:=USER_DEFAULT' or 'negativeAccelEndJerkType:=DIRECT'.
In all other cases, the parameter is irrelevant.
(Regarding acceleration/deceleration/jerk: (See Description of Functions Motion Control Technology Objects Electric/Hydraulic Axis, External Encoder)

velocityProfile (optional)

Direction:	Input parameters
Data type:	EnumPathVelocityProfile
Parameter index:	22
System default:	USER_DEFAULT

EnumPathVelocityProfile

SMOOTH (124)	Continuous acceleration curve
TRAPEZOIDAL (146)	Trapezoidal velocity profile
USER_DEFAULT (149)	User default

Specifies the velocity setpoint profile.
Parameters TRAPEZOIDAL and SMOOTH define the velocity profile directly.
With USER_DEFAULT, the default velocity profile defined in system variable 'userDefault.dynamics.profile' is used.

w (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	23
System default:	0.0

Specifies the target position of the synchronous axis; the evaluation depends on the 'wMode' parameter.

wMode (optional)

Direction:	Input parameters
Data type:	EnumPathWMode
Parameter index:	24
System default:	USER_DEFAULT

EnumPathWMode

ABSOLUTE (1)	Absolute target position
RELATIVE (115)	Relative target position
USER_DEFAULT (149)	User default
OUTPUT_PATH_LENGTH (379)	Absolute path length
OUTPUT_PATH_LENGTH_ADDITIVE (380)	Additive path length

Type of target position specification of the synchronous axis.
With ABSOLUTE, the target position is specified as an absolute value.
With RELATIVE, the path to be traversed is specified.
With OUTPUT_PATH_LENGTH, the current traveled path length is output beginning from zero.
With OUTPUT_PATH_LENGTH_ADDITIVE, the current traveled path length is added to the existing value.
With USER_DEFAULT, the default setting defined in system variable 'userDefault.w.mode' is used.

wDirection (optional)

Direction:	Input parameters
Data type:	EnumPathWDirection
Parameter index:	25
System default:	USER_DEFAULT

EnumPathWDirection

BY_VALUE (24)	From sign of target position
NEGATIVE (85)	Negative
POSITIVE (107)	Positive
SHORTEST_WAY (121)	Shortest path
USER_DEFAULT (149)	User default

Specifies the motion direction of the synchronous axis.
With an absolute position specification ('positioningMode:=ABSOLUTE'), the direction is directly derived from the target position to be approached for linear and rotary axes.
The following applies to modulo axes as well as to linear and rotary axes with relative positioning ('positioningMode:=RELATIVE'):

With BY_VALUE, the direction is determined from the sign of the position specification.
With POSITIVE, motion is in the positive direction relative to the axis coordinate system.
With NEGATIVE, motion is in the negative direction relative to the axis coordinate system.
With SHORTEST_WAY, the shortest path is taken on modulo axes.
With SHORTEST_WAY and relative positioning ('positioningMode:=RELATIVE'), the direction is determined from the sign of the position specification for linear and rotary axes.
With USER_DEFAULT, the default direction defined in system variable 'userDefault.w.direction' is used.

wDynamics (optional)

Direction:	Input parameters
Data type:	EnumPathWDynamics
Parameter index:	26
System default:	MAX_DYNAMICS

EnumPathWDynamics

CURRENT (33)	Dynamic response values of the last command
MAX_DYNAMICS (418)	Maximum axis dynamic response values

Selection of the dynamic response of the synchronous axis.
With MAX_DYNAMICS, the end position with the maximum axis values is approached.
With CURRENT, the end position with the dynamic response values of the last command is approached.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextEnablePathObjectMotionTracking
Parameter index:	27
System default:	IMMEDIATELY

EnumNextEnablePathObjectMotionTracking

IMMEDIATELY (60)	Effective immediately
SYNCHRONIZED (216)	After synchronization of coordinate systems
AT_SYNCHRONIZING_START (419)	Motion start

Specifies condition for transition to next command.
This defines the time at which the command is returned to the program that issued it. Therefore commands issued synchronously and asynchronously are possible.
With IMMEDIATELY, the transition takes place immediately.
With SYNCHRONIZED, transition takes place once 'synchronous' status is reached.
With AT_SYNCHRONIZING_START, transition occurs when the synchronization motion takes effect.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	28
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.66.2 _getPathObjectBcsFromOcsData

This function is used to determine the coordinates in the basic coordinate system using the coordinates in the object coordinate system.

Return value: StructRetGetPathObjectBcsFromOcsData

StructRetGetPathObjectBcsFromOcsData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
position	Position	StructPathVector	-

StructPathVector
Trace data

Structure	Name	Data type	Unit
x	x coordinate	LREAL	-
y	y coordinate	LREAL	-
z	z coordinate	LREAL	-

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

ocsNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the object coordinate system.

x

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

x coordinate in the object coordinate system

y

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

y coordinate in the object coordinate system

z

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5

z coordinate in the object coordinate system

1.3.66.3 _getPathObjectOcsFromBcsData

This function is used to determine the coordinates in the object coordinate system using the coordinates in the basic coordinate system.

Return value: StructRetGetPathObjectOcsFromBcsData

StructRetGetPathObjectOcsFromBcsData

Structure	Name	Data type	Unit
functionResult	Error code	DINT	-
position	Position	StructPathVector	-

StructPathVector
Trace data

Structure	Name	Data type	Unit
x	x coordinate	LREAL	-
y	y coordinate	LREAL	-
z	z coordinate	LREAL	-

Return value 'functionResult'
Description of the return values:
Corresponds to the list of return values under the _move function.

Parameter:
pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

ocsNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the object coordinate system.

x

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

x coordinate in the path coordinate system

y

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4

y coordinate in the path coordinate system

z

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5

z coordinate in the path coordinate system

1.3.66.4 _redefinePathObjectOcs

This function sets the position of the object coordinate system in relation to the basic coordinate system with reference to the x-direction of the basic coordinate system or the motional sequence reference value. This function can only be used with non-synchronized coordinate systems.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

ocsNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the object coordinate system used.

mode (optional)

Direction:	Input parameters
Data type:	EnumPathObjectRedefineOcsMode
Parameter index:	3
System default:	RELATIVE

EnumPathObjectRedefineOcsMode

ABSOLUTE (1)	Absolute value
RELATIVE (115)	Relative value

Specifies the type of assignment for the new position.
With ABSOLUTE, the value is assigned as an absolute position.
With RELATIVE, the value is added to the existing position (offset).

value (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	4
System default:	0.0

Specifies the position, evaluation depends on the 'mode' parameter.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	5
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, transition takes place after the value has been set.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

1.3.66.5 _setPathObjectOcs

The function sets the object coordinate system to the specified frame vector from the basic coordinate system (BCS) in relation to the object coordinate system (OCS), and sets the reference to the technology object whose motion output is the motional sequence reference value.
The OCS is then tracked at the motional sequence reference value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values under the _move function.

Parameter:

pathObject

Direction:	Input parameters
Data type:	_PathObjectType
Parameter index:	1

Specifies the technology object of the 'pathInterpolation' type or a variable of the '_PathObjectType on which the command is to be executed.

ocsNumber

Direction:	Input parameters
Data type:	DINT
Parameter index:	2

Specifies the number of the object coordinate system used.

trackingIn (optional)

Direction:	Input parameters
Data type:	MasterType
Parameter index:	3
System default:	0

Specifies the technology object providing the position value for the motional sequence.

ocsSettingType (optional)

Direction:	Input parameters
Data type:	EnumPathObjectOcsSettingType
Parameter index:	4
System default:	USER_DEFAULT

EnumPathObjectOcsSettingType

DIRECT (40)	Value entry
USER_DEFAULT (149)	User default

Type of specification for the frame vector from the basic coordinate system in relation to the object coordinate system.
With DIRECT, the 'roll', 'pitch' and 'yaw' parameters specified in the 'x', 'y', 'z' parameters take effect.
With USER_DEFAULT, the values set in the userDefaultOcs[i] frame parameters are used.

x (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	5
System default:	0.0

Specifies the offset from the basic coordinate system to the object coordinate system in the x-direction.

y (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	6
System default:	0.0

Specifies the offset from the basic coordinate system to the object coordinate system in the y-direction.

z (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	7
System default:	0.0

Specifies the offset from the basic coordinate system to the object coordinate system in the z-direction.

roll (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	8
System default:	0.0

Specifies the rotation from the basic coordinate system to the object coordinate system about the x-vector.

pitch (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	9
System default:	0.0

Specifies the rotation from the basic coordinate system to the object coordinate system about the already rotated y-vector.

yaw (optional)

Direction:	Input parameters
Data type:	LREAL
Parameter index:	10
System default:	0.0

Specifies the rotation from the basic coordinate system to the object coordinate system about the twice rotated z-vector.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumNextCommandEnable
Parameter index:	11
System default:	IMMEDIATELY

EnumNextCommandEnable

IMMEDIATELY (60)	Immediate command transition
WHEN_COMMAND_DONE (160)	When command is finished or aborted

Specifies condition for transition to next command.
With IMMEDIATELY, the transition takes place immediately.
With WHEN_COMMAND_DONE, transition takes place after the object coordinate system has been set.

commandId (optional)

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	12
System default:	SystemId_low = 0 SystemId_high = 0

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

System Functions - TP TControl 2

2.1 Technology

2.1.1 Temperature channel - Identification

2.1.1.1 _calculateTControllerParameter

This function calculates new controller parameters from the controlled-system parameters determined by self tuning.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

identificationType (optional)

Direction:	Input parameters
Data type:	EnumTControllerIdentificationType
Parameter index:	3
System default:	MODIFIED_TANGENTMETHOD

EnumTControllerIdentificationType

NONE (93)	None
MODIFIED_TANGENTMETHOD (605)	Modified tangent method
STANDARD_TANGENTMETHOD (606)	Standard tangent method

The type of self-tuning is selected, and the result is used to calculate the controller parameters.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.1.2 _forceTControllerIdentification

This function forces the transition to the next stage of parameter identification.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	3

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.1.3 _setTControllerActualIdentificationType

This function sets the current controlled-system-identification type.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

actualIdentificationType

Direction:	Input parameters
Data type:	EnumTControllerIdentificationType
Parameter index:	3

EnumTControllerIdentificationType

NONE (93)	None
MODIFIED_TANGENTMETHOD (605)	Modified tangent method
STANDARD_TANGENTMETHOD (606)	Standard tangent method

Current controlled-system-identification type

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.1.4 _setTControllerIdentificationModifiedTangentMethodParameter

This function sets the parameters for the controlled-system identification according to the modified inflectional tangent method.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

identificationParameter

Direction:	Input parameters
Data type:	StructTControllerIdentificationModifiedTangent-MethodParameter
Parameter index:	3

StructTControllerIdentificationModifiedTangentMethodParameter
Parameter for controlled-system identification according to the modified tangent method

Structure	Name	Data type	Unit
transitionMode	Determines the type of the transition between the selected phases of the identification	EnumTControllerIden-tificationTransiti-onMode	-
minimumStepSize	Minimum step size	LREAL	Unit set by the user
startCondition	Start condition	StructTControllerIden-tificationStaticCondi-tion	-

StructTControllerIdentificationStaticCondition
Condition for a static state (idle system)

Structure	Name	Data type	Unit
permissableTempera-tureChange	Tolerated temperature change in the wait time	LREAL	Unit set by the user
waitingTime	Wait time in which the actual value must remain stable	LREAL	Unit set by the user

EnumTControllerIdentificationTransitionMode

AUTOMATICALLY (607)	Automatic
BY_COMMAND (608)	Forced transition with an ST command

Controlled-system-identification parameter

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	7
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController para-meters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.1.5 _setTControllerIdentificationModifiedTangentMethodProcessParameter

This function sets the controlled-system parameters according to the modified inflectional tangent method.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

processParameter

Direction:	Input parameters
Data type:	StructTControllerIdentificationModifiedTangent-MethodProcessParameter
Parameter index:	3

StructTControllerIdentificationModifiedTangentMethodProcessParameter
Controlled-system parameter

Structure	Name	Data type	Unit
delayTime	Delay time	LREAL	Unit set by the user
maximumGradient	Maximum tempera-ture rise	LREAL	Unit set by the user

Controlled-system parameter

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	5
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController para-meters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.1.6 _setTControllerIdentificationStandardTangentMethodParameter

This function sets the parameters for the controlled-system identification according to the standard inflectional tangent method.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

identificationParameter

Direction:	Input parameters
Data type:	StructTControllerIdentificationStandardTangentMethodParameter
Parameter index:	3

StructTControllerIdentificationStandardTangentMethodParameter
Parameter for identification according to the standard tangent method

Structure	Name	Data type	Unit
transitionMode	Determines the type of the transition between the selected phases of the identification	EnumTControllerIdentificationTransitionMode	-
startCondition	Condition for the start of the identification	StructTControllerIdentificationStaticCondition	-
endCondition	Condition for the end of the identification	StructTControllerIdentificationStaticCondition	-
outputValue	Output value	LREAL	Unit set by the user

StructTControllerIdentificationStaticCondition
Condition for a static state (idle system)

Structure	Name	Data type	Unit
permissableTemperatureChange	Tolerated temperature change in the wait time	LREAL	Unit set by the user
waitingTime	Wait time in which the actual value must remain stable	LREAL	Unit set by the user

EnumTControllerIdentificationTransitionMode

AUTOMATICALLY (607)	Automatic
BY_COMMAND (608)	Forced transition with an ST command

Controlled-system-identification parameter

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	9
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	10

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.1.7 _setTControllerIdentificationStandardTangentMethodProcessParameter

This function sets the controlled-system parameters according to the standard inflectional tangent method.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

processParameter

Direction:	Input parameters
Data type:	StructTControllerIdentificationStandardTangentMethodProcessParameter
Parameter index:	3

StructTControllerIdentificationStandardTangentMethodProcessParameter
Controlled-system parameter

Structure	Name	Data type	Unit
delayTime	Delay time	LREAL	Unit set by the user
riseTime	Rise time	LREAL	Unit set by the user
gain	Process gain	LREAL	Unit set by the user

Controlled-system parameter

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	6
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.2 Temperature channel - Object and Alarm Handling

2.1.2.1 _resetTController

This function resets the technology object without resetting the parameters to the configuration data.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'TemperatureControllerType' on which the command is to be executed.

dataResetMode (optional)

Direction:	Input parameters
Data type:	EnumTControllerDataResetMode
Parameter index:	2
System default:	DO_NOT_CHANGE

EnumTControllerDataResetMode

ACTIVATE_CONFIGURATION_DATA (3)	Resets technology object and loads current configuration data
DO_NOT_CHANGE (43)	Do not change parameters set via a command

Specifies the type of reset.
ACTIVATE_CONFIGURATION_DATA resets the technology objects and loads the configuration parameters.
DO_NOT_CHANGE resets the technology objects without loading the configuration parameters.

activateRestart (optional)

Direction:	Input parameters
Data type:	EnumTControllerRestartActivation
Parameter index:	3
System default:	NO_RESTART_ACTIVATION

EnumTControllerRestartActivation

NO_RESTART_ACTIVATION (293)	Do not perform a TO restart
ACTIVATE_RESTART (294)	Perform a TO restart

Specifies whether or not a TO restart should be executed.
With NO_RESTART_ACTIVATION, the technology object is not restarted.
With ACTIVATE_RESTART, the technology object is restarted.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.2.2 _resetTControllerError

This function resets an alarm.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

errorResetMode (optional)

Direction:	Input parameters
Data type:	EnumTControllerErrorResetMode
Parameter index:	3
System default:	ALL_ERRORS

EnumTControllerErrorResetMode

ALL_ERRORS (10)	All errors are reset
SPECIFIC_ERROR (126)	A specific error is reset

Specifies which errors are reset.
With ALL_ERRORS, all errors are reset.
With SPECIFIC_ERROR, a specific error is reset.

errorNumber (optional)

Direction:	Input parameters
Data type:	DINT
Parameter index:	4
System default:	-1

Specifies the number of the error to be reset.
Requires programming with 'errorResetMode:=SPECIFIC_ERROR'.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.2.3 _setTControllerOperatingMode

This function sets the controller mode.

Return value: DINT

Description of the return value:

0 -
No error

1 -
Illegal command parameter

2 -
Illegal range specification in command parameters

3 -
Command aborted

4 -
Unknown command

5 -
Command cannot be executed due to current object status

6 -
Command aborted due to termination of user task

7 -
Command rejected due to suspension of command interpretation of the addressed technology object

8 -
Command aborted due to full command buffer

9 -
Insufficient memory

10 -
A connection to a technology object required for this operation does not exist

11 -
No object configuration

12 -
The error cannot be reset due to its configuration

13 -
Axis is not homed

14 -
Measurement job not possible on virtual axis

15 -
Ambiguous 'commandId'

16 -
Command not implemented

17 -
Read access denied

18 -
Write access denied

19 -
Command argument not supported

20 -
The cam has already been interpolated and cannot be manipulated

21 -
The interpolation condition was violated

22 -
The programmed jerk is 0

23 -
The alarm to be deleted is not active

24 -
The command is not possible on a virtual axis

25 -
Synchronized start of this command is not possible

26 -
Higher-level command was aborted because it is not permitted by the active command

27 -
Timeout during communication with the drive

28 -
Actual values are not valid

29 -
This command cannot be executed when velocity control is active

30 -
This command cannot be executed when position control is active

31 -
This command cannot be executed in torque-reduced operation or during travel to fixed limit stop

32 -
This command can only be executed when force or pressure control is active

33 -
This command cannot be executed when force or pressure control is active

34 -
This command can only be executed when pressure limiting is active

35 -
Master values are not valid

36 -
Slave values are not valid

37 -
No slave value can be defined for a master value

38 -
No master value can be defined for a slave value

39 -
This command cannot be executed when synchronous operation is inactive

40 -
This command cannot be executed because of a synchronization error.

41 -
This command cannot be executed when gearing or camming is active

42 -
This command cannot be executed when camming is inactive

43 -
This command can only be used for an interpolated cam

44 -
This command cannot be executed when pressure limiting is active

45 -
There are not enough interpolation points for interpolation of the cam

46 -
Specified path location cannot be reached due to kinematic limitations

47 -
Path axis values are not valid

48 -
Maximum number of active commands exceeded

49 -
Command only possible on a technology object local to the CPU

10000 (greater or equal to) internal error

__2000 - Command permitted only if heating and cooling controller has been configured
__2001 - Command permitted only if ADVANCE controller type has been configured
__2002 - Command permitted only if heating controller has been configured
__2100 - Incorrect parameter in command
__2101 - Incorrect 'commandDestination' parameter in command
__2102 - Incorrect 'nextCommand' parameter in command
__2120 - Error in 'OperatingMode' parameter
__2130 - Error when setting 'Setpoint'
__2140 - Error when setting 'ManualOutputValue'
__2150 - Inconsistent parameter when setting 'InputFilterParameter'
__2151 - Error in 'controllerTimeConstant' parameter when setting 'InputFilterParameter'
__2152 - Error in 'displayTimeConstant' parameter when setting 'InputFilterParameter'
__2155 - Inconsistent parameter when setting 'InputDisplayValueParameter'
__2156 - Error in 'initMaxDisplayValue' parameter when setting 'InputDisplayValueParameter'
__2157 - Error in 'initMinDisplayValue' parameter when setting 'InputDisplayValueParameter'
__2158 - 'initMinDisplayValue' parameter greater than 'initMaxDisplayValue'
__2160 - Inconsistent parameter when setting 'InputLimitCheckParameter'
__2161 - Error in 'innerLimitCheck.upperLimitValue' parameter when setting 'InputLimitCheckParameter'
__2162 - Error in 'innerLimitCheck.lowerLimitValue' parameter when setting 'InputLimitCheckParameter'
__2163 - Error in 'outerLimitCheck.upperLimitValue' parameter when setting 'InputLimitCheckParameter'
__2164 - Error in 'outerLimitCheck.lowerLimitValue' parameter when setting 'InputLimitCheckParameter'
__2165 - Error in 'innerLimitCheck.upperLimitMode' parameter when setting 'InputLimitCheckParameter'
__2166 - Error in 'innerLimitCheck.lowerLimitMode' parameter when setting 'InputLimitCheckParameter'
__2167 - Error in 'outerLimitCheck.upperLimitMode' parameter when setting 'InputLimitCheckParameter'
__2168 - Error in 'outerLimitCheck.lowerLimitMode' parameter when setting 'InputLimitCheckParameter'
__2170 - Inconsistent parameter when setting 'InputGradientCheckParameter'
__2171 - Error in 'checkMode' parameter when setting 'InputGradientCheckParameter'
__2172 - Error in 'numberOfToleratedViolations' parameter when setting 'InputGradientCheckParameter'
__2173 - Error in 'maxPositiveGradient' parameter when setting 'InputGradientCheckParameter'
__2174 - Error in 'maxNegativeGradient' parameter when setting 'InputGradientCheckParameter'
__2180 - Inconsistent parameter when setting 'DPIDParameter'
__2181 - Error in 'gain' parameter when setting 'DPIDParameter'
__2182 - Error in 'integrationMode' parameter when setting 'DPIDParameter'
__2183 - Error in 'integrationTime' parameter when setting 'DPIDParameter'
__2184 - Error in 'derivativeActionTime' parameter when setting 'DPIDParameter'
__2185 - Error in 'derivativeActionTime2' parameter when setting 'DPIDParameter'
__2186 - Error in 'integratorResetValue' parameter when setting 'DPIDParameter'
__2187 - Error in 'integrator' parameter when setting the 'Integrator'
__2190 - Error in 'cycletime' parameter when setting 'CycleParameter'
__2200 - Inconsistent parameter when setting 'ControlRangeParameter'
__2201 - Error in 'upperControlRange' parameter when setting 'ControlRangeParameter'
__2202 - Error in 'lowerControlRange' parameter when setting 'ControlRangeParameter'
__2210 - Inconsistent parameter when setting 'ProcessModeParameter'
__2211 - Error in 'processMode' parameter when setting 'ProcessModeParameter'
__2220 - Inconsistent parameter when setting 'PWMParameter'
__2221 - Error in 'upperOperatingThreshold' parameter when setting 'PWMParameter'

__2222 - Error in 'lowerOperatingThreshold' parameter when setting 'PWMParameter'
__2230 - Inconsistent parameter when setting 'PlausibilityParameter'
__2231 - Error in 'checkMode' parameter when setting 'PlausibilityParameter'
__2232 - Error in 'minimalTemperatureGradient' parameter when setting 'PlausibilityParameter'
__2233 - Error in 'upperLimitTemperatureValue' parameter when setting 'PlausibilityParameter'
__2234 - Error in 'lowerLimitTemperatureValue' parameter when setting 'PlausibilityParameter'
__2235 - Error in 'lowerLimitOutputValue' parameter when setting 'PlausibilityParameter'
__2236 - Error in 'delayTime' parameter when setting 'PlausibilityParameter'
__2240 - Error in 'Direction' parameter on command '_copyTControllerShadow'
__2250 - Invalid command
__2260 - Error in logical address
__2270 - Inconsistent parameter when executing command '_calculateTControllerParameter'
__2271 - Error in 'maximumGradient' parameter when executing command '_calculateTControllerParameter'
__2272 - Error in 'delayTime' parameter when executing command '_calculateTControllerParameter'
__2273 - Error in 'gain' parameter when executing command '_calculateTControllerParameter'
__2274 - Error in 'riseTime' parameter when executing command '_calculateTControllerParameter'
__2280 - Inconsistent parameter when setting 'IdentificationModifiedTangentMethodParameter'
__2281 - Error in 'transitionMode' parameter when setting 'IdentificationModifiedTangentMethodParameter'
__2282 - Error in 'minimumStepSize' parameter when setting 'IdentificationModifiedTangentMethodParameter'
__2283 - Error in 'permissableTemperatureChange' parameter when setting 'IdentificationModifiedTangentMethodParameter'
__2284 - Error in 'waitingTime' parameter when setting 'IdentificationModifiedTangentMethodParameter'
__2290 - Inconsistent parameter when setting 'IdentificationModifiedTangentMethodProcessParameter'
__2291 - Error in 'delayTime' parameter when setting 'IdentificationModifiedTangentMethodProcessParameter'
__2292 - Error in 'maximumGradient' parameter when setting 'IdentificationModifiedTangentMethodProcessParameter'
__2300 - Inconsistent parameter when setting 'IdentificationStandardTangentMethodParameter'
__2301 - Error in 'transitionMode' parameter when setting 'IdentificationStandardTangentMethodParameter'
__2302 - Error in 'minimumStepSize' parameter when setting 'IdentificationStandardTangentMethodParameter'
__2303 - Error in 'startCondition.permissableTemperatureChange' parameter when setting 'IdentificationStandardTangentMethodParameter'
__2304 - Error in 'startCondition.waitingTime' parameter when setting 'IdentificationStandardTangentMethodParameter'
__2305 - Error in 'endCondition.permissibleTemperatureChange' parameter when setting 'IdentificationStandardTangentMethodParameter'
__2306 - Error in 'endCondition.waitingTime' parameter when setting 'IdentificationStandardTangentMethodParameter'
__2307 - Error in 'outputValue' parameter when setting 'IdentificationStandardTangentMethodParameter'
__2310 - Inconsistent parameter when setting 'IdentificationStandardTangentMethodProcessParameter'
__2311 - Error in 'delayTime' parameter when setting 'IdentificationStandardTangentMethodProcessParameter'
__2312 - Error in 'riseTime' parameter when setting 'IdentificationStandardTangentMethodProcessParameter'
__2313 - Error in 'gain' parameter when setting 'IdentificationStandardTangentMethodProcessParameter'
__2320 - Switching to next phase is not permitted in the current phase
__2321 - The identification type for this command is not available
__2322 - This command cannot be executed because identification is in process
__2323 - The current type of identification is NONE
__2330 - Unknown identification type
__2340 - Unknown mode for restart activation due to modified configuration
__2341 - Unknown mode for restart activation
__2500 - Minimum and/or maximum values violated more than once
__2501 - Maximum time value violated

__2502 - Maximum time value violated
__2503 - Maximum temperature value violated
__2504 - Minimum temperature value violated
__2505 - Minimum output value violated
__2506 - Maximum output value violated
__2510 - Maximum and/or minimum DPID parameters limits violated more than once
__2511 - Maximum gain limit violated
__2512 - Minimum gain limit violated
__2513 - Maximum integration time limit violated
__2514 - Minimum integration time limit violated
__2515 - Maximum integral-action time limit violated
__2516 - Minimum integral-action time limit violated
__2517 - Maximum integral-action time2 limit violated
__2518 - Minimum integral-action time2 limit violated
__2519 - Maximum cycle limit violated
__2520 - Minimum cycle limit violated
__2530 - Minimum cycle limit violated
__2531 - Maximum process gain limit violated
__2532 - Minimum process gain limit violated
__2533 - Maximum process gradient limit violated
__2534 - Minimum process gradient limit violated
__2535 - Maximum process dead time limit violated
__2536 - Minimum process dead time limit violated
__2537 - Maximum process rise time violated
__2538 - Minimum process rise time violated
__2539 - Command not permitted if a heating controller has been configured
__2540 - Command not permitted if a heating/cooling controller has been configured
__2541 - Command not permitted if a cooling controller has been configured

Parameter:
tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

operatingMode

Direction:	Input parameters
Data type:	EnumTControllerOperatingMode
Parameter index:	3

EnumTControllerOperatingMode

INACTIVE (61)	Closed-loop controller deactivated
CONTROL (601)	Closed-loop controller automatic operation
MEASURING_AND_MANUAL_OUTPUT (602)	Measurement operation, manual control variable output
MEASURING_AND_OUTPUT_ZERO (603)	Measurement operation, output 0
IDENTIFICATION (604)	Identification

Specifies the controller mode setting.
With INACTIVE, the closed-loop controller is deactivated.
With CONTROL, operation is controlled.
With MEASURING_AND_MANUAL_OUTPUT, the closed-loop controller is deactivated, only measuring is performed. The manual output value setting is output.
With MEASURING_AND_OUTPUT_ZERO, the closed-loop controller is deactivated, only measuring is performed. The output value is 0.
With IDENTIFICATION, the controller is deactivated, and the controller is starting identification mode.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3 Temperature channel - Parameter functions

2.1.3.1 _copyTControllerShadow

This function copies a complete data record of the parameters that can be modified by means of commands.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'TemperatureControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

Direction

Direction:	Input parameters
Data type:	EnumTControllerCopyShadow
Parameter index:	3

EnumTControllerCopyShadow

SHADOW_TO_DIRECT (652)	Copies data from the shadow buffer to the utilized TController parameters
DIRECT_TO_SHADOW (653)	Copies data from the TController to the shadow buffer
CONFIG_TO_SHADOW (654)	Copies data from the configuration data to the shadow buffer

Specifies the direction in which the parameters are copied

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	4

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.2 _setTControllerControlRangeParameter

This function sets a control range.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

controlRange

Direction:	Input parameters
Data type:	StructTControllerControlRangeParameter
Parameter index:	3

StructTControllerControlRangeParameter
Control range

Structure	Name	Data type	Unit
upperControlRange	Upper limit	LREAL	Unit set by the user
lowerControlRange	Lower limit	LREAL	Unit set by the user

Specifies the control range.
The parameters are relative to the setpoint.
If the actual value is within the control range, a standard manipulated variable is involved.
If the actual value is outside of the control range,

- the value at 100% is greater (for a heating-cooling controller)
- the value at 0% is less (for a heating controller)

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	5
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.3 _setTControllerCycleParameter

This function sets the scan time.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

cycleParameter

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the controller cycle.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.4 _setTControllerCycleParameterSecondary

This function sets the scan time.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

cycleParameter

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Controller cycle time.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.5 _setTControllerDPIDParameter

This function sets the DPID controller parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

DPIDParameter

Direction:	Input parameters
Data type:	StructTControllerDPIDParameter
Parameter index:	3

StructTControllerDPIDParameter
DPID controller parameter

Structure	Name	Data type	Unit
gain	Proportional component	LREAL	Unit set by the user
integrationMode	Integration mode	EnumTControllerIntegrationMode	-
integrationTime	Integration time	LREAL	Unit set by the user
derivativeActionTime	Derivative-action time (1st order)	LREAL	Unit set by the user
derivativeActionTime2	Derivative-action time (2nd order)	LREAL	Unit set by the user
integratorResetValue	Start value of the integrator after reset	LREAL	Unit set by the user

EnumTControllerIntegrationMode

ACTIVE (4)	Activated
INACTIVE (61)	Deactivated

Specifies the DPID controller parameter setting.
With ACTIVE, the DPID controller parameter is set.
With INACTIVE, the DPID controller parameter is not set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	9
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	10

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.6 _setTControllerDPIDParameterSecondary

This function sets the second DPID controller parameter set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

DPIDParameter

Direction:	Input parameters
Data type:	StructTControllerDPIDParameter
Parameter index:	3

StructTControllerDPIDParameter
DPID controller parameter

Structure	Name	Data type	Unit
gain	Proportional component	LREAL	Unit set by the user
integrationMode	Integration mode	EnumTControllerIntegrationMode	-
integrationTime	Integration time	LREAL	Unit set by the user
derivativeActionTime	Derivative-action time (1st order)	LREAL	Unit set by the user
derivativeActionTime2	Derivative-action time (2nd order)	LREAL	Unit set by the user
integratorResetValue	Start value of the integrator after reset	LREAL	Unit set by the user

EnumTControllerIntegrationMode

ACTIVE (4)	Activated
INACTIVE (61)	Deactivated

Specifies the setting of the second DPID controller parameter set.
With ACTIVE, the DPID controller parameter is set.
With INACTIVE, the DPID controller parameter is not set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	9
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	10

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.7 _setTControllerInputDisplayValueParameter

Set Display Value Parameter

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

displayValueParameter

Direction:	Input parameters
Data type:	StructTControllerInputDisplayValueParameter
Parameter index:	3

StructTControllerInputDisplayValueParameter
Parameter for extreme-value calculation

Structure	Name	Data type	Unit
initMaxDisplayValue	Start value for the maximum display actual value	LREAL	Unit set by the user
initMinDisplayValue	Start value for the minimum display actual value	LREAL	Unit set by the user

Parameter for extreme value calculation

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	5
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.8 _setTControllerInputFilterParameter

This function sets the parameters for the input filter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

filterParameter

Direction:	Input parameters
Data type:	StructTControllerInputFilterParameter
Parameter index:	3

StructTControllerInputFilterParameter
Actual-value-filter parameter

Structure	Name	Data type	Unit
controllerTimeCon-stant	Control actual value time constant	LREAL	Unit set by the user
displayTimeConstant	Display actual value time constant	LREAL	Unit set by the user

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	5
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	6

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.9 _setTControllerInputGradientCheckParameter

This function sets the parameter for the input gradient check.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

checkParameter

Direction:	Input parameters
Data type:	StructTControllerInputGradientCheckParameter
Parameter index:	3

StructTControllerInputGradientCheckParameter
Actual-value-gradient-check parameter

Structure	Name	Data type	Unit
checkMode	Test mode	EnumTControllerInputGradientCheckMode	-
numberOfToleranced-Violations	Number of ignored violations	UDINT	-
maxPositiveGradient	Maximum positive actual value gradient	LREAL	Unit set by the user
maxNegativeGradient	Maximum negative actual value gradient	LREAL	Unit set by the user

EnumTControllerInputGradientCheckMode

ACTIVE (4)	Activated
INACTIVE (61)	Deactivated

Specifies the input plausibility check setting.
ACTIVE means that the input plausibility check is set.
INACTIVE means that the input plausibility check is not set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	7
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	8

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.10 _setTControllerInputLimitCheckParameter

This function sets the parameter for the input tolerance check.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

checkParameter

Direction:	Input parameters
Data type:	StructTControllerInputLimitCheckParameter
Parameter index:	3

StructTControllerInputLimitCheckParameter
Parameter for tolerance check

Structure	Name	Data type	Unit
innerLimitCheck	Inner tolerances	StructTControllerInputSingleLimitCheckParameter	-
outerLimitCheck	Outer tolerances	StructTControllerInputSingleLimitCheckParameter	-

StructTControllerInputSingleLimitCheckParameter
Parameter for tolerance check

Structure	Name	Data type	Unit
upperLimitMode	Upper tolerance mode	EnumTControllerInputLimitCheckMode	-
upperLimitValue	Upper tolerance value	LREAL	Unit set by the user
lowerLimitMode	Lower tolerance mode	EnumTControllerInputLimitCheckMode	-
lowerLimitValue	Lower tolerance value	LREAL	Unit set by the user

EnumTControllerInputLimitCheckMode

ABSOLUTE (1)	Absolute specified value
INACTIVE (61)	Deactivated
RELATIVE (115)	Relative specified value

Specifies the input plausibility check setting.
With ABSOLUTE, the value of the input plausibility check is set to a defined temperature.
With INACTIVE, the input plausibility check is not set.
With RELATIVE, the value of the input plausibility check is set relative to the setpoint.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	11
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	12

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.11 _setTControllerLowerPlausibilityParameter

This function sets the parameters for the controller gradient check (lower limits).

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

PlausibilityParameter

Direction:	Input parameters
Data type:	StructTControllerPlausibilityParameter
Parameter index:	3

StructTControllerPlausibilityParameter
Gradient check parameter

Structure	Name	Data type	Unit
plausibilityMode	Test mode	EnumTControllerPlausibilityCheckMode	-
minimalTemperature-Gradient	Minimum permitted temperature gradient	LREAL	Unit set by the user
upperLimitTemperatureValue	Upper temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitTemperatureValue	Lower temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitOutputValue	Lower control value limit for the plausibility check	LREAL	Unit set by the user
delayTime	Delay time	LREAL	Unit set by the user

EnumTControllerPlausibilityCheckMode

ACTIVE (4)	Activated
INACTIVE (61)	Deactivated

Specifies the controller plausibility check setting (lower limits)
ACTIVE means that the control plausibility check is set.
INACTIVE means that the controller plausibility check check is not set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	9
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	10

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.12 _setTControllerLowerPlausibilityParameterSecondary

This function sets the parameters for the controller gradient check (lower limits, secondary controller).

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

PlausibilityParameter

Direction:	Input parameters
Data type:	StructTControllerPlausibilityParameter
Parameter index:	3

StructTControllerPlausibilityParameter
Gradient check parameter

Structure	Name	Data type	Unit
plausibilityMode	Test mode	EnumTControllerPlausibilityCheckMode	-
minimalTemperature-Gradient	Minimum permitted temperature gradient	LREAL	Unit set by the user
upperLimitTemperatureValue	Upper temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitTemperatureValue	Lower temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitOutputValue	Lower control value limit for the plausibility check	LREAL	Unit set by the user
delayTime	Delay time	LREAL	Unit set by the user

EnumTControllerPlausibilityCheckMode

ACTIVE (4)	Activated
INACTIVE (61)	Deactivated

Specifies the setting for the controller plausibility check (lower limits, secondary controller).
ACTIVE means that the control plausibility check is set.
INACTIVE means that the controller plausibility check check is not set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	9
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	10

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.13 _setTControllerProcessModeParameter

This function sets the process mode for adaptation.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

processMode

Direction:	Input parameters
Data type:	StructTControllerProcessModeParameter
Parameter index:	3

StructTControllerProcessModeParameter
Controller adaptation mode

Structure	Name	Data type	Unit
processMode	Process type	EnumTControllerProcessMode	-

EnumTControllerProcessMode

INACTIVE (61)	Deactivated
CONTINUOUS (621)	Continuous process
DISCONTINUOUS (622)	Discontinuous process

Information regarding process type setting for adaptive control.
With INACTIVE, adaptation is not set.
With CONTINUOUS, adaptation is set for a continuous process mode.
With DISCONTINUOUS, adaptation is set for a discontinuous process mode.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.14 _setTControllerPWMParameter

This function sets the PWM parameter.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

PWMParameter

Direction:	Input parameters
Data type:	StructTControllerPWMParameter
Parameter index:	3

StructTControllerPWMParameter
Parameter for PWM output

Structure	Name	Data type	Unit
upperOperatingThreshold	Upper operating threshold	LREAL	Unit set by the user
lowerOperatingThreshold	Lower operating threshold	LREAL	Unit set by the user
numberOfOutputCycles	Number of PWM blocks	UDINT	-

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	6
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.15 _setTControllerPWMParameterSecondary

This function sets the second PWM parameter set.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

PWMParameter

Direction:	Input parameters
Data type:	StructTControllerPWMParameter
Parameter index:	3

StructTControllerPWMParameter
Parameter for PWM output

Structure	Name	Data type	Unit
upperOperatingThreshold	Upper operating threshold	LREAL	Unit set by the user
lowerOperatingThreshold	Lower operating threshold	LREAL	Unit set by the user
numberOfOutputCycles	Number of PWM blocks	UDINT	-

Specifies the second PWM parameter set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	6
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	7

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.16 _setTControllerUpperPlausibilityParameter

This function sets the parameters for the controller gradient check (upper limits).

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

PlausibilityParameter

Direction:	Input parameters
Data type:	StructTControllerPlausibilityParameter
Parameter index:	3

StructTControllerPlausibilityParameter
Gradient check parameter

Structure	Name	Data type	Unit
plausibilityMode	Test mode	EnumTControllerPlausibilityCheckMode	-
minimalTemperature-Gradient	Minimum permitted temperature gradient	LREAL	Unit set by the user
upperLimitTemperatureValue	Upper temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitTemperatureValue	Lower temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitOutputValue	Lower control value limit for the plausibility check	LREAL	Unit set by the user
delayTime	Delay time	LREAL	Unit set by the user

EnumTControllerPlausibilityCheckMode

ACTIVE (4)	Activated
INACTIVE (61)	Deactivated

Specifies the setting for the controller plausibility check (upper limits).
ACTIVE means that the control plausibility check is set.
INACTIVE means that the controller plausibility check check is not set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	9
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	10

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.3.17 _setTControllerUpperPlausibilityParameterSecondary

This function sets the parameters for the controller gradient check (upper limits, secondary controller).

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

PlausibilityParameter

Direction:	Input parameters
Data type:	StructTControllerPlausibilityParameter
Parameter index:	3

StructTControllerPlausibilityParameter
Gradient check parameter

Structure	Name	Data type	Unit
plausibilityMode	Test mode	EnumTControllerPlausibilityCheckMode	-
minimalTemperature-Gradient	Minimum permitted temperature gradient	LREAL	Unit set by the user
upperLimitTemperatureValue	Upper temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitTemperatureValue	Lower temperature limit for the plausibility check	LREAL	Unit set by the user
lowerLimitOutputValue	Lower control value limit for the plausibility check	LREAL	Unit set by the user
delayTime	Delay time	LREAL	Unit set by the user

EnumTControllerPlausibilityCheckMode

ACTIVE (4)	Activated
INACTIVE (61)	Deactivated

Specifies the setting for the gradient check (upper limits, secondary controller).
ACTIVE means that the control plausibility check is set.
INACTIVE means that the controller plausibility check check is not set.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	9
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	10

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.4 Temperature channel - Control functions

2.1.4.1 _setTControllerIntegrator

This function sets the integrator value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

integrator

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the new integrator value setting.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.4.2 _setTControllerIntegratorSecondary

This function sets the value of the integrator for the secondary controller.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

integrator

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the new integrator value.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.4.3 _setTControllerManualOutputValue

This function sets the manual output value.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

manualOutputValue

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the manual output value set in MEASURING_AND_MANUAL_OUTPUT.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.

2.1.4.4 _setTControllerSetpoint

This function sets the setpoint temperature.

Return value: DINT

Description of the return value:
Corresponds to the list of return values for the _setTControllerOperatingMode function.

Parameter:

tController

Direction:	Input parameters
Data type:	TemperatureControllerType
Parameter index:	1

Specifies the technology object of the 'temperatureController' type or a variable of the 'Temperature-ControllerType' on which the command is to be executed.

nextCommand (optional)

Direction:	Input parameters
Data type:	EnumTControllerNextCommand
Parameter index:	2
System default:	WHEN_COMMAND_DONE

EnumTControllerNextCommand

IMMEDIATLY (60)	Immediate command transition
WHEN_BUFFER_READY (159)	Command transition after entry in the command queue
WHEN_COMMAND_DONE (160)	Command transition when command is finished or aborted
WHEN_DATA_ACTIVE (650)	Return when command has been executed

Specifies condition for transition to next command.
With IMMEDIATELY, the command is only executed if the command buffer is available; otherwise, an immediate transition takes place, and the return value is not equal to zero.
With WHEN_BUFFER_READY, transition takes place when the command has been buffered in the technology object.
With WHEN_COMMAND_DONE, transition takes place when the command data have been received in the technology object.
With WHEN_DATA_ACTIVE, transition takes place when the command data are used for control in the technology object.

setpoint

Direction:	Input parameters
Data type:	LREAL
Parameter index:	3

Specifies the setpoint.

commandDestinantion (optional)

Direction:	Input parameters
Data type:	EnumTControllerCommandDestination
Parameter index:	4
System default:	DIRECT

EnumTControllerCommandDestination

DIRECT (40)	Command targets the utilized TController parameters
SHADOW (651)	Command targets the shadow buffer

DIRECT means that the parameter set used in the TController technology object is received.
SHADOW means that the parameters are received in the shadow buffer.

commandId

Direction:	Input parameters
Data type:	CommandIdType
Parameter index:	5

CommandIdType
System-wide unique ID

Structure	Name	Data type	Unit
SystemId_low	Low_part of ID	UDINT	-
SystemId_high	High_part of ID	UDINT	-

Unique, system-wide 'commandId' to track command status.