BECKHOFF New Automation Technology

Manual | EN

TX1260

TwinCAT | NC I

2023-10-17 | Version: 1.3

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Introduction

The TwinCAT NCI stands for 'numerical control interpolation' and is the NC system for interpolated path movements.
TwinCAT NCI offers 3D interpolation (interpreter, set point generation, position controller), an integrated PLC with an NC interface and an I/O connection for axes via the fieldbus.

NCI can be used to drive 3 path axes and up to 5 auxiliary axes per channel. In addition, master/slave couplings can be formed. In combination with TwinCAT Kinematic Transformation (TS 511x), complex kinematic systems can be controlled via NCI.

Programming is done with a dedicated NC program, based on DIN 66025, with its own language extensions (cf. Interpreter (DIN 66025/G-Code) [▸ 29]) or directly from the PLC with the PLC Library: TcPlcInterpolation [▸ 223].

Installation preconditions

To be able to use TwinCAT NCI select the NCI Level when installing TwinCAT.

Overview

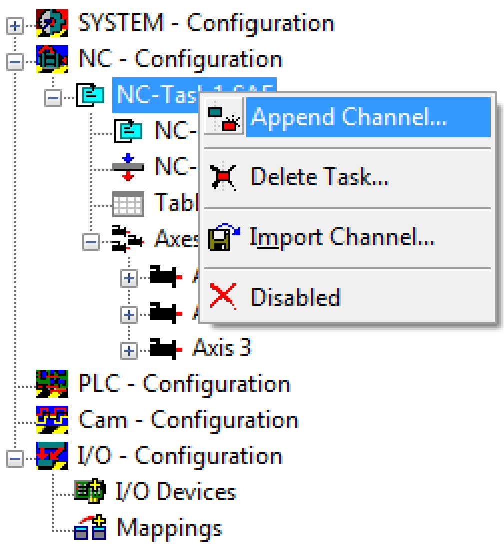
Chapter	Contents
System Manager user interface [▸ 13]	Description of the parameters and functions of the interpreter in the System Manager.
Interpreter [▸ 29]	Interpreter programming instructions.
PLC NCI libraries [▸ 101]	Description of the special NCI libraries.
Sample [▸ 240]	Example using PLC and NC program and System Manager configuration.
Appendix [▸ 247]	Parameterization, cyclic channel interface

3 User interface in the System Manager

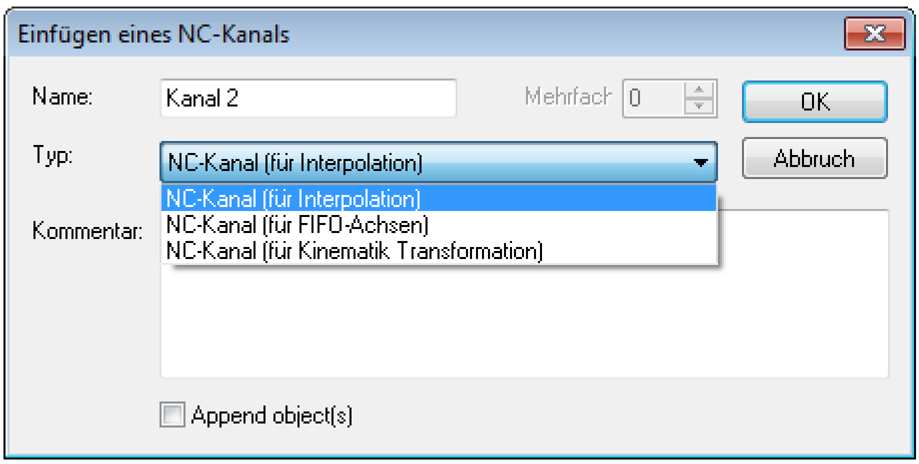
3.1 Outline

In order to be able to use the interpolation, add an interpolation channel in the System Manager. This applies to the interpreter and the library TcPlcInterpolation [▸ 223].

1. Create an NC channel.



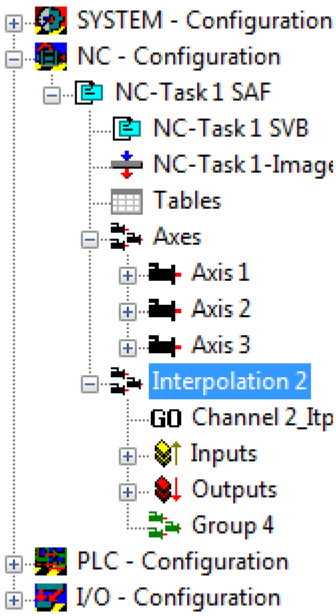
2. In the selection box select the NC channel for the interpolation.



3. Assign PTP axes to it from the PLC via a function block.

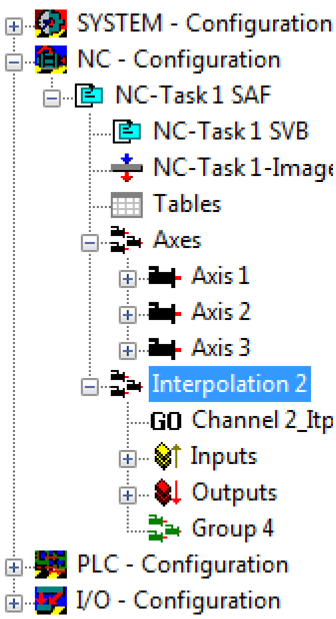
⇨ The created channel consists of the following elements:

Interpolation channel [▸ 14]	Description of the properties pages embedded in the 'interpolation' element
Interpreter element [▸ 16]	Description of the properties pages embedded in the 'Interpreter' element
Group element [▸ 22]	Description of the properties pages embedded in the 'group' element



Axis-specific parameters for NCI can be found in the axis parameterization under subitem 'NCI parameters'.

3.2 Interpolation Channel



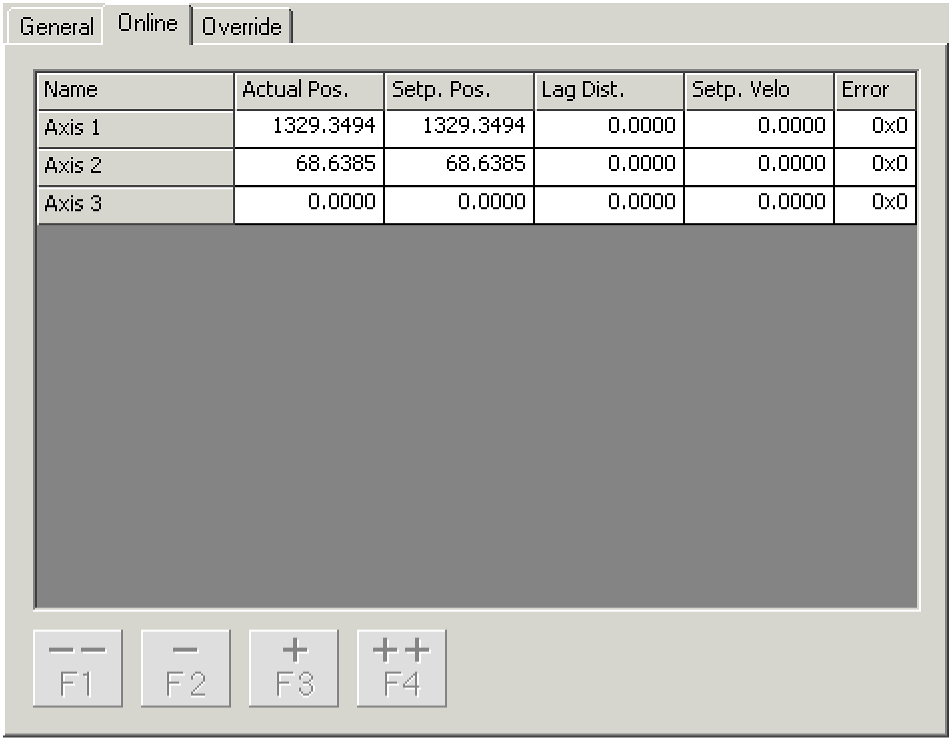
Click on the interpolation channel (in this case, 'Interpolation 2') to display the following dialogues:

"Online" tab

All the axes in the current Interpolation Group [▸ 22] will be listed. The following current parameters will be displayed:

- Actual positions
- Set positions
- Following errors
- Set velocities and

- Error codes

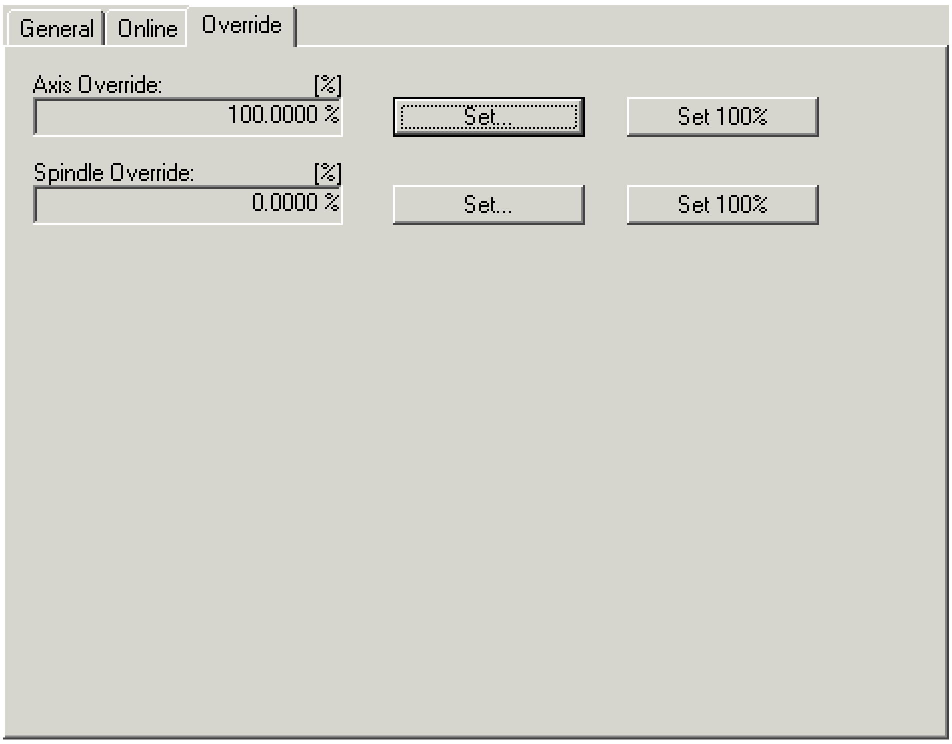


"Override" tab

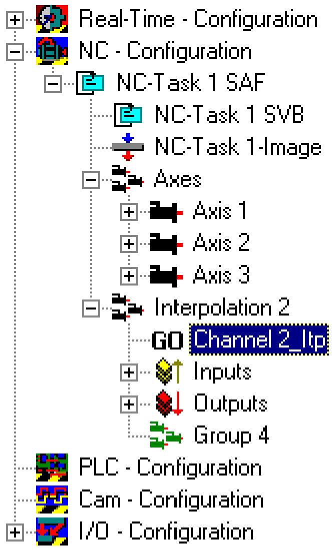
The channel override for the axes can be read and set on the 'Override' page. If PLC is running and the cyclical channel interface [▶ 251] is being written, the override set here will be overwritten by the PLC.

Further information on the override principle can be found under Path override (interpreter override types) [▶ 250].

The spindle override is described by the cyclic channel interface, although it is currently not supported.

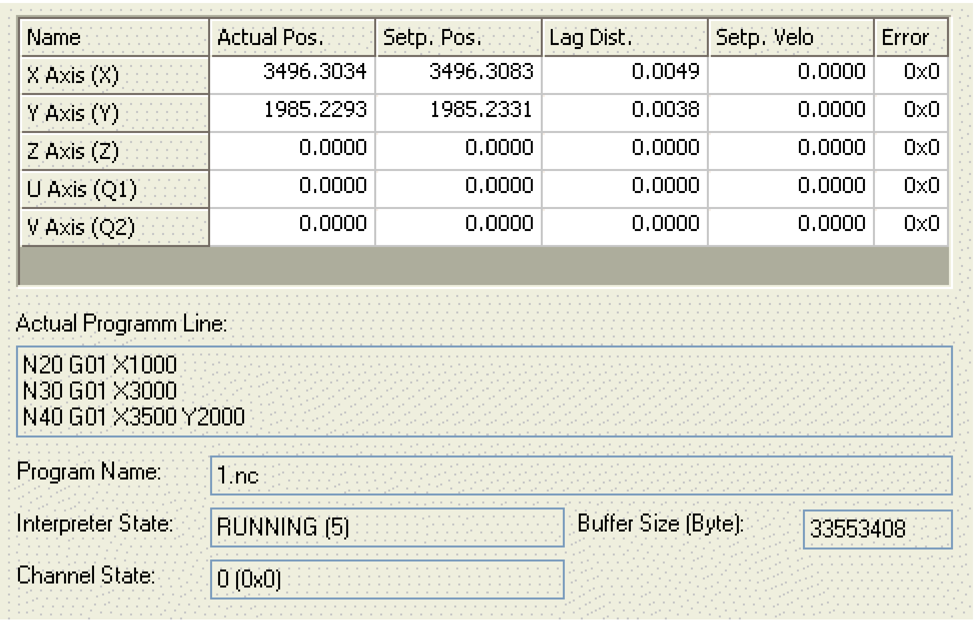


3.3 Interpreter Element



Click on the interpreter element (Channel2_Itp) to view the following properties pages and the online window:

3.3.1 Interpreter online window



Axes

As on the 'Online' properties page in the interpolation channel, this window lists all axes currently included in the interpolation group. The values are shown for:

- Actual positions
- Set positions
- Lag errors
- Set velocities and
- Current error codes

SEC program display

The SEC program display shows the current NC block to be processed in the set execution. The last row in the window is the current block.

As for nearly all the parameters, the program display can be read off via ADS. This can be used, for example, to indicate the current NC blocks within a Visual Basic application (see ADS device documentation - ADS Interface NC [▸ 257].

Program name

Displays the name of the currently loaded program. This does not necessarily have to be the program displayed in Editor.

Interpreter state

The interpreter state reflects the current state of the Interpreter State Machine. The complete list is given below. As PLC evaluation does not require all state information, only the most important parameters are explained.

State	Description
ITP_STATE_IDLE	The interpreter is in idle state if no NC program is loaded yet or if a group reset is executed. The interpreter also goes into idle state when a current program is stopped. In the case a group reset must be executed in order to prevent error 0x42C5. It is therefore recommended to execute a group reset after stopping via the PLC.
ITP_STATE_READY	After successful loading of an NC program, the interpreter is in ready state. After a program has been successfully processed and exited, the interpreter goes into ready state. In the meantime, however, other states are accepted.
ITP_STATE_ABORTED	If a runtime error occurs during the processing of an NC program, the interpreter goes into aborted state. The actual error code is given in Channel State.
ITP_STATE_SINGLESTOP	This state is **only** accepted in Single Block Mode [▸ 36]. As soon as the entry has been sent from the interpreter to the NC core, the interpreter goes into this mode.

Querying the interpreter state during program execution

Since the interpreter state may change between different states during program execution, we recommend querying it with a negative logic. During program execution the interpreter state is not necessarily ITP_STATE_RUNNING. If the program was executed successfully, the interpreter is subsequently always in Ready state. See Samples.

Interpreter state return values

```
0 ITP_STATE_INITFAILED
1 ITP_STATE_IDLE
2 ITP_STATE_READY
3 ITP_STATE_STARTED
4 ITP_STATE_SCANNING
5 ITP_STATE_RUNNING
6 ITP_STATE_STAY_RUNNING
7 ITP_STATE_WRITETABLE
8 ITP_STATE_SEARCHLINE
9 ITP_STATE_END
10 ITP_STATE_SINGLESTOP
11 ITP_STATE_ABORTING
12 ITP_STATE_ABORTED
13 ITP_STATE_FAULT
14 ITP_STATE_RESET
15 ITP_STATE_STOP
16 ITP_STATE_WAITFUNC
17 ITP_STATE_FLUSHBUFFERS
```

Channel state

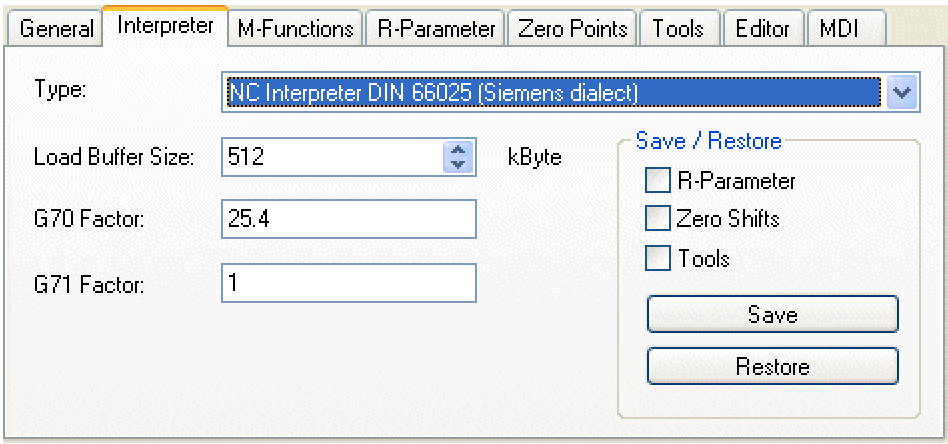
The channel state indicates the current error state of the channel. If an error occurs during NC program loading or runtime, the corresponding error code is displayed here. If, for example, an axis lag error occurs during processing, the NC program is stopped and the channel state will have a value unequal 0. The channel state should therefore always be checked in the PLC, in order to be able to respond to errors. The channel state is always 0 in error-free operation.

Loading buffer

The current size of the loading buffer for the interpreter is displayed here. Select the 'Interpreter' tab to change the value.

3.3.2 "Interpreter" tab

"Interpreter" tab



Type

The interpreter type can be selected in the 'Type' selection box. At present only the DIN 66025-based NC interpreter is available.

Loading buffer size

The loading buffer for the interpreter can be edited here. Note that the memory required in the interpreter is substantially greater than the size of the NC file. The maximum permitted loading buffer size is limited to 64 MB.

Changing the loading buffer size

If the size of the loading buffer is changed it is absolutely necessary to execute a TwinCAT restart.

G70 / G71 factor

If the part program is switched from G71 [▶ 34] (millimeter - default) to G70, the conversion factor is stored here. This only has to be edited if the base reference system is not millimeters.

If, for example, the machine was calibrated based on inches, and G70 is activated in the parts program, the G70 factor should be set to 1 and the G71 factor to 1/25.4

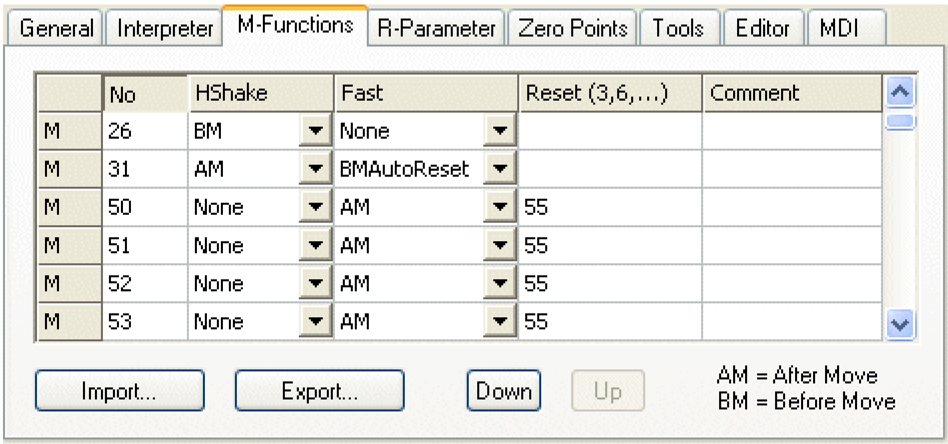
Save/Restore

The Save function can be used to save a 'snapshot' of the current parameters at runtime. The checkboxes can be used to specify the parameters to be saved. The Save function generates the file 'SnapShot.bin' in the TwinCAT\CNC directory.

The Restore function loads the file saved with the Save function.

This function is solely intended for debugging purposes

3.3.3 "M-Functions" tab



Use only with interpreter

This tab is irrelevant for operation with the library TcPlcInterpolation.

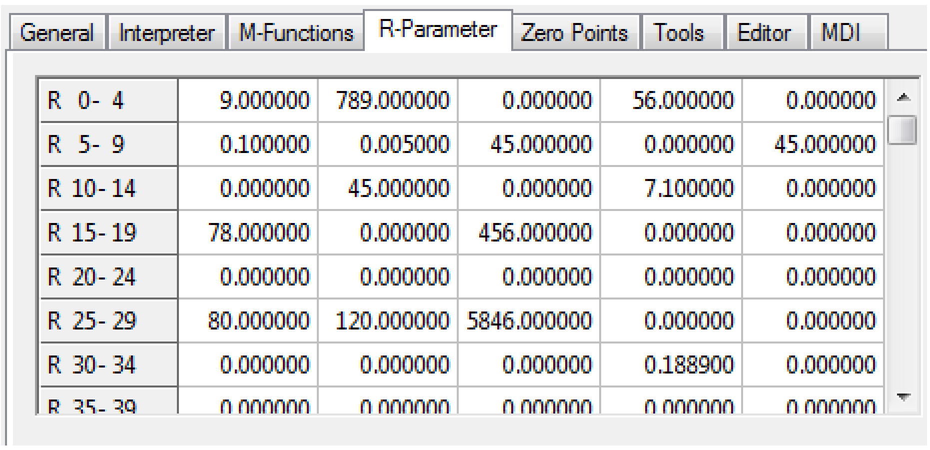
Shows the currently parameterized M-functions. On this page new M-functions can be added, or existing ones modified.

A more detailed description of the available parameters can be found in the interpreter description under M-functions [▶ 67].

Parameterization of M-functions

If M-functions are re-parameterized, subsequent activation of the configuration and a TwinCAT restart is required.

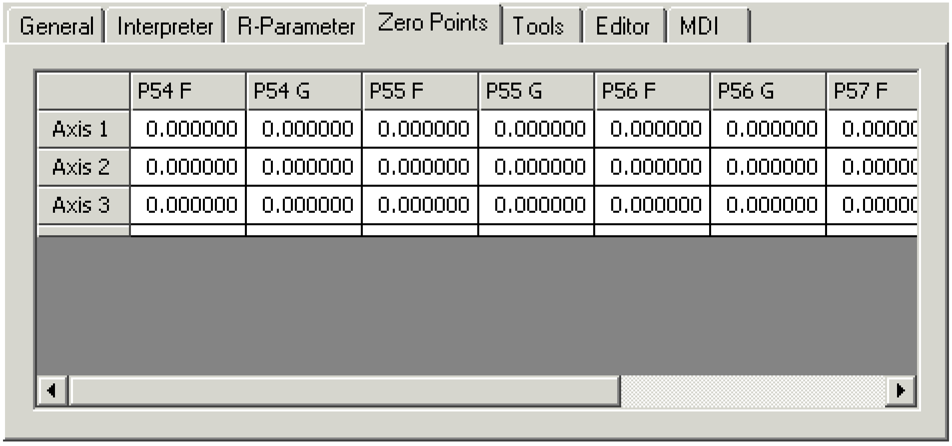
3.3.4 "R parameters" tab



The currently applicable R parameters are displayed on the 'R parameters' properties page. During the test phase it is possible to, for example, initialize or change R parameters here. R parameters are generally edited, however, from the NC program or if necessary, from the PLC.

You can find further information about R parameters in the interpreter description under R Parameters [▶ 37].

3.3.5 "Zero point" tab



	P54 F	P54 G	P55 F	P55 G	P56 F	P56 G	P57 F
Axis 1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
Axis 2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
Axis 3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000

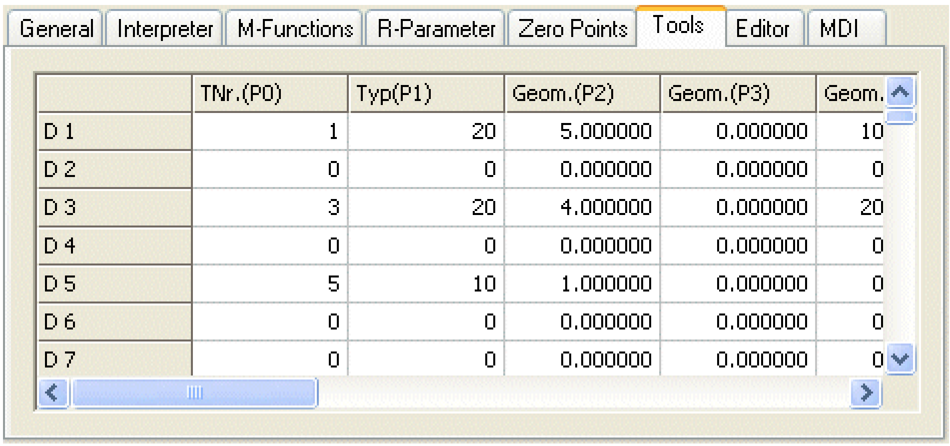
The current zero shift values for the axes within the interpolation group are displayed here. The parameters P54..P59 represent for the corresponding G code. As for the R parameters, the zero shift values can be edited from here.

NOTICE

Columns F & G (e.g. P54 F & P54 G) exist for historical reasons and are added for each parameter.

You can find further details of the effects in the interpreter description under zero shifts [▶ 46].

3.3.6 "Tools" tab

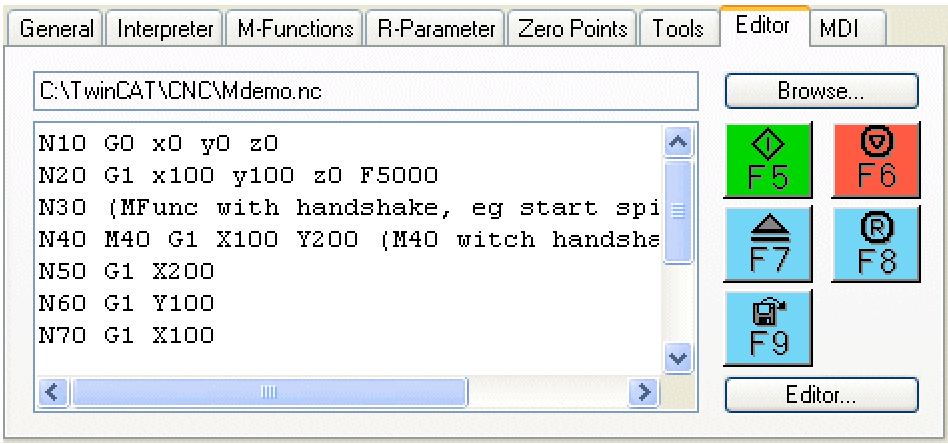


	TNr.(P0)	Typ(P1)	Geom.(P2)	Geom.(P3)	Geom.
D 1	1	20	5.000000	0.000000	10
D 2	0	0	0.000000	0.000000	0
D 3	3	20	4.000000	0.000000	20
D 4	0	0	0.000000	0.000000	0
D 5	5	10	1.000000	0.000000	0
D 6	0	0	0.000000	0.000000	0
D 7	0	0	0.000000	0.000000	0

The data for the tool compensation can be edited on the 'Tools' properties page.

More detailed parameter descriptions can be found in the interpreter description under tool compensations [▶ 83].

3.3.7 "Editor" tab

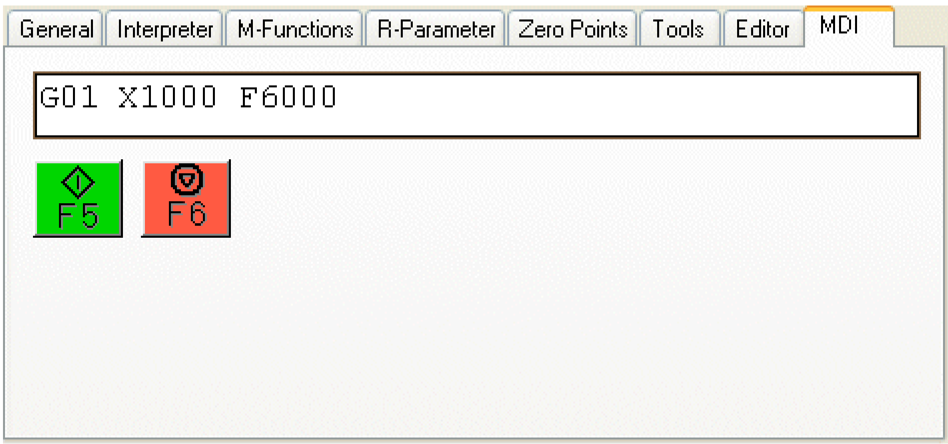


The editor is used to display and edit the NC programs.

Button explanations:

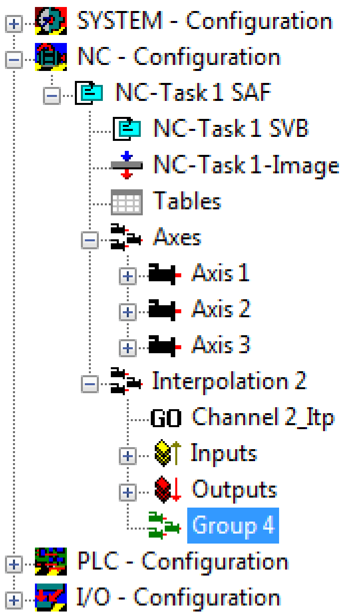
- **Browse...**
  Opens a dialog with which existing NC programs can be selected and displayed.
- **F5**
  Starts the currently loaded NC program.
  The NC program displayed in the editor does not necessarily have to be the currently loaded program.
- **F6**
  Stops the currently running NC program.
- **F7**
  Loads the NC program displayed in the editor.
- **F8**
  Executes a group reset.
- **F9**
  Saves the NC program currently displayed in the editor under the same name.
- **Editor...**
  Opens a larger window in which the NC program is displayed.

3.3.8 "MDI" tab



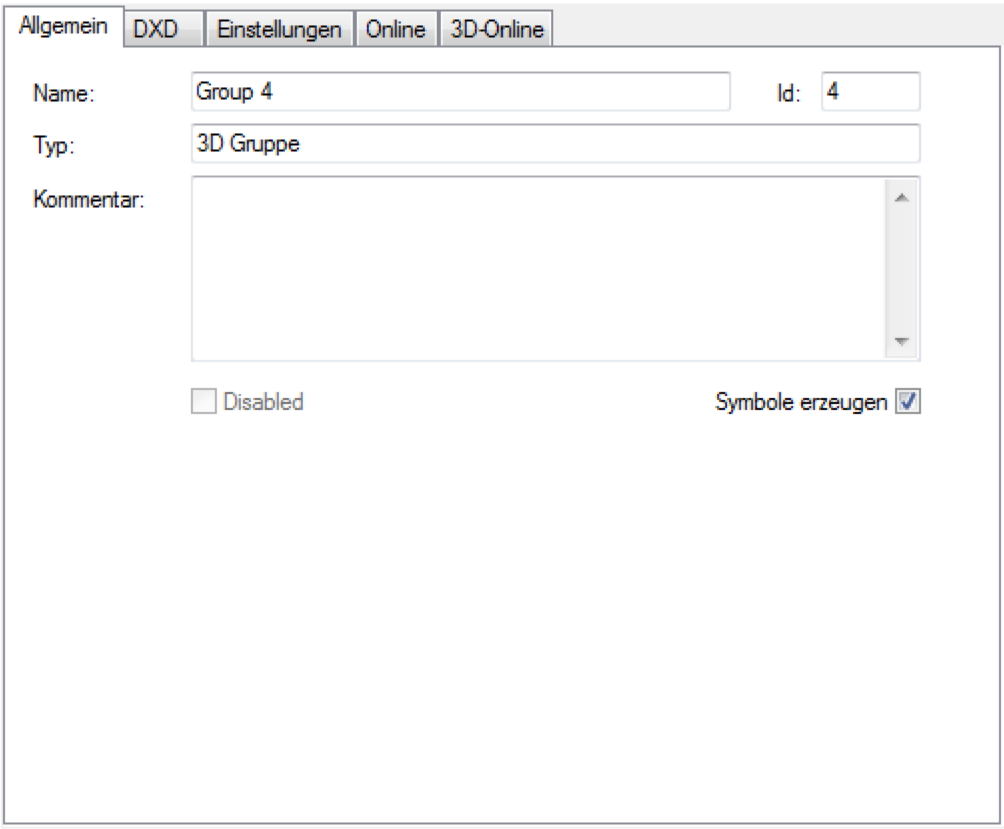
MDI stands for 'manual data interface'. It is used to directly input individual NC blocks from the System Manager. Processing is started and stopped via F5 and F6 respectively.

3.4 Group element



General
DXD [▸ 23]
Settings [▸ 26]
Online [▸ 27]
3D-Online [▸ 28]

3.4.1 "General" tab



group ID

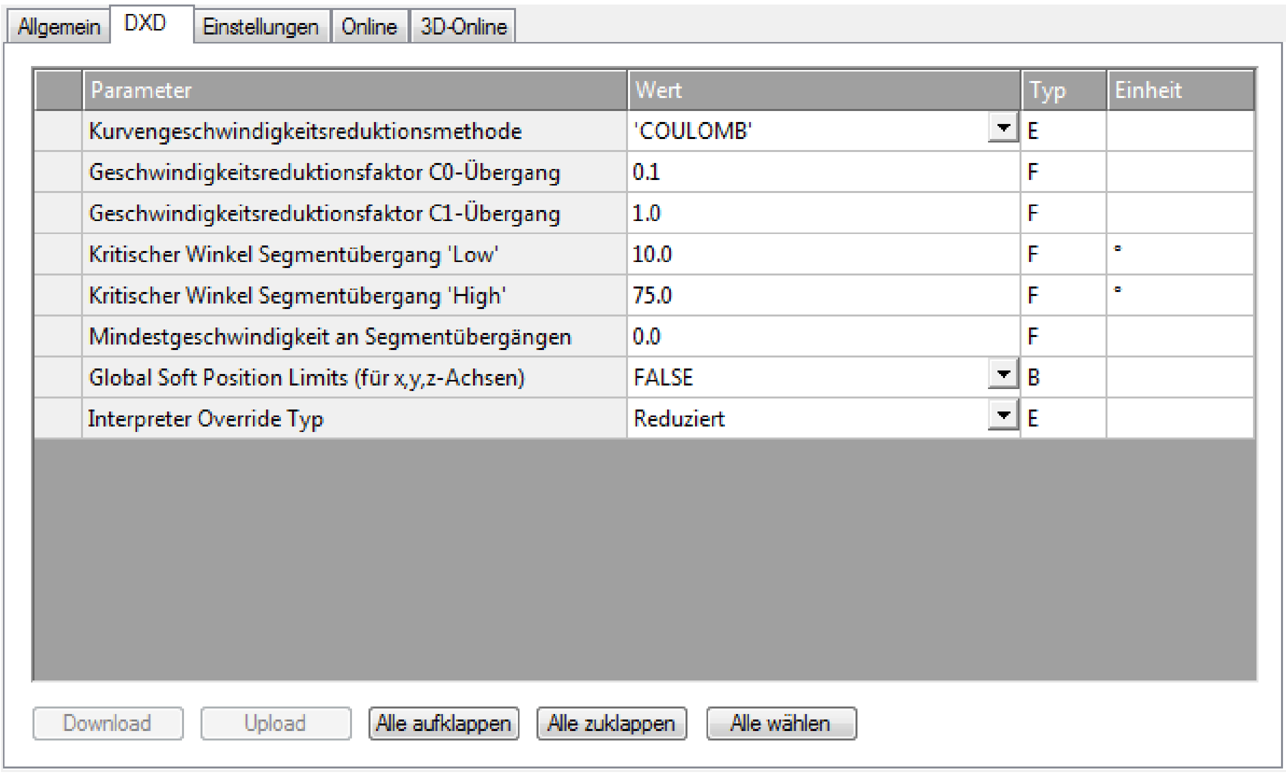
The group ID is given in the 'General' page. This is required for group-specific ADS commands.

Create symbols

In order to be able to access path variables symbolically, select symbol generation for the group here.

3.4.2 "DXD" tab

"DXD" tab



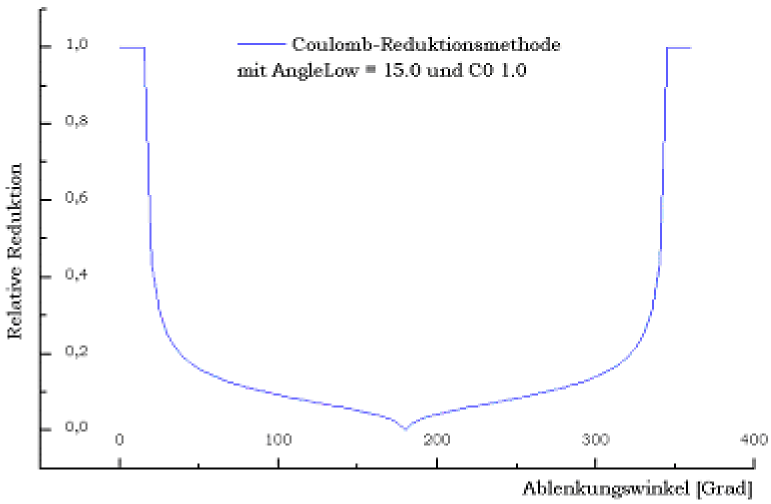
The NCI group parameters are written on the 'DXD' properties page.

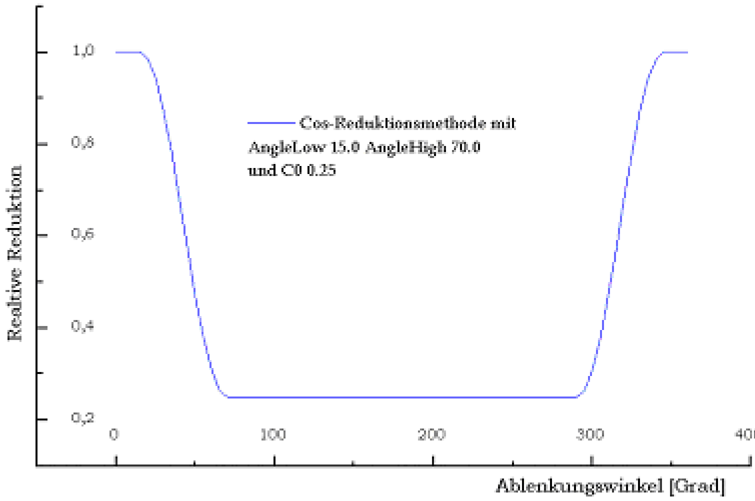
Curve velocity reduction method

The curve velocity reduction method is only effective for C0 transition (see Classification of Segment Transitions [▶ 247])

Defines of the curve velocity reduction method

```
0 Coulomb
1 Cosinus
2 VeloJump
3 DeviationAngle (not yet released)
```

Method	Description
Coulomb	The coulomb reduction method is a dynamic process analogous to the Coulomb scattering. The angle of deflection φ in the transition point is the angle between the tangents of the path at the end of the segment S1 and the tangent of the path at the start of segment S2. The velocity is set analogous to the coulomb scattering proportionally to the velocity at infinity $V_k \propto (\tan(0.5(\pi-\varphi)))^{1/2}$ and then reduced via the C0 factor. $V_k \leftarrow C0\ V_k$. In the case of a movement reversal (φ=180) the reduction is always to V_k = C0. As the reduction in the case of small angles of deflection is drastic, there is an angle $\varphi_{low} \in [0,180]$ from which full reduction takes effect. To avoid reduction, set φ_{low} = 180. For full reduction (down to φ = 0), set C0 = 0.0 and φ_{low} = 0. 

Method	Description
Cosine	The cosine reduction method is a purely geometrical process. It involves: • the C0 factor $\in [0,1]$, • an angle $\varphi_{low} \in [0,180]$, • an angle $\varphi_{high} \in [0,180]$ with $\varphi_{low} < \varphi_{high}$ • Reduction scheme: • $\varphi < \varphi_{low}$: no reduction: $V_k \leftarrow V_{k,}$ • $\varphi_{high} < \varphi$: reduction by the C0 factor: $V_k \leftarrow$ C0 V_k • $\varphi_{low} < \varphi < \varphi_{high}$: partial reduction continuously interpolating between cases 1 and 2, proportional to the cos function in the range $[0,\pi/2]$. For full reduction (up to φ = 0), set C0 = 0.0, φ_{low}=0 and φ_{high} very low but not 0 (e.g. 1.0E-10) 
VeloJump	It is a geometrical procedure for determining the segment transition velocity at a C0 transition. The procedure reduces the path velocity as required, so that the step change in velocity does not exceed the specified limit value. It is calculated based on the following formula: VeloJump factor * cycle time * min(acceleration; deceleration) Further information [▶ 247]

Velocity reduction factor C0 transition

Reduction factor for C0 transitions. The effect depends upon the reduction method.

C0 $\in$ [0.0, 1]

Velocity reduction factor C1 transition

First *V_link* is set equal to the minimum of the two segment set velocities: *V_link*= min(*V_in,V_out*). Depending on the geometry types *G_in* and *G_out* on the segments to be linked and the plane selections on *G_in*and *G_out*, the geometrically induced absolute acceleration jump *AccJump* in the segment transition is calculated under the velocity *V_link*. If this is greater than *C1* times the path acceleration/(absolute)deceleration *AccPathReduced* allowed for the geometries and planes, then the velocity *V_link* is reduced such that the resulting acceleration jump is equal to *AccPathReduced*. If this value is smaller than *V_min*, then *V_min* has priority.

Notice When changing the dynamic parameters, the permissible path acceleration for the geometries and planes and thereby the reaction of the reduction changes automatically.

Reduction factor for *C1* transitions: *C1≥ 0.0*

Critical angle, segment transition 'low'

Parameter for φ_{low}. Description of the operational effects, see curve velocity reduction method [▶ 23]

Critical angle, segment transition 'high'

Parameter for φ_{high}. Description of the operational effects, see curve velocity reduction method [▶ 23]

Minimum velocity at segment transitions

Each NCI group has a minimum path velocity *V_min* ≥ *0.0.* The actual velocity should always exceed this value. User-specified exceptions are: programmed stop at segment transition, path end and override requests which lead to a velocity below the minimum value. A systemic exception is a motion reversal. With the reduction method DEVIATIONANGLE the deflection angle is $\varphi \geq \varphi\_h$, in which case the minimum velocity is ignored. *V_min* must be less than the set value for the path velocity (F word) of each segment.

The minimum velocity can be set to a new value *V_min* ≥ *0.0* in the NC program at any time. The unit is *mm/sec*.

Global soft position limits (for x,y,z-axes)

Parameters for activating the software limit positions for the path

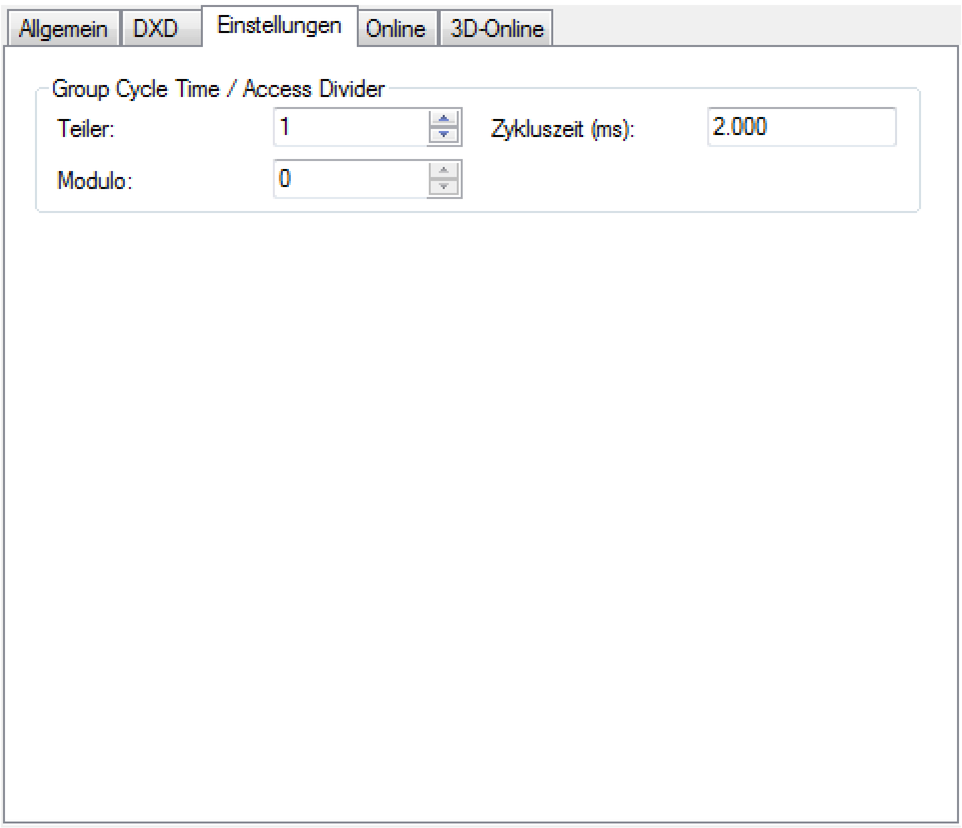
A description can be found in the annex under Parameterization [▶ 249].

Interpreter override type

Parameter for selecting the path override type (for description see Path override (interpreter override types) [▶ 250]).

3.4.3 “Settings” tab

“Settings” tab



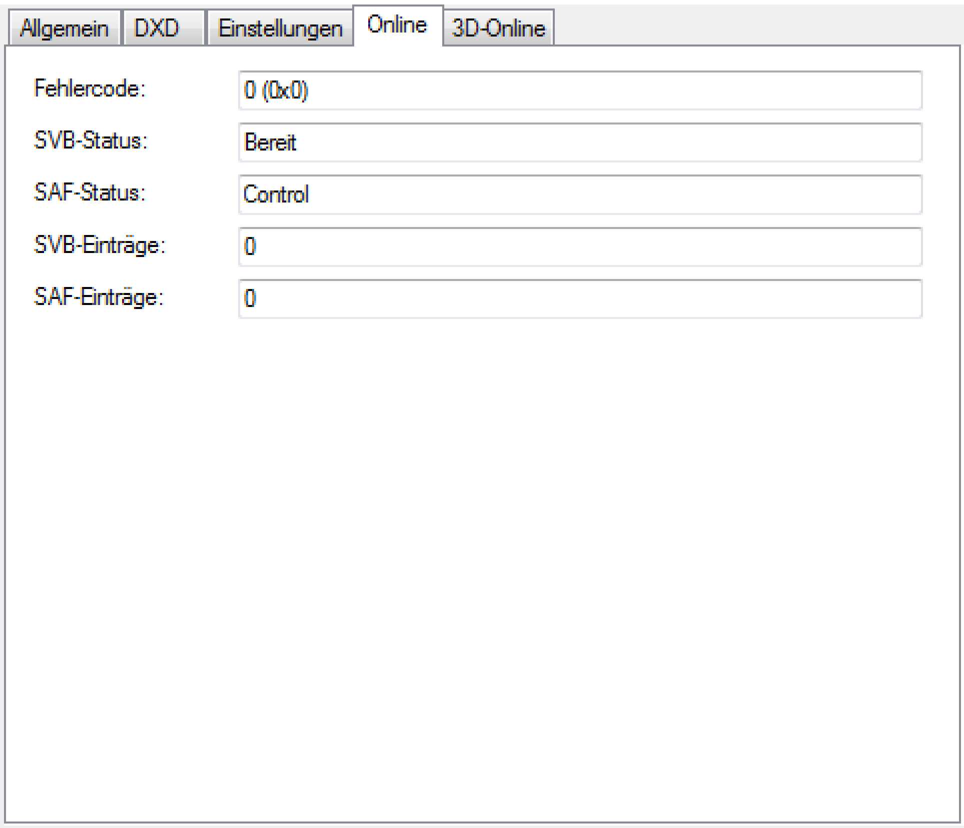
Under the Settings tab you can set the cycle time for the interpolation. The cycle time set here is a multiple of the cycle time of the SEC task.

Using the cycle time in the "Settings" tab

The cycle time setting can be used if you have to select a cycle time for the interpolation that differs from the SEC task. Generally, the cycle time of the SEC task should be adjusted to set the cycle time.

3.4.4 "Online" tab

"Online" tab



Error code

The current error code for the channel is displayed here. The value is the same as the value displayed in the online window of the interpreter under 'Channel Status [▶ 16]'.

SVB status

SVB status displays the current block preparation status (SVB = **S**atz**v**or**b**ereitung). Possible SVB states are:

```
ERROR
IDLE
READY
START
DRIVEOUT
CALIBRATE
MFUNC
SYNCREC
DELAY
MFUNCWAIT
SPINDLEWAIT
```

PLC evaluation of the SVB status is normally not necessary.

SAF status

SAF status displays the current block execution status (SAF = **S**atz**a**us**f**ührung). Possible SAF states are:

```
ERROR
IDLE
CONTROL
```

```
RUN
RUN_DRIVEOUT
WAIT
```

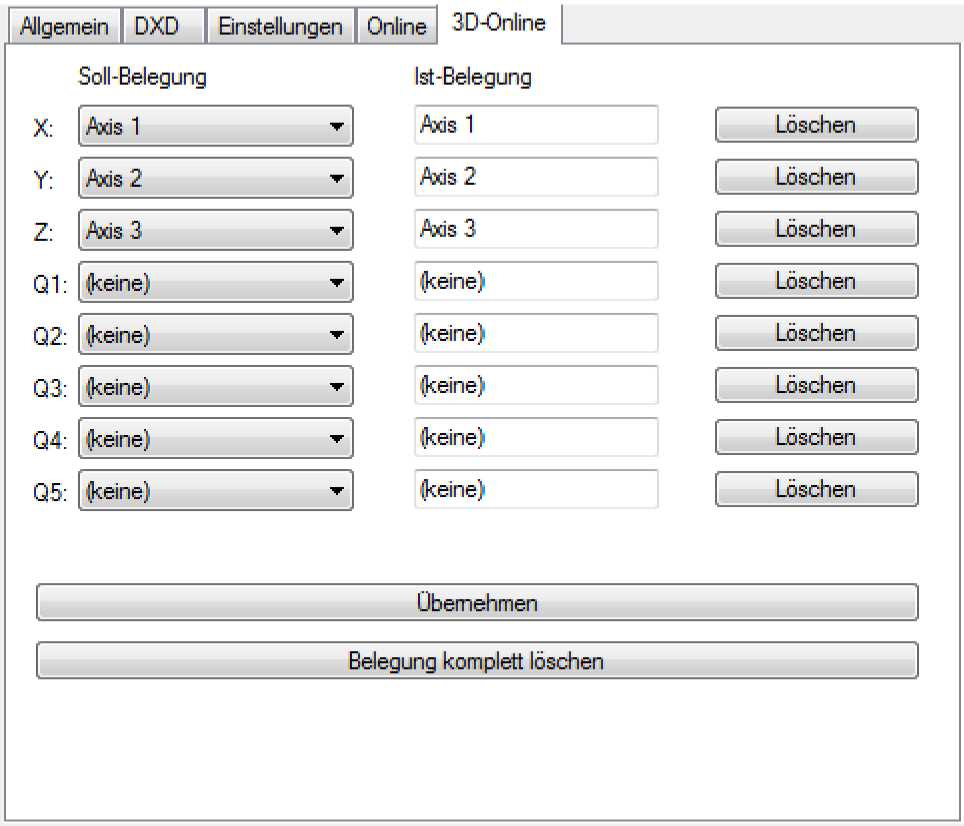
PLC evaluation of the SAF status is normally not necessary.

SVB entries
Number of current SVB entries.

SAF entries
Number of current SAF entries.

3.4.5 "3D-Online" tab

"3D-Online" tab



Target assignment
At this point the **interpolation group** is formed. The movement of the PTP axes, which are assigned to the path axes X, Y and Z, can then be based on interpolation.

Any PTP axes can be selected with the aid of the selection lists for the path axes X, Y and Z. Press the 'Apply' button to form the 3D group.

A corresponding PLC module is also available in the TcNcCfg.lib [▸ 208].

Actual assignment
The current path axis configurations are displayed here. Use 'Delete' to remove individual axes from the 3D group.

Delete whole configuration
Resolves the complete 3D group. You can also use a corresponding PLC module which is available in the TcNcCfg.lib [▸ 210].

4 Interpreter (DIN 66025/G-Code)

4.1 Basic Principles of NC Programming

4.1.1 Structure of an NC Program

An NC program is a text that is normally stored as a sequence of ASCII codes in a file on the hard disk. It consists of a sequence of NC blocks separated by line breaks (Return). Usually it is executed by being interpreted and worked through, character by character and line by line.

Program structure
The NC program is thus composed of three parts

- Program start (optional)
- Number of sets
- Program end

Program start
At the beginning of an NC program the character '%' can represent the start of the program. The name of the program is then found following this character. The block for the program start does not necessarily have to be programmed.

Sample:

```
% Test1 (program start)
N10 G0 X100 Y100 Z0
M30 (program end)
```

NC block
Each NC block consists of one or several NC words, or even of none (an empty line), separated by spaces or tab characters. It is therefore not possible to use a space within a word.

Sample:

```
N10 G0 X100 Y100 Z0
```

NC word
The first character of an NC word specifies its meaning. It is either a letter or a special character.

Upper/lower case has, in general, no significance. Uniform use of upper case is, however, recommended for the sake of better readability. The optional following characters specify the meaning more precisely, or supply parameters for the execution.

In order to manage with such a limited supply of characters, an expression is not available for every variation of every function. It is rather the case that the significance and effect of many NC words is determined partly by the context. This can be a matter of the foregoing words in the block, but it can also depend on previous NC blocks. In a few cases the effect of NC words even depends on the machine data.

Program end
The end of the program is indicated by an M-function. Either M2 or M30 is used for this.

Effective Duration of Words
Commands such as G0 [▸ 39] or G17 [▸ 33], that have effects beyond the end of the block, are known, according to DIN 66025, as **modal**. These commands are effective as long as they are neither canceled nor altered by another command.

Comments
If either parts of an NC block, or the whole of it, is not to be interpreted, the region concerned is to be placed within curved brackets.

Sample:

```
N10 G0 X100 (comment)
```

A comment ends with the closing bracket, or, at the latest, at the end of the block. This means that a comment can't continue over a number of lines. Nested comments are also not possible.

Block number
Each block can be identified by a block number. The block number is identified by an "N" for subsidiary blocks, and with ":" for main blocks.

The block number is not essential. A block not identified by a block number can not, however, be used as the target of a jump command. An error message, moreover, can only approximately report the location of the error (the last block number).

4.1.2 Block Skipping

It is often useful if not all blocks of a program are always executed. This makes it possible to implement similar processes with a single program. In such cases, the blocks that belong to one variant are given a block skipping identifier. This must be written at the start of the block, and consists of a slash "/". If several variants are required, the slash is extended with line information (0..15), for instance "/12". The line information (where "/" is equivalent to "/0") selects a bit from a word in the channel interface from the PLC to the NC. If this bit is **set**, the block is not interpreted.

In the NC the variable *'mSkipLine'* is evaluated for this purpose, which can be found among the inputs in the cyclic channel interface. The counterpart in the PLC can be found in the outputs under 'nSkipLine' [▸ 251] *(previously: nSatzunterdrückung)* (see TwinCAT PLC library: NCI interpreter).

If one of a number of variants is to be active, all the other suppressions must be set. Then only those blocks remain active that have either no identifier, or that have the desired identifier.

Active time of block skipping

The interpreter works an indeterminate number of blocks in advance of the execution. Block skipping can only operate correctly if it is set early enough (perhaps before the program starts), or if the interpreter is synchronized with the execution at a suitable location in the program (interpreter stop [▸ 71]).

4.1.3 Look-Ahead

The actual velocity at the segment transition depends on a range of parameters. These include residual path length, dynamic parameters for the current segment, and (indirectly) the geometric angle at the segment transition.

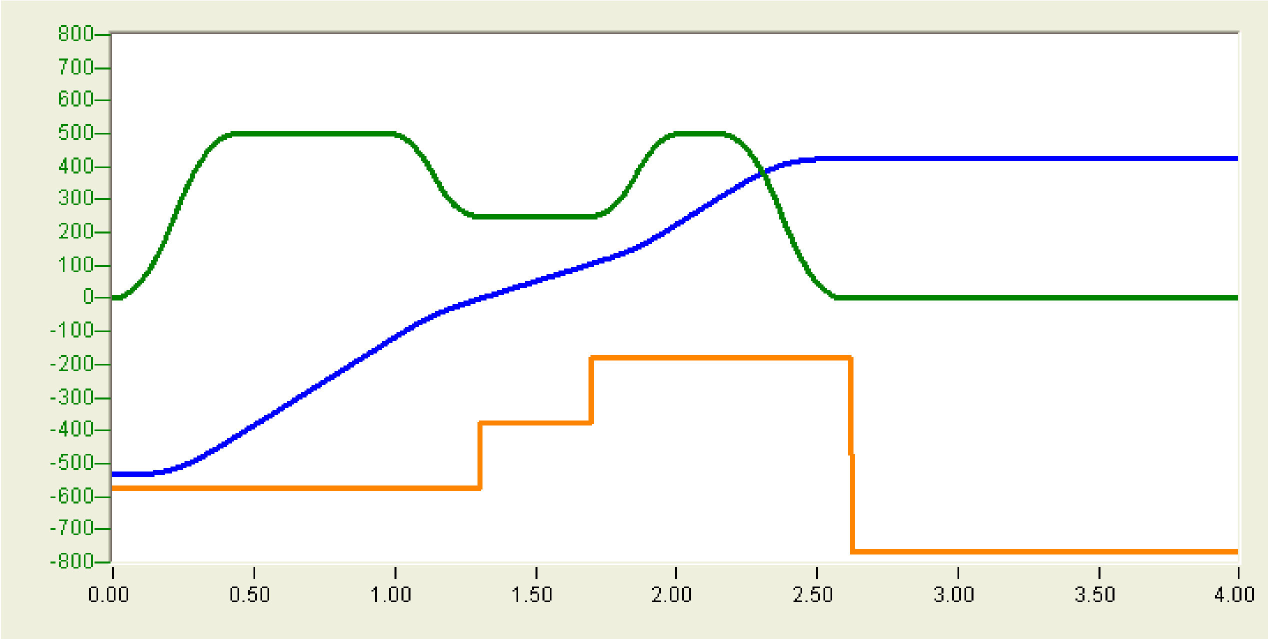
Dynamic look-ahead (referred to as look-ahead below) ensures that the velocity can remain as high as possible at segment transitions. In the standard configuration 128 geometry entries are considered.

Without look-ahead the velocity is reduced to 0 at each segment transition (G60).

Segments with different target velocity

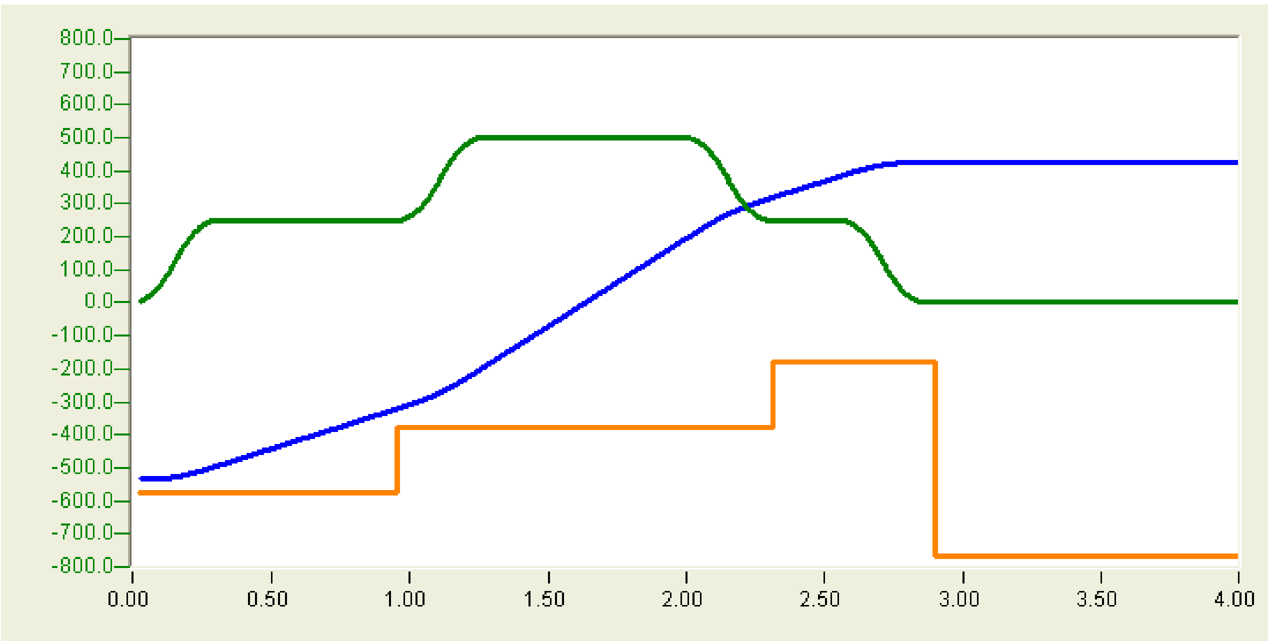
If the target velocity changes from a high velocity level to a lower level (N10 -> N20), the lower velocity will already have been reached at the start of the segment.

If the target velocity changes from a low velocity level to a higher level (N20 -> N30), the higher velocity is initiated with the segment transition. The system therefore always ensures that even at the segment boundary the current velocity does not exceed the programmed velocity.



Green: Path velocity
Blue: Position
Orange: Block numbers

```
N10 G01 X600 F30000
N20 G01 X700 F15000
N30 G01 X900 F30000
M30
```



Green: Path velocity
Blue: Position
Orange: Block numbers

```
N40 G01 X200 F15000
N50 G01 X800 F30000
N60 G01 X900 F15000
M30
```

4.1.4 Smoothing of Segment Transitions

Overview

Segment transitions with no continuous second differential cause instability in the dynamics unless the path velocity is reduced to *0* at those points. For dynamically stable segment transition at a finite speed it is possible to smooth segment transitions with Bezier splines which alter the local geometry and ensure that the complete path has a continuous second differential.

Tolerance spheres

A tolerance sphere is laid around every segment transition within which the path may deviate from its pre-set geometry for smoothing purposes. The radius of the tolerance sphere (parameterization [▶ 247]) is predetermined by the user and applied modally for all segment transitions that imply no exact positioning or stop in the segment transitions. The radii of the tolerance spheres are automatically reset adaptively, thus preventing tolerance spheres from overlapping in the case of small segments.

Dynamic parameters

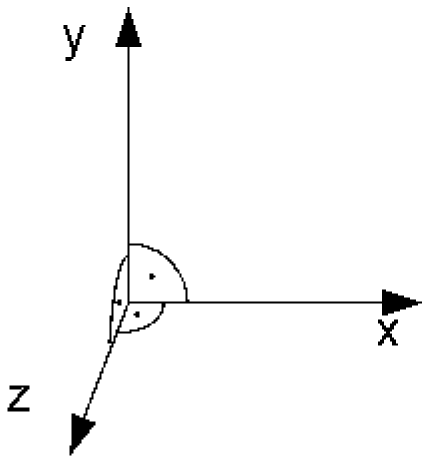
The smoothing enables faster dynamics. The system-determined maximum segment transition velocity *VeloLink* can be influenced by the user insofar as the system parameter C2 velocity reduction *C2* (parameterization [▶ 247]) sets the segment transition velocity to *C2x VeloLink*. The factor can be changed online.

General characteristics at segment transitions

When entering the tolerance sphere, the path acceleration is *0* and the path velocity equals the segment transition velocity. This is maintained within the tolerance sphere. The override is inactive within the tolerance sphere, i.e. the change of the velocity level caused by the override is interrupted within the tolerance sphere and continues after the exit from the tolerance sphere.

4.1.5 Co-ordinate System

The names of the axes of a machine tool are specified by DIN 66217. The letters X, Y and Z are allocated to the axes. These create a right-handed right-angle (Cartesian) co-ordinate system. In many machines, not all three axes are present at every location. In these cases individual letters are allocated in some meaningful way, and the axes that are not present are ignored.



4.1.6 Dimensional Notation

Dimensional data can optionally be referred to an absolute origin or to the current set value.

Absolute dimensions

Command	G90
Cancellation	G91

All positional data in absolute dimensions are always given with reference to the currently valid origin.

In terms of tool movement, this means that, under absolute dimensioning, it is the position to which the tool should move that is described.

Incremental Dimensions

Command	G91
Cancellation	G90

When dimensions are incremental, positional data is related to the immediately preceding point. In addition to the path axes, the auxiliary axes (Q1..Q5) are also taken into account.

For the tool movement this means that the incremental dimension describes by how much the tool is moved.

Units

The units for length, angle etc. are described in the following table:

	Unit
Positions and lengths	mm
Angle	Degrees
Times	sec
Feed	mm/min

4.1.7 Working Plane and Feed Direction

In order to describe circles (except CIP [▸ 41]), and for the compensation of cutter radius [▸ 88] and tool length [▸ 86], it is necessary to specify the working plane.

Working Plane XY

Command	G17
Cancellation	G18 or G19

The function G17 specifies the XY plane as the working plane and the feed direction as the Z direction.

The function acts as:

- Plane for tool radius compensation [▸ 88]
- Feed direction for tool length compensation [▸ 86] (offset)
- Plane for circle interpolation

Changing the working plane

The working plane cannot be changed while tool compensation is active.

Working Plane ZX

Command	G18
Cancellation	G17 or G19

The function G18 specifies the ZX plane as the working plane and the feed direction as the Y direction.

Working Plane YZ

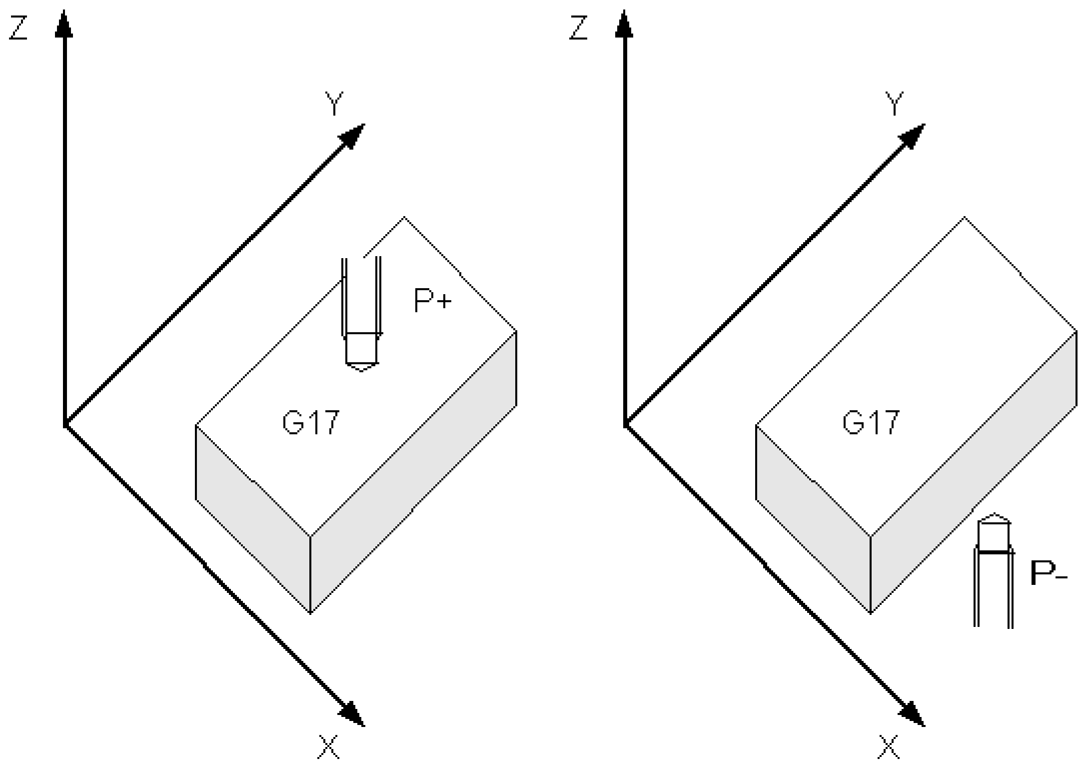
Command	G19
Cancellation	G17 or G18

The function G19 specifies the YZ plane as the working plane and the feed direction as the X direction.

Specification of the feed direction

Command	P
Parameter	+feed direction positive (default) -feed direction negative

Parameterization of the feed direction is required for tool length compensation. It is used to specify whether the tool operates above or below the workpiece.



Sample:

```
N10 G0 X0 Y0 Z0 F6000
N20 D2 P- Z
N30 G01 X100
N40 D0 Z
N50 M30
```

In this sample the length compensation operates below the workpiece.

4.1.8 Inch/metric dimensions

from TwinCAT V2.9, Build 901

G70	Dimensions in inches
G71	Dimensions in millimeters (default)
G700	Dimensions in inches with calculation of the feed

G710	Dimensions in millimeters with calculation of the feed

Dimensions in millimeters (G71) is normally active. Information on whether the coordinates have to be converted is stored in the machine parameters [▶ 16] (Interpreter tab). The basic dimension system in millimeters is also set there as standard.

The effects of the changeover

If the basic dimension system is not the same as the current dimension system (set with G70 or G71), then certain parameters and co-ordinates must be converted. The conversion factor required here is stored in the machine parameters, like the basic dimension system. The changeover has effects on the following parameters:

- Path information for the path axes (X, Y & Z)
- Path information for the auxiliary axes (Q1..Q5)
- Intermediate point co-ordinates (I, J, K)
- Circle radii (B or U)
- Programmable zero shift
- Rounding radius (circle and spline smoothing)

There are also parameters that always remain in the **basic dimension system**, and are not converted. These include the

- adjustable zero shift
- Tool data
- feeds (except under G700 or G710)

Sample 1:

Basic dimension system: inch

```
...
N10 G71      (metric dimensions)
N20 G01 X100 (conversion is carried out)
N30 G70      (dimensions in inches)
N40 G01 Y100 (conversion is not necessary, because)
....         (the basic dimensions are also inches)
```

Sample 2:

Basic dimension system: millimeters

```
...
N10 G71 (metric dimensions)
N20 G01 X100 (conversion is not necessary, because)
    (the basic dimensions are also metric)
N30 G70 (dimensions in inches)
N40 G01 Y100 (conversion is carried out)
```

Zero shifts (NPV)

Adjustable zero shifts (G54-G57) always remain in the basic dimension system, and are not converted. In the case of the programmable zero shifts (G58 & G59) the effect depends on the current dimension system when the shift is selected.

Sample 3:

Basic dimension system: millimeters

```
...
N10 G71     (mm - default)
N20 G54     (activates adjustable zero offset shift)
N30 G58 X100 (programmable zero offset shift)
N40 G01 X0 F6000 (the axis travels to 100 in the machine co-ordinate system)
N50 G70     (inch)
N60 G01 X0    (zero offset shift is programmed under G71 => zero offset shift remains unchanged)
       (i.e. the axis does not move)
N70 G58 X100 (new programmable zero offset shift - now in inches)
N80 G01 X0    (axis moves out by zero offset shift - to 2540 in the machine co-ordinate system)
```

4.1.9 Single Block Operation

from TwinCAT V2.9, Build 901

To test a new NC program, the NCI can be switched to single block mode with the function block ItpSingleBlock [▶ 153]. When single block mode is active, the NC program is stopped after each line. The user has to acknowledge execution of the next line. This can be done by pressing '**NC start (F5)**' in the System Manager under the Editor tab or by setting the input 'bTriggerNext' in the PLC function block ItpSingleBlock [▶ 153].

A distinction is made between two modes:

- Interpreter single block mode
- NC kernel single block mode

Interpreter single block mode

If interpreter single block mode is active, the NC program is stopped after each line in the **interpreter**. This remains true even if the line only contains calculations, and no physical movement is programmed.
This enables re-writing of R-parameters, for example.

Interpreter single block mode should be activated before the NC program is started. If this is not possible, an M-function can be reserved for the activation and combined with a decoder stop.
If interpreter single block mode is activated during processing of the NC program without M-function and decoder stop, it is impossible to predict when it will be active. Theoretically it is possible that the memories in the NC kernel (SVB & SAF) are filled and contain more than 100 geometry entries. The single block can only take effect once these memories have been fully processed.

NC kernel single block mode

Like in interpreter single block mode, in NC kernel single block mode the NC blocks are executed individually. The difference is that in NC kernel single block mode all entries (e.g. geometry entries) have already passed through the interpreter. It is therefore not possible to overwrite R-parameters retrospectively, for example.

This operating mode has the advantage that single block mode can be enabled during processing of the NC program. If a geometry entry is executed (i.e. the axes are moved) during the activation, the system stops at the next possible end of segment. This is usually the current segment. For activation after program startup no M-function with decoder stop is required.

If NC kernel single block mode is used in conjunction with blending, block relaying takes place in the blending sphere. The programmed blending continues to be executed (from TwinCAT V2.10 Build 1301).

Alternatives to activation

We recommend activating single block mode with ItpSingleBlock [▶ 153].

In previous TwinCAT versions single block mode had to be activated via the cyclic channel interface.

from TwinCAT version 2.7 and 2.8:

Single block mode can be selected or deselected in the cyclic channel interface of the PLC. To this end the variable 'nItpMode' has to be masked correctly in the PLC/NC channel interface.

Set bit 14 (0x4000) to switch on interpreter single block mode. Resetting the bit turns single-block mode off again.

It is also possible to trigger the single block from the PLC by means of this interface. Bit 15 must be set for this. The effect is the same as activating NC start in the System Manager.

4.1.10 Arithmetic Parameters

The arithmetic parameters (known as R-parameters for short) are interpreter variables that are named by an expression of the form "R<n>". Since 'n' is an integer in the range 0..999, a total of 1000 R-parameters are available. Of these, the first 900 values (R0..R899) are local variables for the NC channel. They can only be accessed by the channel’s interpreter. The R-parameters R900..R999 are declared globally. They exist only once for each NC, and all channels access the same storage. This makes it possible to exchange data (e.g. for part tracing, collision avoidance etc.) over the channel boundaries.

Mathematical Calculations

The R-parameters (like the axis co-ordinates, feed etc.) are declared as variables of type 'double'. This makes full use of the computer's arithmetic capacity. The number of places before and after the decimal point is not restricted by a format specification. The arithmetical resolution does, nevertheless, have a limit. In practice this is only visible in particularly critical cases. Examples of this include the differences of very large numbers that are almost equal, or trigonometrical functions in particular ranges of angles.

Assignment of R-Parameters

```
N100 R5=17.5
N110 R6=-4
N120 R7=2.5 R8=1
```

As can be seen in the third line, it is quite possible to make more than one assignment in one block. This speeds interpretation slightly, but it can be more difficult to localize an error in the line.

Calculation formula

A calculation formula is an extension of assignment. It consists of a target parameter, an assignment sign and a series of values (R-parameters and constants) separated by arithmetical instructions.

```
N100 R1=R2+R3-17.5*R9/2.5
```

Such a formula, contrary to normal mathematical practice, is processed strictly from left to right.

The illustrated formula is calculated as follows:

1. The contents of R2 is loaded into the arithmetic unit
2. The contents of R3 is loaded into the arithmetic unit
3. The arithmetic unit carries out the + instruction
4. The value 17.5 is loaded into the arithmetic unit
5. The arithmetic unit carries out the - instruction
6. The contents of R9 is loaded into the arithmetic unit
7. The arithmetic unit carries out the * instruction
8. The value 2.5 is loaded into the arithmetic unit
9. The arithmetic unit carries out the / instruction
10. The content of the arithmetic unit is stored in R-parameter R1

Mathematical functions

The interpreter provides standard computing functions. DIN 66025 does not specify any syntax here. The computing functions are called via @6xx (see Annex - @-command overview [▸ 99]).

The trigonometrical functions are always calculated in degrees.

Sample:

```
N10 R2=0 R3=45
N20 @630 R2 R3
```

In this sample the sine of R3 is calculated in degrees. The result is then written into R2.

R-parameter access from the PLC

You can read the R-parameters into the PLC, or write the R-parameters from the PLC. Special PLC function blocks are provided for this purpose

- ItpReadRParams [▸ 185]
- ItpWriteRParams [▸ 202]

During writing of the R-parameters, ensure that the interpreter is ahead of the block execution. In other words, writing of the R-parameters from the PLC should take place before the NC program start or linked with a decoder stop [▸ 71].

from TwinCAT V2.9, Build 1002

For debugging purposes, all R-parameters can be written to a file at any time. This process can be triggered via ADS (see ADS interface - channel functions IndexOffset 0x24 & 0x25 [▸ 264]).

Other functions

RToDwordGetBit

From TwinCAT V2.10 B1308

This function converts an R-parameter to a DWord and then checks whether a particular bit is set. The result is again stored in an R-parameter.

Command	RToDwordGetBit[<dest>; <src>; <bit>]
Parameter <dest>	R-parameter in which the result is entered
Parameter <src>	R-parameter containing the number that is to be converted and checked
Parameter <bit>	Bit to be checked (0..31)

Sample:

```
N10 R1=7
N20 RToDwordGetBit[R2;R1;0]
R10=31
N30 RToDwordGetBit[R3;R1;R10]
```

Enter 1 in R2 and 0 in R3

Initialization of R-parameters

from TwinCAT V2.9, Build 954

'set RParam' is used to assign a value to a contiguous block of R-parameters.

Command	#set RParam(<start index>; <count>; <value>)#
Parameter <start index>	Describes the first R-parameter to be written
Parameter <count>	Number of R-parameters to be written
Parameter <value>	Assigned value

Sample:

```
N10 G01 X100 Y200 F6000
N15 R2=3000
N20 #set RParam( 1; 2; 0.0 )# (R2 is overwritten again here)
N30 G01 X500
```

Saving R-Parameters

If the content of R-parameters [▸ 37] is required for subsequent use, while in the meantime the R-parameters are used for a different purpose, it can temporarily be stored in the values stack of the arithmetic unit.

Two possibilities exist for this:

- enumeration of the R-parameters
- giving the range of R-parameters

Saving the values:

Command	@40 <number> R<n> R<m>... @41 <1st R-parameter> <last R-parameter>

Restoring the values:

Command	@42 <number> R<n> R<m> @43 <last R-parameter> <1st R-parameter>

When restoring the values, call the parameters in reverse order.

Sample 1:

```
(saving the data)
N100 @40 K4 R800 R810 R823 R4

N110 R800=4711
N120 ...

(restoring the data)
N200 @42 K4 R4 R823 R810 R800
```

Sample 2:

```
(saving the data)
N100 @41 R800 R805

N110 R800=4711
N120 ...

(restoring the data)
N200 @43 R805 R800
```

Stack size

The arithmetic unit's stack has a limited capacity. If it overflows, the NC program is interrupted with an error message. That can occur as the value is saved, but can also occur in the course of subsequent formula evaluation.

4.2 Programming Movement Statements

4.2.1 Referencing

By default, axis referencing (homing) should take place before the 3D-group is formed from the PTP channel. Or it can be done from the NC program.

If axes are referenced in PTP mode, it can be done for several axes simultaneously. If axes are referenced from the NC program, it can only be done for one axis at a time.

Command	G74
Cancellation	End of block

Sample:

```
N10 G74 X
N20 G74 Y
```

Referencing with own block

Referencing must be carried out within its own block. G74 may only refer to one axis. This command is only applicable for the main axes (X,Y,Z).

4.2.2 Rapid Traverse

Command	G0

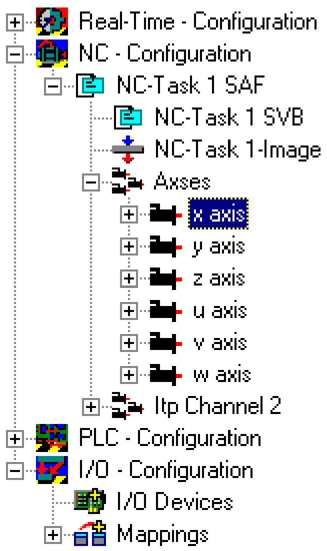
Cancellation	G1 [▸ 41], G2 [▸ 41] or G3 [▸ 41]

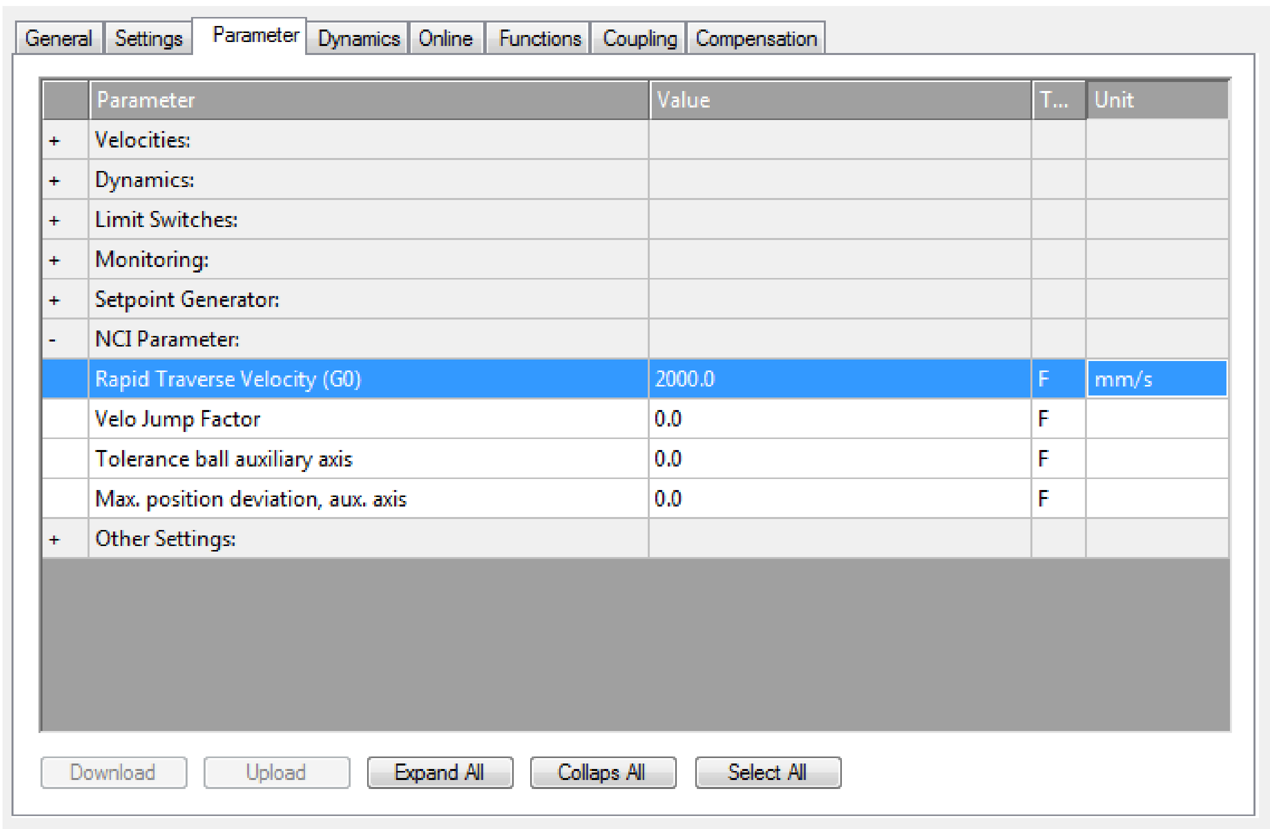
Rapid traverse is used to position the tool quickly, and is not to be used for machining the workpiece. The axes are driven at maximum velocity.

If a number of axes are to be driven in rapid traverse, the velocity is determined by that axis that requires the most time for its movement.

An accurate stop (G60 [▸ 45]) is cancelled with G0.

The rapid traverse velocity is set individually for each axis. It can be edited in the axis parameters in the System Manager under NCI parameters.





4.2.3 Linear Interpolation

Command	G1 or G01 (default)
Cancellation	G0 [▸ 39], G2 [▸ 41] or G3 [▸ 41]

Under linear interpolation the tool moves, with feedrate F, along a straight line that can be freely located in space. The movement of the axes involved is completed at the same moment.

The feedrate (short: feed), F, describes the rate of displacement in millimeters per minute. This value is effective globally, so that it is not necessary to program it again if the same feed is to be used later for other geometrical movements.

Sample:

```
N10 G90
N20 G01 X100.1 Y200 F6000
```

In this example the axes are moved linearly to the position described. The Z axis is not mentioned in this program, and therefore retains its old position.

4.2.4 Circular Interpolation

Circular arcs can be programmed in a number of ways. Two types must be distinguished. One of these is an arc in the working plane [▸ 33] (e.g. the XY plane), and the other is an arc that can be freely located in space (a CIP circle).

Clockwise arc interpolation

Command	G2 or G02
Cancellation	G0 [▸ 39], G1 [▸ 41] or G3 [▸ 42]

Function G2 describes the path of a circular arc clockwise. This function requires the working plane [▸ 33] to have already been defined (G17 [▸ 33] is standard).

In order to describe the circle unambiguously, further parameters are required in addition to the end point. A choice is available between center point programming and radius programming.

Radius programming

In radius programming the radius of the circle is programmed as well as the end point. Either of the letters *'B'* or *'U'* may be used for the radius.

Since the direction is prescribed with G2, the circle is also unambiguously described. The starting point is determined by the foregoing geometrical movements.

Sample 1:

```
N10 G01 G17 X100 Y100 F6000
N20 G02 X200 B200
```

Angle programming for angles >180°

If an angle of more than 180° is to be traversed, the radius must be stated negatively.

Full circle programming

The start and the end points must be different, so that the center can be calculated. Radius programming can therefore not be used for programming a full circle. Centre point programming can be used for this purpose.

Centre point programming

Centre point programming represents an alternative to the method that has just been described. The advantage of center point programming is that complete circles can also be described in this way.

Under the standard settings, the center point is always given relatively to the starting point of the arc. The parameters I, J and K are used for this purpose, with

- I for the X-component
- J for the Y-component and
- K for the Z-component.

At least one of these parameters is 0, and does not therefore have to be programmed.

Sample 2:

```
N10 G01 G17 X100 Y100 F6000
N20 G02 I50 J0 (J is optional) X200
N30 M30 (program end)
```

Sample 3:

```
N10 G01 G18 X100 Y100 Z100 F6000
N20 G02 I0 K50 X150 Z150 (quarter circle in ZX plane)
N30 M30
```

By programming an item of machine data it is however also possible to enter the center point absolutely. The command @402 is required for write access to a machine data bit.

In the following example, the circle from the first example is programmed using the absolute circle center.

Sample 4:

```
N10 G01 G17 X100 Y100 F6000
N20 @402 K5003 K5 K1 (center point programming absolute)
N30 G02 I150 J100 X200
N40 @402 K5003 K5 K0 (center point programming relative)
N50 M30
```

Anticlockwise Circular Interpolation

Command	G3 or G03
Cancellation	G0 [▸ 39], G1 [▸ 41] or G2 [▸ 41]

The function G3 describes a circular arc anticlockwise. The parameters and entry possibilities are the same as under G2.

Circular accuracy

from TwinCAT V2.9 Build 1022

Command	#set paramRadiusPrec(<param>)#
Parameter	param: maximum allowed radius tolerance 0.001 < param < 1.0 (default 0.1)

The 'set paramRadiusPrec' function is used to parameterize the required circular accuracy. This parameter affects circles programmed with G02 or G03.

With center point programming, an error is generated if the difference in radius length is greater than <param>.

Centre point correction

from TwinCAT V2.10, Build 1243

Command	CPCON (standard setting)
Cancellation	CPCOF

In centre point programming the circle is overdetermined. For data consistency, the centre point is usually corrected. Normally only a marginal modification of the centre point is required. After the centre point correction, the magnitude of the input radius equals the output radius.

It the start and end point are very close together, the centre point offset may be large. This may lead to problems with automatically generated G-Code (postprocessor). For manually written G-Code, the CPCON setting (centre point correction on) is recommended.

CIP circle

Command	CIP
Cancellation	End of block

The circles discussed so far can only be used in the principal planes. The CIP circle can also be used to program an arc anywhere in space. For this purpose it is necessary to program not only an end point but also some other point on the path.

So that the circle can be described unambiguously, it is necessary that the 3 points (the starting point is given implicitly) must not be collinear. It is thus not possible to program a full circle in this way.

I, J and K are available as path point parameters. By default their values are relative to the starting point of a circular path.

Sample 5:

```
N10 G01 X100 Y100 F6000
N20 CIP X200 Y200 I50 J50 K50
```

In order to be able to follow a CIP circle it is necessary that the cutter radius compensation [▸ 88] is not active.

4.2.5 Helix

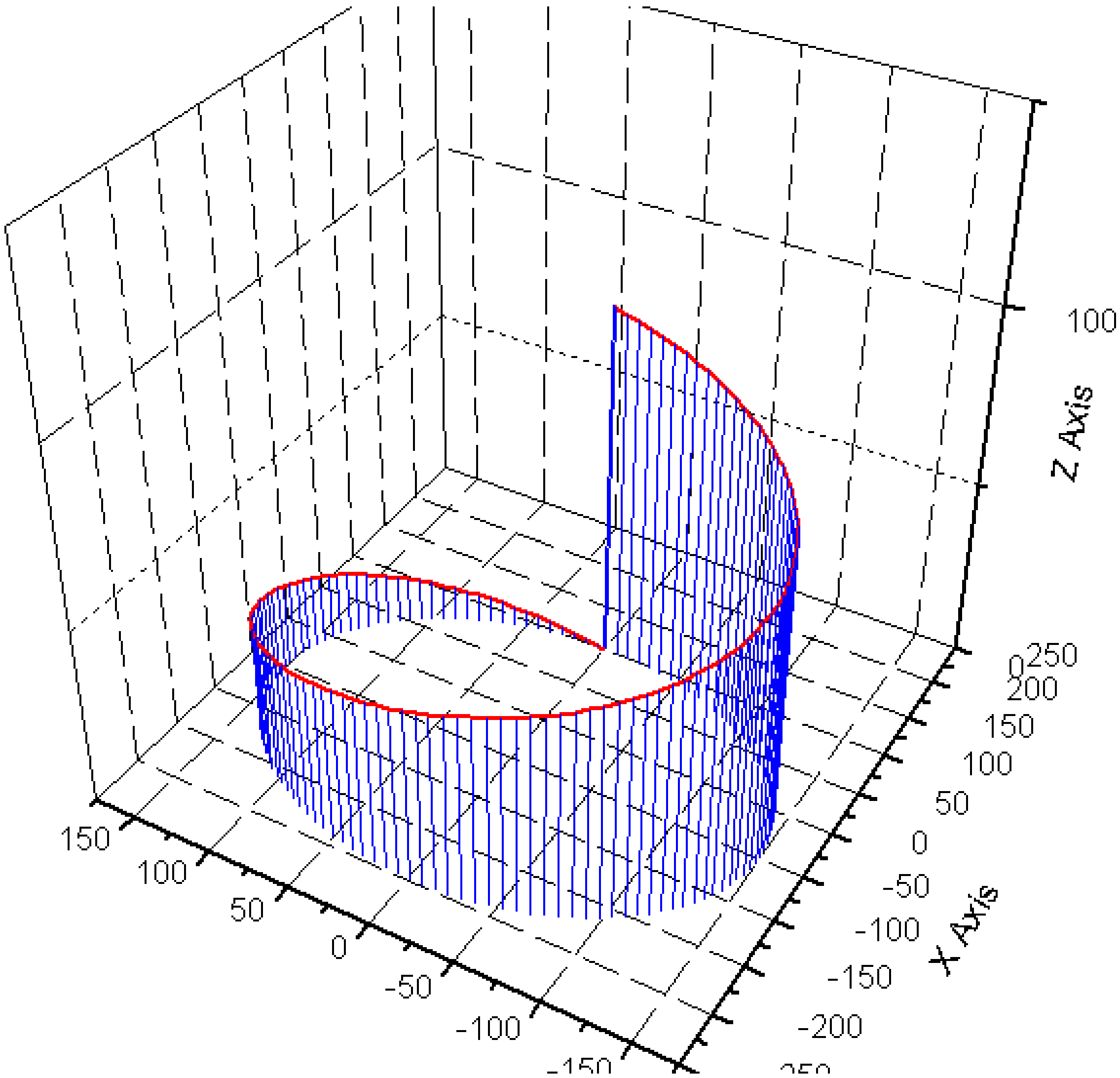
If a circular motion is superimposed onto a perpendicular linear movement, a helix is obtained. A helix can only be programmed in the principal planes. The same parameters as are used for a circle in the principal plane are used. At the same time the axis that is perpendicular to the plane is driven.

from TwinCAT 2.11 Build 2235

The helix can be used together with the cutter radius compensation [▸ 88].

Sample:

```
N10 G01 G17 X100 Y0 Z0 F6000
N20 G03 I-50 Z100
M30
```



4.2.6 Dwell Time

Command	G4 or G04
Cancellation	End of block
Parameter	F or X

G4 is used to switch on a dwell time. It is used to interrupt workpiece machining between two NC blocks for a programmed time (in seconds).

Sample:

```
N10 G01 X100 F6000
N20 G04 X0.5 (pause in sec)
N30 G02 X300
...
```

The dwell time must be programmed in a dedicated block, and the parameters (X or F) must be programmed after G04.

4.2.7 Accurate Stop

block-by-block

Command	G9 or G09
Cancellation	End of block

The accurate stop instruction is used, for example, when sharp contour corners must be manufactured. At the contour transition the set path velocity is reduced to zero and then increased again. This ensures that the programmed position is approached precisely.

G09 acts only on the set value side. The actual values can be checked with TPM (target position monitoring), for example.

modal

Command	G60
Cancellation	G0 [▸ 39]

Description:

see above

See also target position monitoring [▸ 48] (TPM)

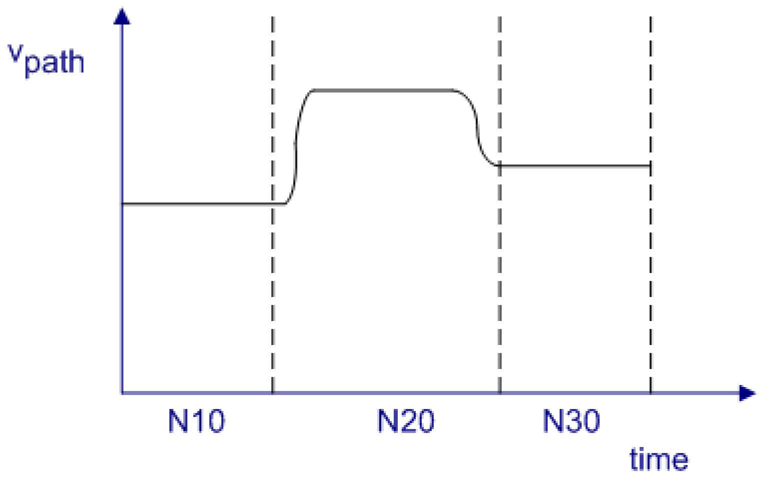
4.2.8 Feed interpolation

from TwinCAT V2.10 B1308

Constant feed interpolation

Command	FCONST (standard setting)
Cancellation	FLIN

The programmed velocity is applied as fast as possible with the constant feed interpolation (default).



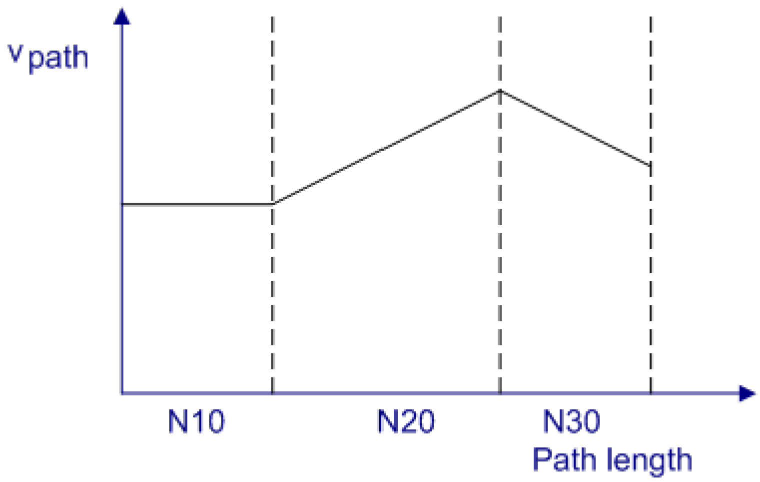
Sample 1:

```
N05 FCONST
N10 G01 X1000 F50000
N20 G01 X2500 F80000
N30 G01 X3500 F60000
...
```

Linear feed interpolation

Command	FLIN
Cancellation	FCONST

The linear feed interpolation transfers the velocity linearly over the path from v_start to v_end.



Sample 1:

```
N05 FCONST
N10 G01 X1000 F50000
N15 FLIN
N20 G01 X2500 F80000
N30 G01 X3500 F60000
...
```

If the velocity on the segment transition has to be reduced more drastically than the programmed segment velocity, due to the geometry or M-function for example, then the linear velocity is maintained as long as possible. The reduced segment velocity will delayed, only if required dynamically.

4.2.9 Zero Offset Shifts

A range of zero offset shifts are available in TwinCAT NC I. They describe the distance between the origins of the workpiece and of the machine.

Zero shift suppression

Command	G53
Cancellation	G54 [▶ 46] to G59 [▶ 47]

The zero shift is suppressed modally with G53. The suppression affects both the adjustable and the programmable zero shift.

Adjustable zero offset shift

Command	G54 G55 G56 G57
Cancellation	G53 [▶ 46] or selection of another configurable zero offset shift

The commands G54 to G57 can be used within the NC program to switch back and forth between the zero offset shifts.

Parameterization

The configurable zero offset shift can be parameterized in different ways

1. PLC function block ItpWriteZeroShiftEx [▸ 161] (recommended standard)
2. System Manager [▸ 16]
3. from the DIN program (from TwinCAT V2.9 Build 1031)

The parameters are saved for each interpolation channel. This means that the adjustable zero offset shifts are channel dependent.

The selection of a zero offset shift must be made in its own block. In order for the movement corresponding to the shift to be actually made it is necessary that at least the axes involved are named in a following geometrical block.

Sample 1:

```
N10 G01 X100 Y0 Z0 F6000
N20 G54 (activates adjustable zero offset shift (NPV))
N30 G01 X Y Z
N40 M30
```

In example 1, line 30 mentions all the axes involved. The effect of this is that the zero offset shifts are applied to all the axes.

Sample 2:

```
N10 G01 X100 Y0 Z0 F6000
N20 G54 (activates adjustable zero offset shift (NPV))
N30 G01 X200 Y
```

In line 30 of sample 2 the X axis is taken to position 200 + shift in the X direction. The Y axis only moves to accommodate the shift, and the Z axis is not moved.

Parameterization from the DIN program

from TwinCAT V2.9, Build 1031

Command	#set paramZeroShift(G<n>; <value x>; <value y>; <value z>)#
Parameter G<n>	Zero shift to be parameterized (G54..G59)
Parameter <value>	Coordinates of the zero shift

'*#set paramZeroShift(..)#*' parameterizes the zero shift but does not activate it. This requires explicit programming of the G-Code.

Sample 3:

```
N10 G01 X100 Y0 Z0 F6000
N20 R12=200
N30 #set paramZeroShift( G54; 100.0; R12; -20)#
N40 G54 (activates adjustable zero offset shift (NPV))
N50 G01 X200 Y Z
```

Programmable zero shift

Command	G58 or G59
Cancellation	G53 [▸ 46]

Programmable zero shifts exist in addition to the adjustable ones. This type of zero shift is directly described from the NC program.

Addition of zero shifts

The programmable zero shift is only effective when the adjustable zero shift is active. This means that the total shift is the sum of

- set zero shift (G54, G55, G56 or G57)
- first programmable zero shift (G58)
- second programmable zero shift (G59)

Sample 4:

```
N10 G01 X100 Y0 Z0 F6000
N20 G54 (activates adjustable zero offset shift (NPV))
N30 G58 X0.5 Y0.5 Z0.5 (1st prg. zero offset shift)
N50 X Y Z (movements for the zero offset shift)
...
M30
```

Behavior with incremental dimension notation

Default behavior, independent of TwinCAT version

Changing the origin also affects the incremental dimension.

Sample 5:

```
N10 G01 X100 Y0 Z0 F6000
N20 G54 (activates adjustable zero offset shift (NPV))
N25 G58 X10 Y10 Z0
N30 G91 (Incr. dimensions)
N40 G01 X200 Y0
N50 ...
```

In N40 Y moves to 10 in the basic coordinate system. A shift in origin also shifts the point of reference for incremental dimension programming, resulting in a travel path for Y.

In this way a contour, which is fully programmed based on the incremental dimension, can be positioned at any point through a zero shift.

from TwinCAT version 2.10 Build 1308

The characteristic can be parameterized under G91 from this version onwards.

Command	Description
ZeroShiftIncOn	The zero shifts are also applied under G91 once the axis is named
ZeroShiftIncOff	The zero shift is not applied under G91

Sample 6:

```
N10 G01 X100 Y0 Z0 F6000
N15 ZeroShiftIncOff
N20 G54 (activates adjustable zero offset shift (NPV))
N25 G58 X10 Y10 Z0
N30 G91 (Incr. dimensions)
N40 G01 X200 Y
N50 ...
```

Since 'ZeroShiftInc**Off**' is set in example 6 , the X-axis in N40 is moved by 200 mm independently of the new zero shift. The Y-axis does not move as no target coordinate is programmed for it.

See also ToolOffsetIncOn/Off [▸ 86]

4.2.10 Target Position Monitoring

from TwinCAT V2.9 Build 1026

Command	TPM
Cancellation	End of block

The command 'TPM' is used to trigger target position monitoring from the NC program. At the end of the geometry this always leads to an accurate stop on the set value side and subsequent checking of the target position window. Block relaying takes place when the monitoring conditions are met for all axes in the group.

Like for PTP, this function is enabled and parameterized individually for each axis. This means that different limits can be selected for auxiliary axes than for the path axes, for example.

Sample 1:

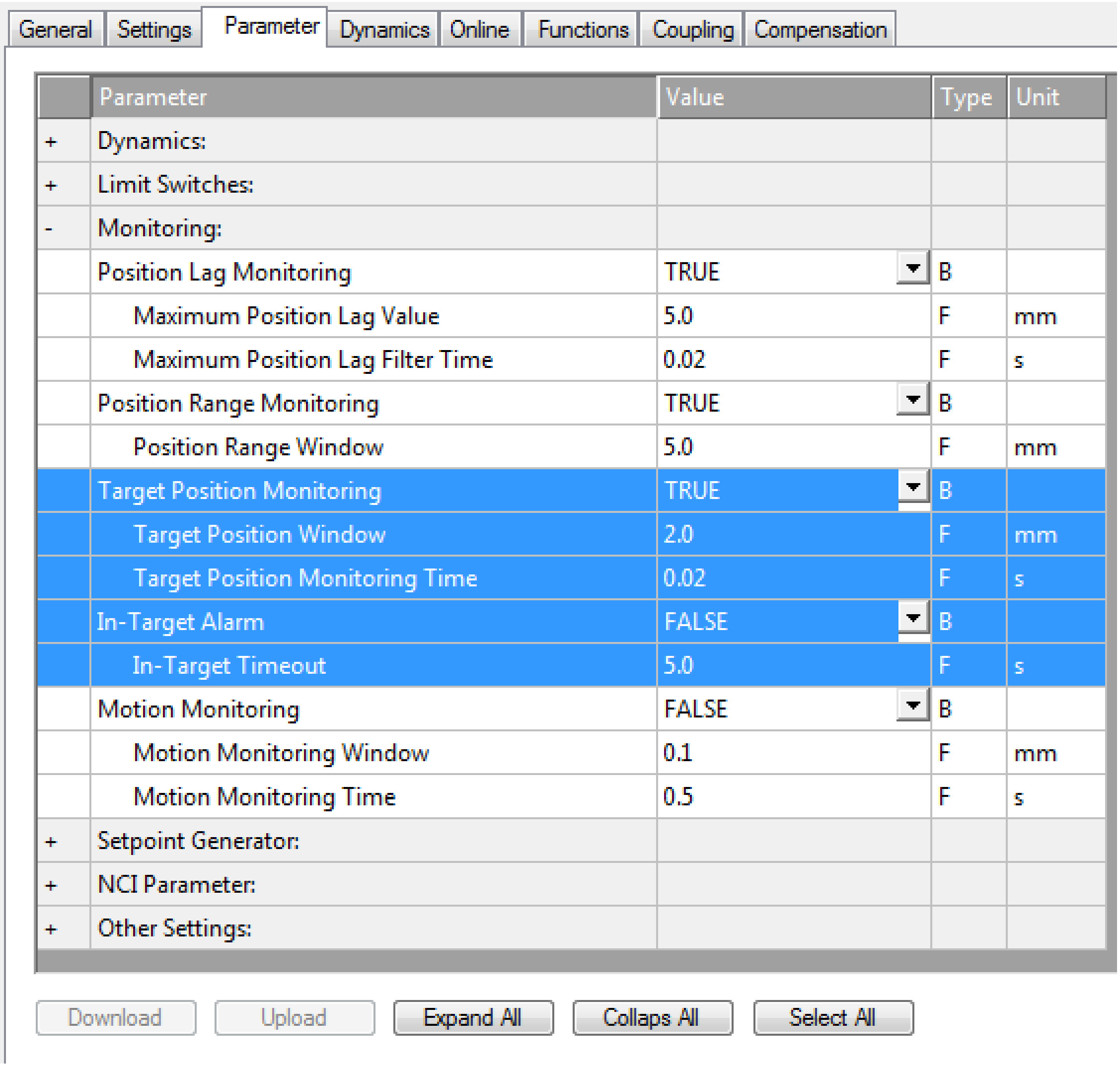
```
N10 G01 X100 Y100 F6000
N20 G01 X300 Y100 TPM
...
```

At the end of the motion of N20, target position monitoring is performed both for the X axis and for Y axis (provided target position monitoring is enabled for both axes)

Sample 2:

```
N10 G01 X100 Y100 F6000
N20 G01 X300 Y100
N30 M61 (Type Handshake)
N40 TPM
...
```

TPM can also be programmed in a dedicated block. In this case the last positioning is checked (of N20 in this case).



	Parameter	Value	Type	Unit
+	Dynamics:			
+	Limit Switches:			
-	Monitoring:			
	Position Lag Monitoring	TRUE	B	
	Maximum Position Lag Value	5.0	F	mm
	Maximum Position Lag Filter Time	0.02	F	s
	Position Range Monitoring	TRUE	B	
	Position Range Window	5.0	F	mm
	Target Position Monitoring	TRUE	B	
	Target Position Window	2.0	F	mm
	Target Position Monitoring Time	0.02	F	s
	In-Target Alarm	FALSE	B	
	In-Target Timeout	5.0	F	s
	Motion Monitoring	FALSE	B	
	Motion Monitoring Window	0.1	F	mm
	Motion Monitoring Time	0.5	F	s
+	Setpoint Generator:			
+	NCI Parameter:			
+	Other Settings:			

If target position monitoring is enabled for an axis, the target position alarm (PEH) should also be active. Time monitoring results in a channel error after the timeout (or before), if the axis is not yet in the target position window. In order to avoid unnecessary channel errors, a sufficiently large timeout value should be selected (e.g. 5 - 10 s). If no PEH time monitoring is active and the axis is permanently outside the position window, no block relaying takes place and the NC remains stationary when viewed from outside. The SEC is in Waiting state (not to be confused with Interpreter state).

See also Accurate stop [▸ 45] (G09)

4.2.11 Contour definitions

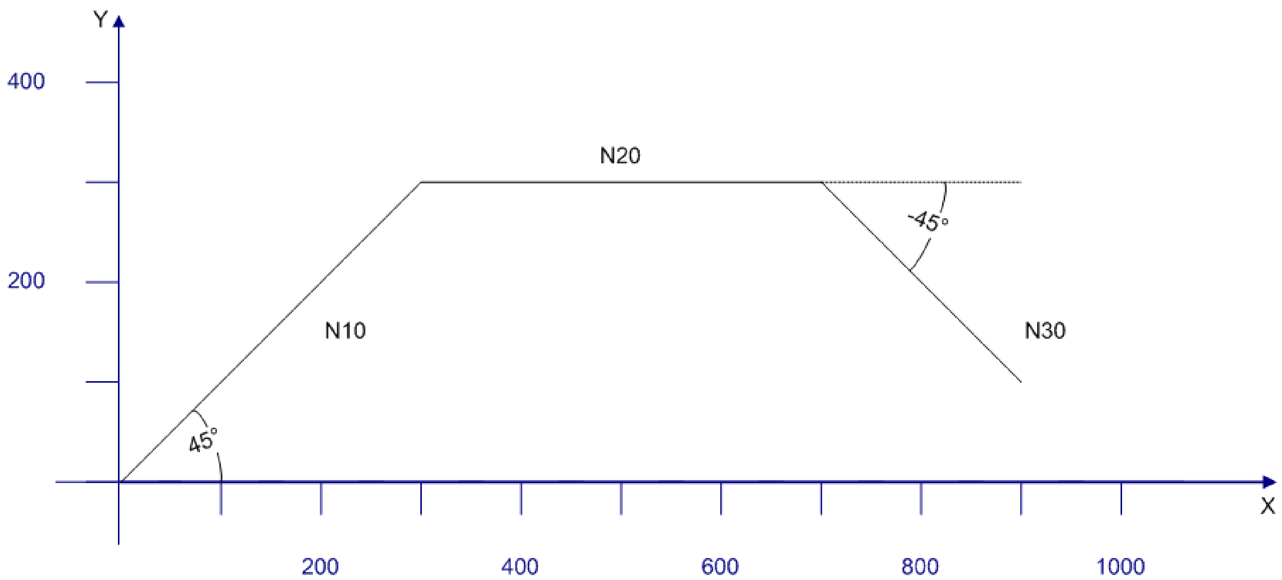
from TwinCAT V2.9, Build 959

Angle and segment length

In this type of programming the angle and the magnitude (segment length) are always quoted, similarly to polar co-ordinates.

Parameter	Description
ANG	Angle in degrees with reference to the abscissa (-360 ≤ ang ≤ 360)
SEG	Magnitude of the segment length

Sample 1:




```
N10 G01 ANG=45  SEG=424.264 F60000
N20 G01 ANG=0   SEG=400
N30 G01 ANG=-45 SEG=282.843
```

or

```
N10 G01 ANG=45  SEG=424.264 F60000
N20 G01 X700 Y300
N30 G01 ANG=-45 SEG=282.843
```

Restrictions:

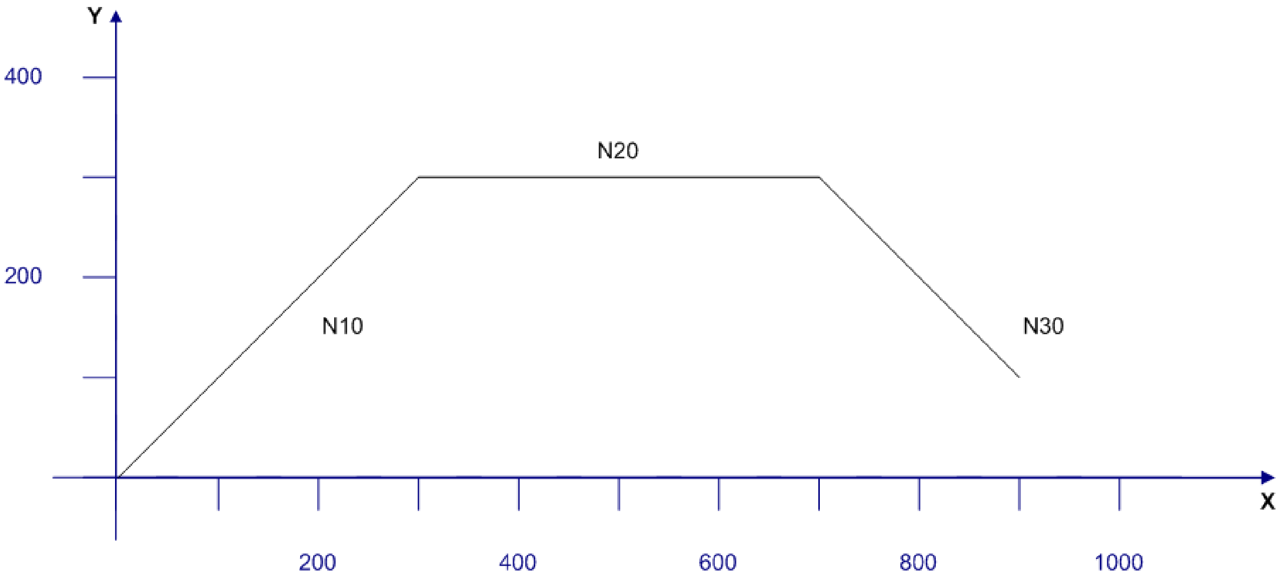
- The programming may only be done in the chosen principal plane.
- The length of the segment must be greater than zero, and refers to the projection in the principal plane.

It is additionally possible to program rounding or chamfering. The ANG and SEG parameters must be programmed in every block. The assignment may use R parameters, but formulas cannot be programmed.

Angle and one component in the plane

As above, an angle is programmed, but the length of the segment is no longer specified directly. It is calculated from a component in the selected principal plane.

Sample 2:




```
N10 G01 ANG=45 X300

N20 G01 ANG=0 Z700
R10=100
N30 G01 ANG=315
X=R10
```

Runtime error

If either two components in the plane are quoted or none at all, the result is a run-time error. A run-time error is also generated if the movement is parallel to the abscissa or to the ordinate, and there is therefore no intersection.

4.2.12 Rotation

It is also possible to program a rotation as well as the zero offset shift [▸ 46]. A distinction is drawn between absolute and additive rotation.

The rotation can turn the co-ordinate axes (X, Y and Z) in the workpiece co-ordinate system.

This makes it possible to machine inclined surfaces (in the plane or in space).

Absolute Rotation

Command	ROT X<value(x)> Y<value(y)> Z<value(z)>
Cancellation	ROT (without parameters)

The rotation instructions must be programmed in their own block. Angles must always be stated in degrees.

Direction of Rotation

A positive angle describes rotation in the direction of the positive co-ordinate axis, the rotation being anti-clockwise.

Carrying Out the Rotation

The sequence of rotations is of critical importance when a co-ordinate system is being rotated. In TwinCAT NC I rotations are always carried out in the following sequence:

1. Rotation about the Z axis
2. Rotation about the Y axis
3. Rotation about the X axis

This sequence is maintained even if the parameters are programmed in a different order.

The origin of the tool co-ordinate system is always used as the centre point of the rotation. This means that the total zero offset shift currently active describes the rotation centre.

Additive Rotation

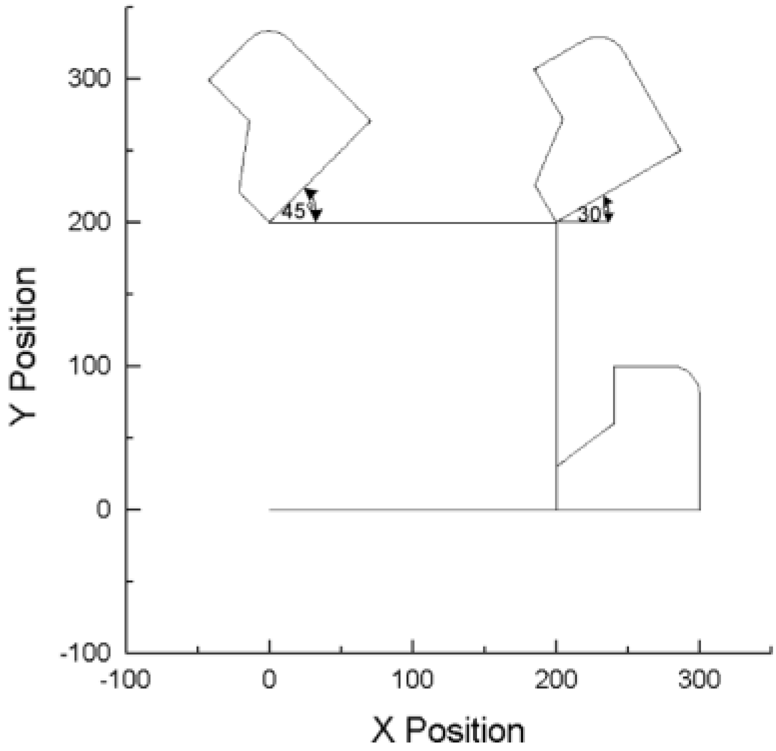
In addition to absolute programming of rotation it is also possible to carry this out additively. The same conditions apply to this as do to absolute rotation.

Command	AROT X <value(x)> Y<value(y)> Z<value(z)>
Cancellation	ROT (without parameters)

Sample:

```
N10 G01 G17 X0 Y0 Z0 F60000
N20 G55
N30 G58 X200 Y0
N50 L47
N60 G58 X200 Y200
N65 ROT Z30
N70 L47
N80 G58 X0 Y200
N90 AROT Z15
N100 L47
N110 M30
```

```
L47
N47000 G01 X0 Y0 Z0 (movements for zero shift & rotation)
N47010 G91 (incremental dimensions)
N47020 G01 X100
N47030 G01 Y80
N47040 G03 X-20 Y20 I-20 J0
N47050 G01 X-40
N47060 G01 Y-40
N47070 G01 X-40 Y-30
N47080 G01 Y-30
N47090 G90
N47100 M17
```



In this sample, the same contour is traversed under different rotations. Since the contour (L47) is programmed in incremental dimensions, and the starting point is described by means of the programmed zero offset shift, the rotation is clear to see.

Once the ROT or AROT command has been programmed, the complete path vector (X, Y & Z) must be assigned.

Rotation extensions

from TwinCAT V2.9, Build 1031

Up to TwinCAT V2.9 Build 1030, after each ROT command the complete path vector has to be programmed. Since this is difficult to realize in some applications, this calculation can optionally be performed automatically in the interpreter. To use this option, 'RotExOn' should be included at the start of the NC program.

Command	RotExOn
Cancellation	RotExOff

Sample:

```
N10 RotExOn
...
N100 G54 (activate zero point & point of rotation)
N110 ROT X90
N120 G0 Z3 (preposition the tool)
N130 G01 Z-10 F6000 (lower to cutting depth)
N140 G01 X100
N150 G01 Z3 (raise to preposition)
...
N1000 RotExOff
N1010 M30
```

Calculate rotation

from TwinCAT V2.9 Build 931

Command	CalcRot[R<s>; R<t>; R<u>]
	CalcInvRot[R<s>; R<t>; R<u>]
Parameter	The 3 R parameters describe the vector to be calculated. The calculation will write the result into this R parameter, and the original value will therefore be overwritten

The function **CalcRot** rotates a three-dimensional vector through the current rotation angle. The rotation angles had been determined by ROT or AROT. The sequence of the calculation is the same as is used for the rotation itself, that is Z, Y and X.

The **CalcInvRot** function behaves in precisely the opposite way. The signs of the currently valid rotation angles are inverted, and the order of calculation is X, Y and Z. In other words, the vector is turned back, so to speak.

Neither CalcRot nor CalcInvRot generate any geometry, but merely carry out the calculation of the vector.

Sample:

```
N10 G01 X40 Y10 Z0 F6000 (the axes are moved
without rotation)
N20 R1=40 R2=10 R3=0

N30 ROT Z45

(What is the position to which X, Y, must be taken so that no
movement is executed?)
N40 CalcInvRot[R1; R2; R3]
N50 G01 X=R1 Y=R2 Z=R3 (R1=35.35 R2=-21.21 R3=0)
N60 ...
```

from TwinCAT V2.9 Build 948

Command	RotVec[R<x>; R<y>; R<z>; R<α>; R<β>; R<γ>]
Parameter	The 3 R parameters (x..z) describe the vector to be rotated through. The calculation will write the result into this R parameter, and the original value will therefore be overwritten. The last 3 R parameters describe the angles.

The function **RotVec** rotates a three-dimensional vector through the specified angle. As with ROT, the rotation sequence is Z, Y and X. RotVec is a pure calculation routine for rotating a vector, and has no effect on either ROT or AROT.

4.2.13 Mirror

from TwinCAT V2.11 Build 2016

The mirror functionality changes the sign of named axes. This enables subroutines to be reused.

Mirroring

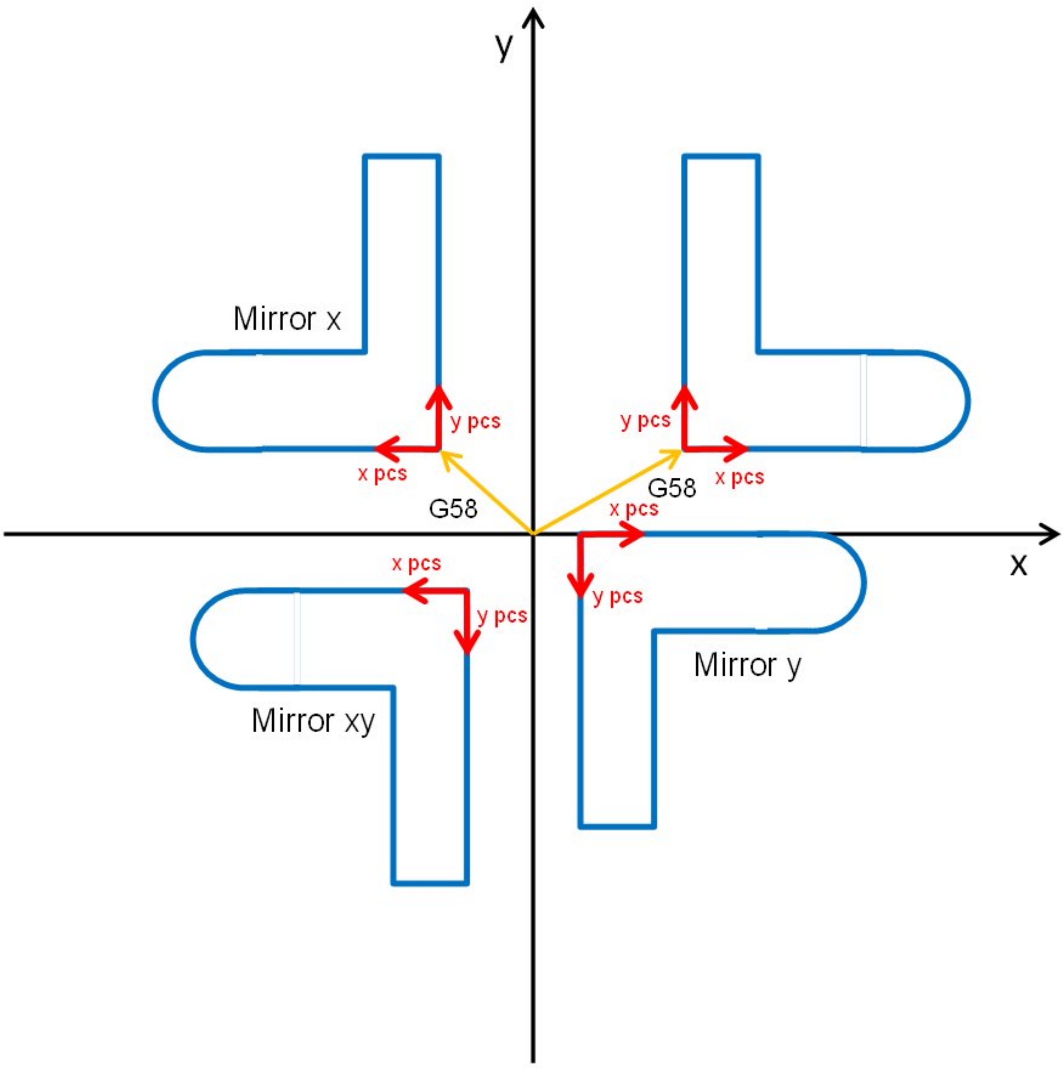
Command	Mirror <opt. X> <opt. Y> <opt. Z>
Cancellation	Mirror (without parameters)

The mirror instructions must be programmed in a dedicated block. Mirrored axes must be named without further parameters.

Sample:

```
N20 G54
N30 G58 X100 Y100
N40 L100

N50 G58 X-100 Y100
N60 Mirror X
N70 L100
```

```
N80 G58 X-50 Y-50
N90 Mirror X Y
N100 L100

N110 G58 X10 Y-10
N120 Mirror Y
N130 L100

N140 Mirror (turn off mirror)
N150 G0 X0 y0
M02

L100
N1000 G0 X200 Y0 Z10 F60000 (move to start pos)
N1020 G01 Z0
N1030 G03 X200 Y100 J50
N1040 G01 X50
N1050 G01 Y400
N1060 G01 X0
N1070 G01 Y0
N1080 G01 X200
N1090 G01 Z10
M17
```



If a zero offset shift is present (G54...G59), the mirror functionality depends on the currently programmed coordinate system.

4.2.14 Smoothing of segment transitions

4.2.14.1 Overview

Overview

Polygon functions (G01 blocks) generally do not consist of continuously differentiable segment transitions, but contain „kinks“ in the contour. Instantaneous velocity changes are generally not possible. To avoid having to reduce the path velocity to 0 at such transitions, the contour can be modified through blending at the segment transition.

	Design	Supported segment transitions	Acceleration of axis components	Max. tolerance	Adaptive tolerance radius	Command
Circular smoothing [▸ 61]	Interpreter	Straight line/ straight line	Step change in acceleration (value parameterizable via the C1 factor)	1/2 of the input or output segment	no	paramCircularSmoothing(...)
Parabola smoothing [▸ 57]	NC kernel	Straight line/ straight line	Step change in acceleration to a constant level (value parameterizable via the C1 factor)	1/3 of the input or output segment	can be selected	paramVertexSmoothing(...)
Bi-quad smoothing [▸ 57]	NC kernel	Straight line/ straight line	Constant acceleration - the acceleration is 0 at the entry and exit - no intermediate point required	1/3 of the input or output segment	can be selected	paramVertexSmoothing(...)
Bezier blending function of the 3rd order [▸ 58]	NC kernel	all	Step change in acceleration to a linear level (can be parameterized with the C1 factor)	1/3 of the input or output segment	can be selected, effective for straight line transitions	paramVertexSmoothing(...)
Bezier blending function of the 5th order [▸ 58]	NC kernel	all	Constant acceleration - the acceleration is 0 at the entry and exit - no intermediate point required	1/3 of the input or output segment	can be selected, effective for straight line transitions	paramVertexSmoothing(...)

	Design	Supported segment transitions	Acceleration of axis components	Max. tolerance	Adaptive tolerance radius	Command
'Old' Bezier blending [▶ 59]	NC kernel	all	Constant acceleration - the acceleration is 0 at the entry, the exit and at the symmetric intermediate point	1/4 of the input or output segment	no	paramSplineSmoothing(...) paramVertexSmoothing(...)

Blending takes effect from the transition between the subsequent two segments.

Principle of blending

The radius of the tolerance sphere can be altered at any time within the NC program, and can be switched off again by setting the radius to 0. Blending remains active until the next reset of the interpreter or TwinCAT restart.

4.2.14.2 Parabolic smoothing

Parabolic smoothing

from TwinCAT V2.10 B1248

Command	#set paramVertexSmoothing(<type>; <subtype>; <radius>)#
Parameter <type>	For parabolic smoothing: **2**
Parameter <subtype>	1: Constant tolerance radius [▶ 60] 2: Distance between intersection and vertex [▶ 60] 3: Adaptive tolerance radius [▶ 61]
Parameter <radius>	Max. radius of the tolerance sphere

For parabolic smoothing a parabola is inserted geometrically into the segment transition. This ensures a steady velocity transition within the tolerance radius.

The parabola is only inserted for straight line/straight line transitions.

4.2.14.3 Biquadratic smoothing

Biquadratic smoothing

from TwinCAT V2.10 B1248

Command	#set paramVertexSmoothing(<type>; <subtype>; <radius>)#
Parameter <type>	For parabolic smoothing: **3**
Parameter <subtype>	1: Constant tolerance radius [▶ 60] 2: Distance between intersection and vertex [▶ 60] 3: Adaptive tolerance radius [▶ 61]
Parameter <radius>	Max. radius of the tolerance sphere

With biquadratic smoothing there is no step change in acceleration in the axis components. With the same radius, a smaller input velocity may therefore be required than for parabolic smoothing.

The operating principle of the subtypes is identical to that of the parabolic subtypes.

4.2.14.4 Bezier curve of the 3rd order

Bezier curve of the 3rd order

from TwinCAT V2.10 B1308

Command	#set paramVertexSmoothing(<type>; <subtype>; <radius>)#
Parameter <type>	for the Bezier curve of the 3rd order: **4**
Parameter <subtype>	1: Constant tolerance radius [▸ 60] 2: Distance between intersection and vertex [▸ 60] 3: Adaptive tolerance radius [▸ 61]
Parameter <radius>	Max. radius of the tolerance sphere

In case of the 3rd order Bezier curve a step change in acceleration appears in the axis components when the tolerance sphere is entered. The max. size is limited by the acceleration of the axis components and the C1 factor.

This curve can be used for all segment transitions. The subtypes 2 and 3 only work for straight line / straight line transitions.

Acute angles at the segment transition

The Bezier splines are generated by default, even at very acute angles. In order to avoid the dynamic values being exceeded, a considerable reduction velocity is required in this case. However, since the dynamics are held constant in the spline, the movement across the spline can be quite slow. In this case it is often practical to start the segment transition with an exact positioning. The command AutoAccurateStop [▸ 61] can be used to avoid having to calculate the angles manually.

4.2.14.5 Bezier curve of the 5th order

Bezier curve of the 5th order

from TwinCAT V2.10 B1308

Command	#set paramVertexSmoothing(<type>; <subtype>; <radius>)#
Parameter <type>	for the Bezier curve of the 5th order: **5**
Parameter <subtype>	1: Constant tolerance radius [▸ 60] 2: Distance between intersection and vertex [▸ 60] 3: Adaptive tolerance radius [▸ 61]
Parameter <radius>	Max. radius of the tolerance sphere

With 5th order Bezier blending, **no** step change in acceleration occurs in the axis components on entry into the tolerance sphere. In other words, the path axis acceleration is always constant if blending is selected.

This curve can be used for all segment transitions. The subtypes 2 and 3 only work for straight line / straight line transitions.

Acute angles at the segment transition

The Bezier splines are generated by default, even at very acute angles. In order to avoid the dynamic values being exceeded, a considerable reduction velocity is required in this case. However, since the dynamics are held constant in the spline, the movement across the spline can be quite slow. In this case it is often practical to start the segment transition with an exact positioning. The command AutoAccurateStop [▸ 61] can be used to avoid having to calculate the angles manually.

4.2.14.6 Old Bezier blending type

Functions for compatibility with existing projects

These functions are provided for compatibility reasons. For new projects, the Bezier curve of the 3rd order [▸ 58] or the Bezier curve of the 5th order [▸ 58] should be used.

Old Bezier blending with paramVertexSmoothing

from TwinCAT V2.10 B1243

Command	#set paramVertexSmoothing(<type>; <subtype>; <radius>)#
Parameter <type>	for Bezier Spline smoothing: **1**
Parameter <subtype>	for Bezier Spline smoothing: **1**
Parameter <radius>	Radius of the tolerance sphere

Sample 1:

```
N10 R57=100
#set paramVertexSmoothing(1; 1;R57)#
```

Old Bezier blending with paramSplineSmoothing

from TwinCAT version 2.7 Build 423

With the aid of smoothing, it is possible to insert a Bezier spline automatically between two geometrical entries. It is only necessary to program the radius of the tolerance sphere. This describes the maximum permissible deviation from the programmed contour in the segment transition. The advantage of this type of smoothing as opposed to rounding with an arc is that there are no step changes in acceleration at the segment transitions.

The radius of the tolerance sphere can be altered at any time within the NC program, and can be switched off again by setting the radius to 0. If the radius is not reset to 0, it remains active until the next interpreter reset or TwinCAT restart.

Command	#set paramSplineSmoothing(<radius>)#
Parameter <radius>	Radius of the tolerance sphere

or alternatively

#set paramVertexSmoothing(...)

Sample 1:

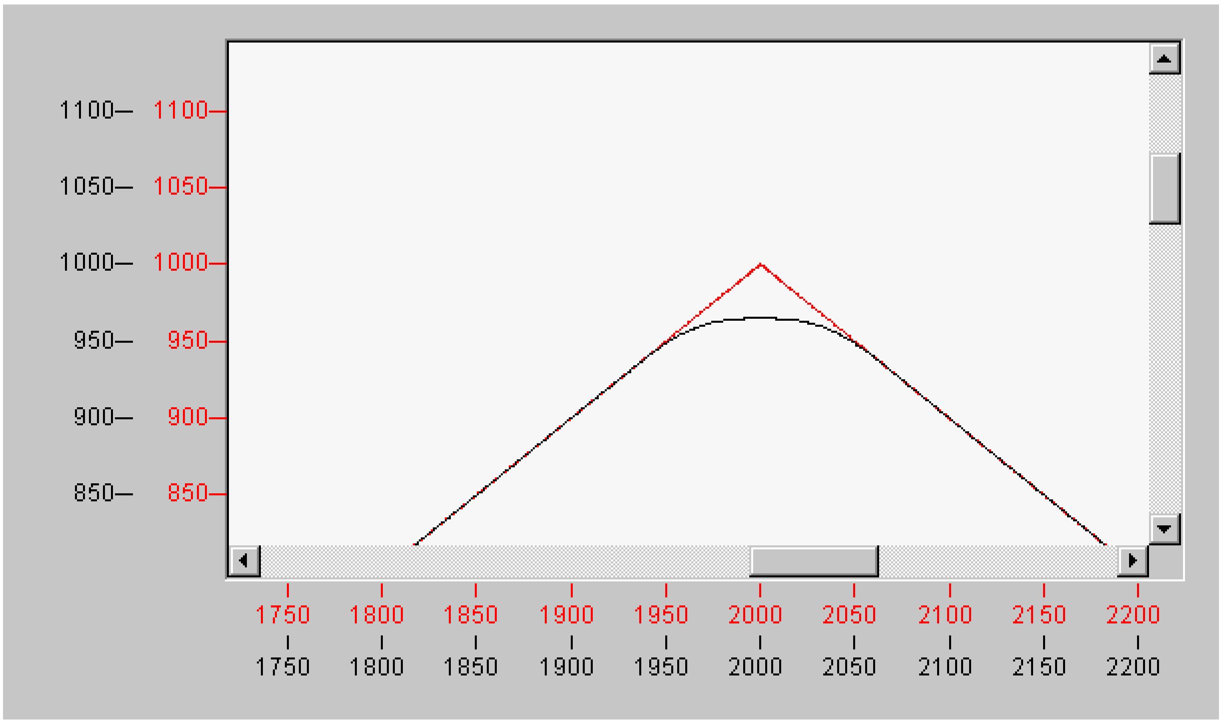
```
N10 R57=100
#set paramSplineSmoothing(R57)#
```

Sample 2:

```
N10 G01 X0 Y0 F6000
N20 X1000
#set paramSplineSmoothing(100)#
N30 X2000 Y1000
N40 X3000 Y0
M30
```

The new parameter is valid from the transition between the subsequent two segments. In example 2, the new value for the tolerance sphere is applicable at the segment transition from N30 to N40.

Figure: Contour with and without spline in the segment transition

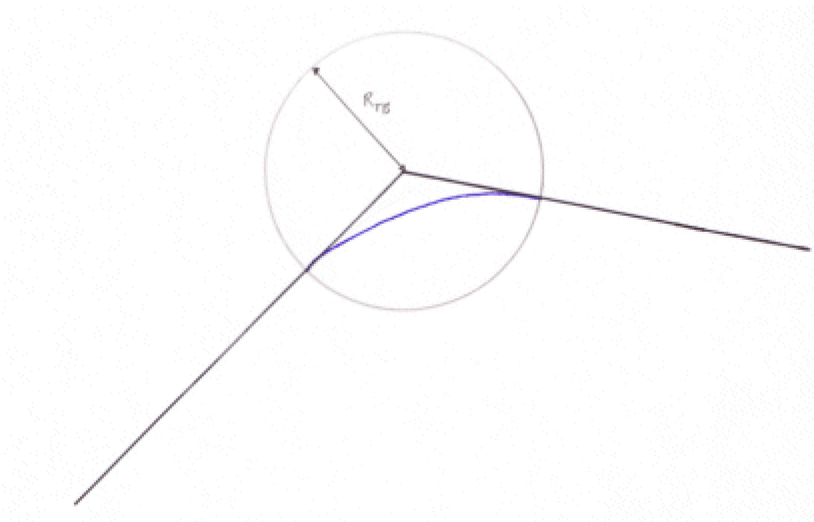


The splines are generated even at very sharp angles as standard. In order to avoid the dynamic values being exceeded, a considerable reduction velocity is required in this case. However, as the dynamics are held constant, the movement across the spline can be quite slow. In this case it is often practical to start the segment transition with an accurate stop. In order to avoid manual calculation of the angles, an 'AutoAccurateStop [▶ 61]' command is available which can also be initiated via the NC program.

4.2.14.7 Subtypes

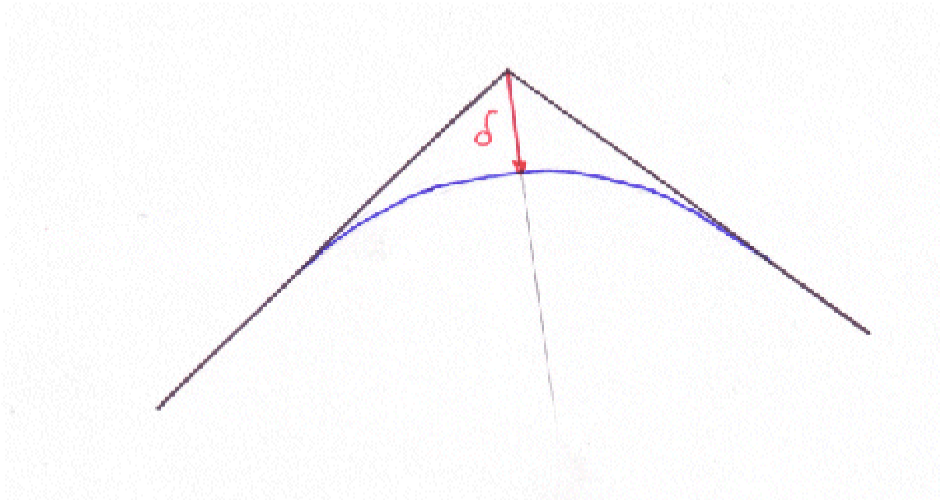
Constant tolerance radius (subtype 1)

If subtype 1 is selected, the maximum tolerance radius (R_{TB}) is used for smoothing. R_{TB} is reduced if and only if the input or output segment is less than 3*R_{TB}.



Distance between intersection and vertex (subtype 2)

The distance between the programmed segment transition and the vertex of the parabola is specified with the subtype 2. The tolerance radius (R_{TB}) results from this. If a segment is too short, then the distance is shortened so that the tolerance radius is a max. of 1/3.



Adaptive tolerance radius (subtype 3)

Within the tolerance radius (including constant tolerance radius) the system ensures that the maximum permissible acceleration is not exceeded. Depending on the deflection angle and the velocity, the maximum axis acceleration within the smoothing segment may be different. The aim of an adaptive tolerance radius is maximum acceleration during smoothing. In order to achieve this, the smoothing radius is reduced based on the programmed velocity and dynamics. In other words, if the programmed velocity is changed, the tolerance radius can also change. The override has no influence on the radius.

4.2.15 Circular Smoothing

from TwinCAT version 2.6 Build 323

It is possible with the aid of circular smoothing to insert an arc automatically between two straight lines. It is only necessary to program the radius of the arc.

The radius of the circular smoothing can be altered at any time within the NC program, and can be switched off again by setting the radius to 0. Rounding must be switched off before the end of the program or a decoder stop [▸ 71].

Command	#set paramCircularSmoothing(<radius>)#
Parameter <radius>	Radius of the circular smoothing arc

Sample:

```
N10 R57=4.5
#set paramCircularSmoothing(R57)#
…
#set paramCircularSmoothing(0)#
N1000 M02
```

When combined with cutter radius compensation, please note that first the radius compensation is calculated, then the circular smoothing is added. The smoothing radius thus refers to the TCP.
The old command `paramGroupVertex` is still supported. However, it cannot be used to transfer R parameters.

Syntax:

```
#set paramGroupVertex(<grp>,<radius>)#
```

The first parameter describes the group to which the circular smoothing refers. This value is currently always 1. The second parameter is used to specify the circular smoothing radius.

4.2.16 Automatic Accurate Stop

from TwinCAT version 2.7 Build 423

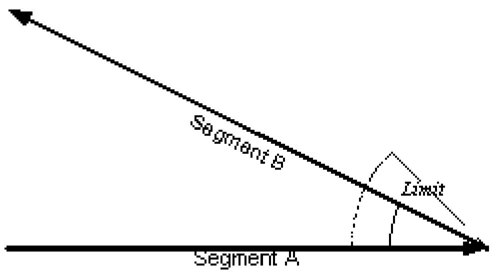
Command	#set paramAutoAccurateStop(<angle>)#
Parameter <angle>	Limit angle (in degrees) after which an accurate stop is inserted
Deselect	#set paramAutoAccurateStop(**0**)#

An accurate stop after a defined limit angle is inserted between 2 segments with the aid of the 'AutoAccurateStop' command.

For circle segments, the angle is calculated from the tangents at the points of entry and leaving.

Sample:

```
#set paramAutoAccurateStop(45)# (angle in
degrees)
N10 G01 X1000 Y0 Z0 F60000 (start position: X0 Y0 Z0)
N20 X0 Y500
...
```



An accurate stop is inserted between segments A and B in this example.

Application field:

This command should be used in conjunction with Bezier blending, if acute angles are programmed in the NC program.

See also:

- Bezier blending function of the 3rd order [▶ 58]
- Bezier blending function of the 5th order [▶ 58]
- 'Old' Bezier blending [▶ 59]

> This function has not yet been implemented for segment transitions with a helix.

4.2.17 Delete Distance to Go

Command	DelDTG
Cancellation	End of block

DelDTG (**del**ete **d**istance **t**o **g**o) is activated block by block via the NC program. This command enables deleting of the residual distance of the current geometry from the PLC (TcNci.lib ItpDelDtgEx [▶ 105]). In other words, if the command is issued while the block is processed, the motion is stopped with the usual deceleration ramps. The NC program then processes the next block. An error message is generated if the PLC command is not issued during the execution of a block with "delete distance to go" selected.

The "delete distance to go" command always effects an implicit decoding stop, i.e. an exact positioning always occurs at the end of the block.

Sample:

```
N10 G01 X0 Y0 F6000
N20 DelDTG G01 X2000
N30 G01 X0
```

DelDTG must not be active with active cutter radius correction.
From TwinCAT V2.10 B1303 a zero offset shift may be active.

4.2.18 Modulo Movements

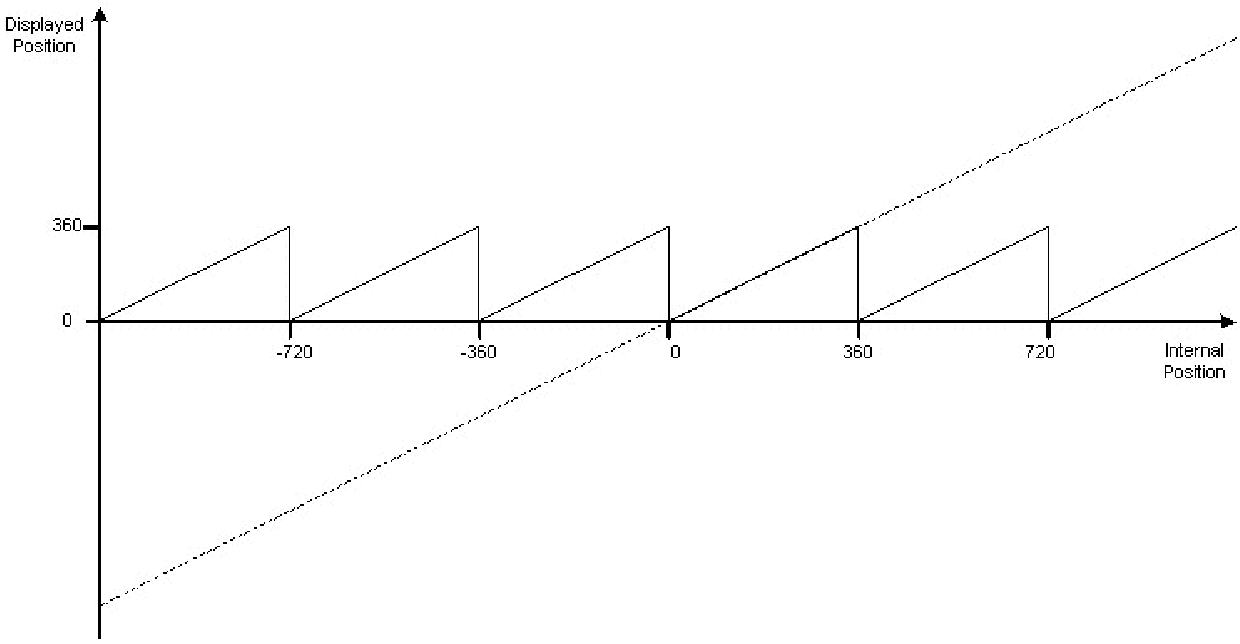
from TwinCAT V2.7 Build 451

Command	MOD[<axis and target modulo position>]
Cancellation	End of block
Parameter 1	Axis for modulo operation
Parameter 2	Arithmetic sign for the direction of rotation (optional)
Parameter 3	Modulo position

The modulo position is programmed in the same way as normal positioning.

The MOD command is effective for specific blocks, and must be explicitly programmed for every axis that is selected for modulo operation. The modulo position's arithmetic sign specifies the direction of rotation.

- Positive sign: The axis moves in the 'greater' direction
- Negative sign: The axis moves in the 'smaller' direction
- Exception: The axis cannot move to modulo -0, since 0 has no sign



Sample 1:

```
N10 G90
N20 G01 MOD[X200] Y30 F600
N30 G01 X200
```

N20 specifies a move in a positive direction for X to modulo position 200. Y is taken to absolute position 30. In block N30, X is moved to absolute (in other words **not** modulo) position 200.

Modulo movements of more than 360 degrees

The MOD command also allows movements of more than 360 degrees to be made.

Modulo position = number of necessary rotations * 360 + modulo position

Sample 2:

```
N10 G90
N20 G01 X3610 F6000
N30 R1=360
N40 G01 MOD[X=R1+20]
```

In this example, the X axis moves 370 degrees to modulo position 20.

Restrictions for modulo movements. Please note:

- No radius compensation may be active for the modulo axis.
- No zero offset shift may be active for the modulo axis.
- During relative programming (G91 [▶ 32]) the modulo command is not evaluated, so that the axis referred to in square brackets is treated as if the MOD command had not been given.

Modulo factor

The modulo factor is constant, and is 360.

4.2.19 Auxiliary axes

from TwinCAT 2.8

Auxiliary axes (also known as Q axes) can be added to an interpolation group in addition to the actual path axes (X, Y & Z). The auxiliary axis can be seen as a type of slave for the path, i.e. it has no direct influence on the path velocity. In addition to the 3 path axes, 5 auxiliary axes can also be interpolated for each channel.

The function block 'CfgBuildExt3DGroup' from TcNcCfg.lib, for example, may be used for adding to the interpolation group from the PLC.

Syntax

The auxiliary axes are addressed as Q1..Q5 from the part program. The numerical value may be assigned directly, or an R-parameter.

Sample 1:

```
(start position X=Y=Z=Q1=0)
N10 G01 X100 Q1=47.11 F6000
...
```

If an NC block is programmed with one or more path axes and an auxiliary axis, both axes start **simultaneously** and also reach the destination **together**.

Swivelling of the auxiliary axes

The term "swivelling of the auxiliary axes" is used if the path length within a motion set is zero. This is often the case during 'swivelling' of a tool, with the feed angle relative to the contour being changed.

Since the path length is zero, there is no link to the path, and the movements of the auxiliary axes are calculated via a virtual path. However, this has no influence on the real path of X, Y and Z, but here too all auxiliary axes are started simultaneously and also arrive at the destination simultaneously.

Here too, the velocity is specified via the F-parameter and now refers to the auxiliary axis with the greatest travel distance.

Sample 1:

```
(start position X=Y=Z=Q1=Q1=0)
N10 G01 X100 F6000
N20 Q1=100 Q2=200 F3000
...
```

In N20, the velocity of Q2 is now 3000 and that of Q1 is 1500, since the travel distance is Q1=Q2/2.

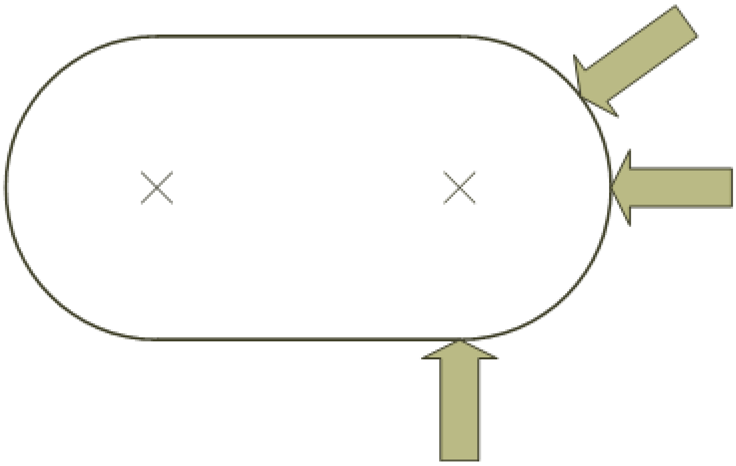
4.2.19.1 Calculation of the velocity

Initially, only the path axes (X, Y and Z) are considered for the calculation of the path velocity.
The path and the travel distance of the individual auxiliary axes result in a fixed coupling ratio for each auxiliary axis within a segment. The target velocity of the auxiliary axis is thus also known. If this velocity is greater than the permitted maximum velocity for this auxiliary axis, the path velocity is reduced until the upper speed limit is adhered to. In other words, exceeding of the velocity limits of the auxiliary axes also has an indirect effect on the path velocity.

4.2.19.2 Path velocity at segment transitions

The reduction of the path velocity is explained below by means of an example. The contour of a stadium is particularly suitable for this purpose. The aim is for the feed angle of a tool relative to the path tangent to remain constant.

On the stadium straight, the orientation of the tool remains constant, i.e. the tool is not turned. In contrast, the orientation relative to the base coordinate system must be changed continuously within the circle. Assuming the path velocity in the transition between straight and circle is not reduced to zero, a step change in velocity is inevitably generated for the swivelling axis (but not for the path axes!).



This step change in velocity of the auxiliary axis is freely parameterizable and depends on the machine. Extreme cases would be for the path velocity at such segment transitions to be reduced to zero, or for the velocity not to be reduced at all.

The global axis parameter 'VeloJumpFactor', which can be set individually for each axis, is used for the parameterization. The resulting velocity and the calculation is described in more detail in the TwinCAT NCI appendix on page Parameterization [▶ 247].

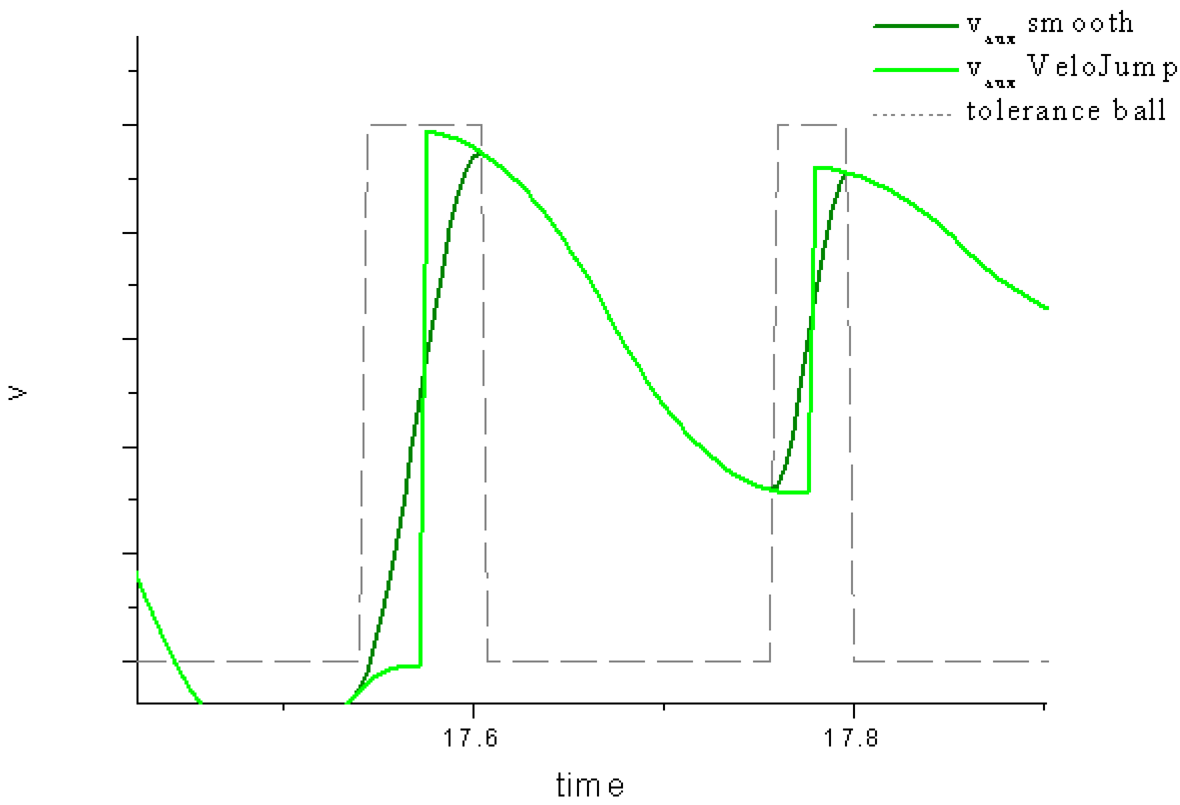
Smoothing of the velocity at segment transitions

from TwinCAT 2.9 Build 940

As has been described above, step changes in velocity can occur at the segment transitions. The size of these steps can be affected by the VeloJump parameter.

It is further possible for a tolerance sphere to be specified for every auxiliary axis. This sphere is symmetrical with the path at a segment transition. On entering this sphere, the velocity of the auxiliary axis is continuously modified to reach the set velocity at the exit of the sphere. The step changes in velocity are, in other words, eliminated. This does imply that the auxiliary axis is subject to a positional error when it is within the sphere. On entering the sphere the change to the new target velocity of the axis starts immediately. This avoids an overshoot in position, and the position is again precise at the borders of the sphere.

If it happens that the specified sphere is larger than 1/3 of the path, its radius is automatically restricted to that value.



Selection and Deselection

The tolerance sphere of the auxiliary axis is an axis parameter (IO: 0x108). It can be set in the axis interface in the System Manager or via ADS [▸ 282].

Parameterization of the axis parameters

The parameters described here only take effect for axes that are in the interpolation group as auxiliary axes (Q1..Q5). For path axes (x,y,z), the parameters 'Veloc. discontinuity factor', 'Tolerance sphere auxiliary axis' and 'Max. positional deviation, auxiliary axis' have no influence.

Diagnostics

It is possible to record the tolerance sphere of each auxiliary axis and the positional error that results from this for diagnostic purposes. It is also possible to access the variables via ADS. They are to be found in the group status [▸ 272] (IO: 0x54n and 0x56n).

Effect on VeloJump, if the size of the tolerance sphere is reduced

from TwinCAT V2.9, Build 1013

If the size of the tolerance sphere has to be reduced due to the given geometry, the VeloJump parameter is automatically adjusted for this segment, i.e. the path velocity in the transition is reduced more strongly, so that the dynamics of the auxiliary axis is not exceeded for smaller tolerance spheres.

Positional deviation of the auxiliary axis if the tolerance sphere has to be reduced

The parameter 'maximum permitted positional deviation of the auxiliary axis' **only** takes effect if the tolerance sphere would have to be reduced due to the geometry.

The aim is to keep the path velocity high despite the smaller tolerance sphere, as long as the resulting position error does not exceed a threshold value. To this end the velocity of the auxiliary axis is kept constant and the position error is calculated. If the error is smaller than the maximum positional deviation the velocity is maintained for this segment transition, and the resulting position error is compensated in the next segment (the tolerance sphere then becomes unnecessary for this segment transition).

In the event that the position error would exceed the maximum deviation, the reduced tolerance sphere takes effect, including the VeloJump factor, and the path velocity is reduced if necessary.

Sample 1:

Initial conditions:

- Set tolerance sphere: 5
- Max. positional deviation: 1
- The given geometry results in an effective tolerance sphere of 0.2, for example
- The potential positional deviation is 0.3

Resultant behavior:

- The path velocity remains at a constant high level
- The velocity of the auxiliary axis is kept constant
- For this transition no tolerance sphere is required
- The resulting positional deviation is compensated in the subsequent segment

Example2:

Initial conditions:

- Set tolerance sphere: 5
- Max. positional deviation: 1
- The given geometry results in an effective tolerance sphere of 1.2, for example
- The potential positional deviation is 1.1

Resultant behavior:

- The tolerance sphere is adjusted
- The VeloJump parameter is adjusted
- The path velocity is reduced at the segment transition
- There is **no** positional deviation that has to be compensated

Parameterization

The parameterization of the maximum permitted positional deviation is an axis parameter [▸ 282]. By default this feature is switched off (tolerance = 0.0)

4.3 Supplementary Functions

4.3.1 M-Functions

Task: Signal exchange between NC and PLC

A range of equipment, such as collet chucks, drill drives, transport equipment etc. is best not driven directly by the NC, but indirectly, using the PLC as an adapting and linking controller. This makes it easy to consider feedback or safety conditions, without having to adapt the NC program, or even the NC system. The NC's M-functions involve digital signal exchange: functions are switched on or off, activated or deactivated. The transfer of numerical working parameters is not provided for here, but can be implemented in other ways (H-functions [▸ 71], T-numbers [▸ 71] etc.).

4.3.1.1 Available M-functions

Number of M-functions

A total of 160 M-functions are available per channel

M function	Meaning
0..159	Freely definable M-functions (except 2, 17, 30)
2	Program end
17	End of subroutine
30	Program end with deletion of all fast M functions

All M-functions (apart from the 3 pre-defined M-functions - M2, M17, M30) are freely definable. This means that, depending on the machine type, M8 can be used to switch on a cooling medium or indeed for any other functionality, for example. The machine manufacturer can select the function as required.

Like any other rules, the rules for reserved M-functions are read when TwinCAT is started. Additionally, an internal code is generated for these functions, which is responsible for the behavior described. These 3 M-functions therefore do not have to be described in the table. It makes sense to parameterize M2 and M30, even if M-functions are used.

Types of M-functions

In principle, two types of signal exchange are available: fast signal bits, or transfer secured by handshake.

Secured Handshakes

M-functions that require feedback must be processed using bi-directional signal exchange between the NC and the PLC. If an M-function of type handshake is programmed, the velocity is reduced to 0 at this point. The PLC uses the ItpIsHskMFunc [▸ 131] function to check whether an M-function with handshake is present, in which case the number of the M-function is determined via ItpGetHskFunc [▸ 131]. The NC is in a waiting state and will not process further NC commands until the PLC has acknowledged the M-function. Processing of the NC program continues once acknowledgement has been received from the PLC (ItpConfirmHsk [▸ 104]).

This procedure permits the operation of the equipment controlled by the NC to be securely coordinated with the equipment controlled by the PLC. It is therefore advisable to acknowledge the M-function for starting the spindle (e.g. M3) once a minimum speed has been reached.

Since this kind of M-function involves synchronous functions, it is only ever possible for one M-function with handshake to be active in the NC program.

Fast signal bits

If no feedback is required from the PLC, fast signal bits can be used for activating M-functions. Since the NC does not have to wait for the PLC with these M-functions, look-ahead [▸ 30] can combine the segments. In this way it is possible to apply an M-function without velocity reduction.

This type of M-function is useful for in-motion activation of a nozzle for applying adhesive, for example.

A combination of fast signal bits and handshake is also possible. Since a handshake always requires acknowledgement from the PLC, the velocity has to be reduced to 0 in this case.

4.3.1.2 Resetting of M-functions

Resetting fast signal bits

The signal bits are active until they are reset explicitly, or until an M30 (end of program) or channel reset is executed.

Resetting with reset list

Each M-function can reset up to 10 fast M-functions. If cooling medium is switched on with M8, for example, the cooling medium can be switched off again with M9. To this end simply enter M8 in the reset list for M9.

Automatic reset

During parameterization of the M-function an 'auto-reset flag' can be set. This means that the M-function is automatically reset at the end of the block.

In order for the PLC to be able to see the signal, the duration of the motion block must be long enough, or this M-function is combined with a handshake. The handshake may come from the same or a different M-function.

Reset from the PLC

The fast M-functions can be reset from the PLC via the 'ItpResetFastMFunc [▸ 192]' function block. For reasons of transparency, mixed resets using via PLC and NC should be avoided.

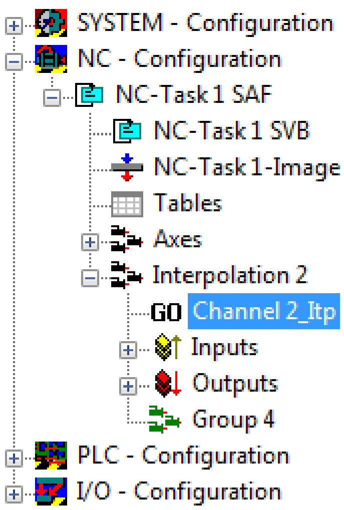
Delete all pending M functions

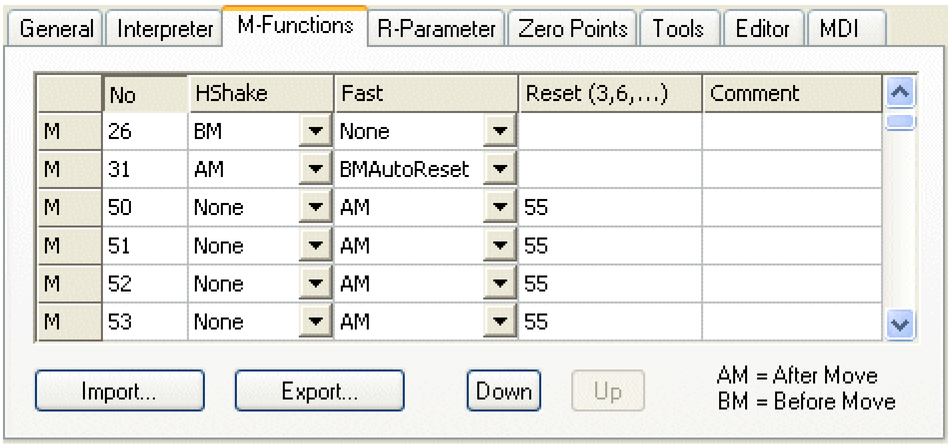
A channel stop and a channel reset are used to reset all pending M functions. This is true for the 'handshake' type M functions, and also for the fast signal bits. If the NC program is terminated properly with M30, all M-functions are also cleared.

4.3.1.3 Parameterization of M-functions

The M-functions are parameterized via the TwinCAT system manager. A dedicated M-function table is used for each interpolation channel.

A restart of the TwinCAT configuration is required to activate a configuration of M-functions.





No

Number of M-function to be parameterized. The value must be between 0 and 159

HShake

If a value other than 'None' is entered, the M-function is of type 'Handshake'

- *None*: no handshake is executed

- *BM* (Before Move)**:** If a movement is programmed in the same block, the handshake occurs **before** the movement
- *AM* (After Move): If a movement is programmed in the same block, the handshake occurs **after** the movement

Fast

If a value other than 'None' is entered, a 'fast signal bit' type M-function is executed

- *None***:** no fast M-function is executed
- *BM* (Before Move): If a movement is programmed in the same block, the output occurs **before** the movement.
- *AM* (After Move): If a movement is programmed in the same block, the output occurs **after** the movement.
- *BMAutoReset* (Before Move & automatic reset): If a movement is programmed in the same block, the output occurs **before** the movement. In addition, the M-function is automatically canceled at the end of the block, i.e. the M-function is active on a per-block basis. In order to ensure that the PLC recognizes the M-function, the duration of the associated motion block must be long enough (at least 2 PLC cycles), or an additional M-function with handshake should be programmed.
- *AMAutoReset* (After Move & automatic reset): This parameterization is only meaningful if either an M-function of type handshake is programmed at the same time (or parameterized), or if the M-function is only used for resetting other M-functions. Without an additional handshake the PLC will usually not be able to detect this M-function.
- All other combinations can be selected for compatibility reasons.

Reset

Up to 10 M-functions can be entered for cancellation when a reset is called.

In the event that no reset-signal-bit is in fact set, the bits to be cleared are reset immediately before setting the new signal bits.

Import/Export

The M-functions are parameterized individually for each channel. The parameterization can be transferred to other channels via the import/export function.

4.3.1.4 Combination of M functions

- Within each line, only **one** 'handshake' type M function must be programmed!
- Within a single line, up to 10 'signal bit' M functions may be programmed
- A combination of the two options above is allowed

Sample:

```
N10 G01 X1000 F60000
N20 M10 M11 M12 X2000 (M10 & M11 are signal bits)
(M12 is of type handshake)
M30
```

Examples of meaningful and practically applicable rule combinations:

- An M-function is to be active for the duration of a movement and then be automatically cleared. Select 'None' in the HShake column and 'BMAutoReset' in the Fast column. The signal bit generated could, for instance, control a glue application valve.
- An M-function starts a drill motor, and the subsequent movements may only be started after an appropriate run-up time, and then only when the drill is ready for operation. Select 'BM' in the HShake column. The PLC acknowledges the request after a certain delay time and only if the frequency converter is ready for operation.

- A drill motor is started with an M-function. In order not to have to wait for the drive to run up, the M-function is programmed in the block before the one for the drill movement. In the following movement (the drill movement itself) it is however still essential to ensure that the drive has reached its full rotation speed. For this variant either two different M-functions have to be used (lead signal as signal bit, safety query as handshake) or a Fast 'BMAutoReset' and HShake 'AM' M-function is used.

4.3.1.5 Behavior in case of an error

If a run-time error occurs during the execution of an NC program (e.g. following error monitoring is activated), the NC program is interrupted. In this case the M functions, provided they are set, remain pending. This means that the PLC program may have to ensure that M functions are not executed.

4.3.2 H, T and S Parameters

H-, T- and S-parameters are used to transfer parameters from the NC interpreter to the PLC.

In this context the H-parameter stands for auxiliary parameter and is of type DINT (32 bit signed).

The T and S parameters are of type WORD, and stand for Tool and Spindle.

Sample:

```
H=4711
R1=23
S=R1
T4711
```

No R parameter can be assigned for the T parameter. Furthermore, the assignment is made without assignment operator ('=').

T and S parameters take effect at the start of a block, H parameters take effect at the end of the programmed block.

4.3.3 Decoder stop

Code	Function
@714 [▸ 71]	Decoder stop
@716 [▸ 72]	Decoder stop with axis position rescan
@717 [▸ 72]	Decoder stop with trigger event, conditional decoder stop

4.3.3.1 Decoder stop (@714)

The interpreter offers the option to execute a decoder stop in the NC program. In this case the interpreter waits until a certain external event occurs. Execution of the NC program does not continue until this event has taken place.

A decoder stop can be used, for instance, to switch block skipping [▸ 30] on or off from the PLC, or to re-assign R parameters [▸ 37].

Two events are available for continuing processing:

- Acknowledgment of an M-function [▸ 67]
- SEC task is empty

Acknowledgment of an M-function

Decoding of the NC program is interrupted until the M-function [▸ 67], which is programmed immediately prior to the decoder stop, is acknowledged. In other words, the M-function must be of type "handshake".

Sample 1:

```
N10...
N20 M43 (M-function with handshake)
N30 @714 (decoder stop)
N40 ...
```

SEC task is empty

The decoder stop does not necessarily have to be programmed in conjunction with an M-function. If the SEC task runs out of travel commands, an event is sent to the interpreter. This event causes the interpreter to start up again.

The decoder stop must not be programmed with active tool compensation or circular smoothing, as these will then no longer function.

4.3.3.2 Decoder Stop with Axis Position Rescan (@716)

In addition to the usual decoder stop (see Decoder stop (@714) [▶ 71]), there is also a decoder stop in which the axis positions of the interpolation channel are read again. This stop is required, if, for example, axes are moved during a tool change via PTP and are subsequently not returned to the old position. Another possible application is a change in axis configuration via an M-function (with handshake).

If a decoder stop with rescan is programmed, it is essential to program an M-function with handshake immediately before it.

Sample 2:

```
N10...
N20 M43 (M function with handshake carries out a tool change, for
example)
N30 @716 (Decoder stop with rescan)
N40 ...
```

The decoder stop must not be programmed with active tool compensation or circular smoothing, as these will then no longer function.
From TwinCAT V2.10 B1303, a zero offset shift may be active.

4.3.3.3 Decoder Stop with external trigger event (@717)

Sometimes the question of whether the NC part of the program must wait or can continue may depend, for instance, on events in the PLC. With the 2 types of M-functions [▶ 67] this can give rise to the following problems:

- Handshake: Because of the M-function's handshake the path velocity must be brought to 0 at the location where the M-function is programmed, after which confirmation is awaited from the PLC.
- On The Fly (also known as a fast M-function): Because no confirmation from the PLC is waited for, there is also no way for the partial program to wait for the PLC.
- Even a combination of the two types of M-function does not help here.

Sample:

During positioning with a flying M-function, a process A is initiated by the NC partial program. It is assumed here that the block of processes in the NC program is typically long enough for process A to be completed in the PLC. If A is ready, then the NC partial program should execute the next segment with look-ahead. In case A is not ready, however, then the NC should stop at the end of the segment and wait until process A has finished. It is exactly this scene that can be implemented with the command @717. The PLC here sends the so-called 'GoAhead [▶ 127]' command when process A has finished.

```
N10 ...
N20 G0 X0 Y0 Z0
N30 G01 X500 F6000
N40 M70 (flying M-function that triggers process A)
N50 G01 X700
N60 @717 (decoder stop with external trigger event)
N70 G01 X1000
N80 ...
```

If the GoAhead signal reaches the PLC early enough, then blocks N50 and N70 are linked by look-ahead, and the path velocity is not then reduced. If the signal arrives during the deceleration phase of N50, then the velocity is once more increased. Otherwise, the machine waits for the signal from the PLC.

The decoder stop must not be programmed with active tool compensation or circular smoothing, as these will then no longer function.

from TwinCAT V2.10 B1319:

The function block 'ItpGoAheadEx' returns the error code 0x410A, if no @717 is present in the interpreter at the time of the call.

4.3.4 Jumps

Code	Function
@100 [▸ 73]	Unconditional jump
@121 [▸ 73]	Jump if unequal
@122 [▸ 73]	Jump if equal
@123 [▸ 74]	Jump if less or equal
@124 [▸ 74]	Jump if less
@125 [▸ 74]	Jump if greater or equal
@126 [▸ 74]	Jump if greater
@111 [▸ 74]	Case block

Unconditional jump

Command	@100
Parameter	K or R

The parameter describes the jump destination. This must have an indication of direction ('+' or '-').

Sample 1:

```
N10 ..
...
N120 @100 K-10
```

In this example, execution continues from line 10 after line 110 has been interpreted. The sign indicates the direction in which the line to be searched can be found.

Jump if unequal

Command	@121	
Parameter 1	R<n>	Comparison value
Parameter 2	K or R<m>	Comparison value
Parameter 3	K	Jump destination with direction indication

Sample 2:

```
N10 ..
...
R1=14
N120 @121 R1 K9 K-10
N130 ...
```

Jump if equal

cf. Jump if not equal [▸ 73]

Jump if less or equal

cf. Jump if not equal [▸ 73]

Jump if less

cf. Jump if not equal [▸ 73]

Jump if greater or equal

cf. Jump if not equal [▸ 73]

Jump if greater

cf. Jump if not equal [▸ 73]

Case block

Command	@111	
Parameter 1	R<n>	Comparison value
Parameter 2	K or R<m>	First comparison value
Parameter 3	K	First jump destination
Parameter 4	K or R<m>	Second comparison value
...		

Sample 3:

```
N100 R2=12 (R2=13) (R2=14)
N200 @111 R2 K12 K300
K13 K400
K14 K500

N300 R0=300
N310 @100 K5000

N400 R0=400
N410 @100 K5000

N500 R0=500
N510 @100 K5000

N5000 M30
```

A case block is made in line 200. If R2 = 12 a jump is made to line 300.

If R2 = 13, the jump destination is line 400. If R2 = 14, the jump destination is line 500.

In the event that none of the conditions is satisfied, execution simply continues with the next line (in this case, line 300).

4.3.5 Loops

The various types of loop are described below.

Code	Loop type	Aborting condition
@131	While Loop [▸ 75]	while equal
@132	While Loop [▸ 75]	while not equal
@133	While Loop [▸ 75]	while greater
@134	While Loop [▸ 75]	while greater or equal
@135	While Loop [▸ 75]	while less
@136	While Loop [▸ 75]	while less or equal
@141	Repeat Loop [▸ 75]	repeat until equal

Code	Loop type	Aborting condition
@142	Repeat Loop [▶ 75]	repeat until not equal
@143	Repeat Loop [▶ 75]	repeat until greater
@144	Repeat Loop [▶ 75]	repeat until greater or equal
@145	Repeat Loop [▶ 75]	repeat until less
@146	Repeat Loop [▶ 75]	repeat until less or equal
@151	For-To Loop [▶ 76]	
@161	For-DownTo Loop [▶ 76]	

Loops can be nested.

While loops

Command	@13<n>	where 1<= n <= 6
Parameter 1	R<m>	Comparison value
Parameter 2	K or R<k>	Comparison value
Parameter 3	K	Jump destination for the case that the condition is not met

A while loop is executed for as long as the condition is satisfied. The test is made at the beginning of the loop. If the condition is not or no longer met, a jump to the specified line takes place (parameter 3).

At the end of the While loop an unconditional jump (@100 [▶ 73]) must be programmed. The target of this jump is the line number of the while loop.

The loop’s exit condition is specified with <n>.

Sample 1:

```
N100 R6=4
N200 @131 R6 K4 K600 (K600 is the target of the jump, when the condition is no longer satisfied)
N210 ...
N220 @100 K-200

N600 ...

N5000 M30
```

The loop (lines 200 to 220) is repeated for as long as R6 = 4. Once the condition is no longer satisfied, execution jumps to line 600.

Repeat loops

Command	@14<n>	where 1<= n <= 6
Parameter 1	R<m>	Comparison value
Parameter 2	K or R<k>	Comparison value
Parameter 3	K	Jump destination at the start of the loop

In a repeat loop, the interrogation takes place at the end of the loop. This means that the loop is executed at least once. The loop is only ended, to continue with the rest of the program, when the condition is satisfied.

Sample 2:

```
N200 ...
N210 ...

N300 @141 R6 K25 K200
```

The loop is repeated until R6 = 25. The second constant in line 300 gives the jump target (the start of the loop).

For-To loops

Command	@151 <variable> <value> <constant>

A for-to loop is a counting loop that is executed until the *variable* equals the *value*. The test is made at the beginning of the loop. If that condition is satisfied, execution jumps to the line specified by the *constant*.

The variable must be incremented (@620) at the end of the loop, and there must be an unconditional jump to the start of the loop.

Sample 3:

```
N190 R6=0
N200 @151 R6 K20 K400
N210 ...
N290 @620 R6 (increment R6)
N300 @100 K-200
```

For-Downto Loops

Command	@161 <variable> <value> <constant>

A for-downto loop is a counting loop. The behaviour is similar to that of a for-to loop. The difference is merely that the variable is decremented (@621) by 1 at the end of the loop.

4.3.6 Subroutine techniques

As in other fields, it is also valuable in NC programming to organize frequently used command sequences as subroutines. This makes it possible to employ pre-prepared and tested functions in various workpiece programs.

Subroutines are identified within a program by a number. This number must be unique: there must be only one subroutine with a particular number (1..>2.000.000.000).

As interpretation proceeds, the calling program is suspended. The text of the subroutine is worked through, as often as necessary. Processing then returns to the calling program after the call location.

It is of course possible for one subroutine to call another subroutine. This call is executed in a similar way. This causes a stack of return information to be created. For technical reasons this nesting of subroutines is presently limited to 20 levels.

Definition of a Subroutine

The code for a subroutine can be written to the same file as the calling program. In this case the subroutine is linked directly: it is automatically also loaded as the file is read. If it is to be made generally available then it must be written in its own file that must be located in the CNC directory.

The name of the file begins with the letter 'L', and continues with a string of digits. This digit string must repeat the subroutine number, without any leading '0's.

The code should contain a label to indicate the starting point of the subroutine. Like the file name, it consists of the letter 'L' and the digit sequence described above.

The interpreter starts immediately after this label.

Subroutine syntax:

```
(Datei L2000.NC)
L2000
N100...
N110...
...
N5000 M17 (return command)
```

Calling a Subroutine

The following syntax must be used to call a subroutine from some block within the NC program. It is important that the expression "L2000" does not stand at the start of the line, in order to avoid confusion with a subroutine label.

```
(syntax of the subroutine call)
N100 L2000
```

In the following sample the expression "P5" causes the subroutine to be repeated 5 times.

```
(n-fold subroutine call (here: 5- fold))
N100 L2000 P5
```

Dynamic subroutine call

In some cases the subroutine to be called is not known until runtime. In this case the subroutine can be called with an R-parameter, thereby avoiding the need for a CASE instruction. The value for R must be allocated or calculated in a dedicated line.

```
(Dynamic call of a subroutine)
N099 R47=R45+1
N100 L=R47
```

Parameter passing

Parameters are passed to subroutines with the aid of R-parameters [▶ 37]. Note that R-parameters are not saved automatically (see Rescuing R-parameters [▶ 37]).

Use of Parameters

R-parameters can, in general, be freely used within subroutines. This has a number of consequences that can lead to errors if they are not borne in mind. On the other hand their careful use offers the NC-programmer a range of useful working techniques.

Results of Subroutines

If an R-parameter is changed without its contents being saved and restored, the change is effective after a return from the subroutine. If this was not intended, the result can be machine behavior that was not planned.

This feature can however be deliberately used in order to make the continuation of the processing dependent on the results of a subroutine. No restriction need be considered here other than those on the R-parameters.

Sample:

```
N100 L2000
N110 R2=R3+R4
...
N999 M30
```

```
L2000
N10 R3=17.5
N20 R4=1
N99 M17
```

Values are specified here in a subroutine. The values are then used in the calling program.

Ending a Subroutine

A subroutine is ended with M17.

4.3.7 Dynamic Override

from TwinCAT V2.6, Build 320

Command	DynOvr=<value> or DynOvr = R<n>

Cancellation	DynOvr=1

Sample:

```
N10 G01 X100 Y200 F6000
N20 DynOvr=0.4
N30 G01 X500
```

'DynOvr' can be used to make percentage changes to the dynamic parameters of the axes in the group while the NC program is running. This also results in new values for the motion dynamics. The new dynamic values become valid, without any stop, when the line is executed. This means, for the example illustrated above, that in block 10 the old values will still be used for the deceleration, while the new values will be used for acceleration in block 20.

Scope of Definition

0 < DynOvr ≤ 1

See also change in path dynamics [▸ 78].

4.3.8 Altering the Motion Dynamics

from TwinCAT version 2.6 Build 323

Command	#set paramPathDynamics
Parameter <acc>	Value of the maximum permitted path acceleration in mm/s^2
Parameter <dec>	Value of the maximum permitted deceleration in mm/s^2
Parameter <jerk>	Value of the maximum permitted jerk in mm/s^3.

Sample:

```
N10 G01 X100 Y200 F6000
N15 R4=3000
N20 #set paramPathDynamics( 700; 700; R4 )#
N30 G01 X500
```

The 'paramPathDynamics' command can be used to change the path dynamics as the NC program is running. The new dynamic values become effective as from the line in which they are programmed. For the example illustrated, this means that the whole of block 10 is still treated with the default values. The new parameters are used for block 30 from the start of the segment.

This command limits all path axes to the parameterized dynamic values, although the path itself can have higher dynamics, depending on its orientation. The dynamics of auxiliary axes remains unchanged.

See also Dynamic override [▸ 77].

The dynamics values changed via the NC program remain active until the next reset of the interpreter or restart of TwinCAT.
The old command 'paramGroupDynamics' remains effective. However, it cannot be used to transfer R parameters.

Command	#set paramGroupDynamics(<grp>,<acc>,<dec>,<jerk>)#
Parameter <grp>	Group for which the alteration of the motion dynamics is to be effective. Presently always 1.
Parameter <acc>	Value of the maximum permitted path acceleration in mm/s^2
Parameter <dec>	Value of the maximum permitted deceleration in mm/s^2
Parameter <jerk>	Value of the maximum permitted jerk in mm/s^3.

Sample:

```
N10 G01 X100 Y200 F6000
N20 #set paramGroupDynamics( 1, 700, 700, 3000 )#
N30 G01 X500
```

Change in axis dynamics

from TwinCAT version 2.9 Build 948

Command	#set paramAxisDynamics
Parameter <axis>	Axis in the interpolation group: X: 0 Y: 1 Z: 2 Q1: 3 ... Q5: 7
Parameter <acc>	Value of the maximum permitted acceleration in mm/s^2
Parameter <dec>	Value of the maximum permitted deceleration in mm/s^2
Parameter <jerk>	Value of the maximum permitted jerk in mm/s^3.

Sample:

```
N10 G01 X100 Y200 F6000
N15 R4=30000
N20 #set paramAxisDynamics( 0; 1500; 1400; R4 )#
N30 G01 X500
```

'paramAxisDynamics' can be used to change the axis dynamics at runtime. Generally the behavior is the same as for 'paramPathDynamics', except that here the dynamics can be specified individually for each axis.

4.3.9 Change of the Reduction Parameters

C0 reduction [▶ 79]
C1 reduction [▶ 80]
C2 reduction [▶ 81]

C0 reduction

TwinCAT Version 2.6, Build 323 and higher

In some types of machine it is not absolutely necessary to reduce the path velocity to 0 at knee-points. 2 reduction methods are available

- VeloJump
- DeviationAngle

VeloJump

Command	#set paramVeloJump(<C0X>; <C0Y>; <C0Z>)#
Parameter <C0X>	Reduction factor for *C0* transitions: *X* axis: *C0X* ≥ *0.0*
Parameter <C0Y>	Reduction factor for *C0* transitions: *Y* axis: *C0Y* ≥ *0.0*
Parameter <C0Z>	Reduction factor for *C0* transitions: *Z* axis: *C0Z* ≥ *0.0*

The 'paramVeloJump' command can be used to alter the velocity step change factors as the NC program is running. The new values come into effect via the block execution in the programmed line. You can find further details of the means of operation in the appendix under Parameterization [▶ 247].

Sample:

```
N10 G01 X100 Y200 F6000
N20 R2=4.5
N30 #set paramVeloJump( 1.45; R2; R2 )#
N40 G01 X500
```

Resetting parameters

The VeloJump parameters changed via the NC program remain active until the interpreter is next reset and/or TwinCAT has been restarted.

DeviationAngle (not yet released)

Command	#set paramDevAngle(<C0Factor>; <AngleLow>; <AngleHeigh>)#
Parameter <C0Factor>	Path reduction factor for *C0* transitions: 1.0 ≥ *C0* ≥ *0.0*
Parameter <AngleLow>	Angle in degrees from which reduction takes effect: $0 \le \varphi\_l < \varphi\_h \le \pi$
Parameter <AngleHeigh>	Angle in degrees from which reduction to $v\_link = 0.0$ takes effect: $0 \le \varphi\_l < \varphi\_h \le \pi$

The 'paramDevAngle' command is used to describe the parameters for the C0 reduction. In contrast to the VeloJump reduction method, in which the velocity step change is influenced directly, in the DeviationAngle method the velocity step change depends upon the angle. You can find further details of the means of operation in the appendix under Parameterization [▶ 247].

Sample:

```
N10 G01 X100 Y200 F6000
N20 #set paramDevAngle(0.15; 5; 160 )#
N30 G01 X500
```

Resetting parameters

The DeviationAngle parameters changed via the NC program remain active until the interpreter is next reset and/or TwinCAT has been restarted.

C1 reduction factor

TwinCAT Version 2.6, Build 323 and higher

Command	#set paramC1ReductionFactor(<C1Factor>)#
Parameter <C1Factor>	C1 reduction factor

The 'paramC1ReductionFactor' command is used to change the C1 reduction factor while the NC program is running.

The new parameter is applied in the segment transition between which the reduction factor is programmed. In the example shown, the new value for the C1 reduction is therefore already effective in the segment transition from N10 to N30.

A floating point value or an 'R parameter' can be provided as parameter.

You can find further details of the means of operation in the appendix under Parameterization [▶ 247].

Sample:

```
N10 G01 X100 Y200 F6000
N20 #set paramC1ReductionFactor( 0.45 )#
N30 G01 X500
```

Resetting parameters

The C1 reduction factor changed via the NC program remains active until the interpreter is next reset and/or TwinCAT has been restarted.

C2 reduction factor

TwinCAT Version 2.7, Build 422 and higher

Command	#set paramC2ReductionFactor(<C2Factor>)#
Parameter <C2Factor>	C2 reduction factor

The 'paramC2ReductionFactor' command is used to change the C2 reduction factor while the NC program is running.

The command takes effect in the segment transition for which the reduction factor is programmed. In the example shown, the new value for the C2 reduction is therefore already effective in the segment transition from N10 to N30.

A floating point value or an 'R parameter' can be provided as parameter.

Sample:

```
N10 G01 X100 Y200 F6000
N20 #set paramC2ReductionFactor( 1.45 )#
N30 G01 X500
```

Resetting parameters

The C2 reduction factor changed via the NC program remains active until the interpreter is next reset and/or TwinCAT has been restarted.

4.3.10 Change of the Minimum Velocity

TwinCAT Version 2.6, Build 323 and higher

Command	#set paramVeloMin(<VeloMin>)#
Parameter <VeloMin>	Minimum path velocity

The 'paramVeloMin' command can be used to alter the minimum path velocity while the NC program is running. The new velocity comes into effect via the block execution in the programmed line.

A floating point value or an 'R parameter' can be provided as parameter.

Sample:

```
N10 G01 X100 Y200 F6000
N20 #set paramVeloMin( 2.45 )#
N30 G01 X500
```

Resetting parameters

The minimum velocity changed via the NC program remains active until the interpreter is next reset and/or TwinCAT has been restarted.

Programming the velocity

The unit of velocity is mm/sec and is therefore equivalent to the usual System Manager units.

4.3.11 Read Actual Axis Value

Command	@361	
Parameter 1	R<n>	R parameter to which the actual axis value is assigned
Parameter 2	K<m>	Constant for the axis coordinate to be read 0: X axis 1: Y axis 2: Z axis 3: Q1 axis

		4: Q2 axis . . . 7: Q5 axis

Sample 1:

```
N10 G0 X0 Y0 Z0 F24000
N30 G01 X1000
N40 @361 R1 K0 (read position of x axis)
N50 R0=X
N60 G01 X=R0+R1
N70 M30
```

A decoder stop is implicitly executed by @361 command. This ensures that, in this example, the position is read when block N30 has been processed.

One possible application is, for example, the combination with the residual path deletion.

Read actual axes value without decoder stop

from TwinCAT V2.9 Build 947

Command	#get PathAxesPos(R<a>; R<b>; R<c>)#	
Parameter 1	R<a>	R parameter to which the actual axes value of the X axis is assigned
Parameter 2	R<b>	R parameter to which the actual axes value of the Y axis is assigned
Parameter 3	R<c>	R parameter to which the actual axes value of the Z axis is assigned

The command #get PathAxesPos( )# reads the current actual positions of the path axes (X, Y & Z). It behaves similarly to @361, with the difference that this command does not trigger an implicit decoder stop. This means that the programmer must himself ensure that at the time when the command is being processed in the interpreter the axes have not yet moved, or else a decoder stop (@714) must be programmed in the block before this command.

#get PathAxesPos( )# is an alternative to @361, but it is linked to certain specific conditions.

Sample 2:

```
@714(optional)
N27 #get PathAxesPos( R0; R1; R20 )#
```

If a path axis is not assigned (e.g. no axis is assigned to Z) the value 0 is passed to the associated R parameter.

4.3.12 Skip virtual movements

from TwinCAT V2.9 Build 1022

from 2.10 Build 1341 for auxiliary axes

Command	#skip VirtualMovements(<parameter>)#
Parameter	0 (default): virtual movements are 'extended' 1: virtual movements are skipped

Movements of unavailable but programmed main axes (X, Y & Z) can be skipped with the command 'skip VirtualMovements'.

Sample:

The interpolation group (CfgBuildGroup) contains only assignments for the X and Y axis. The Z axis is **not** assigned, but programmed in the parts program.

```
(Startposition X0 Y0 Z0)
N10 #skip VirtualMovements(1)#
N20 G01 X100 Y200 F6000
N30 G01 Z1000 (virtual movement, because z is not assigned)
N40 G01 X500
```

Segment N30 is skipped during execution of this program.

4.3.13 Messages from NC program

from TwinCAT V2.9 Build 1031

Command	#MSG (<message level>; <mask>; "<text>")#
<message level>	• ITP The message is issued from the interpreter. This means that the message generally appears well before the execution in the NC kernel. • NCK The message is issued from the NC kernel when the NC block is executed. This means it appears at the same time as the block execution (SEC)
<mask>	STRING
<text>	the text to be displayed

```
N10 G0 X0 Y0
N20 G01 X100 Y0 F6000

N30 #MSG( NCK; STRING; "this is a text")#

N40 G01 X200 Y-100
```

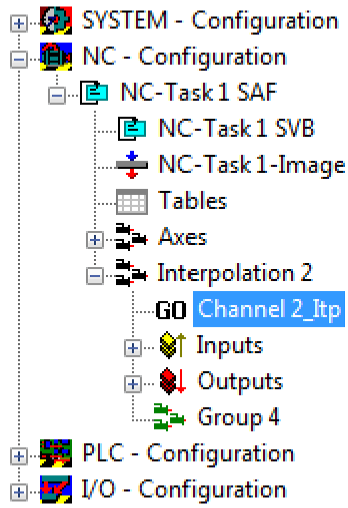
The text can **not** be used to transfer further parameters (e.g. R parameters).

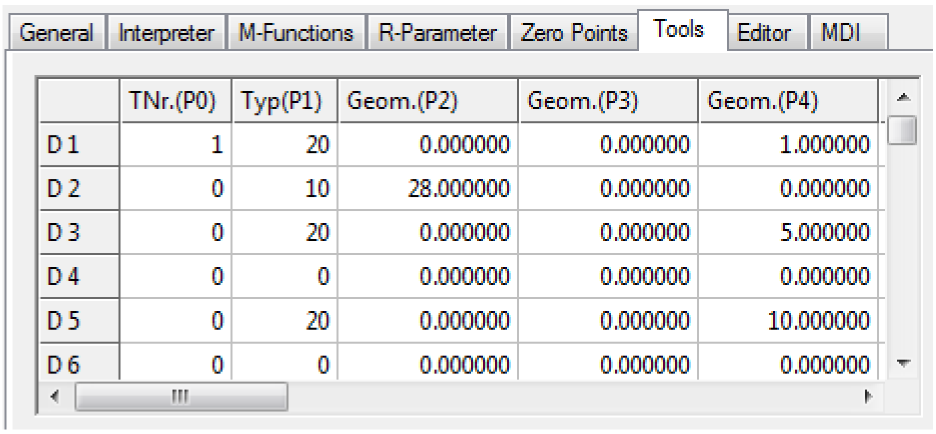
Internally the message is handled like a note.

4.4 Tool Compensation

4.4.1 Tool Data

The NC has 255 memory locations (D1..D255) available for each channel for tool data. The parameters for the tool data can be described directly in the System Manager. The data is saved as an ASCII file (<channel ID>.wz) which is kept in the TwinCAT\CNC directory. These files are automatically loaded when TwinCAT is started.





Currently two tool types are supported:

- Drills
- Shaft Cutters

The relevant columns (parameters) for this type of tool are described below.

Drills

Parameter	Meaning
0	**Tool number** When this D-word is called, a tool number that is specified here can be given at the same time.
1	**Tool type** The drill is **type 10**
2	**Geometry: Length** Describes the length of the drill
5	**Wear: Length** Describes the wear of the drill. The wear has to be given as a negative value, since it is added to the length.
8	Cartesian tool displacement [▶ 86] in X direction
9	Cartesian tool displacement in Y direction
10	Cartesian tool displacement in Z direction

Shaft cutters

Parameter	Meaning
0	**Tool number** When this D-word is called, a tool number that is specified here can be given at the same time.
1	**Tool type** The shaft cutter is **type 20**
2	**Geometry: Length** Length of the shaft cutter
4	**Geometry: Radius**
5	**Wear: Length**
7	**Wear: Radius**
8	Cartesian tool displacement [▶ 86] in X direction
9	Cartesian tool displacement in Y direction
10	Cartesian tool displacement in Z direction

Writing of tool data

Editing tool data with the System Manager

As already mentioned, the tool data can be written directly from the System Manager. To do this, edit the window shown above.

Parameterization of tool data via the PLC

Furthermore, tool data can be read and written from the PLC via the function block ItpWriteToolDescEx [▶ 159].

Writing tool data from the parts program

(from TwinCAT V2.9 Build 932)

In some applications, it is more convenient to write the tool data directly from the part program.

The tool set to be overwritten must not be active during the write process. This means, for example, if tool radius compensation with parameter set D10 is active, this cannot be overwritten, as long as D10 is still selected.

Command	#set ToolParam(<line>; <column>;<value>)#
Parameter <line>	Writes to the tool parameter line (1..255) This corresponds to the D number
Parameter <column>	Column to be written (0..15)
Parameter <value>	Parameter value to be transmitted

Sample:

```
N10 G0 X0 Y0 Z0
N20 G01 X100 F60000
N30 R1=10 R2=4 R3=20.3
N40 #set ToolParam(10; 0; 5)# #set ToolParam(10;1;20)#
N50 #set ToolParam(R1; R2; R3)#
N60 G41 X200 Y D10
...
```

No formulas may be transmitted as parameters. Writing of the tool data does not require a decoder stop.

Reading tool data from the parts program

(from TwinCAT V2.10 B1329)

This command can be used to assign tool data to an R parameter.

Command	#get ToolParam(<line>; <column>;<R-Param>)#
Parameter <line>	Writes to the tool parameter line (1..255), this corresponds to the D number
Parameter <column>	Column to be written (0..15)
Parameter <R-Param>	R parameter in which the date is entered

Sample:

```
N10 G0 X0 Y0 Z0
N20 G01 X100 F60000
N30 R1=10 R2=4
N40 #get ToolParam(10; 0; R5)# #getToolParam(10;1;R20)#
N50 #get ToolParam(R1; R2; R3)#
N60 G41 X200 Y D10
...
```

No formulas may be transmitted as parameters. Reading of the tool data does not require a decoder stop.

4.4.2 Selecting and Deselecting the Length Compensation

Length compensation can only be selected when G0 [▶ 39] or G1 [▶ 41] are in effect. The working plane [▶ 33] must be selected to which the length compensation is perpendicular.

The feed direction is defined with P (see Working Plane and Feed Direction [▶ 33]).

To effect the movement corresponding to the length compensation, the axis concerned must at least be mentioned.

Sample:

```
N10 G17 G01 X0 Y0 Z0 F6000
N20 D1 X10 Y10 Z
N30 ...
N90 M30
```

When the cutter radius compensation [▶ 88] is selected, the length compensation is also automatically selected. D0 must be programmed in order to deselect the length compensation again. It is again here necessary to at least mention the axis concerned in order to move to the new position.

4.4.3 Cartesian Tool Translation

Cartesian tool displacement refers to an offset between the reference point of the tool carrier and the reference point of the tool itself. In many cases, these reference points have the same location, so that a 0 can be entered for the tool displacement.

Parameters

The parameters for a translation are entered into the tool data [▶ 83] in the same way as the tool length etc. Parameters 8 to 10 are available for this purpose. Here

- P8 always describes the X-component
- P9 always describes the Y-component
- P10 always describes the Z-component

independently of the choice of level.

General | Interpreter | M-Functions | R-Parameter | Zero Points | Tools | Editor | MDI

	TNr.(P0)	Typ(P1)	Geom.(P2)	Geom.(P3)	Geom.(P4)
D 1	1	20	0.000000	0.000000	1.000000
D 2	0	10	28.000000	0.000000	0.000000
D 3	0	20	0.000000	0.000000	5.000000
D 4	0	0	0.000000	0.000000	0.000000
D 5	0	20	0.000000	0.000000	10.000000
D 6	0	0	0.000000	0.000000	0.000000

Selecting and deselecting Cartesian tool displacement

As in the case of length compensation, tool displacement is switched on with D<n> (n>0). In order to travel to the translated location, the axes must at least be named. This means that the displacement affects the positioning when the axis is called for the first time. It is also possible for a new final position to be entered for the axis.

The function is switched off with D0. Here again, it is necessary for the axes at least to be named, if the axes are to travel to their new co-ordinates.

Sample 1:

```
N10 G17 G01 X0 Y0 Z0 F6000
N20 D1 X10 Y10 Z (Z-Axis is repositioned)
N30 ...
N90 M30
```

Sample 2:

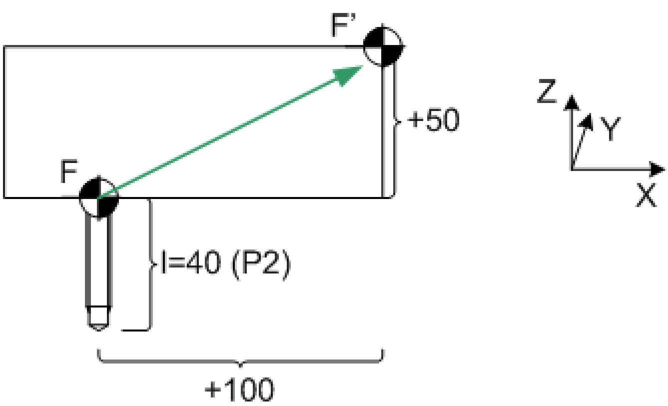
```
N10 G17 G01 X0 Y0 Z0 F6000
N20 D1 X10 Y10 (Z-Axis is not moved)
N30 ...
N90 M30
```

Using tool displacement and rotation

If the Cartesian tool displacement is used in combination with rotation [▶ 51], then the compensation will only be correct if the aggregate (the tool carrier) is also rotated through the same angle.

Application example

It often happens that a processing machine's tool carrier contains a number of tools. The appropriate tool is pneumatically activated according to the kind of machining required. Since, however, the tools are located at different positions, Cartesian tool displacement is required.



Tool parameters

Parameter	Value
0	0..65535
1	10
2	40
5	0
8	100.0
9	0.0
10	50

Behavior with incremental dimension notation

Default behavior, independent of TwinCAT version

If a new tool offset (and also length compensation) is selected in incremental dimensions (G91), then the compensation is applied once the axis is named.

Sample 3:

```
(Tooloffset D1: X10 Y20 Z30)
N10 G01 D1 X100 Y0 Z0 F6000
N20 G91 (incremental dimension)
N30 D2 (Tooloffset D2: X100
Y200 Z300)
N30 Z10
N40 ...
```

from TwinCAT version 2.10 Build 1308

The characteristic can be parameterized under G91 from this version onwards.

Command	Description
ToolOffsetIncOn	The tool displacements and length compensations are also applied under G91 once the axis is named
ToolOffsetIncOff	The tool displacement and length compensation are **not** applied under G91

Sample 4:

```
(Tooloffset D1: X10 Y20 Z30)
N05 ToolOffsetIncOff
N10 G01 D1 X100 Y0 Z0 F6000
N20 G91 (incremental dimension)
N30 D2 (Tooloffset D2: X100
Y200 Z300)
N30 Z10
N40 ...
```

In N10 the Tooloffset is applied to all 3 axes. I.e. the axes move in the machine coordinate system (MCS) to X110 Y10 Z30.

In N30 the new Tooloffset of the Z-axis is **not** applied. This results in MCS X110 Y10 **Z40.**

See also ZeroShiftIncOn/Off [▸ 46]

4.4.4 Cutter Radius Compensation

4.4.4.1 Miller/Cutter Radius Compensation Off

Miller/Cutter Radius Compensation Off

Command	G40
Cancellation	G41 [▸ 89] or G42 [▸ 89]

The G40 function switches the miller/cutter radius compensation off. The length radius compensation [▸ 86] will still remain active until it is switched off with D0.

4.4.4.2 Miller/cutter radius compensation left

Miller/cutter radius compensation left

Command	G41
Cancellation	G40 [▸ 88]

The function G41 switches on the miller/cutter radius compensation. The tool is located to the **left** of the workpiece in the direction of movement.

As has already been seen for the length compensation [▸ 86], the cutter radius compensation can only be activated when G0 [▸ 39] or G1 [▸ 41] is in effect. The axes of the plane must be driven when the cutter radius compensation is selected.

Sample:

```
N10 G17 G01 X0 Y0 Z0 F6000
N20 G41 X10 Y20 Z D1
N30 X30
N40 G40 X10 Y10 Z
N50 M30
```

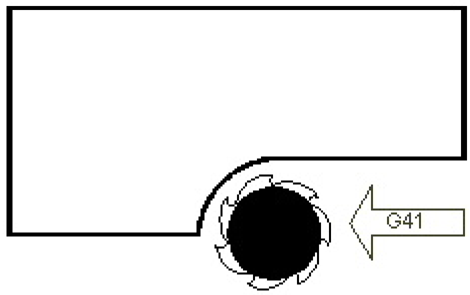


Figure 1: Radius compensation left

Cutter radius compensation does not apply to full circles

The cutter radius compensation does not support full circles. Full circles have to be split into semicircles, for example.

Please note:

- The cutter radius compensation should be deactivated before the end of the NC program, in order to close it properly.
- If a decoder stop [▸ 71] is programmed, cutter radius compensation has to be disabled first.
- Due to the radius compensation, the path velocity at the contour can change for circles, see'Path velocity for circles [▸ 94]'.
- See Orthogonal contour approach/departure [▸ 94]

4.4.4.3 Miller/cutter radius compensation right

Miller/cutter radius compensation right

Command	G42
Cancellation	G40 [▸ 88]

The function G42 switches on the miller/cutter radius compensation. The tool is located to the **right** of the workpiece in the direction of movement.

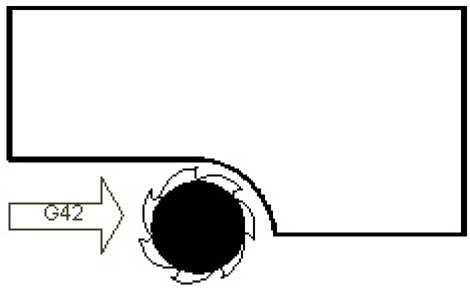


Figure 2: Radius compensation right

Cutter radius compensation does not apply to full circles

- The cutter radius compensation does not support full circles. Full circles have to be split into semicircles, for example.
- If a change is to be made from G41 to G42, then a G40 should be programmed between the two movements.

4.4.4.4 Departure and approach behavior of the miller/cutter radius compensation

Departure and approach behavior of the miller/cutter radius compensation

This chapter describes the approach and departure behavior when the miller/cutter radius compensation is switched on or off. This behavior depends on the start position and cannot be influenced in any other way.

After the radius compensation is switched on, it must be applied. This means the milling cutter is at one point P1 (without radius compensation) and travels to P2', with the cutter radius being compensated at point P2'.

Point P2' depends on the start position P1 within the plain. A distinction is made between 3 basic cases. These cases are exemplified below during application of the radius compensation with a programmed G42 (right compensation).

Similar rules apply for the deactivation of the compensation, except that the tangent t is determined at the end of the path segment, with similar conditions being derived.

Case 1

If the starting point P1 is to the right of the path tangent t, P2' is orthogonal to the tangent (see Figure 3). This approach behavior applies to the hatched green region in Figure 4.

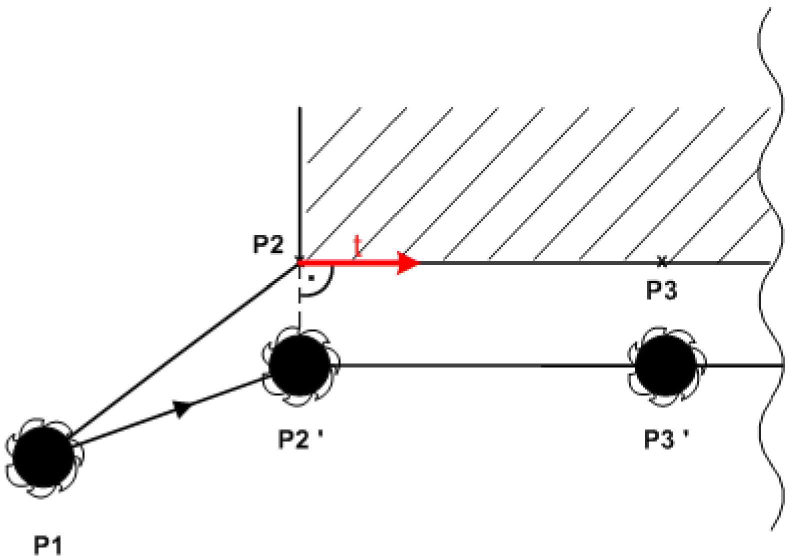


Figure 3: Example of start position to the right of the path tangent

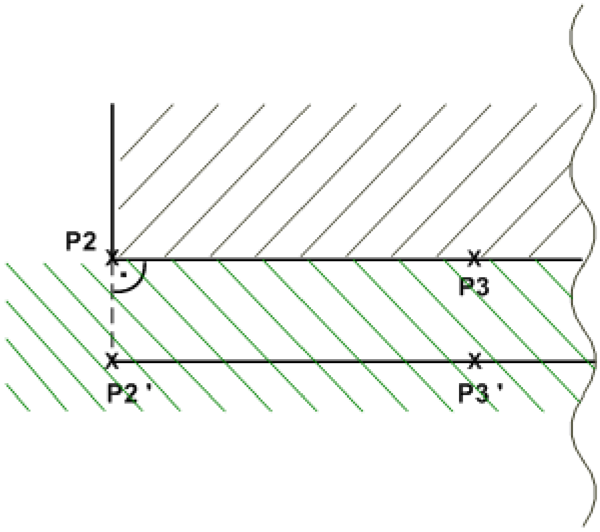


Figure 4: Start position to the right of the path tangent (general)

Case 2

For the case of the start position P1 being on the right of the normal n and to the left of the path tangent t, P2' is offset (see Figure 5). P2' results from the intersection of the parallel of P1P2 and the offset distance P2P3. Both straight lines are offset by radius R.

This behavior applies to the hatched green region in Figure 6.

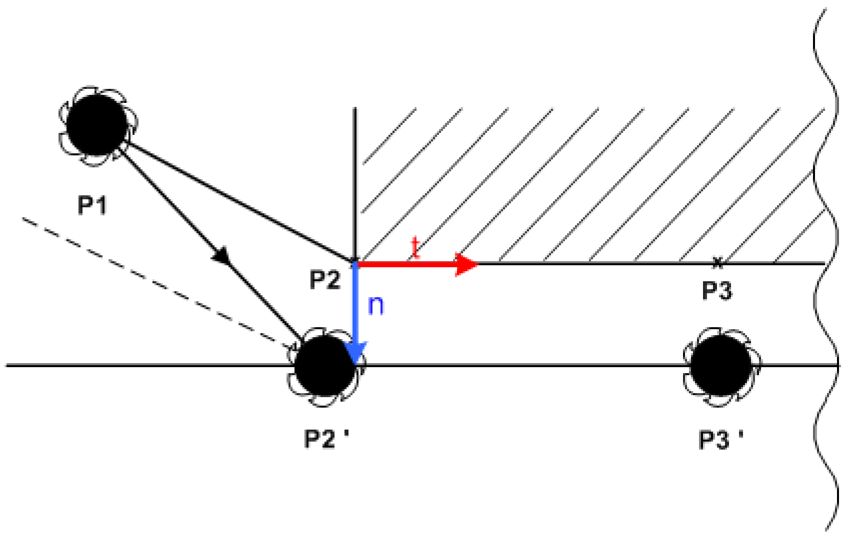


Figure 5: Example of start position to the right of normal n

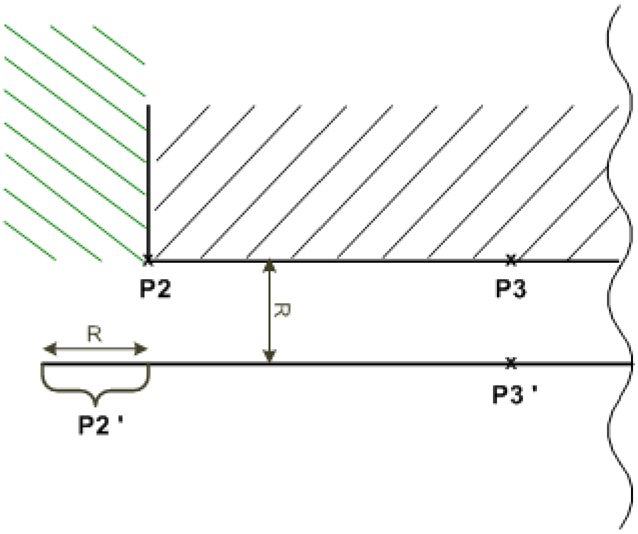


Figure 6: Start position to the right of normal n (general)

Case 3:

If the start position P1 is to the left of the normal n and also to the left of the path tangent t, an additional circle segment is inserted during approaching of P2'. In order to avoid free cutting at P2, P2' is not orthogonal to the start tangent of the section P2P3.

The additional circle segment is inserted for all start positions within the hatched green region in Figure 8.

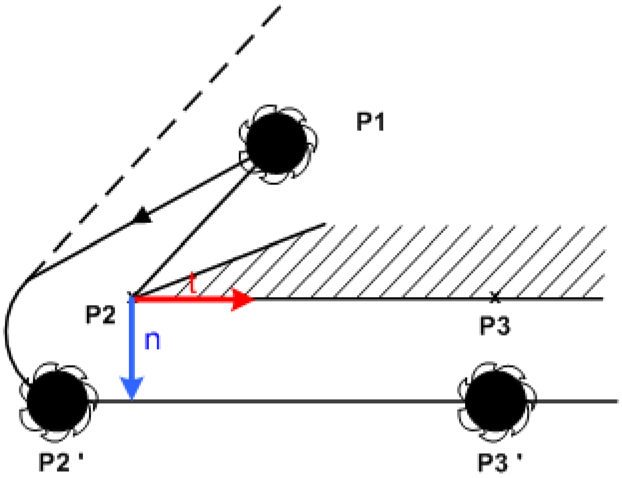


Figure 7: Example of start position to the left of normal n

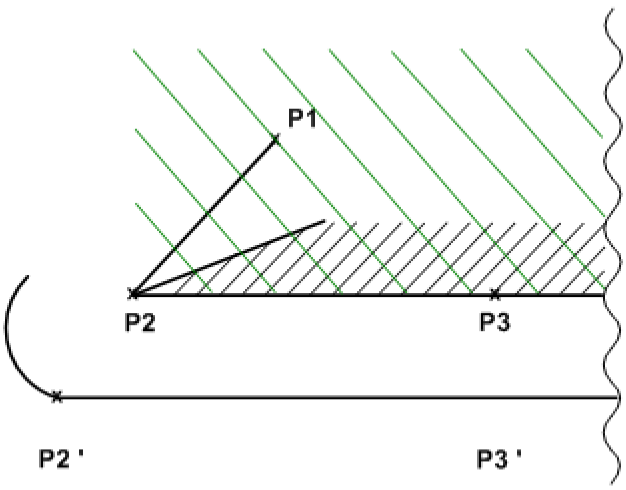


Figure 8: Start position to the left of normal n (general)

A circle segment follows after the offset

The radius compensation is invariably applied via a straight line. (This must be set in the part program, since otherwise a runtime error will be generated). The contour can then start with a circle. The rules for the approach and departure are the same as before, i.e. here too the path tangent of the contour for P2 is determined, and the 3 cases described are distinguished.

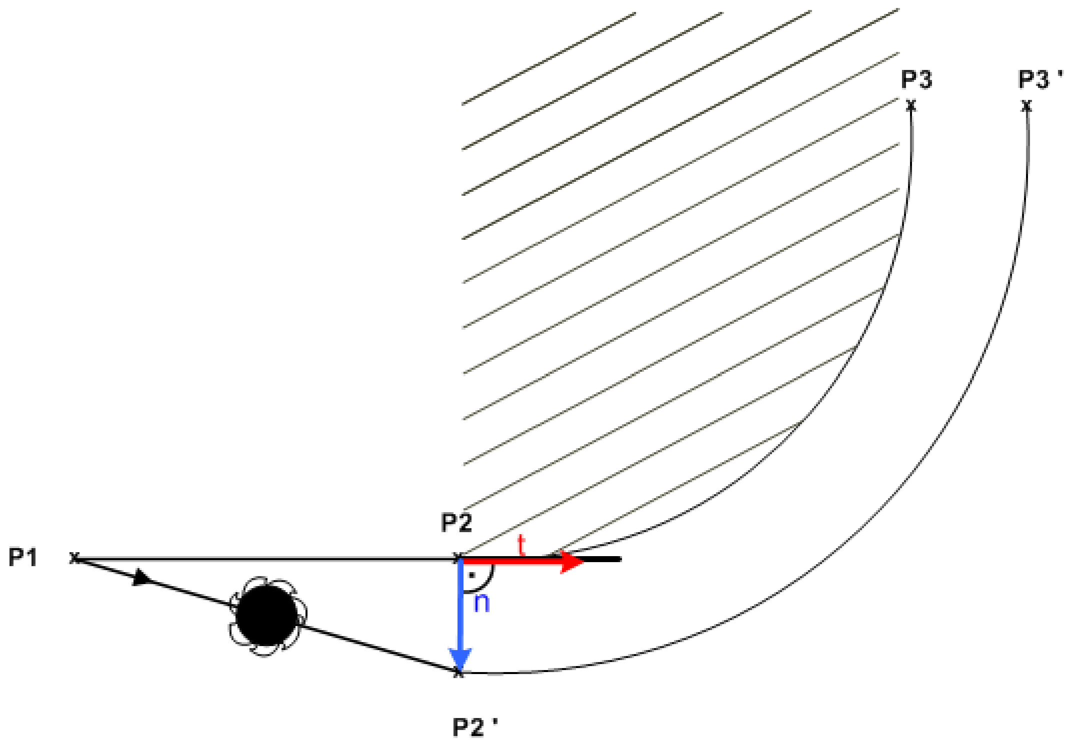


Figure 9: Example of circle segment

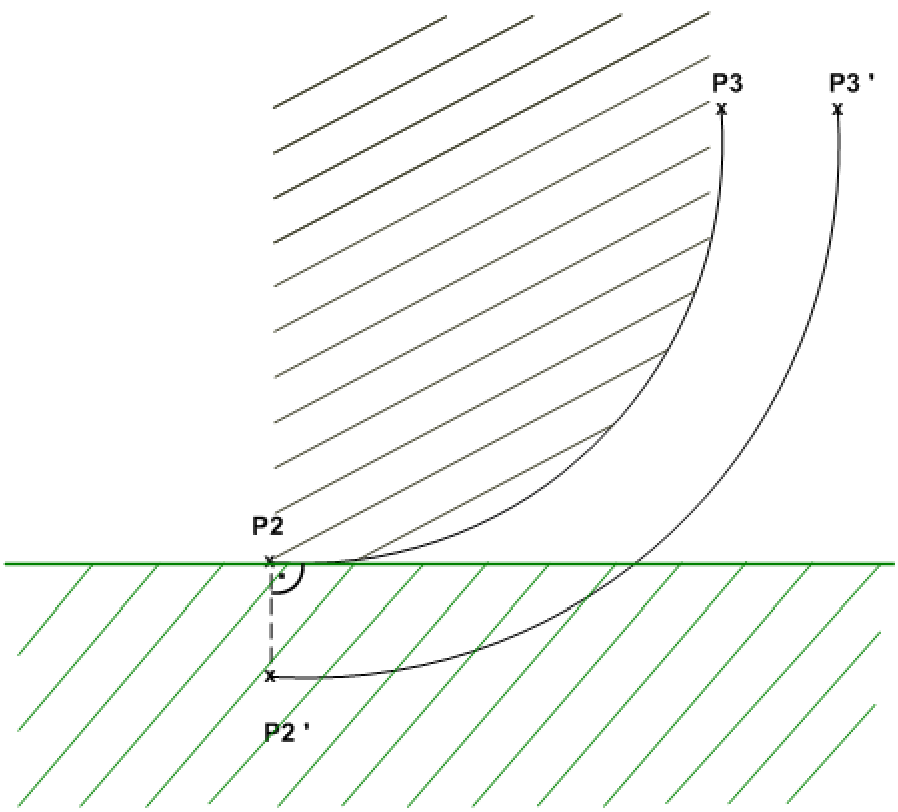


Figure 10: Circle segment (general)

If P2' is always to be approached orthogonal to the path tangent of P2, independent of the starting point, this can be realized with an additional command (see Orthogonal contour approach/departure [▶ 94]).

4.4.5 Orthogonal Contour Approach/Departure

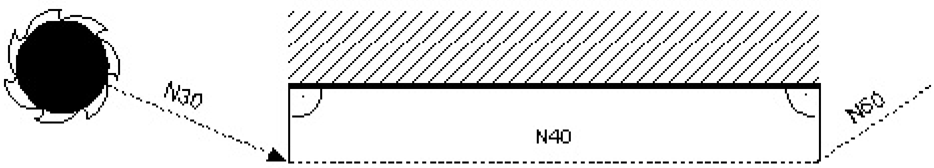
from TwinCAT V2.7 Build 427

Command	NORM
Cancellation	End of block
Programmable with	G40 [▶ 88] G41 [▶ 88] G42 [▶ 88]

The 'NORM' command has the effect that the contour is approached orthogonally when cutter radius compensation is switched on. The actual position of the cutter is irrelevant. When de-selecting, the last segment with active compensation is also left orthogonally.

Sample:

```
N10 G17
N20 G01 X0 Y0 Z0 F6000
N30 G42 NORM X100 Y0 D5
N40 X200
N50 G40 NORM X220 Y0
N60 M30
```



The Norm command has hitherto only been implemented for straight line/straight line transitions.

4.4.6 Path Velocity in Arcs

When the cutter radius compensation [▶ 88] is active, the programmed radius changes for arcs. This in turn alters the velocity. The following commands are used to specify whether the feed value refers to the contour or the tool center point.

Constant Feed at the Contour

Command	CFC
Cancellation	CFIN or CFTCP

With CFC (constant feed contour) the feedrate at the contour is held constant.

Constant Feed at the Internal Radius

Command	CIN
Cancellation	CFC or CFTCP

With CFIN (constant feed internal radius) the feedrate at internal radii is reduced. This results in a constant velocity at the contour. The velocity at external radii is not increased.

Constant Feed of the Tool Centre Point

Command	CFTCP
Cancellation	CFC or CFIN

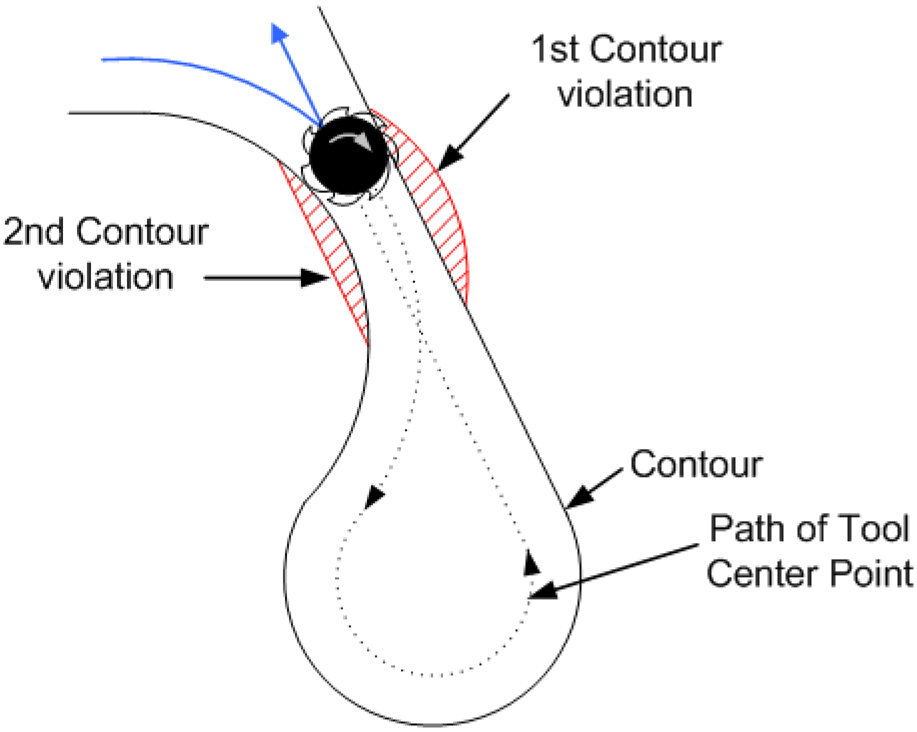
With CFTCP (constant feed tool center point) the feedrate of the tool's center point is kept constant. This means that at internal radii the velocity at the contour is increased, and that it is correspondingly reduced at external radii.

4.4.7 Bottle Neck Detection

from TwinCAT Version 2.8

If the cutter radius is not considered when a part program is created, the cutter may inadvertently process the opposite side of the workpiece. This leads to a contour collision with the workpiece, or, in other words, a bottleneck was programmed.

Command	CDON
Cancellation	CDOF



In this form, this behavior can only occur in combination with cutter radius compensation (G41/G42). In order to prevent such contour collisions, monitoring can be switched on from the part program via **CDON**. For it to be active, cutter radius compensation must also be selected.

The response of the NCI when a bottleneck is detected can be parameterized via the PLC. 3 cases are distinguished:

- Error and abort
  If a bottleneck is detected, TwinCAT generates a run-time error and aborts the program execution.
- Notification and modification of the contour
  If a bottleneck is detected, the contour is modified such that a contour collision is avoided (see Figure 1: blue line). However, this also means that segments may be left out, depending on the program. Furthermore, a note is entered in the application viewer to say that a bottleneck was detected.
- Notification and contour collision
  If a bottleneck is detected, the contour is not changed and no error is generated. Only a message is entered in the application viewer.

Significant computing resources are required for contour collision monitoring. It should therefore only be selected if it is actually required. Furthermore, the amount of look-ahead for the bottleneck detection should be specified. This requires the number of future segments to be determined that are monitored relative to the n-th segment, in order to check for bottlenecks. The selected number of segments should not be too large, since this would put unnecessary strain on the system. The value for the look-ahead is also parameterized from the PLC.

Function blocks for parameterizing the bottleneck detection:

- ItpSetBottleNeckModeEx [▸ 144]
- ItpGetBottleNeckModeEx [▸ 112]
- ItpSetBottleNeckLookAheadEx [▸ 142]
- ItpGetBottleNeckLookAheadEx [▸ 110]

Sample:

```
N10 G0 X0 Y0 Z0
N20 CDON
N30 G01 G41 D3 X100 F6000 (cutter radius 30mm)
...
N40 G01 X200
N50 G02 X220 Y-74.641 I0 J-40
N60 G01 X300 Y-104
N70 G01 X230 Y120
N80 G40 D0 Y200
N90 CDOF
...
M30
```

4.5 Command overview

4.5.1 General command overview

Command	Description	block-by-block / modal	Default
ANG [▸ 50]	Contour line programming (angle)	s	
AROT [▸ 51]	Rotation additive	m	
CalcInvRot [▸ 51]	Calculates the inverse rotation of a vector	s	
CalcRot [▸ 51]	Calculates the rotation of a vector	s	
CDOF [▸ 95]	Bottleneck detection off	m	Default
CDON [▸ 95]	Bottleneck detection on	m	
CFC [▸ 94]	Constant velocity at the contour	m	Default
CFIN [▸ 94]	Constant velocity in the interior circle	m	
CFTCP [▸ 94]	Constant velocity of tool center point	m	
CIP [▸ 41]	Circular interpolation	s	
CPCOF [▸ 41]	Centre point correction off	m	
CPCON [▸ 41]	Centre point correction on	m	Default
DelDTG [▸ 62]	Delete Distance to Go	s	
DYNOVR [▸ 77]	Dynamic Override	m	
FCONST [▸ 45]	Constant feed programming	m	Default

Command	Description	block-by-block / modal	Default
FLIN [▸ 45]	Linear feed programming	m	
G00 [▸ 39]	Rapid traverse	m	
G01 [▸ 41]	Straight line interpolation	m	Default
G02 [▸ 41]	Clockwise arc interpolation	m	
G03 [▸ 41]	Anticlockwise arc interpolation	m	
G04 [▸ 44]	Dwell time	s	
G09 [▸ 45]	Accurate stop	s	
G17 [▸ 33]	Plane selection XY	m	Default
G18 [▸ 33]	Plane selection ZX	m	
G19 [▸ 33]	Plane selection YZ	m	
G40 [▸ 88]	No miller/cutter radius compensation	m	Default
G41 [▸ 88]	Miller/cutter radius compensation left	m	
G42 [▸ 88]	Miller/cutter radius compensation right	m	
G53 [▸ 46]	Zero shift suppression	m	Default
G54 [▸ 46]	1. adjustable zero shift	m	
G55 [▸ 46]	2. adjustable zero shift	m	
G56 [▸ 46]	3. adjustable zero shift	m	
G57 [▸ 46]	4. adjustable zero shift	m	
G58 [▸ 46]	1. programmable zero offset shift	m	
G59 [▸ 46]	2. programmable zero offset shift	m	
G60 [▸ 45]	Accurate stop	m	
G70 [▸ 34]	Dimensions inch	m	
G71 [▸ 34]	Dimensions metric	m	Default
G74 [▸ 39]	Programmed traverse to reference point	s	
G90 [▸ 32]	Reference dimension notation	m	Default
G91 [▸ 32]	Incremental dimension notation	m	
G700 [▸ 34]	Dimensions in inches with calculation of the feed	m	
G710 [▸ 34]	Dimensions metric with calculation of the feed	m	
Mirror [▸ 54]	Mirroring coordinate system	m	
MOD [▸ 63]	Modulo movement	s	
MSG [▸ 83]	Messages from the NC program	s	
NORM [▸ 94]	Orthogonal approach of and departure from the contour	s	
P+ [▸ 33]	Feed direction positive	m	Default

Command	Description	block-by-block / modal	Default
P- [▸ 33]	Feed direction negative	m	
paramAutoAccurateStop [▸ 61]	Automatic Accurate Stop	m	
paramAxisDynamics [▸ 78]	Parameterization of the axis dynamics	m	
paramC1ReductionFactor [▸ 79]	C1 reduction factor	m	
paramC2ReductionFactor [▸ 79]	C2 reduction factor	m	
paramCircularSmoothing [▸ 61]	Circular smoothing	m	
paramDevAngle [▸ 79]	C0 reduction - deflection angle	m	
paramGroupVertex [▸ 61]	Circular smoothing (old)	m	
paramGroupDynamic [▸ 78]	Pathway dynamics (old)	m	
paramPathDynamics [▸ 78]	Pathway dynamics	m	
paramRadiusPrec [▸ 42]	Circular accuracy	m	
paramSplineSmoothing [▸ 59]	Smoothing with Bezier Splines	m	
paramVertexSmoothing [▸ 56]	Smoothing of Segment Transitions	m	
paramVeloJump [▸ 79]	C0 reduction - max. step change in velocity	m	
paramVeloMin [▸ 81]	Minimum velocity	m	
paramZeroShift [▸ 46]	Parameterization of the configurable zero shift	m	
PathAxesPos [▸ 82]	Reads the actual position	s	
ROT [▸ 51]	Absolute rotation	m	
RotExOff [▸ 51]	Extended rotation function off	m	Default
RotExOn [▸ 51]	Extended rotation function on	m	
RotVec [▸ 51]	Calculation routine for rotating a vector	s	
RParam [▸ 37]	Initialization of R-parameters	s	
RToDwordGetBit [▸ 37]	Converts an R-parameter to DWord and checks whether a defined bit is set	m	
SEG [▸ 50]	Contour line programming (segment length)	s	
skip VirtualMovements [▸ 82]	Skip virtual movements	m	
ToolOffsetIncOff [▸ 86]	Cartesian tool displacement and length compensation is not applied under G91	m	

Command	Description	block-by-block / modal	Default
ToolOffsetIncOn [▶ 86]	Cartesian tool displacement and length compensation is applied under G91	m	Default
ToolParam [▶ 83]	Writing and reading of tool parameters	m	
TPM [▶ 48]	Target position monitoring	s	
ZeroShiftIncOff [▶ 46]	Zero shift is not applied under G91	m	
ZeroShiftIncOn [▶ 46]	Zero shift is applied under G91	m	Default

Address	Description
Q<n> [▶ 64]	Axis label for auxiliary axis (1 <= n <= 5)

4.5.2 @-Command Overview

Several variations of these commands are often possible, since K for a constant, R for an R-parameter or P for an R-parameter used as a pointer can be used for parameters. For example, the notation K/R/Pn should be understood to mean "either a number or an R-parameter or a pointer".

The following @-commands are available:

Command	Versions	Function
@40 [▶ 37]	@40 Kn Rn Rm	Save register on the stack
@41 [▶ 37]	@41 Rn Rm	Save register on the stack
@42 [▶ 37]	@42 Kn .... Rm Rn	Restore register from stack
@43 [▶ 37]	@43 Rm Rn	Restore register from stack
@100 [▶ 73]	@100 K±n @100 Rm	Unconditional jump
@111 [▶ 73]	@111 Rn K/Rn Km ...	Case block
@121 [▶ 73]	@121 Rn K/Rn Kn	Jump if unequal
@122 [▶ 73]	@122 Rn K/Rn Kn	Jump if equal
@123 [▶ 73]	@123 Rn K/Rn Kn	Jump if less or equal
@124 [▶ 73]	@124 Rn K/Rn Kn	Jump if less
@125 [▶ 73]	@125 Rn K/Rn Kn	Jump if greater or equal
@126 [▶ 73]	@126 Rn K/Rn Kn	Jump if greater
@131 [▶ 74]	@131 Rn K/Rn Kn	Loop while equal
@132 [▶ 74]	@132 Rn K/Rn Kn	Loop while unequal
@133 [▶ 74]	@133 Rn K/Rn Kn	Loop while greater
@134 [▶ 74]	@134 Rn K/Rn Kn	Loop while greater or equal
@135 [▶ 74]	@135 Rn K/Rn Kn	Loop while less
@136 [▶ 74]	@136 Rn K/Rn Kn	Loop while less or equal
@141 [▶ 74]	@141 Rn K/Rn Kn	Repeat until equal
@142 [▶ 74]	@142 Rn K/Rn Kn	Repeat until unequal
@143 [▶ 74]	@143 Rn K/Rn Kn	Repeat until greater
@144 [▶ 74]	@144 Rn K/Rn Kn	Repeat until greater or equal
@145 [▶ 74]	@145 Rn K/Rn Kn	Repeat until less
@146 [▶ 74]	@146 Rn K/Rn Kn	Repeat until less or equal

Command	Versions	Function
@151 [▸ 74]	@151 Rn K/Rn Kn	FOR_TO loop
@161 [▸ 74]	@161 Rn K/Rn Kn	FOR_DOWNTO loop
@200	@200 Rn	Delete a variable
@202	@202 Rn Rm	Swap two variables
@302	@302 K/R/Pn K/R/Pn R/Pn	Read machine data bit
@361 [▸ 81]	@361 Rn Km	Read machine-related actual axis value
@372	@372 Rn	Set number of the NC channel in variable
@402 [▸ 41]	@402 K/R/Pn K/R/Pn K/R/Pn	Write machine data bit
@610	@610 Rn Rn	Find absolute value of a variable
@613	@613 Rn Rn	Find square root of a variable
@614	@614 Rn Rm Rm	Find square root of the sum of the squares of two variables x = sqrt(a^2 + b^2)
@620 [▸ 74]	@620 Rn	Increment variable
@621	@621 Rn	Decrement variable
@622	@622 Rn	Find integer part of a variable
@630 [▸ 37]	@630 Rn Rm	Find sine of a variable
@631 [▸ 37]	@631 Rn Rm	Find cosine of a variable
@632 [▸ 37]	@632 Rn Rm	Find tangent of a variable
@633 [▸ 37]	@633 Rn Rm	Find cotangent of a variable
@634 [▸ 37]	@634 Rn Rm	Find arcsine of a variable
@635 [▸ 37]	@635 Rn Rm	Find arccosine of a variable
@636 [▸ 37]	@636 Rn Rm	Find arctangent of a variable
@714 [▸ 71]	@714	Decoder stop
@716 [▸ 71]	@716	Decoder stop with rescan of the axis positions
@717 [▸ 71]	@717	Decoder stop with external trigger event

Machine data

Access to the following machine data is supported:

Byte	Bit	Action
5003 [▸ 41]	5	0: IJK words specify the distance between the center of the circle and the starting point. 1: IJK are absolute data giving the center of the circle.

5 PLC NCI Libraries

Prerequisites

Overview of PLC NCI libraries	Description
TwinCAT PLC Library NCI Interpreter [▶ 101]	Function blocks for operating the interpreter (G-Code (DIN 66025)). These include loading and starting.
TwinCAT PLC Library NC Configuration [▶ 207]	Blocks for configuration of the interpolation group (e.g. creation of the 3D group)
TwinCAT PLC Library NCI Utilities [▶ 212]	Auxiliary function blocks for interpolation (e.g. generate ASCII line for straight movement)
TwinCAT PLC Library TcPlcInterpolation [▶ 223]	Programming of multi-dimensional movements from the PLC (alternative to using G-Code (DIN 66025))

5.1 PLC Library: NCI Interpreter

The TwinCAT library **TcNci.lib** contains function blocks for operation of the NC interpreter from the PLC.

The following function blocks are contained in the TcNci.lib library.

Function block	Description
ItpConfirmHsk [▶ 104]	Acknowledges an M-function of type handshake
ItpDelDtgEx [▶ 105]	Triggers “Delete Distance to go” in the NC
ItpEnableDefaultGCode [▶ 106]	Executes a user-defined standard G-Code before the start of each NC program
ItpEStopEx [▶ 108]	Triggers the NCI EStop
ItpGetBlockNumber [▶ 109]	Provides the block number of the NC program for the cyclic interface
ItpGetBottleNeckLookAheadEx [▶ 110]	Provides the value of the look-ahead for bottleneck detection
ItpGetBottleNeckModeEx [▶ 112]	Provides the response mode for bottleneck detection
ItpGetChannelId [▶ 113]	Provides the channel ID
ItpGetChannelType [▶ 114]	Provides the channel type of the cyclic interface
ItpGetCyclicLRealOffsets [▶ 115]	Provides the index offset of the LREAL variables used in the cyclic channel interface
ItpGetCyclicUdintOffsets [▶ 117]	Provides the index offset of the UDINT variables used in the cyclic channel interface
ItpGetError [▶ 118]	Provides the error number
ItpGetGroupAxisIds [▶ 119]	Provides the axis IDs that were configured for the group
ItpGetGroupId [▶ 120]	Provides the group ID
ItpGetHParam [▶ 121]	Provides the current H-parameter from the NC
ItpGetHskMFunc [▶ 122]	Provides the current M-function number of type handshake
ItpGetItfVersion [▶ 122]	Provides the current version of the cyclic interface
ItpGetOverridePercent [▶ 123]	Provides the channel override in percent
ItpGetSParam [▶ 124]	Provides the current S-parameter from the NC
ItpGetStateInterpreter [▶ 125]	Provides the current interpreter state
ItpGetTParam [▶ 126]	Provides the current T-parameter from the NC
ItpGetVersion [▶ 127]	Provides the version number of this library
ItpGoAheadEx [▶ 127]	Triggers the GoAhead function (decoder stop with external trigger event)

Function block	Description
ItpHasError [▸ 128]	Determines whether an error is present
ItpIsFastMFunc [▸ 129]	Determines whether the M-function number provided is present in the form of a fast M-function
ItpIsEStopEx [▸ 130]	Determines whether an EStop is executed or pending
ItpIsHskMFunc [▸ 131]	Determines whether an M-function of type handshake is present
ItpLoadProgEx [▸ 132]	Loads an NC program using program names
ItpReadCyclicLRealParam1 [▸ 133]	Reads the first LReal parameter from the cyclic channel interface
ItpReadCyclicUdintParam1 [▸ 134]	Reads the first UDINT parameter from the cyclic channel interface
ItpReadRParamsEx [▸ 135]	Reads arithmetic parameters
ItpReadToolDescEx [▸ 136]	Reads the tool description from the NC
ItpReadZeroShiftEx [▸ 138]	Reads the zero offset shift from the NC
ItpResetEx2 [▸ 140]	Carries out a reset of the interpreter or of the NC channel
ItpResetFastMFuncEx [▸ 141]	Resets a fast signal bit
ItpSetBottleNeckLookAheadEx [▸ 142]	Sets the value of the look-ahead for bottleneck detection
ItpSetBottleNeckModeEx [▸ 144]	Sets the response mode when bottleneck detection is switched on
ItpSetCyclicLRealOffsets [▸ 145]	Sets the index offsets of the LREAL variables used in the cyclic channel interface
ItpSetCyclicUdintOffsets [▸ 147]	Sets the index offsets of the UDINT variables used in the cyclic channel interface
ItpSetOverridePercent [▸ 149]	Sets the channel override in percent
ItpSetSubroutinePathEx [▸ 150]	Optionally sets the search path for subroutines
ItpSetToolDescNullEx [▸ 151]	Sets all tool parameters (including number and type) to zero
ItpSetZeroShiftNullEx [▸ 152]	Set all zero offset shifts to null
ItpSingleBlock [▸ 153]	Enables or disables the single block mode in the NCI.
ItpStartStopEx [▸ 155]	Starts or stops the interpreter (NC channel)
ItpStepOnAfterEStopEx [▸ 155]	Enables further processing of the parts program after an NCI EStop
ItpWriteRParamsEx [▸ 157]	Writes arithmetic parameters
ItpWriteToolDescEx [▸ 159]	Writes the tool description into the NC
ItpWriteZeroShiftEx [▸ 161]	Writes the zero offset shift into the NC
Block search (for description of the functionality see Blocksearch [▸ 162])	
ItpBlocksearch [▸ 163]	Sets the interpreter to a user-defined location, so that NC program execution continues from this point.
ItpGetBlocksearchData [▸ 166]	Reads the current state after an NC program interruption.
ItpStepOnAfterBlocksearch [▸ 166]	Starts the motion after a block search.
Retrace	
ItpEnableFeederBackup [▸ 167]	Enables the backup list for retracing
ItpIsFeederBackupEnabled [▸ 169]	Reads whether the backup list for retracing is active
ItpIsFirstSegmentReached [▸ 171]	Reads whether the start position is reached during retracing

Function block	Description
ItpIsFeedFromBackupList [▸ 170]	Reads whether feeder entries were sent from the backup list
ItpIsMovingBackwards [▸ 171]	Reads whether backward movement occurs on the current path
ItpRetraceMoveBackward [▸ 172]	Performs a backward movement on the path
ItpRetraceMoveForward [▸ 173]	Performs a forward movement on the path. This is called to cancel retracing.

Requirements

The library to be integrated depends on the TwinCAT version. From TwinCAT version 2.8 **TcNci.lib** should be used. For older versions **TcNciItp.lib** should be used. The interface for the function blocks etc. of both libraries is the same.

Development environment	Target platform	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
from TwinCAT v2.8.0	PC (i386)	TcNci.lib

Function blocks for compatibility with existing programs:

Function blocks for compatibility

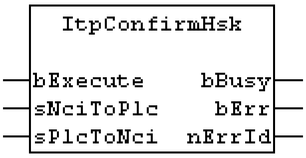
The purpose of the function blocks listed is to ensure compatibility with existing projects. It is **not** advisable to use these function blocks for new projects. Instead, the equivalent function blocks shown in the table above should be used.

Function block	Description
ItpDelDtg [▸ 175]	Triggers “Delete Distance to go” in the NC
ItpEStop [▸ 176]	Triggers the NCI EStop
ItpGetBottleNeckLookAhead [▸ 178]	Provides the value of the look-ahead for bottleneck detection
ItpGetBottleNeckMode [▸ 179]	Provides the response mode for bottleneck detection
ItpGoAhead [▸ 181]	Triggers the GoAhead function
ItpIsEStop [▸ 182]	Determines whether an EStop is executed or pending
ItpLoadProg [▸ 184]	Loads an NC program using program names
ItpReadRParams [▸ 185]	Reads arithmetic parameters
ItpReadToolDesc [▸ 186]	Reads the tool description from the NC
ItpReadZeroShift [▸ 188]	Reads the zero offset shift from the NC
ItpReset [▸ 190]	Carries out a reset of the interpreter or of the NC channel
ItpResetEx [▸ 190]	Carries out a reset of the interpreter or of the NC channel.
ItpResetFastMFunc [▸ 192]	Resets a fast signal bit
ItpSetBottleNeckLookAhead [▸ 193]	Sets the value of the look-ahead for bottleneck detection
ItpSetBottleNeckMode [▸ 195]	Sets the response mode when bottleneck detection is switched on
ItpSetSubroutinePath [▸ 196]	Optionally sets the search path for subroutines
ItpSetToolDescNull [▸ 198]	Sets all tool parameters (including number and type) to zero
ItpSetZeroShiftNull [▸ 199]	Sets all origins to zero
ItpStartStop [▸ 200]	Starts or stops the interpreter (NC channel)

Function block	Description
ItpStepOnAfterEStop [▸ 200]	Enables further processing of the parts program after an NCI EStop
ItpWriteRParams [▸ 202]	Writes arithmetic parameters
ItpWriteToolDesc [▸ 204]	Writes the tool description into the NC
ItpWriteZeroShift [▸ 205]	Writes the zero offset shift into the NC

5.1.1 ItpConfirmHsk

from library version 4.0



The **ItpConfirmHsk** function block confirms the currently present M-function.

If the channel override is set to 0 or an E-stop is active, no M-functions are acknowledged during this time. The busy signal of ItpConfirmHsk therefore remains active and must continue to be called.

Interface

```
VAR_INPUT
bExecute   : BOOL;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc : NciChannelToPlc;
sPlcToNci : NciChannelFromPlc;
END_VAR
```

NciChannelToPlc [▸ 251]
NciChannelFromPlc [▸ 253]

```
VAR_OUTPUT
bBusy     : BOOL;
bErr      : BOOL;
nErrId    : UDINT;
END_VAR
```

Parameter

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.
sPlcToNci	NciChannelFromPlc	The structure of the cyclic channel interface from the PLC to the NCI.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE for as long as the request bit of the M-function is set to FALSE in the channel interface.

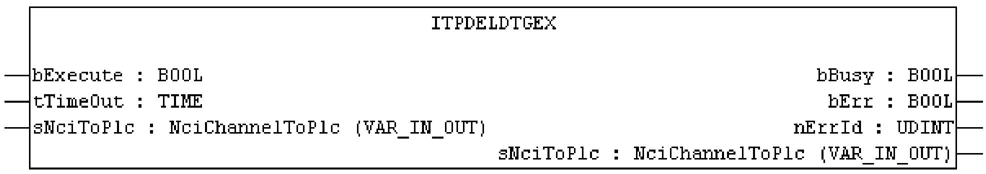
Output	Data type	Description
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.2 ItpDelDtgEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



Interface

```
VAR_INPUT
bExecute   : BOOL;
tTimeOut   : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy      : BOOL;
bErr       : BOOL;
nErrId     : UDINT;
END_VAR
```

Description

The function block ItpDelDtgEx triggers residual distance deletion. There is a more detailed description in the Interpreter [▶ 62] documentation.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

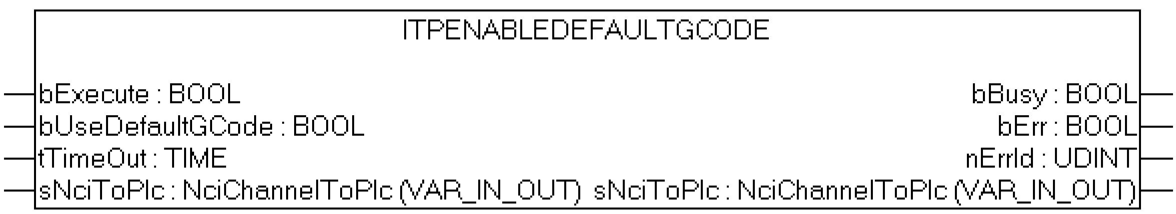
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.3 ItpEnableDefaultGCode

from library version 4.0



The function block ItpEnableDefaultGCode enables execution of a user-defined G-Code before the start of each NC program from the PLC. The default program is executed before the loaded program when the actual NC program starts.

This function block enables rotation of the coordinate system for all NC programs to be executed, for example.

The standard G-Code must be stored as „DefaultGCode<Channel-Number>.def“ in the TwinCAT\CNC directory.

Interface

```
VAR_IN
bExecute            : BOOL;
bUseDefaultGCode    : BOOL;
tTimeOut            : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc           : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUT
bBusy               : BOOL;
bErr                : BOOL;
nErrId              : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input
bUseDefaultGCode	BOOL	If this variable is TRUE, the default G-Code is activated through a rising edge at bExecute. If the variable is FALSE, the default G-Code is deactivated.
tTimeOut	TIME	ADS Timeout-Delay

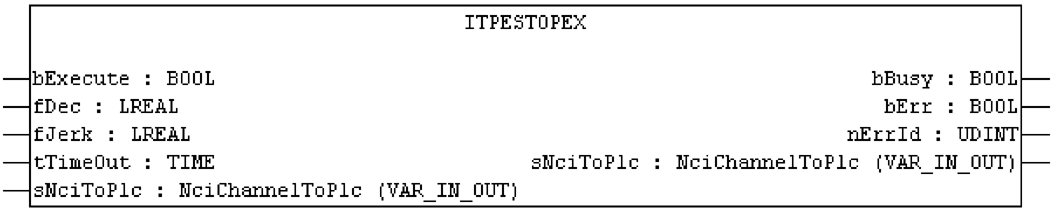
Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at

Output	Data type	Description
		the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

5.1.4 ItpEStopEx

from library version 6.1.21and TwinCAT version 2.10 Build 1304



Interface

```
VAR_INPUT
bExecute        : BOOL;
fDec            : LREAL;
fJerk           : LREAL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The function block ItpEStopEx triggers the NCI EStop and enables a controlled stop on the path. The limit values for the deceleration and the jerk are transferred as parameters. If these should be smaller than the currently active dynamic parameters, the transferring parameters are rejected.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input
fDec	LREAL	Max. deceleration during stopping. If fDec is smaller than the currently active deceleration, then fDec is not applied. This ensures that the deceleration occurs with the standard ramp as a minimum.
fJerk	LREAL	Max. jerk during stopping If fJerk is smaller than the currently active jerk, fJerk is not applied.

Input	Data type	Description
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

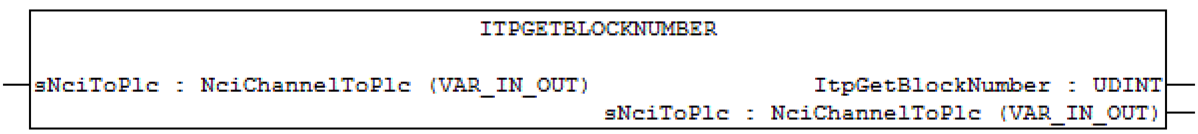
See also ItpStepOnAfterEStopEx [▶ 155].

Prerequisites

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.5 ItpGetBlockNumber

from library version 6.1.32



ItpGetBlockNumber is a function that returns the block number of the NC program for the cyclic interface.

Interface

```
Function ItpGetBlockNumber

VAR_IN_OUT
sNciToPlc    : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
UDINT	Block number of the active geometry segment

Sample

```
VAR
nBlockNumber   : UDINT;
sNciToPlc      : NciChannelToPlc;
END_VAR
```

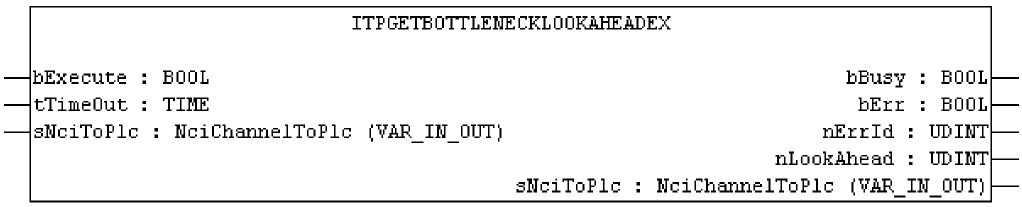
```
nBlockNumber := ItpGetBlockNumber(sNciToPlc);
```

Prerequisites

Development environment	Target system	PLC libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.6 ItpGetBottleNeckLookAheadEx

from library version 6.1.21and TwinCAT version 2.10 Build 1304



Interface

```
VAR_INPUT
bExecute        : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
```

```
nErrId         : UDINT;
nLookAhead     : UDINT;
END_VAR
```

Description

The function block ItpGetBottleNeckLookAheadEx determines the maximum size of the look-ahead for the bottleneck detection (contour collision monitoring).

A more detailed description can be found in the Interpreter [▶ 95] documentation.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

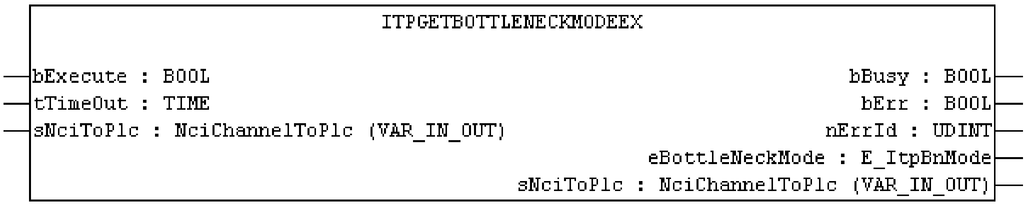
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. If the function block has a timeout error, 'Error' is TRUE and 'nErrId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).
nLookAhead	UDINT	Value of the look-ahead for bottleneck detection

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.7 ItpGetBottleNeckModeEx

from library version 6.1.21and TwinCAT version 2.10 Build 1304



Interface

```
VAR_INPUT
bExecute        : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
eBottleNeckMode : E_ItpBnMode
END_VAR
```

```
TYPE E_ItpBnMode:
(
ItpBnm_Abort := 0,
ItpBnm_Adjust := 1,
ItpBnm_Leave := 2
);
END_TYPE
```

Description

The function block ItpGetBottleNeckModeEx reads the behavior in the event of a contour collision (bottleneck).

A more detailed description can be found in the Interpreter [▸ 95] documentation.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the

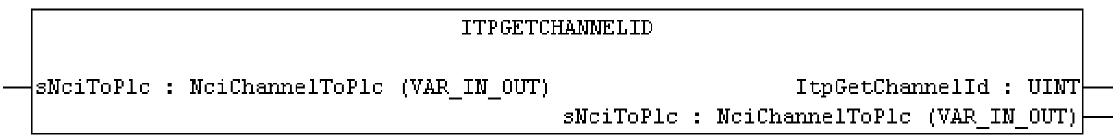
Output	Data type	Description
		duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. If the function block has a timeout error, 'Error' is TRUE and 'nErrId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).
eBottleNeckMode	E_ItpBnMode	Enum for the behavior in the event of a contour collision

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.8 ItpGetChannelId

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



ItpGetChannelId is a function that determines the channel ID from the cyclic interface

Interface

```
FUNCTION ItpGetChannelId

VAR_IN_OUT
sNciToPlc    : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
UDINT	Channel ID

Sample

```
VAR
nChnId     : UINT;
sNciToPlc : NciChannelToPlc;
END_VAR
```

```
nChnId := ItpGetChannelId( sNciToPlc );
```

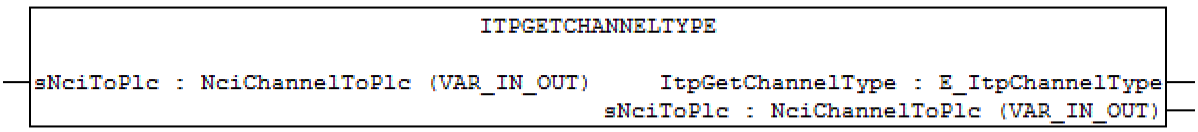
see also: ItpGetGroupId [▸ 120]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.9 ItpGetChannelType

from library version 6.1.32



ItpGetChannelType is a function that returns the channel type of the cyclic interface.

Interface

```
FUNCTION ItpGetChannelType

VAR_IN_OUT
sNciToPlc     : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
E_ItpChannelType	Channel type

```
TYPE E_ItpChannelType :
(
ItpChannelTypeNone,
ItpChannelTypeInterpreter,
ItpChannelTypeKinematic,

ItpChannelType_InvalidItfVer := 16#4B14 (*ErrTcNciItp_ItfVersion the cyclic channel interface does n
ot match to the requested function/fb *)
);
END_TYPE
```

Sample

```
VAR
nChannelType     : E_ItpChannelType;
sNciToPlc        : NciChannelToPlc;
END_VAR
```

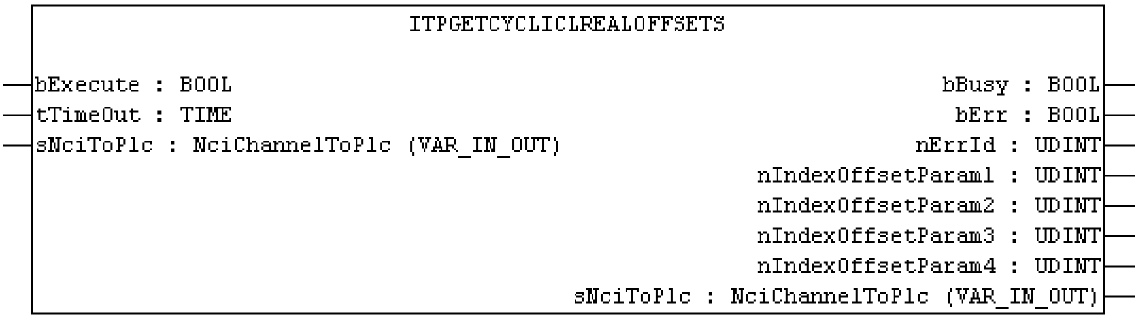
```
nChannelType := ItpGetChannelType( sNciToPlc );
```

Prerequisites

Development environment	Target system	PLC libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.10 ItpGetCyclicLrealOffsets

from library version 6.1.25 and TwinCAT version 2.10 Build 1320



The function block ItpGetCyclicLRealOffsets is used to read the current configuration of the cyclic channel interface for LREAL variables.

Interface

```
VAR_INPUT
bExecute             : BOOL;
tTimeOut             : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc            : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy                : BOOL;
bErr                 : BOOL;
nErrId               : UDINT;
nIndexOffsetParam1 : UDINT;
nIndexOffsetParam2 : UDINT;
nIndexOffsetParam3 : UDINT;
nIndexOffsetParam4 : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).
nIndexOffsetParam1	UDINT	Group state (index offset) for parameter 1
nIndexOffsetParam2	UDINT	Group state (index offset) for parameter 2
nIndexOffsetParam3	UDINT	Group state (index offset) for parameter 3
nIndexOffsetParam4	UDINT	Group state (index offset) for parameter 4

see also:

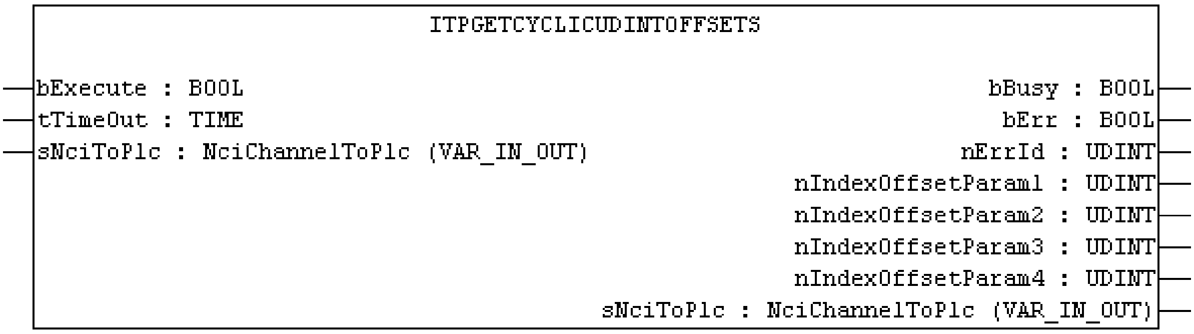
- ItpReadCyclicLRealParam1 [▸ 133]
- ItpSetCyclicLRealOffsets [▸ 145]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.11 ItpGetCyclicUDintOffsets

from library version 6.1.25 and TwinCAT version 2.10 Build 1320



The function block ItpGetCyclicUDintOffsets is used to read the current configuration of the cyclic channel interface for UDINT variables.

Interface

```
VAR_INPUT
bExecute                 : BOOL;
tTimeOut                 : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc           : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy               : BOOL;
bErr                : BOOL;
nErrId              : UDINT;
nIndexOffsetParam1 : UDINT;
nIndexOffsetParam2 : UDINT;
nIndexOffsetParam3 : UDINT;
nIndexOffsetParam4 : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the

Output	Data type	Description
		execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).
nIndexOffsetParam1	UDINT	Group state (index offset) for parameter 1
nIndexOffsetParam2	UDINT	Group state (index offset) for parameter 2
nIndexOffsetParam3	UDINT	Group state (index offset) for parameter 3
nIndexOffsetParam4	UDINT	Group state (index offset) for parameter 4

see also:

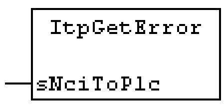
- ItpReadCyclicUDintParam1 [▶ 134]
- ItpSetCyclicUdintOffsets [▶ 147]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.12 ItpGetError

from library version 4.0



ItpGetError is a function that returns the error number. A description of the NC error codes can be found here.

Interface

```
FUNCTION ItpGetError

VAR_IN_OUT
sNciToPlc: NciChannelToPlc;
END_VAR
```

Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC.

Return value

Data type	Description
UINT	Error number

ItpGetError evaluates the variable 'nItpErrCode' from the cyclic interface.

Sample

```
VAR
bItpError: BOOL;
nErrId: UINT;
sNciToPlc: NciChannelToPlc;
END_VAR
```

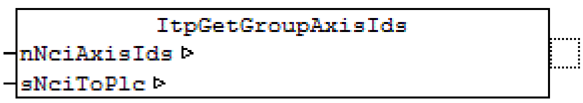
```
bItpError := ItpHasError( sNciToPlc );
IF bItpError THEN
nErrId := ItpGetError( sNciToPlc );
…
END_IF
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.13 ItpGetGroupAxisIds

from library version 6.1.29



ItpGetGroupAxisIds is a function that returns an array of axes IDs that were configured for the group.

Interface

```
FUNCTION ItpGetGroupAxisIds

VAR_IN_OUT
sNciToPlc     : NciChannelToPlc;
nNciAxisIds   : ARRAY[1..8] OF DWORD;
END_VAR
```

NciChannelToPlc [▸ 251]

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.
sNciAxisIds	ARRAY[1..8] OF DWORD	Array of axis IDs

Return value

Data type	Description
DWORD	Error number - this function is only valid for a cyclic interface of version 6 or higher.

ItpGetGroupAxisIds evaluates the information of the variable 'nAcsAxisIDs[8]' from the cyclic interface.

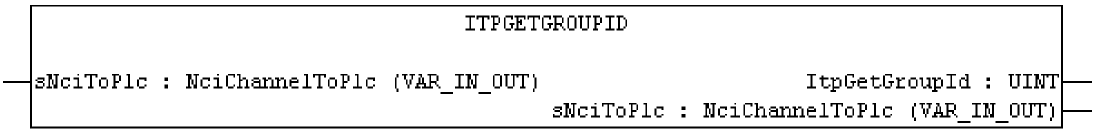
Sample

```
VAR
nNciAxisIds : ARRAY[1..8] OF DWORD;
sNciToPlc : NciChannelToPlc;
nVersionErr : DWORD;
END_VAR
```

```
nVersionErr := ItpGetGroupAxisIds(nNciAxisIds, sNciToPlc );
```

5.1.14 ItpGetGroupId

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



ItpGetGroupId is a function that determines the group ID from the cyclic interface.

Interface

```
FUNCTION ItpGetGroupId

VAR_IN_OUT
sNciToPlc: NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
UINT	Group ID

Sample

```
VAR
nGrpId: UINT;
sNciToPlc: NciChannelToPlc;
END_VAR
```

```
nGrpId := ItpGetGroupId( sNciToPlc );
```

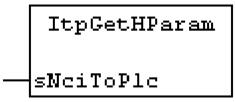
See also: ItpGetChannelId [▶ 113]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.15 ItpGetHParam

from library version 4.0



ItpGetHParam is a function that returns the current H-parameter.

Interface

```
FUNCTION ItpGetHParam

VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
DINT	H parameter

ItpGetHParam evaluates the variable 'nHFuncValue' from the cyclic interface.

Sample

```
VAR
nHParam: DINT;
sNciToPlc: NciChannelToPlc;
END_VAR
```

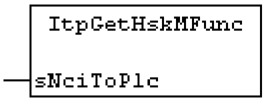
```
nHParam := ItpGetHParam( sNciToPlc );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.16 ItpGetHskMFunc

from library version 4.0



ItpGetHskMFunc supplies the number of the M function of type handshake.

Interface

```
FUNCTION ItpGetHskMFunc

VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
INT	Number of the M-function

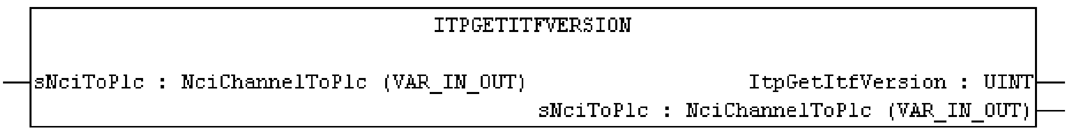
ItpGetHskMFunc evaluates the variable 'nHskMFuncNo' from the cyclic interface.

Prerequisites

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.17 ItpGetItfVersion

from library version 6.1.21and TwinCAT version 2.10 Build 1304



ItpGetItfVersion is a function to determine the version number of the cyclic interface.

Interface

```
FUNCTION ItpGetItfVersion

VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
UINT	Version number of the cyclic interface

Sample

```
VAR
nItfVer: UINT;
sNciToPlc: NciChannelToPlc;
END_VAR
```

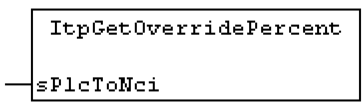
```
nItfVer := ItpGetItfVersion( sNciToPlc );
```

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.18 ItpGetOverridePercent

from library version 4.0



The **ItpGetOverridePercent** function returns the axis channel override as a percentage. It is essential to remember that this is not a value from the NC. The value involved is that which is passed on to the NC as a set value.

Interface

```
FUNCTION ItpGetOverridePercent

VAR_IN_OUT
sPlcToNci     : NciChannelFromPlc;
END_VAR
```

NciChannelFromPlc [▶ 253]

Input parameters

Input & output	Data type	Description
sPlcToNci	NciChannelFromPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed here for reading.

Return value

Data type	Description
LREAL	Override in percent

Sample

```
VAR
sPlcToNci: NciChannelFromPlc;
fOverride: LREAL;
END_VAR
```

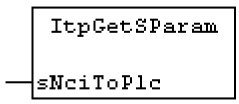
```
fOverride := ItpGetOverridePercent( sPlcToNci );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.19 ItpGetSParam

from library version 4.0



ItpGetSParam is a function that returns the current S-parameter.

Interface

```
FUNCTION ItpGetSParam

VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
UINT	S parameter

ItpGetSParam evaluates the variable 'nSpindleRpm' from the cyclic interface.

Sample

```
VAR
nSParam: UINT;
sNciToPlc: NciChannelToPlc;
END_VAR
```

```
nSParam := ItpGetSParam( sNciToPlc );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciltp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.20 ItpGetStateInterpreter

from library version 4.1

ItpGetStateInterpreter is a function that returns the interpreter state.

Interface

```
FUNCTION ItpGetStateInterpreter : UDINT

VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC

Return value

Data type	Description
UDINT	Current interpreter state

ItpGetStateInterrpreter evaluates the variable 'nItpState' from the cyclic interface.

Sample

```
VAR
nItpState: UDINT;
sNciToPlc: NciChannelToPlc;
END_VAR
```

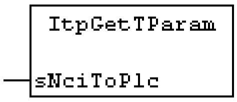
```
nItpState := ItpGetStateInterpreter( sNciToPlc );
```

Requirements

Development Environment	Target system	PLC Libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.21 ItpGetTParam

from library version 4.0



ItpGetTParam is a function that returns the current T parameter.

Interface

```
FUNCTION ItpGetTParam

VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Data type	Description
UINT	T parameter

ItpGetTParam evaluates the variable 'nTool' from the cyclic interface.

Sample

```
VAR
nTParam: UINT;
sNciToPlc: NciChannelToPlc;
END_VAR
```

```
nTParam := ItpGetTParam( sNciToPlc );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.22 ItpGetVersion

from library version 4.0

ItpGetVersion is a function that provides the current version number of this PLC library as a string.

Interface

```
FUNCTION ItpGetVersion

VAR_INPUT
END_VAR
```

Return value

Data type	Description
STRING(20)	Version number

Sample

```
VAR
strVersion: STRING(20);
END_VAR
```

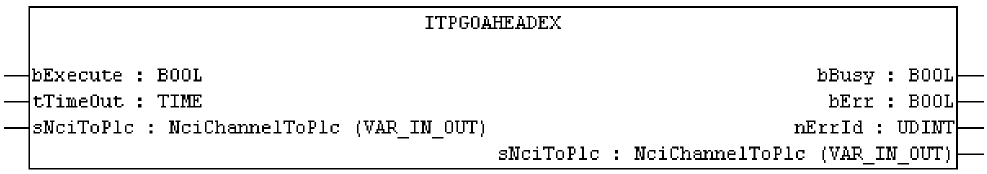
```
strVersion := ItpGetVersion();
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.23 ItpGoAheadEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The ItpGoAheadEx function block may only be used in association with the decoder stop '@717' [▶ 71]. There is a more detailed description of this decoder stop in the interpreter documentation.

Interface

```
VAR_INPUT
bExecute       : BOOL;
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy         : BOOL;
bErr          : BOOL;
nErrId        : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

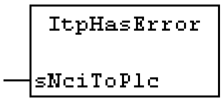
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. The output is reset to FALSE when a new command is executed.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.24 ItpHasError

from library version 4.0



ItpHasError is a function that determines whether the interpreter is in an error state.

Interface

```
FUNCTION ItpHasError : BOOL

VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Table 1: Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC

Return value

If there is an error, the function returns TRUE.

ItpHasError evaluates the variable 'nItpErrCode' from the cyclic interface. If this value does not equal 0, TRUE is returned.

Sample

```
VAR
bItpError: BOOL;
sNciToPlc: NciChannelToPlc;
END_VAR
```

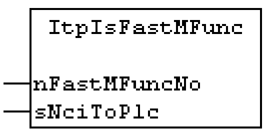
```
bItpError := ItpHasError( sNciToPlc );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.25 ItpIsFastMFunc

from library version 4.0



ItpIsFastMFunc is a function that determines whether the fast M-function is set for the supplied M-function number.

Interface

```
FUNCTION ItpIsFastMFunc

VAR_IN
nFastMFuncNo    : INT;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Input parameters

Input	Data type	Description
nFastMFuncNo	INT	Number of the M-function that is to be checked.

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC

Return value

The function returns TRUE if the fast bit of the M-function is set.

ItpIsFastMFunc evaluates the variable 'nFastMFuncMask' from the cyclic interface.

Sample

```
(*this enum is defined by the user *)

TYPE FastMFuncs:
(
M10_CoolingFluidOn := 10, (*fast M-Funktion M10*)
M11_CoolingFluidOff := 11,
M12_FanOn := 12,
M13_FanOff := 13
);
END_TYPE
```

```
VAR
sNciToPlc: NciChannelToPlc
enFastMFuncs: FastMFuncs;
bTurnFanOn: BOOL;
END_VAR
```

```
bTurnFanOn := ItpIsFastMFunc( M12_FanOn,sNciToPlc );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNcItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.26 ItpIsEStopEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304

The function ItpIsEStopEx indicates whether an EStop command was triggered.

Interface

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Table 2: Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC

Return value

If the return value is TRUE, the function was preceded by an EStop (e.g. ItpEStopEx). The flag does **not** provide information as to whether the axes have already stopped or are still on the deceleration ramp.

After execution of ItpStepOnAfterEStopEx, ItpIsEStopEx returns FALSE again.

ItpIsEStopEx evaluates the cyclic interface.

see also:

ItpEStopEx [▸ 108]

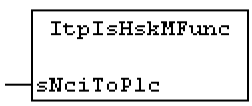
ItpStepOnAfterEStopEx [▸ 155]

Prerequisites

Version 2 of the cyclic channel interface

5.1.27 ItpIsHskMFunc

from library version 4.0



ItpIsHskMFunc determines whether an M-function of type handshake is present.

Interface

```
FUNCTION ItpIsHskMFunc

VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Table 3: Input parameters

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC

Return value

The function returns TRUE if an M-function of type handshake is present.

ItpIsHskFunc evaluates the variable 'nHskMFuncReq' from the cyclic interface.

Sample

```
VAR
bMFuncRequest: BOOL;
END_VAR
```

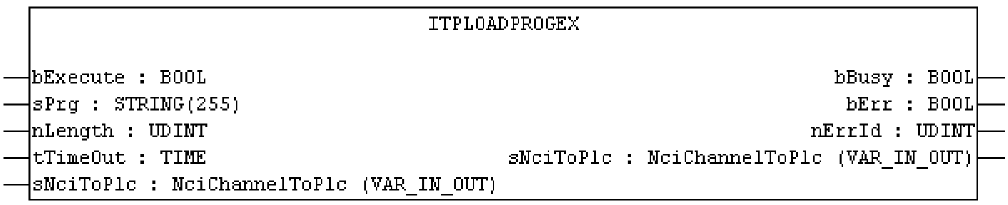
```
bMFuncRequest := ItpIsHskMFunc( sNciToPlc );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.28 ItpLoadProgEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



Interface

```
VAR_INPUT
bExecute        : BOOL;
sPrg            : STRING;
nLength         : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

On a rising edge at the **bExecute** input, the block loads the NC program whose name is given at the **sPrg** input. The string length of the program name is provided at the **nLength** input.

The NC program is sought in the “TwinCAT\cnc” directory if no other information is provided. It is however also possible to give an absolute path.

The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs. The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Sample

```
VAR
in_stItpToPlc AT %I*          : NciChannelToPlc;
fbLoadProg                    : ItpLoadProgEx;
sProgramPath                  : STRING (255):= 'TestIt.nc';
END_VAR
```

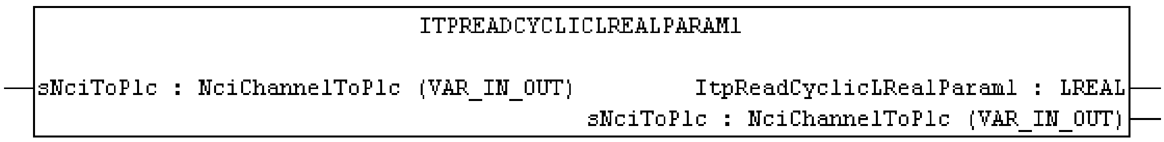
```
fbLoadProg(
bExecute := TRUE,
sPrg := sProgramPath,
nLength := LEN(sProgramPath),
tTimeOut := t#200ms,
sNciToPlc := in_stItpToPlc
);
```

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.29 ItpReadCyclicLRealParam1

from library version 6.1.25 and TwinCAT version 2.10 Build 1320



This function reads the first LREAL parameter from the cyclic channel interface. This parameter is configured previously with ItpSetCyclicLRealOffsets [▸ 145].

Parameter 2 to 4 are read via the same mechanism (e.g. ItpReadCyclicLRealParam2).

Interface

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Parameter 1 of type LREAL.

see also:

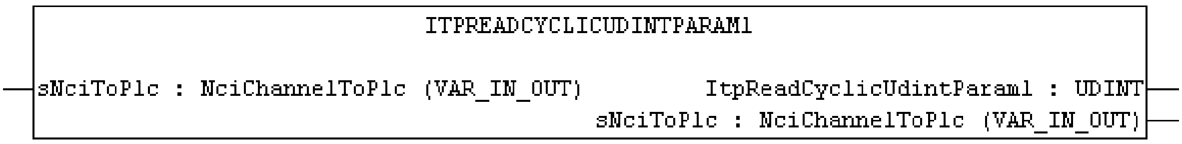
- ItpReadCyclicUdintParam1 [▶ 134]
- ItpSetCyclicLRealOffsets [▶ 145]
- ItpGetCyclicLRealOffsets [▶ 115]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.30 ItpReadCyclicUdintParam1

from library version 6.1.25 and TwinCAT version 2.10 Build 1320



Interface

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Description

This function reads the first UDINT parameter from the cyclic channel interface. This parameter is configured previously with ItpSetCyclicUdintOffsets [▶ 147].

Parameter 2 to 4 are read via the same mechanism (e.g. ItpReadCyclicUdintParam2).

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Return value

Parameter 1 of type UDINT

see also:

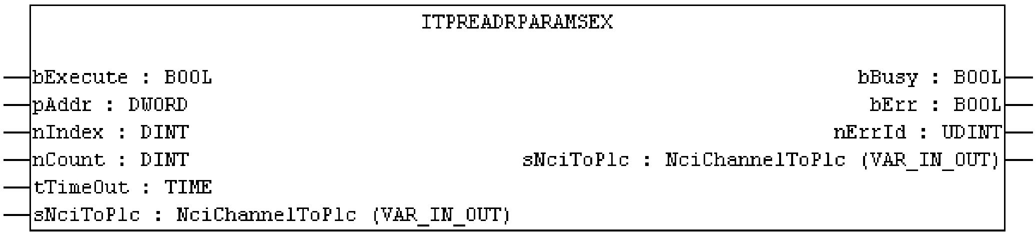
- ItpReadCyclicLRealParam1 [▶ 133]
- ItpSetCyclicUdintOffsets [▶ 147]
- ItpGetCyclicUdintOffsets [▶ 117]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.31 ItpReadRParamsEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The ItpReadRParamsEx function block reads the NC's calculation parameters, also known as R-parameters. A more detailed description of the calculation parameters can be found here [▶ 37]. A total of 1000 R-parameters are available, of which the first 900 (0..899) are local, so that they are only visible in the current NC channel. The other 100 (900..999) R-parameters are global, and are thus visible from anywhere in the NC.

Interface

```
VAR_INPUT
bExecute        : BOOL;
pAddr           : DWORD;
nIndex          : DINT;
nCount          : DINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	A rising edge starts the read operation.
pAddr	DWORD	Address of the target variables of the data to be read. Data are written directly from the specified address, i.e. nIndex is **not** to be interpreted as offset from pAddr. The data are usually stored in an array of type LREAL, which has to be defined by the user.
nIndex	DINT	Describes the index of the R-parameter to be read from an NC perspective.

Input	Data type	Description
nCount	DINT	Number of R-parameters to be read
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

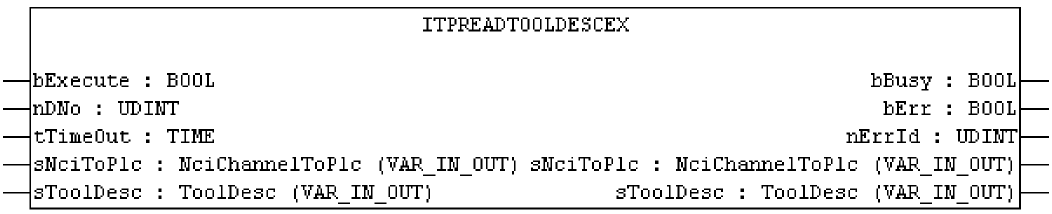
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.32 ItpReadToolDescEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The **ItpReadToolDescEx** function block reads the tool parameters for the supplied D-word.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nDNo            : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
sToolDesc       : ToolDesc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

```
TYPE ToolDesc:
STRUCT
nToolNumber     : UDINT; (*valid range from 0 .. 65535*)
nToolType       : UDINT;
fParam          : ARRAY [2..15] OF LREAL;
END_STRUCT
END_TYPE
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nDNo	UDINT	D-word for which the tool parameters are to be read. nDNo can take values from 1 up to and including 255 (up to library version 5.4.15 only 50 tool parameters are supported).
tTimeOut	TIME	ADS Timeout-Delay
Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.
sToolDesc	ToolDesc	A structure into which the tool parameters of nDNo are written. The meaning of the parameters depends on the tool type, and can be found in the tool data [▸ 83].

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is

Output	Data type	Description
		contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

See also:

ItpWriteToolDescEx [▸ 159]

ItpSetToolDescNullEx [▸ 151]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.33 ItpReadZeroShiftEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304

```
                                  ITPREADZEROSHIFTEX
bExecute : BOOL                                                           bBusy : BOOL
nZsNo : UDINT                                                              bErr : BOOL
tTimeOut : TIME                                                         nErrId : UDINT
sNciToPlc : NciChannelToPlc (VAR_IN_OUT)        sNciToPlc : NciChannelToPlc (VAR_IN_OUT)
sZeroShiftDesc : ZeroShiftDesc (VAR_IN_OUT) sZeroShiftDesc : ZeroShiftDesc (VAR_IN_OUT)
```

The **ItpReadZeroShiftEx** function block reads the offset shift components X, Y and Z for the given zero offset shift.

Interface

```
VAR_INPUT
bExecute       : BOOL;
nZsNo          : UDINT;
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
sZeroShiftDesc : ZeroShiftDesc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

```
TYPE ZeroShiftDesc:
STRUCT
fShiftX         : LREAL;
fShiftY         : LREAL;
fShiftZ         : LREAL;
END_STRUCT
END_TYPE
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nZsNo	UDINT	Number of the zero offset shift on the NC side G54 to G59 are zero offset shifts. The valid range of values for 'nZsNo' is therefore from 54 to 59.
tTimeOut	TIME	ADS Timeout-Delay
Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.
sZeroShiftDesc	ZeroShiftDesc	The structure containing the components of the zero offset shift.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

For reasons of compatibility, there are two entries (coarse and fine) for each axis in each zero offset shift (e.g. G54). These two entries must be added together. This function block evaluates both the entries and adds them together automatically.

See also:

ItpWriteZeroShiftEx [▶ 161]

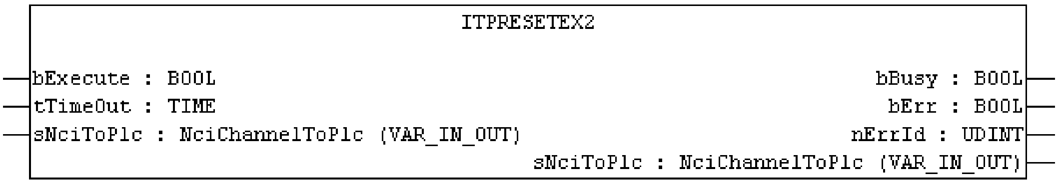
ItpSetZeroShiftNullEx [▶ 152]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.34 ItpResetEx2

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The function block 'ItpResetEx2' executes a channel reset, which deletes all existing tables of the NC channel. In contrast to the outdated function block ItpReset, an active channel is stopped first, before the reset is executed. This simplifies programming in the PLC, since no explicit check is necessary to ascertain whether the axes are still in motion.

Interface

```
VAR_INPUT
bExecute        : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS timeout delay (the bBusy signal can be active for longer than tTimeOut)

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE as long as the axes are stopped according to their set value and the function block has executed the reset

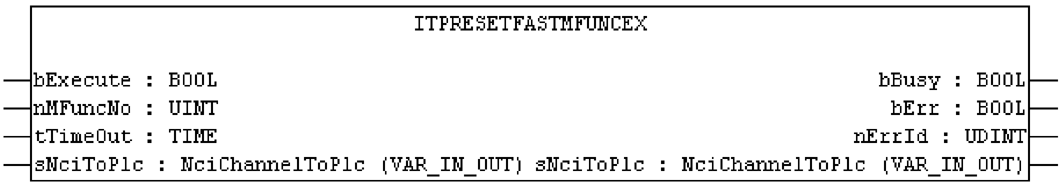
Output	Data type	Description
		successfully, or an error is present at bErr. While Busy = TRUE, no new command will be accepted at the inputs.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.35 ItpResetFastMFuncEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The fast M-function [▶ 67] **nMFuncNo** is reset with an increasing edge at input **bExecute**. In the event of the M-function not being available, **no** error is returned.

This function block represents an alternative to Auto-reset or reset with another M-function (reset list during parameterization of the M-function). For reasons of transparency, mixed resets using an M-function and this function block should be avoided.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nMFuncNo        : UINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nMFuncNo	UINT	Flying M-function that is to be reset
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.36 ItpSetBottleNeckLookAheadEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304

```
                         ITPSETBOTTLENECKLOOKAHEADEX
bExecute : BOOL                                                   bBusy : BOOL
nLookAhead : UDINT                                                 bErr : BOOL
tTimeOut : TIME                                                 nErrId : UDINT
sNciToPlc : NciChannelToPlc (VAR_IN_OUT) sNciToPlc : NciChannelToPlc (VAR_IN_OUT)
```

The function block ItpSetBottleNeckLookAheadEx determines the maximum number of segments the system may look ahead for bottleneck detection (contour collision monitoring). Note that segments, which were added as a result of radius compensation (e.g. additional segments at acute angles) are taken into account.

A more detailed description can be found in the Interpreter [▶ 95] documentation.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nLookAhead      : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nLookAhead	UDINT	Specifies the look-ahead value
tTimeOut	TIME	ADS Timeout-Delay
Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.

Output	Data type	Description
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.37 ItpSetBottleNeckModeEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304

```
                            ITPSETBOTTLENECKMODEEX
bExecute : BOOL                                                          bBusy : BOOL
eBottleNeckMode : E_ItpBnMode                                             bErr : BOOL
tTimeOut : TIME                                                        nErrId : UDINT
sNciToPlc : NciChannelToPlc (VAR_IN_OUT) sNciToPlc : NciChannelToPlc (VAR_IN_OUT)
```

The function block ItpSetBottleNeckModeEx specifies the behavior in the event of a contour collision (bottleneck).

There is a more detailed description in the Interpreter [▶ 95] documentation.

Interface

```
VAR_INPUT
bExecute       : BOOL;
eBottleNeckMode: E_ItpBnMode
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

```
TYPE E_ItpBnMode:
(
ItpBnm_Abort   := 0,
ItpBnm_Adjust  := 1,
ItpBnm_Leave   := 2
);
END_TYPE
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.

Input	Data type	Description
eBottleNeckMode	E_ItpBnMode	Enum for the behavior in the event of a contour collision
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

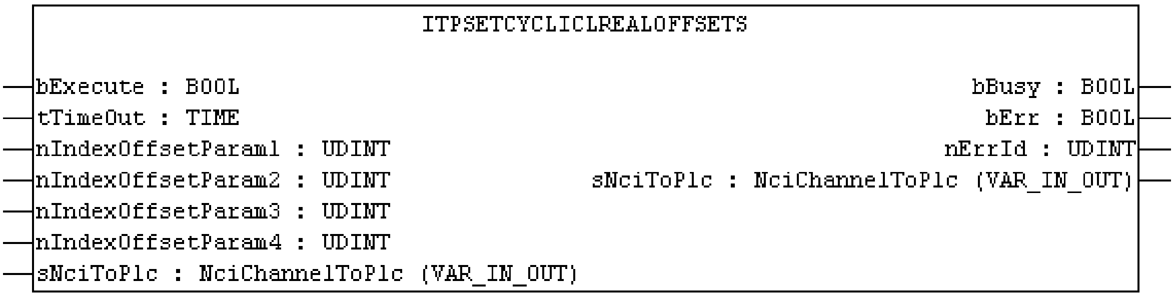
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.38 ItpSetCyclicLrealOffsets

from library version 6.1.25 and TwinCAT version 2.10 Build 1320



The function block ItpSetCyclicLrealOffsets is used to describe the cyclic channel interface for the 4 freely configurable LREAL variables. Variables (index offsets) can be selected from the group state [▸ 272].

The functionality is only active if nIndexOffsetParam1 is not equal 0.

Interface

```
VAR_INPUT
bExecute           : BOOL;
tTimeOut           : TIME;
nIndexOffsetParam1 : UDINT;
nIndexOffsetParam2 : UDINT;
nIndexOffsetParam3 : UDINT;
nIndexOffsetParam4 : UDINT;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc          : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy              : BOOL;
bErr               : BOOL;
nErrId             : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay
nIndexOffsetParam1	UDINT	Group state (index offset) for parameter 1
nIndexOffsetParam2	UDINT	Group state (index offset) for parameter 2
nIndexOffsetParam3	UDINT	Group state (index offset) for parameter 3
nIndexOffsetParam4	UDINT	Group state (index offset) for parameter 4

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the

Output	Data type	Description
		duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

See also:

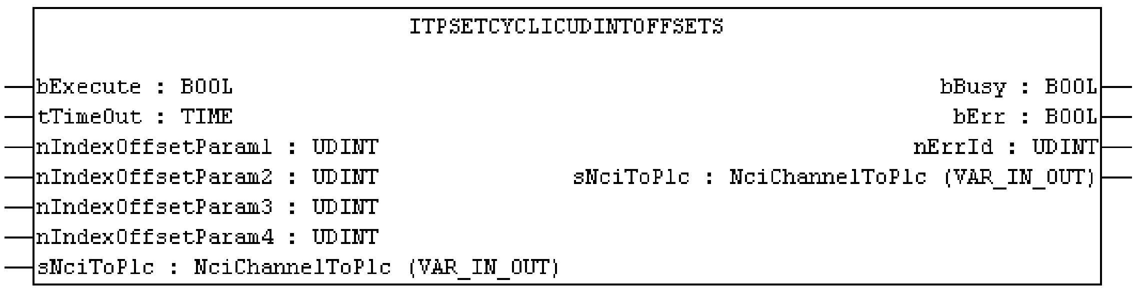
- ItpReadCyclicLRealParam1 [▸ 133]
- ItpGetCyclicLRealOffsets [▸ 115]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.39 ItpSetCyclicUDintOffsets

from library version 6.1.25 and TwinCAT version 2.10 Build 1320



The function block ItpSetCyclicUDintOffsets is used to describe the cyclic channel interface for the 4 freely configurable UDINT variables. Variables (index offsets) can be selected from the group state [▸ 272].

The functionality is only active if nIndexOffsetParam1 is not equal 0.

Interface

```
VAR_INPUT
bExecute            : BOOL;
tTimeOut            : TIME;
```

```
nIndexOffsetParam1 : UDINT;
nIndexOffsetParam2 : UDINT;
nIndexOffsetParam3 : UDINT;
nIndexOffsetParam4 : UDINT;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc          : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy              : BOOL;
bErr               : BOOL;
nErrId             : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay
nIndexOffsetParam1	UDINT	Group state (index offset) for parameter 1
nIndexOffsetParam2	UDINT	Group state (index offset) for parameter 2
nIndexOffsetParam3	UDINT	Group state (index offset) for parameter 3
nIndexOffsetParam4	UDINT	Group state (index offset) for parameter 4

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

See also:

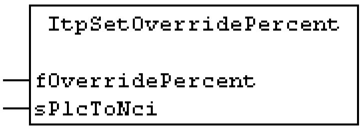
- ItpReadCyclicUDintParam1 [▶ 134]
- ItpGetCyclicUdintOffsets [▶ 117]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.40 ItpSetOverridePercent

from library version 4.0



The **ItpSetOverridePercent** function writes the axis channel override to the cyclic interface to the NCI. The override is passed as a percentage.

Interface

```
FUNCTION ItpSetOverridePercent

VAR_INPUT
fOverridePercent : LREAL;
END_VAR

VAR_IN_OUT
sPlcToNci        : NciChannelFromPlc;
END_VAR
```

NciChannelFromPlc [▶ 253]

Table 4: Parameter

Input	Data type	Description
fOverridePercent	LREAL	Axis channel override as a percentage
Input & output	Data type	Description
sPlcToNci	NciChannelFromPlc	The structure of the cyclic channel interface from the PLC to the NCI

Table 5: Return value

Data type	Description
BOOL	always TRUE

Sample

```
VAR
sPlcToNci: NciChannelFromPlc;
fOverride: LREAL;
END_VAR
```

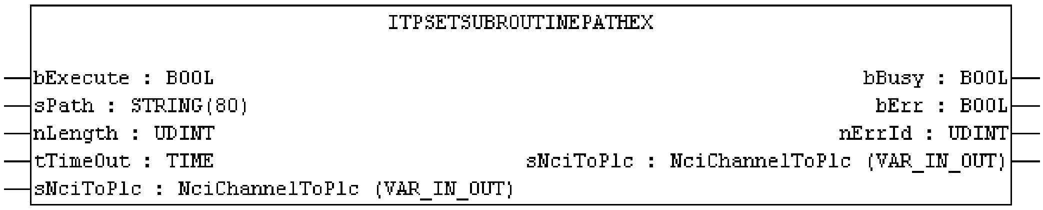
```
fOverride := 47.11;
ItpSetOverridePercent( fOverride, sPlcToNci );
```

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.41 ItpSetSubroutinePathEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



With block ItpSetSubroutinePathEx, the search path for subroutines can optionally be set.

If a subroutine still has to be integrated, the file is searched in the following order:

1. optional search path (ItpSetSubroutinePath)
2. path from which the main program was loaded
3. TwinCAT\CNC folder

Only one optional path can take effect, which remains active until it is overwritten with another path or an empty string.

After a TwinCAT restart, the path has to be re-assigned.

Interface

```
VAR_INPUT
bExecute        : BOOL;
sPath           : STRING;
nLength         : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPath	STRING	Optional path for subroutines. Is deactivated with an empty string
nLength	UDINT	String length
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

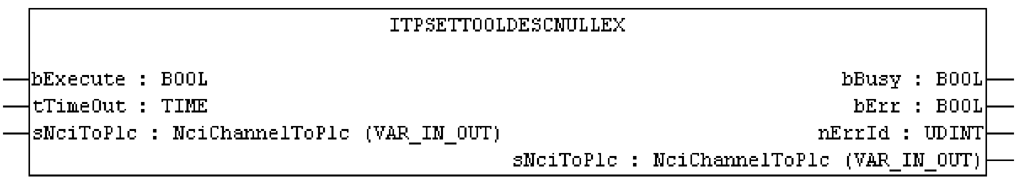
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.42 ItpSetToolDescNullEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



FB ItpSetToolDescNullEx overwrites all tool parameters (incl. number & type) of the channel with zero.

Interface

```
VAR_INPUT
bExecute        : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

A rising edge at input **bExecute** results in overwriting of all tool parameters of the NC channel with zero.

The bBusy output remains TRUE until the function block has executed the command, with the maximum duration specified by the time associated with the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

See also:

ItpWriteToolDescEx [▶ 159]

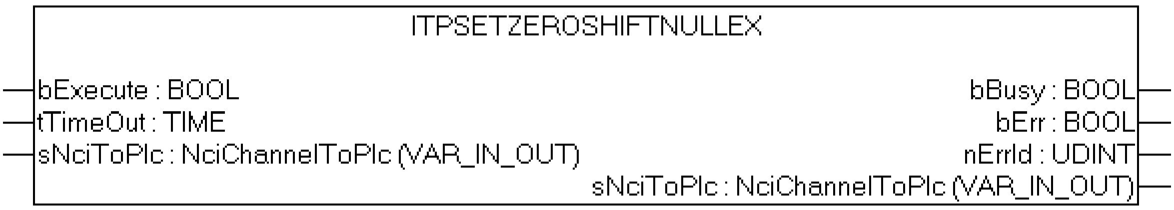
ItpReadToolDescEx [▶ 136]

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.43 ItpSetZeroShiftNullEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The function block ItpSetZeroShiftNullEx overwrites all zero shifts of the channel with zero.

Interface

```
VAR_INPUT
bExecute        : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

A rising edge at input **bExecute** results in overwriting of all zero offset shifts of the NC channel with zero.

The bBusy output remains TRUE until the function block has executed the command, with the maximum duration specified by the time associated with the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

See also:

- ItpWriteZeroShiftEx [▶ 161]
- ItpReadZeroShiftEx [▶ 138]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.44 ItpSingleBlock

from library version 6.1.26

```
                              ITPSINGLEBLOCK
bExecuteModeChange : BOOL                                        bBusy : BOOL
nMode : E_ItpSingleBlockMode                                      bErr : BOOL
bTriggerNext : BOOL                                            nErrId : UDINT
tTimeOut : TIME                    sNciToPlc : NciChannelToPlc (VAR_IN_OUT)
sNciToPlc : NciChannelToPlc (VAR_IN_OUT)
```

The ItpSingleBlock block activates or deactivates single block mode in the NCI. Block relaying can be triggered directly from the PLC with the input 'bTriggerNext'. Alternatively the Start button of the interpreter (F5) can be used in the System Manager.

A more detailed description can be found in the interpreter documentation [▶ 36].

Interface

```
VAR_INPUT
bExecuteModeChange   : BOOL;
nMode                : E_ItpSingelBlockMode;
bTriggerNext         : BOOL
tTimeOut             : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc            : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy                : BOOL;
bErr                 : BOOL;
nErrId               : UDINT;
END_VAR
```

```
TYPE E_ItpSingleBlockMode:
(
ItpSingleBlockOff    := 0,
ItpSingleBlockNck    := 1,
```

```
ItpSingleBlockIntp  := 16#4000
);
END_TYPE
```

Description

The block has the following inputs:

Input	Data type	Description
bExecuteModeChange	BOOL	Single block mode (nMode) is activated through a rising edge at this input.
nMode	E_ItpSingleBlockMode	Operation mode for single block (cf. single block mode [▸ 36]): • ItpSingleBlockOff: single block off • ItpSingleBlockNck: single block in NC kernel • ItpSingleBlockIntp: single block in interpreter
bTriggerNext	BOOL	Block relaying is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

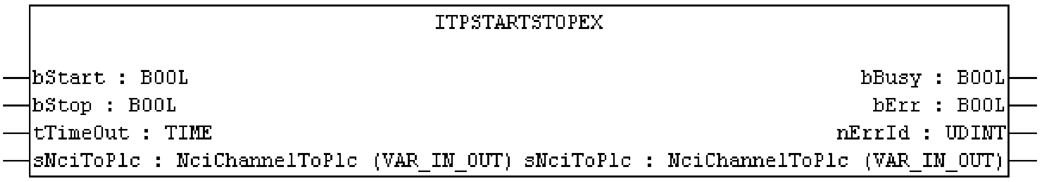
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development environment	Target system	PLC libraries to be included
TwinCAT v2.7.0	PC (i386)	not implemented yet
TwinCAT v2.8.0	PC (i386)	TcNci.lib (ab Lib. V 6.1.26)

5.1.45 ItpStartStopEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



Description

The function block ItpStartStopEx starts the NC channel, if a positive edge is applied at input **bStart**. On a positive edge at the **bStop** input, the NC channel is stopped. The bStop input has a higher priority than the bStart input, i.e. if both inputs have a positive edge, a channel stop is executed.
With a stop command, all tables in the NC are deleted and the axes are stopped in a controlled manner.

The bBusy output remains TRUE until the function block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Interface

```
VAR_INPUT
bStart          : BOOL;
bStop           : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

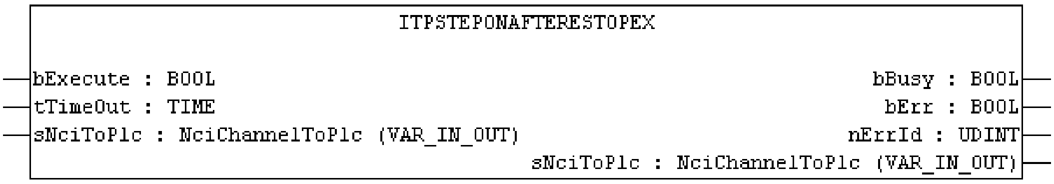
```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.46 ItpStepOnAfterEStopEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The function block ItpStepOnAfterEStopEx enables further processing of the parts program after a programmed EStopEx.

Interface

```
VAR_INPUT
bExecute        : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.

Output	Data type	Description
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

See also:

ItpEStopEx [▶ 108]

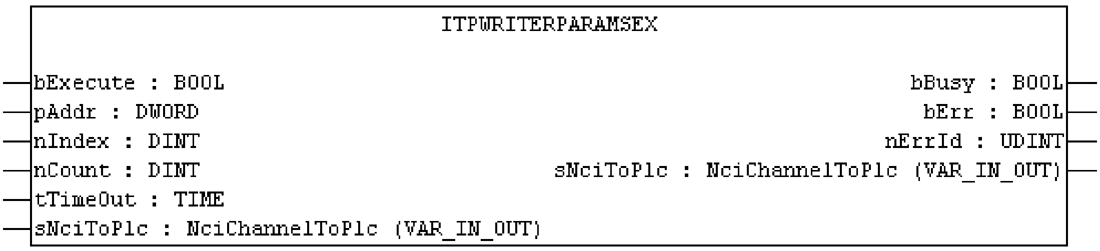
ItpIsEStopEx [▶ 130]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.47 ItpWriteRParamsEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The function block ItpWriteRParamsEx writes R-parameters into the NC.

Interface

```
VAR_INPUT
bExecute        : BOOL;
pAddr           : DWORD;
nIndex          : DINT;
nCount          : DINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

Input	Data type	Description
bExecute	BOOL	A rising edge starts the write operation.
pAddr	DWORD	Address of the variables containing the data to be written. Data are used directly from the specified address, i.e. nIndex is **not** to be interpreted as offset from pAddr. The data are usually read from an array of type LREAL, which has to be defined by the user.
nIndex	DINT	Describes the index of the R-parameter to be written from an NC perspective.
nCount	DINT	Number of R-parameters to be written
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Sample

```
VAR
arrfRParam90to99  : ARRAY[0..9] OF LREAL;
fbWriteRParamEx   : ItpWriteRParamsEx;
n                 : INT := 0;
bWriteParam       : BOOL := FALSE;
END_VAR
```

```
FOR n:=0 TO 9 DO
arrfRParam90to99[n] := 90 + n;
END_FOR

fbWriteRParam(
 bExecute := bWriteParam,
 pAddr := ADR( arrfRParam90to99[0] ),
 nIndex := 90,
 nCount := 10,
 tTimeOut := T#200ms,
 sNciToPlc := g_sNciToPlc );
```

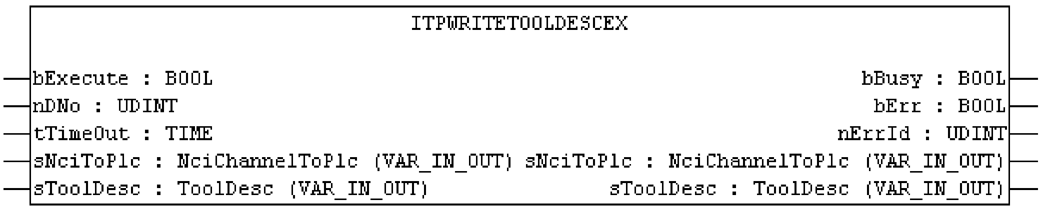
In this example the parameters R90 to R99 are written from an NC perspective.

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.48 ItpWriteToolDescEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The function block **ItpWriteToolDescEx** writes a block of tool parameters.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nDNo            : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
sToolDesc       : ToolDesc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

```
TYPE ToolDesc:
STRUCT
nToolNumber     : UDINT; (*valid range from 0 .. 65535*)
nToolType       : UDINT;
fParam          : ARRAY [2..15] OF LREAL;
END_STRUCT
END_TYPE
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.

Input	Data type	Description
nDNo	UDINT	D-word for which the tool parameters are to be read. nDNo can assume values between 1 and 50 (the library TcNci.lib from version 5.4.16 supports 255 tool parameters).
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.
sToolDesc	ToolDesc	The structure that contains the new tool parameters. This structure is only accessed for reading. The meaning of the parameters depends on the tool type, and can be found in the tool data [▸ 83].

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

See also:

ItpReadToolDescEx [▸ 136]

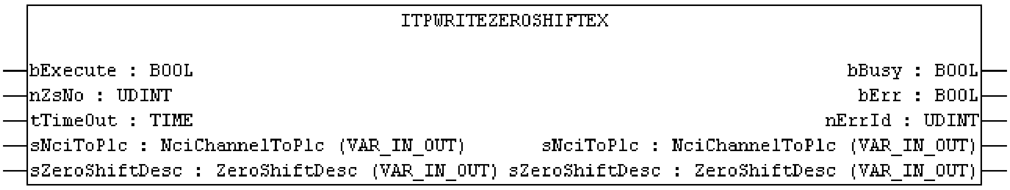
ItpSetToolDescNullEx [▸ 151]

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.49 ItpWriteZeroShiftEx

from library version 6.1.21 and TwinCAT version 2.10 Build 1304



The function block **ItpWriteZeroShiftEx** writes the shift components X, Y and Z for the specified zero shift.

Interface

```
VAR_INPUT
bExecute       : BOOL;
nZsNo          : UDINT;
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
sZeroShiftDesc : ZeroShiftDesc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

```
TYPE ZeroShiftDesc:
STRUCT
fShiftX        : LREAL;
fShiftY        : LREAL;
fShiftZ        : LREAL;
END_STRUCT
END_TYPE
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nZsNo	UDINT	Number of the zero offset shift on the NC side G54 to G59 are zero offset shifts. G58 and G59 can only be edited from the NC program. The valid range of values for 'nZsNo' is therefore from 54 to 57.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.
sZeroShiftDesc	ZeroShiftDesc	The structure containing the components of the zero offset shift. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Notice For reasons of compatibility every zero offset shift that can be set has two parameters (coarse and fine) for each axis. When using this function block to write a new zero offset shift, the new value is written into the 'fine parameter'. A value of 0.0 is entered into the 'coarse parameter'.

This makes it possible to use a function block such as ItpReadZeroShiftEx [▸ 138] to read and modify a zero offset shift and to send it back to the NC.

See also:

- ItpReadZeroShiftEx [▸ 138];
- ItpSetZeroShiftNullEx [▸ 152]

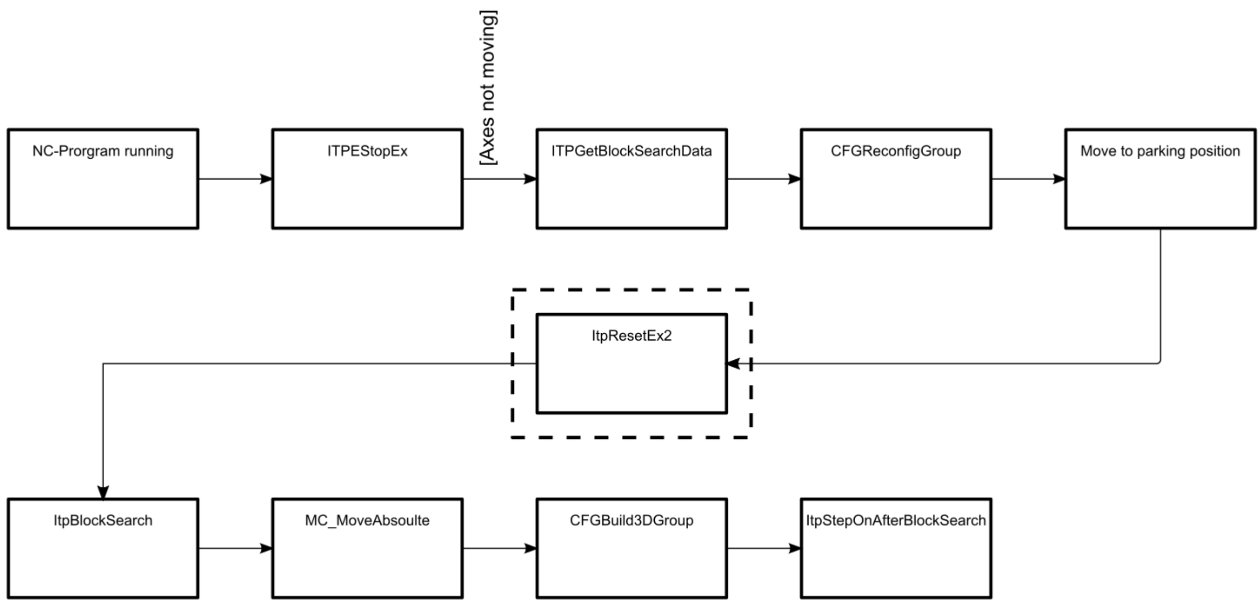
Requirements

Devlopment Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

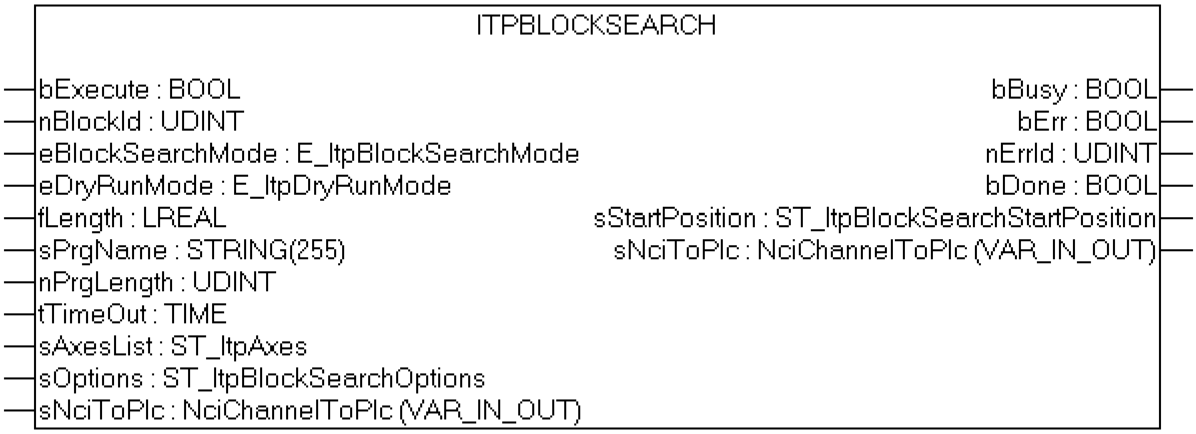
5.1.50 Blocksearch

Block search can be used to interrupt a program for a tool change or at the end of a shift. After the interruption the program can continue at the previous position.

The diagram illustrates how the block search is used.



5.1.50.1 ItpBlocksearch



The function block ItpBlocksearch sets the interpreter to the point defined at the inputs. The input values can be taken from function block ItpGetBlocksearchData [▶ 166] or set manually. Once the interpreter has been set to the defined location with ItpBlocksearch, the motion can continue with ItpStepOnAfterBlocksearch [▶ 166] at the position indicated at output sStartPosition.

```
VAR_INPUT
bExecute            : BOOL;
nBlockId            : UDINT;
eBlockSearchMode    : E_ItpBlockSearchMode;
eDryRunMode         : E_ItpDryRunMode;
fLength             : LREAL;
sPrgName            : STRING(255);
nPrgLength          : UDINT;
tTimeOut            : TIME;
sAxesList           : ST_ItpAxes;
sOptions            : ST_ItpBlockSearchOptions;
END_VAR
```

bExecute: The command is triggered by a rising edge at this input.

nBlockId: Block number or EntryCounter of the segment in the NC program used as starting point.

eBlockSearchMode: Defines whether the specified nBlockId is a block number (e.g. N4711) or continuous EntryCounter. A prerequisite for using the block number is that it is unique. See E_ItpBlockSearchMode [▶ 164]

eDryRunMode: Defines which program lines are executed and which are skipped. See E_ItpDryRunMode [▶ 164]

fLength: Remaining length within the segment selected with nBlockId in percent.

sPrgName: Name or path of the program to be executed.

nPrgLength: Indicates the length of string *sPrgName*.

tTimeOut: ADS timeout delay

sAxesList: Definition of the axes in the NCI group. See ST_ItpAxes [▶ 165]

sOptions: Provides information on retrace.

```
VAR_IN_OUT
sNciToPlc         : NciChannelToPlc;
END_VAR
```

sNciToPlc: The structure of the cyclic channel interface between NCI and PLC. This structure is only accessed for reading. See NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy             : BOOL;
bErr              : BOOL;
nErrId            : UDINT;
bDone             : BOOL;
sStartPosition    : ST_ItpBlockSearchStartPosition;
END_VAR
```

bBusy: Remains TRUE until the function block has executed a command request, but no longer than the time specified at the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs.

bErr: Becomes TRUE if an error occurs while executing the command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.

nErrId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

bDone: The output becomes TRUE when the command was executed successfully.

sStartPosition: Indicates the start position from which the NC program continues. The individual axes should be moved to this position before ItpStepOnAfterBlocksearch [▶ 166] is executed. See ST_ItpBlockSearchStartPosition [▶ 165]

E_ItpBlockSearchMode

E_ItpBlockSearchMode is used to define in which way the block search is executed.

```
TYPE E_ItpBlockSearchMode :
(
ItpBlockSearchMode_Disable        := 0,
ItpBlockSearchMode_BlockNo        := 1,
ItpBlockSearchMode_EntryCounter   := 2
);
END_TYPE
```

ItpBlockSearchMode_Disable: Block search disabled (initial value).

ItpBlockSearchMode_BlockNo: The block search is executed via the block number (e.g. N4711) programmed by the user in the NC program. A prerequisite is that the user-defined block number is unique.

ItpBlockSearchMode_EntryCounter: The block search is executed via a unique EntryCounter. This EntryCounter is implicitly unique, but it is not visible to the user in the NC program.

E_ItpDryRunMode

E_ItpDryRunMode defines how blocks are handled, which are located before the block in which the block search is started, when the program is executed.

```
TYPE E_ItpDryRunMode :
(
ItpDryRunMode_Disable               := 0,
ItpDryRunMode_SkipAll               := 1,
ItpDryRunMode_SkipMotionOnly        := 2,
ItpDryRunMode_SkipDwellAndMotion    := 3
);
END_TYPE
```

ItpDryRunMode_Disable: DryRun disabled (initial value).

ItpDryRunMode_SkipAll: All previous blocks are skipped. R-parameters are written.

ItpDryRunMode_SkipMotionOnly: Only movement blocks are skipped. R-parameters are written, and dwell times and M-functions are executed.

ItpDryRunMode_SkipDwellAndMotion: Movement blocks and dwell times are skipped. R-parameters are written and M-functions are executed.

ST_ItpAxes

The structure ST_ItpAxes contains the axes that were in the NCI group during program execution.

```
TYPE ST_ItpAxes :
STRUCT
nAxisIds            : ARRAY[1..8] OF UDINT;
END_STRUCT
END_TYPE
```

nAxisIds: Array of axes that were in the NCI group. The order is nAxisIds[1]=X, nAxisIds[2]=Y, nAxisIds[3]=Z, nAxisIds[4]=Q1, nAxisIds[5]=Q2… The axis ID can be read from the cyclic axis interface.

St_ItpBlockSearchOptions

The structure contains information on the retrace functionality.

```
TYPE ST_ItpBlockSearchOptions :
STRUCT
bIsRetrace          : BOOL:= FALSE;
bRetraceBackward    : BOOL:= FALSE;
END_STRUCT
END_TYPE
```

bIsRetrace: Indicates whether the retrace functionality is active.

bRetraceBackward: Indicates whether reverse movement took place on the path.

ST_ItpBlockSearchStartPosition

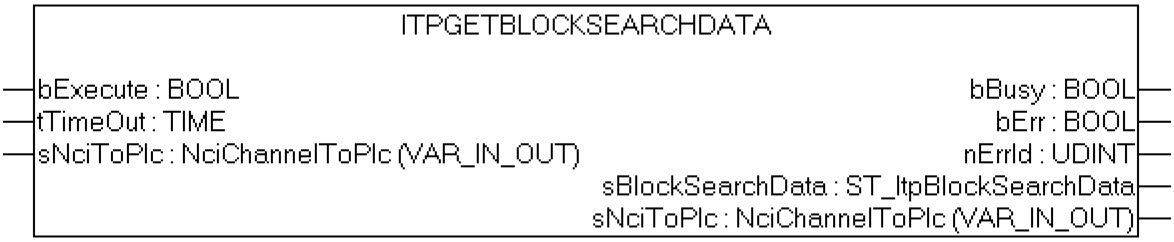
The structure indicates the position at which the NC program continues after a block search. The user is responsible for moving the axes to the corresponding positions.

```
TYPE ST_ItpBlockSearchStartPosition :
STRUCT
fStartPosition      : ARRAY[1..8] OF LREAL;
END_STRUCT
END_TYPE
```

fStartPosition: Array of axis positions at which the NC program continues.

The order is fStartPosition[1]=X, fStartPosition [2]=Y, fStartPosition [3]=Z, fStartPosition [4]=Q1, fStartPosition [5]=Q2…

5.1.50.2 ItpGetBlocksearchData



The function block ItpGetBlocksearchData reads the current position on the path. Usually this command is called at standstill. Subsequently ItpBlockSearch [▶ 163] can be used to set the interpreter to the position stored in sBlockSearchData.

```
VAR_INPUT
bExecute            : BOOL;
tTimeOut            : TIME;
END_VAR
```

bExecute: The command is triggered by a rising edge at this input.

bTimeOut: ADS Timeout-Delay

```
VAR_IN_OUT
sNciToPlc           : NciChannelToPlc;
END_VAR
```

sNciToPlc: The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading. See NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy               : BOOL;
bErr                : BOOL;
nErrId              : UDINT;
sBlockSearchData    : ST_ItpBlockSearchData;
END_VAR
```

bBusy: Remains TRUE until the function block has executed a command request, but no longer than the time specified at the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs.

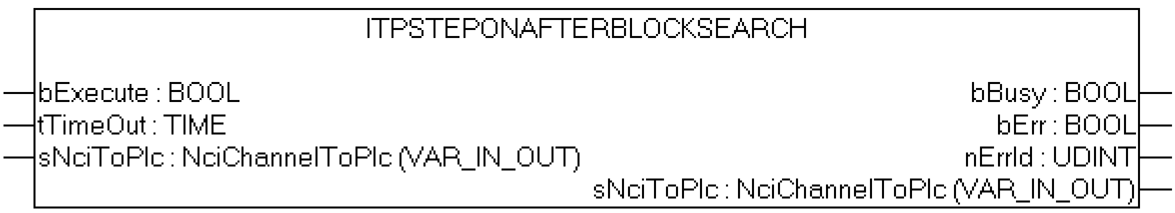
bErr: Becomes TRUE if an error occurs during command execution. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.

nErrId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

sBlockSearchData: Contains information on the current position on the path.

```
TYPE ST_ItpBlockSearchData :
STRUCT
fLength             : LREAL;(* remaining distance of actual movement block in percent*)
nBlockNo            : UDINT;(* number of the actual block *)
nBlockCounter       : UDINT;(* counter value of the actual block *)
bIsRetrace          : BOOL;(* e.g. by activ Retrace *)
bRetraceBackward    : BOOL;(* e.g. by activ Retrace Backward *)
END_STRUCT
END_TYPE
```

5.1.50.3 ItpStepOnAfterBlocksearch



Starts the motion after a block search.

The axes first have to be moved to the positions output by ItpBlocksearch [▸ 163].

```
VAR_INPUT
bExecute            : BOOL;
tTimeOut            : TIME;
END_VAR
```

bExecute: The command is triggered by a rising edge at this input.

bTimeOut: ADS Timeout-Delay

```
VAR_IN_OUT
sNciToPlc           : NciChannelToPlc;
END_VAR
```

sNciToPlc: The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading. See NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy               : BOOL;
bErr                : BOOL;
nErrId              : UDINT;
END_VAR
```

bBusy: Remains TRUE until the function block has executed a command request, but no longer than the time specified at the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs.

bErr: Becomes TRUE if an error occurs during command execution. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.

nErrId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

5.1.51 Retrace

5.1.51.1 ItpEnableFeederBackup

from library version 6.1.24 and anticipated with TwinCAT version 2.10 Build 1308

```
                              ITPENABLEFEEDERBACKUP

bEnable : BOOL                                                  bBusy : BOOL
bExecute : BOOL                                                  bErr : BOOL
tTimeOut : TIME                                               nErrId : UDINT
sNciToPlc : NciChannelToPlc (VAR_IN_OUT) sNciToPlc : NciChannelToPlc (VAR_IN_OUT)
```

The function block ItpEnableFeederBackup enables storing of the path for reversing. It has to be activated once before the NC program (G-Code) is started. If the Blocksearch [▸ 162] functionality is used, ItpEnableFeederBackup has to be activated before ItpBlocksearch [▸ 163] is called. Feeder backup is executed as long as a TwinCAT restart or bEnable = FALSE is triggered with a rising edge at bExecute.

If feeder backup is not enabled, reversing does not work. This can be verified via ItpIsFeederBackupEnabled [▸ 169].

Interface

```
VAR_INPUT
bEnable        : BOOL;
bExecute       : BOOL;
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

Input	**Data type**	**Description**
bEnable	BOOL	TRUE: enables feeder backup FALSE: disables feeder backup
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input/output	**Data type**	**Description**
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

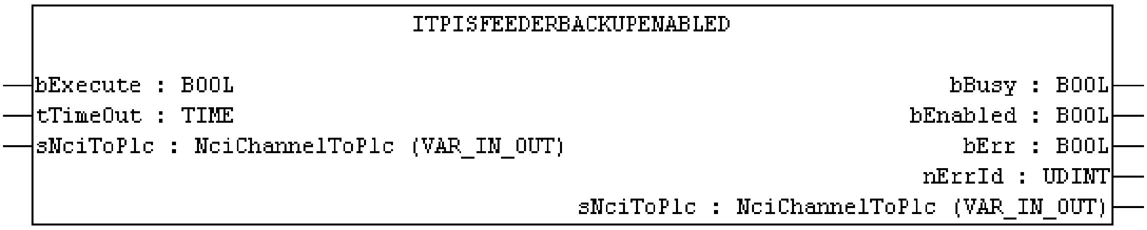
Output	**Data type**	**Description**
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	**Target System Type**	**PLC Libraries to include**
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.51.2 ItpIsFeederBackupEnabled

from library version 6.1.24 and anticipated with TwinCAT version 2.10 Build 1308



The function block ItpIsFeederBackupEnabled indicates whether feeder backup is enabled. Reversing requires feeder backup to be enabled first. This activates storing of the path.

Interface

```
VAR_INPUT
bExecute       : BOOL;
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

```
VAR_OUTPUT
bBusy          : BOOL;
bEnabled       : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input/output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	The bBusy output remains TRUE until the function block has executed a command, with the maximum duration specified by the time associated with the ‘Timeout’ input. While bBusy = TRUE, no new instruction will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.

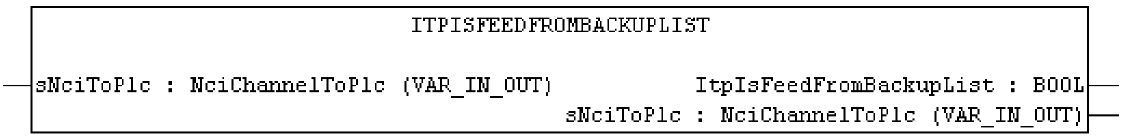
Output	Data type	Description
bEnabled	BOOL	TRUE: Backup list for tracing is enabled FALSE: Backup list for tracing is disabled
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. If the function block has a timeout error, 'Error' is TRUE and 'nErrId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.51.3 ItpIsFeedFromBackupList

from library version 6.1.24 and anticipated with TwinCAT version 2.10 Build 1308



The function ItpIsFeedFromBackupList becomes TRUE when the feed entries (SAF & SVB) were sent from the backup list. During reversing all entries are sent from the backup list. If the program is executed in forward mode, the first entries usually also originate from the backup list. This is dependent of the number of retraced entries and the number of entries in the SVB and SAF tables at the time at which tracing was called. All further commands originate from the ‚original' code.

While the NCI is processing the backup list, not all functions are available or meaningful. Here are a few examples:

- Decoder stops such as @714 are not evaluated
- Modifications of R-parameters do not take effect as long as the motion takes place on the backup path (forward or reverse). R-parameters modifications take effect again as soon as the path data no longer come from the backup list.

Interface

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

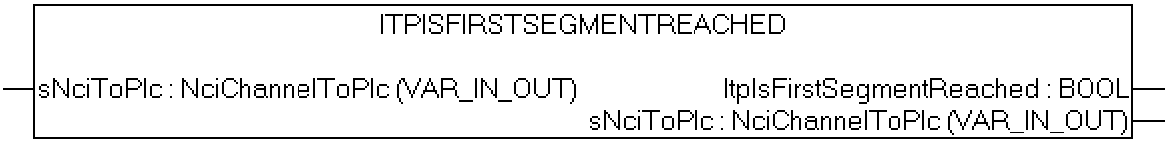
Input/output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.51.4 ItpIsFirstSegmentReached

from version 6 of the cyclic channel interface



ItpIsFirstSegmentReached is a function that determines whether the program starting position is reached during reversing, based on the cyclic channel interface.

Function ItpIsFirstSegmentReached: BOOL

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Table 6: Input parameters

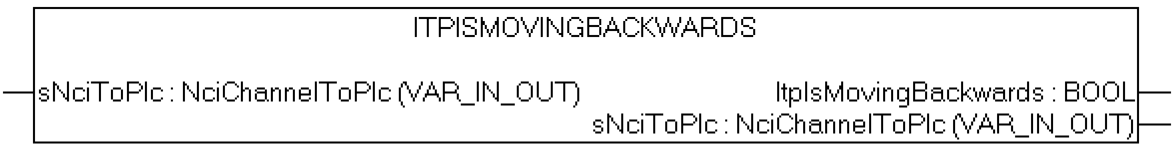
Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC.

Return value

The function returns TRUE when the starting position of the G-Code program is reached. If the version number of the cyclic channel interface is less than 6, the return value is always FALSE.

5.1.51.5 ItpIsMovingBackwards

from version 6 of the cyclic channel interface



ItpIsMovingBackwards is a function that determines whether reverse movement takes place on the path of the current G-Code program, based on the cyclic channel interface.

ItpIsMovingBackwards: BOOL

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▶ 251]

Table 7: Input parameters

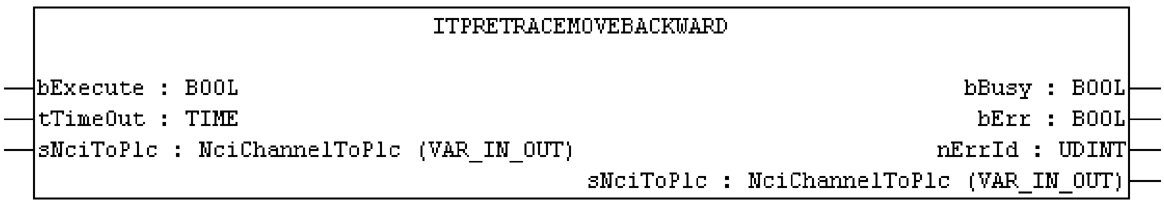
Input & output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC.

Return value

The function returns TRUE when reverse movement takes place on the path. If the version number of the cyclic channel interface is less than 6, the return value is always FALSE.

5.1.51.6 ItpRetraceMoveBackward

from library version 6.1.24 and anticipated with TwinCAT version 2.10 Build 1308



The function block ItpRetraceMoveBackward deals with the geometric entries at the actual position at the start of the part program (G-Code).

Procedure

1. Activate feeder backup list (see ItpEnableFeederBackup [▶ 167])
   ⇨ The NC program is stopped with ItpEStopEx [▶ 108]
2. Wait and ensure that all axes in the group are at standstill
3. Call ItpRetraceMoveBackward
4. Stop backward movement with ItpEStop, otherwise the program returns to the start
5. Call ItpRetraceMoveForward [▶ 173] to move forward again
6. Call ItpEStopEx and ItpRetraceMoveBackward etc., if required.

Notice **Do not use in conjunction with vertex blending. M-functions are suppressed during reversing.**

The function block has the following inputs

Interface

```
VAR_INPUT
bExecute       : BOOL;
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc      : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input/output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.51.7 ItpRetraceMoveForward

from library version 6.1.24 and anticipated with TwinCAT version 2.10 Build 1308

```
                         ITPRETRACEMOVEFORWARD
bExecute : BOOL                                            bBusy : BOOL
tTimeOut : TIME                                             bErr : BOOL
sNciToPlc : NciChannelToPlc (VAR_IN_OUT)                 nErrId : UDINT
                        sNciToPlc : NciChannelToPlc (VAR_IN_OUT)
```

The function block ItpRetraceMoveForward transfers all entries from the current block (e.g. position) in forward travel direction to the NC kernel. It is called to reverse the direction after ItpRetraceMoveBackward [▸ 172] was called.

Interface

```
VAR_INPUT
bExecute        : BOOL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sNciToPlc       : NciChannelToPlc;
END_VAR
```

NciChannelToPlc [▸ 251]

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
tTimeOut	TIME	ADS Timeout-Delay

Input/output	Data type	Description
sNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface from the NCI to the PLC. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.

Output	Data type	Description
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

see also: ItpRetraceMoveBackward [▶ 172]

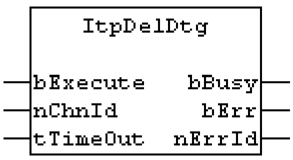
Requirements

Development Environment	Target System Type	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcNci.lib

5.1.52 Blocks for compatibility with existing programs

5.1.52.1 ItpDelDtg

from library version 4.0



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpDelDtgEx [▶ 105].

Interface

```
VAR_INPUT
bExecute       : BOOL;
nChnId         : UDINT;
tTimeOut       : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

Description

The ItpDelDtg block triggers deletion of the remaining travel. There is a more detailed description in the Interpreter [▶ 62] documentation.

The block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID

Input	Data type	Description
tTimeOut	TIME	ADS Timeout-Delay

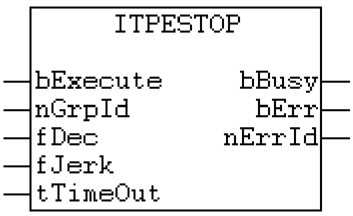
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.2 ItpEStop

from library version 5.3.9 and
TwinCAT V2.8 B540 or V2.9 B905



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpEStopEx [▶ 108].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGrpId          : UDINT;
```

```
fDec            : LREAL;
fJerk           : LREAL;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The function block ItpEStop triggers the NCI EStop and enables a controlled stop on the path. The limit values for the deceleration and the jerk are transferred as parameters. If these should be smaller than the currently active dynamic parameters, the transferring parameters are rejected.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input
nGrpId	UDINT	Group ID
fDec	LREAL	Max. deceleration during stopping. If fDec is smaller than the currently active deceleration, then fDec is not applied. This ensures that the deceleration occurs with the standard ramp as a minimum.
fJerk	LREAL	Max. jerk during stopping If fJerk is smaller than the currently active jerk, fJerk is not applied.
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

see also:

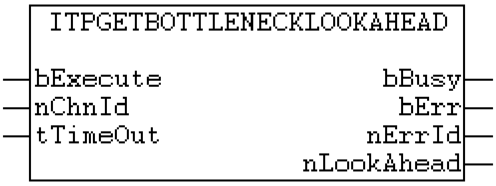
ItpStepOnAfterEStop [▸ 200]

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib
TwinCAT v2.9.0	PC (i386)	TcNci.lib

5.1.52.3 ItpGetBottleNeckLookAhead

from library version 5.3.4



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpGetBottleNeckLookAheadEx [▸ 110].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
nLookAhead      : UDINT;
END_VAR
```

Description

The function block ItpGetBottleNeckLookAhead determines the maximum size of the look-ahead for the bottleneck detection (contour collision monitoring).

There is a more detailed description in the Interpreter [▸ 95] documentation.

The block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new

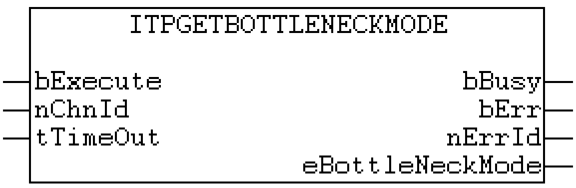
Output	Data type	Description
		command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. If the block has a timeout error, ‘Error’ is TRUE and ‘nErrId’ is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).
nLookAhead	UDINT	Value of the look-ahead for bottleneck detection

Requirements

Development environment	Target System	PLC libraries to inlcude
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib (from Lib. V 5.3.4)

5.1.52.4 ItpGetBottleNeckMode

from library version 5.3.4



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpGetBottleNeckModeEx [▶ 112].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
eBottleNeckMode: E_ItpBnMode
END_VAR
```

```
TYPE E_ItpBnMode:
(
ItpBnm_Abort  := 0,
ItpBnm_Adjust := 1,
ItpBnm_Leave  := 2
);
END_TYPE
```

Description

The function block ItpGetBottleNeckMode reads the behavior in the event of a contour collision (bottleneck).

There is a more detailed description in the Interpreter [▶ 95] documentation.

The block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. If the block has a timeout error, ‘Error’ is TRUE and ‘nErrId’ is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

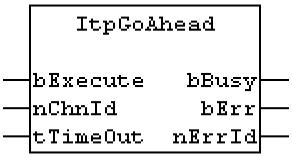
Output	Data type	Description
eBottleNeckMode	E_ItpBnMode	Enum for the behavior in the event of a contour collision

Requirements

Development environment	Target system	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib (from Lib. V 5.3.4)

5.1.52.5 ItpGoAhead

from library version 4.0



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpGoAheadEx [▸ 127].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The function block ItpGoAhead may only be used in association with the decoder stop '@717' [▸ 71]. There is a more detailed description of this decoder stop in the interpreter documentation.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the

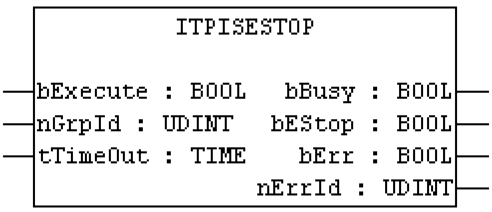
Output	Data type	Description
		execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.6 ItpIsEStop

from library version 5.4.15 and
TwinCAT V2.9 B956



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpIsEStopEx [▸ 130].

Interface

```
VAR_INPUT
bExecute       : BOOL;
nChnId         : UDINT;
tTimeOut       : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy          : BOOL;
bEStop         : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

Description

Via bEStop, the function block ItpIsEStop provides information as to whether an EStop command was triggered. If bEStop is TRUE, then an EStop was initiated (e.g. ItpEStop). The flag does **not** provide information as to whether the axes have already stopped or are still on the deceleration ramp.

After the execution of ItpStepOnAfterEStop, ItpIsEStop will once again return FALSE.

The ItpIsEStop function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input
nGrpId	UDINT	Group ID
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bEStop	BOOL	TRUE: EStop command was executed FALSE: No EStop is present
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

see also:

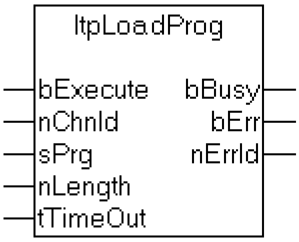
ItpEStop [▸ 176]

ItpStepOnAfterEStop [▸ 200]

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.7.0	PC (i386)	not supported
TwinCAT v2.8.0	PC (i386)	TcNci.lib
TwinCAT v2.9.0	PC (i386)	TcNci.lib

5.1.52.7 ItpLoadProg



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpLoadProgEx [▶ 132].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
sPrg            : STRING;
nLength         : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

On a rising edge at the **bExecute** input, the function block loads the NC program whose name is given at the **sPrg** input. The string length of the program name is applied to the input **nLength**. **nChnId** is the ID of the associated NC channel.

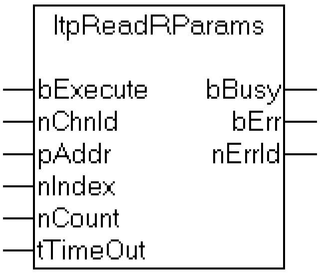
The NC program is sought in the “TwinCAT\cnc” directory if no other information is provided. It is however also possible to give an absolute path.

The bBusy output remains TRUE until the function block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.8 ItpReadRParams



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpReadRParamsEx [▶ 135].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
pAddr           : DWORD;
nIndex          : DINT;
nCount          : DINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The ItpReadRParams block reads the NC's calculation parameters, also known as R-parameters. A more detailed description of the calculation parameters can be found here [▶ 37]. A total of 1000 R-parameters are available, of which the first 900 (0..899) are local, so that they are only visible in the current NC channel. The other 100 (900..999) R-parameters are global, and are thus visible from anywhere in the NC.

Input	Data type	Description
bExecute	BOOL	A rising edge starts the read operation
nChnId	UDINT	ID of the NC channel whose R-parameters are to be read
pAddr	DWORD	Address of the target variables of the data to be read. The data are written by the NC directly from the specified address, i.e. nIndex is **not** to be interpreted as offset from pAddr. The data are usually in an array of type LREAL, which has to be defined by the user.
nIndex	DINT	Describes the index of the R-parameter to be read from an NC perspective.
nCount	DINT	Number of R-parameters to be read
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

see also:

ItpWriteRParams [▸ 202]

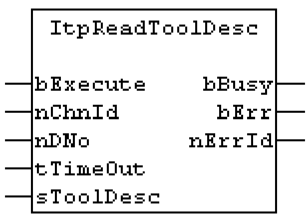
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.7.0	PC (i386)	TcNcIltp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.9 ItpReadToolDesc

from library version 4.0

from TwinCAT V2.7, Build 500



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpReadToolDescEx [▸ 136].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
nDNo            : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sToolDesc       : ToolDesc;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

```
TYPE ToolDesc:
STRUCT
nToolNumber     : UDINT; (*valid range from 0 .. 65535*)
nToolType       : UDINT;
fParam          : ARRAY [2..15] OF LREAL;
END_STRUCT
END_TYPE
```

The **ItpReadToolDesc** block reads the tool parameters for the supplied D-word.

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
nDNo	UDINT	D-word for which the tool parameters are to be read. nDoNo can assume values between 1 and 50 (the library TcNci.lib from version 5.4.16 supports 255 tool parameters).
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sToolDesc	ToolDesc	A structure into which the tool parameters of nDNo are written. The meaning of the parameters depends on the tool type, and can be found in the tool data [▶ 83].

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.

Output	Data type	Description
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

see also:

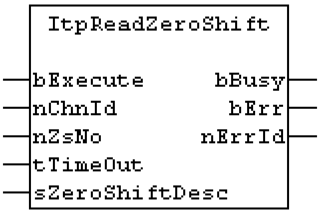
ItpWriteToolDesc [▶ 204]; ItpSetToolDescNull [▶ 198]

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.10 ItpReadZeroShift

from library version 4.0



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpReadZeroShiftEx [▶ 138].

Interface

```
VAR_INPUT
bExecute       : BOOL;
nChnId         : UDINT;
nZsNo          : UDINT;
tTimeOut       : TIME;
END_VAR
```

```
VAR_IN_OUT
sZeroShiftDesc : ZeroShiftDesc;
END_VAR
```

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

```
TYPE ZeroShiftDesc:
STRUCT
fShiftX        : LREAL;
fShiftY        : LREAL;
fShiftZ        : LREAL;
END_STRUCT
END_TYPE
```

The **ItpReadZeroShift** function block reads the offset shift components X, Y and Z for the given zero offset shift.

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input
nChnId	UDINT	Channel ID
nZsNo	UDINT	Number of the zero offset shift on the NC side G54 to G59 are zero offset shifts. The valid range of values for 'nZsNo' is therefore from 54 to 59.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sZeroShiftDesc	ZeroShiftDesc	The structure containing the components of the zero offset shift.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

For reasons of compatibility, there are two entries (coarse and fine) for each axis in each zero offset shift (e.g. G54). These two entries must be added together. This function block evaluates both the entries and adds them together automatically.

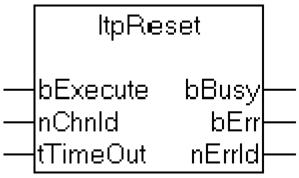
see also:

ItpWriteZeroShift [▸ 205]; ItpSetZeroShiftNull [▸ 199]

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.11 ItpReset



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpResetEx2 [▶ 140].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

On a rising edge at the **bExecute** input, a reset is carried out on the NC channel whose ID is **nChnId**. This deletes all the tables in the NC. The axes are halted immediately. For this reason a reset should only be carried out either in the event of an error or when the axes are stationary.

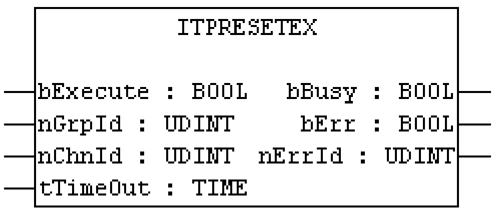
The bBusy output remains TRUE until the function block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.12 ItpResetEx

from TwinCAT V2.10, Build 1245



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpResetEx2 [▸ 140].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGrpId          : UDINT;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The function block 'ItpResetEx' executes a channel reset, which deletes all existing tables of the NC channel. In contrast to the conventional ItpReset [▸ 190], an active channel is stopped first, before the reset is executed. This simplifies programming in the PLC, since no explicit check is necessary to ascertain whether the axes are still in motion.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nGrpId	UDINT	group ID
nChnId	UDINT	Channel ID
tTimeOut	TIME	ADS timeout delay (the bBusy signal can be active for longer than tTimeOut)

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS

Output	Data type	Description
		error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

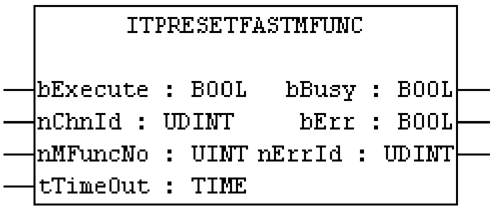
see also: ItpStartStop [▶ 200]

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.13 ItpResetFastMFunc

from library version 5.4.19



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpResetFastMFuncEx [▶ 141].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
nMFuncNo        : UINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

A rising edge at input **bExecute** resets the fast M-function [▶ 67] **nMFuncNo**. In the event of the M-function not being available, **no** error is returned.

This function block represents an alternative to Auto-reset or reset with another M-function (reset list during parameterization of the M-function). For reasons of transparency, mixed resets using an M-function and this function block should be avoided.

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
nMFuncNo	UINT	Flying M-function that is to be reset

Input	Data type	Description
tTimeOut	TIME	ADS Timeout-Delay

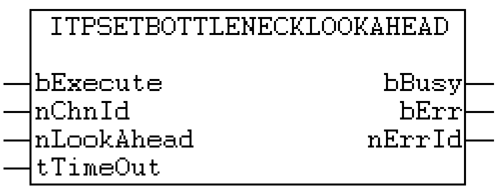
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.14 ItpSetBottleNeckLookAhead

from library version 5.3.4



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpSetBottleNeckLookAheadEx [▸ 142].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
nLookAhead      : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The function block ItpSetBottleNeckLookAhead determines the maximum number of segments the system may look ahead for bottleneck detection (contour collision monitoring). Note that segments, which were added as a result of radius compensation (e.g. additional segments at acute angles) are taken into account.

There is a more detailed description in the Interpreter [▶ 95] documentation.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
nLookAhead	UDINT	Specifies the look-ahead value
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

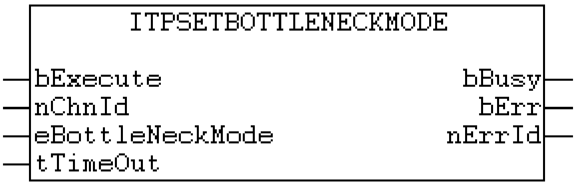
Requirements

Development environment	Target system	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	not implemented

Development environment	Target system	PLC libraries to include
TwinCAT v2.8.0	PC (i386)	TcNci.lib (from Lib. V 5.3.4)

5.1.52.15 ItpSetBottleNeckMode

from library version 5.3.4



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpSetBottleNeckModeEx [▶ 144].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
eBottleNeckMode: E_ItpBnMode
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

```
TYPE E_ItpBnMode:
(
ItpBnm_Abort    := 0,
ItpBnm_Adjust   := 1,
ItpBnm_Leave    := 2
);
END_TYPE
```

Description

The function block ItpSetBottleNeckMode specifies the behavior in the event of a contour collision (bottleneck).

There is a more detailed description in the Interpreter [▶ 95] documentation.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
eBottleNeckMode	E_ItpBnMode	Enum for the behavior in the event of a contour collision
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development environment	TArget system	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	not implemeted
TwinCAT v2.8.0	PC (i386)	TcNci.lib (from Lib. V 5.3.4)

5.1.52.16 ItpSetSubroutinePath

from library version 5.4.17

```
          ITPSETSUBROUTINEPATH

--bExecute : BOOL        bBusy : BOOL--
--nChnId : UDINT          bErr : BOOL--
--sPath : STRING(80) nErrId : UDINT--
--nLength : UDINT
--tTimeOut : TIME
```

Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpSetSubroutinePathEx [▸ 150].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
sPath           : STRING;
```

```
nLength         : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

With block ItpSetSubroutinePath, the search path for subroutines can optionally be set.

If a subroutine still has to be integrated, the file is searched in the following order:

- optional search path (ItpSetSubroutinePath)
- path from which the main program was loaded
- TwinCAT\CNC folder

Only one optional path can be active at any one time. It remains active until it is

- overwritten with another path or
- with an empty string.
- After a TwinCAT restart, the path has to be re-assigned.

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
sPath	STRING	Optional path for subroutines. Is deactivated with an empty string
nLength	UDINT	String length
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

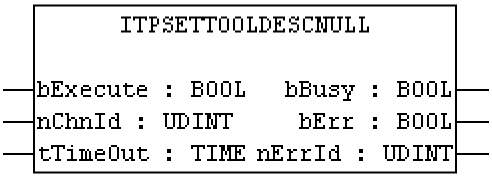
Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.7.0	PC (i386)	not supported
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.17 ItpSetToolDescNull

from library version 5.5

from TwinCAT V2.9, Build 1031



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpSetToolDescNullEx [▶ 151].

FB ItpSetToolDescNull overwrites all tool parameters (incl. number & type) of the channel with zero.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

A rising edge at input **bExecute** overwrites all tool parameters of the NC channel with ID **nChnId** with zero.

The bBusy output remains TRUE until the function block has executed the command, with the maximum duration specified by the time associated with the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

See also:

- ItpWriteToolDesc [▶ 204],
- ItpReadToolDesc [▶ 186]

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.18 ItpSetZeroShiftNull

from library version 5.5

from TwinCAT V2.9, Build 1031

```
          ITPSETZEROSHIFTNULL
bExecute : BOOL    bBusy : BOOL
nChnId : UDINT      bErr : BOOL
tTimeOut : TIME nErrId : UDINT
```

Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpSetZeroShiftNullEx [▶ 152].

FB ItpSetZeroShiftNull overwrites all zero shifts of the channel with zero.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

A rising edge at input **bExecute** overwrites all zero offset shifts of the NC channel with ID **nChnId** with zero.

The bBusy output remains TRUE until the function block has executed the command, with the maximum duration specified by the time associated with the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

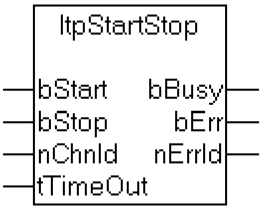
See also:

- ItpWriteZeroShift [▶ 205]
- ItpReadZeroShift [▶ 188]

Requirements

Development environment	Target system type	PLC libraries to be linked
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.19 ItpStartStop



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpStartStopEx [▶ 155].

Interface

```
VAR_INPUT
bStart          : BOOL;
bStop           : BOOL;
nChnId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The ItpStartStop function block starts the NC channel whose ID is passed to the **nChnId** input when a positive edge is applied to the **bStart** input. On a positive edge at the **bStop** input, the NC channel is stopped. The bStop input has a higher priority than the bStart input, i.e. if both inputs have a positive edge, a channel stop is executed.
With a stop command, all tables in the NC are deleted and the axes are stopped in a controlled manner.

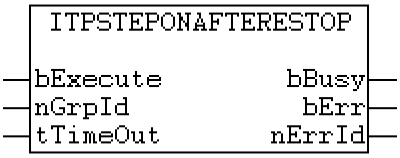
The bBusy output remains TRUE until the function block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE no new command is accepted at the inputs.
The output bErr is switched to TRUE if an error occurred during the execution of the command. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.20 ItpStepOnAfterEStop

from library version 5.3.9 and
TwinCAT V2.8 B540 or V2.9 B905



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpStepOnAfterEStopEx [▶ 155].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGrpId          : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Description

The function block ItpStepOnAfterEStop enables further processing of the parts program after a programmed EStop.

The function block has the following inputs:

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nGrpId	UDINT	group ID
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in

Output	Data type	Description
		ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

See also:

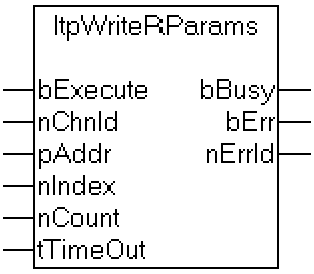
ItpEStop [▶ 176]

ItpIsEStop [▶ 182]

Requirements

Development environment	Target system	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.21 ItpWriteRParams



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpWriteRParamsEx [▶ 157].

Interface

```
VAR_INPUT
bExecute         : BOOL;
nChnId           : UDINT;
pAddr            : DWORD;
nIndex           : DINT;
nCount           : DINT;
tTimeOut         : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy            : BOOL;
bErr             : BOOL;
nErrId           : UDINT;
END_VAR
```

Description

The ItpWriteRParams block writes R-parameters into the NC.

Input	Data type	Description
bExecute	BOOL	A rising edge starts the write operation
nChnId	UDINT	ID of the NC channel whose R-parameters are to be written

Input	Data type	Description
pAddr	DWORD	Address of the variables containing the data to be written. Data are used directly from the specified address, i.e. nIndex is **not** to be interpreted as offset from pAddr. The data are usually in an array of type LREAL, which has to be defined by the user.
nIndex	DINT	Describes the index of the R-parameter to be written from an NC perspective.
nCount	DINT	Number of R-parameters to be written
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Sample

```
VAR
arrfRParam90to99 : ARRAY[0..9] OF LREAL;
fbWriteRParam    : ItpWriteRParams;
n                : INT := 0;
bWriteParam      : BOOL := FALSE;
END_VAR
```

```
FOR n:=0 TO 9 DO
arrfRParam90to99[n] := 90 + n;
END_FOR

fbWriteRParam(
 bExecute := bWriteParam,
 nChnId := 2,
 pAddr := ADR( arrfRParam90to99[0] ),
 nIndex := 90,
 nCount := 10,
 tTimeOut := T#200ms );
```

In this example the parameters R90 to R99 are written from an NC perspective.

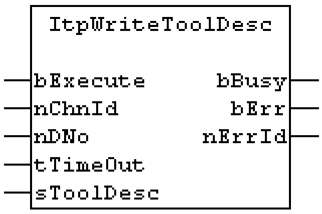
Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.22 ItpWriteToolDesc

from library version 4.0

from TwinCAT V2.7, Build 500



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpWriteToolDescEx [▶ 159].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
nDNo            : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
sToolDesc       : ToolDesc;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

```
TYPE ToolDesc:
STRUCT
nToolNumber     : UDINT; (*valid range from 0 .. 65535*)
nToolType       : UDINT;
fParam          : ARRAY [2..15] OF LREAL;
END_STRUCT
END_TYPE
```

The **ItpWriteToolDesc** block writes a block of tool parameters.

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
nChnId	UDINT	Channel ID
nDNo	UDINT	D-word for which the tool parameters are to be read. nDoNo can assume values between 1 and 50 (the library TcNci.lib from version 5.4.16 supports 255 tool parameters).
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sToolDesc	ToolDesc	The structure that contains the new tool parameters. This structure is only accessed for reading. The meaning of the parameters depends on the tool type, and can be found in the tool data [▸ 83].

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

see also:

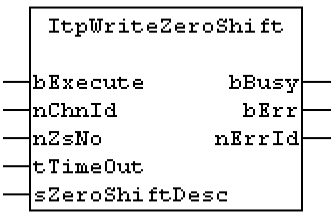
ItpReadToolDesc [▸ 186], ItpSetToolDescNull [▸ 198]

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

5.1.52.23 ItpWriteZeroShift

from library version 4.0



Outdated version

The sole purpose of the function block is to ensure compatibility with existing projects. For new projects please use the function block ItpWriteZeroShiftEx [▶ 161].

Interface

```
VAR_INPUT
bExecute        : BOOL;
nChnId          : UDINT;
nZsNo           : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

```
VAR_IN_OUT
sZeroShiftDesc : ZeroShiftDesc;
END_VAR
```

```
TYPE ZeroShiftDesc:
STRUCT
fShiftX : LREAL;
fShiftY : LREAL;
fShiftZ : LREAL;
END_STRUCT
END_TYPE
```

The **ItpWriteZeroShift** function block reads the offset shift components X, Y and Z for the given zero offset shift.

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input
nChnId	UDINT	Channel ID
nZsNo	UDINT	Number of the zero offset shift on the NC side G54 to G59 are zero offset shifts. G58 and G59 can only be edited from the NC program. The valid range of values for 'nZsNo' is therefore from 54 to 57.
tTimeOut	TIME	ADS Timeout-Delay

Input & output	Data type	Description
sZeroShiftDesc	ZeroShiftDesc	The structure containing the components of the zero offset shift. This structure is only accessed for reading.

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The

Output	Data type	Description
		command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

For reasons of compatibility every zero offset shift that can be set has two parameters (coarse and fine) for each axis. When using this function block to write a new zero offset shift, the new value is written into the 'fine parameter'. A value of 0.0 is entered into the 'coarse parameter'.

This makes it possible to use a function block such as ItpReadZeroShift to read and modify a zero offset shift and to send it back to the NC.

see also:

ItpReadZeroShift [▸ 188]; ItpSetZeroShiftNull [▸ 205]

Requirements

Development Environment	Target System	PLC libraries to include
TwinCAT v2.7.0	PC (i386)	TcNciItp.lib
TwinCAT v2.8.0	PC (i386)	TcNci.lib

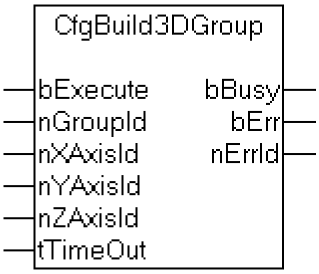
5.2 PLC Library: NC Configuration

The TcNcCfg.lib library provides the function blocks for general NC axis configuration. This makes it possible to configure or to reconfigure axes in a simple way directly from the PLC.

The following function blocks are included in the library TcNcCfg.lib and can be instantiated repeatedly.

Function Block	Description
CfgBuild3DGroup [▸ 208]	Groups up to 3 PTP axes into a 3D group
CfgBuildExt3DGroup [▸ 208]	Groups up to 3 PTP axes and 5 auxiliary axes into a 3D group
CfgAddAxisToGroup [▸ 209]	Configures a single axis at a particular location within a group (PTP, 3D, FIFO)
CfgReconfigGroup [▸ 210]	Removes 3D (or FIFO) axis assignments and returns of the axes to their personal PTP group
CfgReconfigAxis [▸ 210]	Returns a single axis from, for example, a 3D group, to its personal PTP group
CfgRead3DAxisIds [▸ 211]	Reads the axis IDs (axis assignment) of a 3D group
CfgReadExt3DAxisIds [▸ 211]	Reads the axis IDs (axis assignment) of a 3D group with auxiliary axes

5.2.1 CfgBuild3DGroup



This block configures a 3D group with up to 3 PTP axes (X, Y and Z). The IDs of the PTP axes are applied at inputs **nXAxisId**, **nYAxisId** and **nZAxisId**. **nGroupId** contains the ID of the 3D group. The command is executed at a rising edge at input **bExecute**.

The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs.

The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Interface

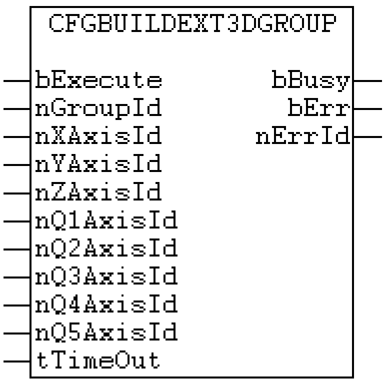
```
VAR_INPUT
bExecute        : BOOL;
nGroupId        : UDINT;
nXAxisId        : UDINT;
nYAxisId        : UDINT;
nZAxisId        : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

5.2.2 CfgBuildExt3DGroup

from TwinCAT 2.8

from TcNcCfg.lib version 3.1



This block configures a 3D group with up to 3 path axes (X, Y and Z). Additionally, up to 5 auxiliary axes (Q1..Q5) can be configured. The axis IDs of the PTP axes that are to be included in the interpolation group are applied at the inputs **nXAxisId** to **nQ5AxisId**.

The assignment of the auxiliary axes must start with **nQ1AxisId**. No gaps between auxiliary axes are permitted. For example, if nQ**3**AxisId is to be assigned, nQ2AxisId must also be assigned a valid Axis ID.

nGroupId contains the ID of the 3D group. The command is executed at a rising edge at input **bExecute**.

The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs.

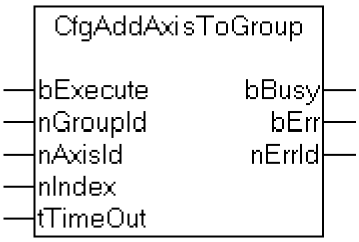
The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGroupId        : UDINT;
nXAxisId        : UDINT;
nYAxisId        : UDINT;
nZAxisId        : UDINT;
nQ1AxisId       : UDINT;
nQ2AxisId       : UDINT;
nQ3AxisId       : UDINT;
nQ4AxisId       : UDINT;
nQ5AxisId       : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

5.2.3 CfgAddAxisToGroup



The CfgAddAxisToGroup block configures a single axis at a particular location within an existing group (PTP, 3D, FIFO). The ID of the target group is specified at the **nGroupId** input. Die ID of the axis that is to be configured is passed to the **nAxisId** input. **nIndex** contains the location of the axis within the group. nIndex can assume values from 0..n-1. Depending on the type of group, n has the following significance:

- PTP: n = 1
- 3D: n = 3
- FIFO: n = 8

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGroupId        : UDINT;
nAxisId         : UDINT;
nIndex          : UDINT;
tTimeOut        : TIME;
END_VAR
```

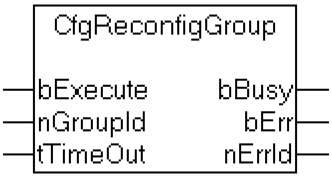
```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

The command is executed with a rising edge at the input **bExecute**.

The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs.

The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

5.2.4 CfgReconfigGroup



Description

The block CfgReconfigGroup removes the axis assignment of an existing group (NCI or FIFO), returning the axes to their personal PTP groups. The ID of the group that is to be removed is supplied to the **nGroupId** input. The command is executed with a rising edge at the input **bExecute**.

The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs.

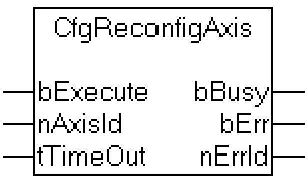
The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGroupId        : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

5.2.5 CfgReconfigAxis



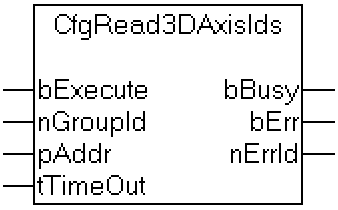
The block CfgReconfigAxis returns a single axis from, for example, a 3D group, to its personal PTP group. To do this, the ID of the axis that is to be returned is supplied to the **nAxisId** input. The command is executed with a rising edge at the input **bExecute**.
The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs.
The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nAxisId         : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

5.2.6 CfgRead3DAxisIds



The block CfgRead3DAxisIds reads the axis configuration of a 3D group. The ID of the 3D group is supplied to input **nGroupId**. The address of the variables into which the block will write the AchsIds of the group assignment is supplied to input **pAddr**. This variable is an array with three elements of type UDINT.

The command is executed with a rising edge at the input **bExecute**.

The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs.

The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGroupId        : UDINT;
pAddr           : DWORD;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

Sample:

```
VAR
(* instance *)
ReadAxIds : CfgRead3DAxisIds;
AxIds : ARRAY[1..3] OF UDINT;
END_VAR
```

```
ReadAxIds( bExecute := TRUE,
nGroupId := 4,
pAddr := ADR( AxIds ),
tTimeOut := T#1s );
```

AxIds now contains the three axis IDs for the 3D group with the group ID 4.

5.2.7 CfgReadExt3DAxisIds

TwinCAT Version 2.8, Build 723 and higher from TcNcCfg.lib version 3.2

```
        CFGREADEXT3DAXISIDS
 bExecute                    bBusy
 nGroupId                     bErr
 tTimeOut                   nErrId
 stExt3dGroup         stExt3dGroup
```

The function block CfgReadExt3DAxisIds reads the axis configuration of the extended 3D group. The ID of the 3D group is supplied to input **nGroupId**. An instance of the structure NCI_EXT3DGROUP is applied at input & output stExt3dGroup. The axis IDs of the current interpolation group are entered in this structure.

The command is executed with a rising edge at the input **bExecute**.

The bBusy output remains TRUE until the block has executed a command, although only for as long as the time specified at the tTimeOut input. While bBusy = TRUE, no new instruction will be accepted at the inputs.

The output bErr goes TRUE if an error occurs as the command is being executed. The command-specific error code is contained in nErrId. The outputs are reset by carrying out a command at the inputs.

Interface

```
VAR_INPUT
bExecute        : BOOL;
nGroupId        : UDINT;
tTimeOut        : TIME;
END_VAR
```

```
VAR_IN_OUT
stExt3dGroup    : NCI_EXT3DGROUP;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

```
TYPE NCI_EXT3DGROUP :
STRUCT
nXAxisId        : UDINT;
nYAxisId        : UDINT;
nZAxisId        : UDINT;
nQ1AxisId       : UDINT;
nQ2AxisId       : UDINT;
nQ3AxisId       : UDINT;
nQ4AxisId       : UDINT;
nQ5AxisId       : UDINT;
END_STRUCT
END_TYPE
```

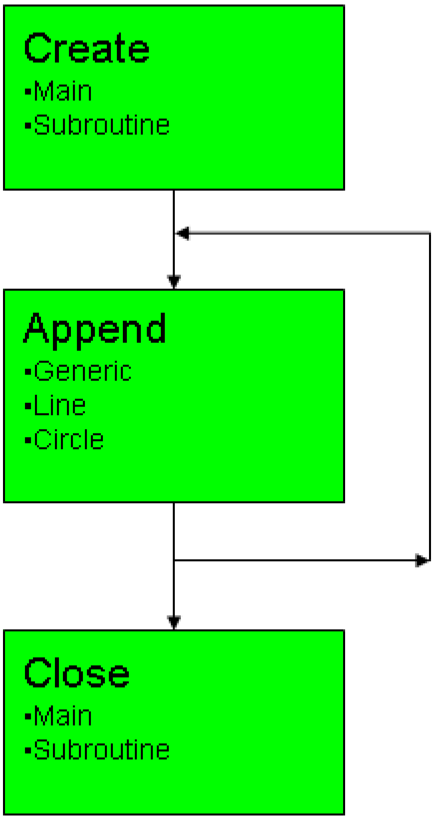
5.3 PLC Library: NCI Utilities

The function blocks ItpPpg* provide an option for creating a parts program (G-Code file) from the PLC. During program generation a distinction is made between a main program (ItpPpgCreateMain [▸ 220]) and a subroutine (ItpPpgCreateSubroutine [▸ 222]).

Subsequently ItpPpgAppend* can be used to add various NC lines. The following function blocks are available:

- ItpPpgAppendGeoLine [▸ 217] adds a linear motion.
- ItpPpgAppendGeoCircleByRadius [▸ 215] adds a circle with radius specification.
- ItpPpgAppendGenericBlock [▸ 214] inserts a self-defined line, such as activation of rounding or M-functions.

Once the parts program is complete, it is closed with the routines ItpPpgCloseMain [▸ 218] or ItpPpgCloseSubroutine [▸ 219].



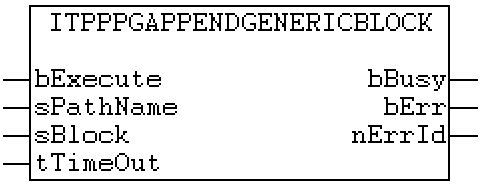
The following function blocks can be used:

Function Block	Description
ItpPpgAppendGenericBlock [▶ 214]	Appends a generic NC line to a specified parts program
ItpPpgAppendGeoCircleByRadius [▶ 215]	Adds a circle to a specified parts program
ItpPpgAppendGeoLine [▶ 217]	Adds a linear motion to a specified parts program
ItpPpgCloseMain [▶ 218]	Closes a previously opened parts program
ItpPpgCloseSubroutine [▶ 219]	Closes a previously opened subroutine
ItpPpgCreateMain [▶ 220]	Opens or generates a parts program
ItpPpgCreateSubroutine [▶ 222]	Opens or generates a subroutine

Prerequisites

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0 Build >746 TwinCAT v2.9.0 Build >947	PC (i386)	TcNciUtilities.lib

5.3.1 ItpPpgAppendGenericBlock



The function block ItpPpgAppendGenericBlock adds a generic line to the parts program. It can be used to activate an M-function or rounding, for example.

Before it is called, ItpPpgCreateMain [▸ 220] or ItpPpgCreateSubroutine [▸ 222] should be called.

The function block ItpPpgAppendGenericBlock has the following inputs:

Interface

```
VAR_INPUT
    bExecute        : BOOL;
    sPathName       : STRING;
    sBlock          : STRING;
    tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
    bBusy           : BOOL;
    bErr            : BOOL;
    nErrId          : UDINT;
END_VAR
```

Description

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPathName	STRING	Name of the parts program including path name
sBlock	STRING	Generic line to be added to the parts program
tTimeOut	TIME	ADS Timeout-Delay

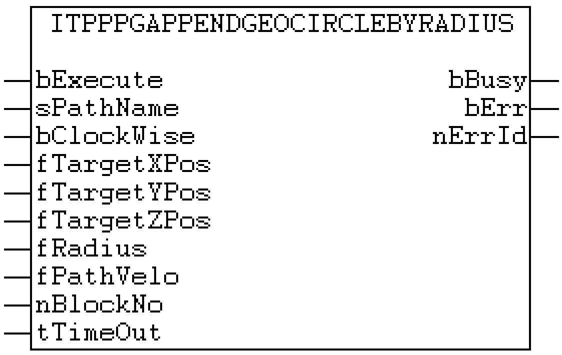
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.

Output	Data type	Description
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target system	PLC LIbraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0 Build > 746 TwinCAT v2.9.9 Build > 947	PC (i386)	TcNciUtilities.lib

5.3.2 ItpPpgAppendGeoCircleByRadius



The function block ItpPpgAppendGeoCircleByRadius adds a circular motion to the parts program. The circle is parameterized by the radius.

Before it is called, ItpPpgCreateMain [▸ 220] or ItpPpgCreateSubroutine [▸ 222] should be called.

The function block ItpPpgAppendGeoCircleByRadius has the following inputs:

Interface

```
VAR_INPUT
    bExecute        : BOOL;
    sPathName       : STRING;
    bClockWise      : BOOL;
    fTargetXPos     : LREAL;
    fTargetYPos     : LREAL;
    fTargetZPos     : LREAL;
    fRadius         : LREAL;
    fPathVelo       : LREAL;
    nBlockNo        : UDINT;
    tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
    bBusy           : BOOL;
    bErr            : BOOL;
    nErrId          : UDINT;
END_VAR
```

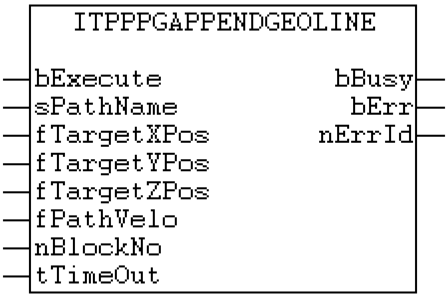
Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPathName	STRING	Name of the parts program including path name
bClockwise	BOOL	If TRUE, the movement along the circle is clockwise, otherwise counter-clockwise
fTargetXPos	LREAL	Target position of the X axis
fTargetYPos	LREAL	Target position of the Y axis
fTargetZPos	LREAL	Target position of the Z axis
fRadius	LREAL	Circle radius
fPathVelo	LREAL	Path velocity
nBlockNo	UDINT	Line number in the parts program
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target system	PLC LIbraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0 Build > 746 TwinCAT v2.9.9 Build > 947	PC (i386)	TcNciUtilities.lib

5.3.3 ItpPpgAppendGeoLine



The function block ItpPpgAppendGeoLine adds a linear motion to the parts program. In addition to the actual target position, the path velocity and the line number are transferred.

Before it is called, ItpPpgCreateMain [▶ 220] or ItpPpgCreateSubroutine [▶ 222] should be called.

The function block ItpPpgAppendGeoLine has the following inputs:

Interface

```
VAR_INPUT
bExecute        : BOOL;
sPathName       : STRING;
fTargetXPos     : LREAL;
fTargetYPos     : LREAL;
fTargetZPos     : LREAL;
fPathVelo       : LREAL;
nBlockNo        : UDINT;
tTimeOut        : TIME;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPathName	STRING	Name of the parts program including path name
fTargetXPos	LREAL	Target position of the X axis
fTargetYPos	LREAL	Target position of the Y axis
fTargetZPos	LREAL	Target position of the Z axis
fPathVelo	LREAL	Path velocity
nBlockNo	UDINT	Line number in the parts program
tTimeOut	TIME	ADS Timeout-Delay

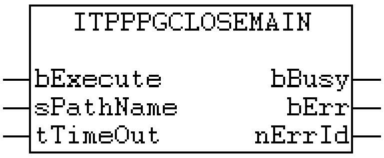
Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is

Output	Data type	Description
		contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target system	PLC LIbraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0 Build > 746 TwinCAT v2.9.9 Build > 947	PC (i386)	TcNciUtilities.lib

5.3.4 ItpPpgCloseMain



The function block ItpPpgCloseMain completes the main program with the corresponding code for the interpreter (M02).

Before it is called, ItpPpgCreateMain [▸ 220] should be called.

The function block has the following inputs:

Interface

```
VAR_INPUT
bExecute        : BOOL;
sPathName       : STRING;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

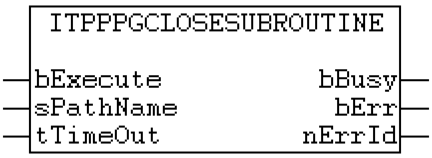
Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPathName	STRING	Name of the parts program including path name
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Requirements

Development Environment	Target system	PLC LIbraries to include
TwinCAT v2.7.0	PC (i386)	not implemented
TwinCAT v2.8.0 Build > 746 TwinCAT v2.9.9 Build > 947	PC (i386)	TcNciUtilities.lib

5.3.5 ItpPpgCloseSubroutine



The function block ItpPpgCloseSubroutine completes the subroutine with the corresponding code for the interpreter (M17).

Before it is called, ItpPpgCreateSubroutine [▸ 222] should be called.

The function block has the following inputs:

Interface

```
VAR_INPUT
bExecute        : BOOL;
sPathName       : STRING;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

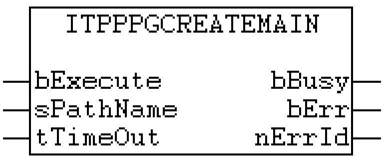
Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPathName	STRING	Name of the parts program including path name
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Prerequisites

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.8.0 Build >746 TwinCAT v2.9.0 Build >947	PC (i386)	TcNciUtilities.lib

5.3.6 ItpPpgCreateMain



The function block ItpPpgCreateMain generates a new file, which can later be processed as main program. If the file does not yet exist, it is created, otherwise it is overwritten.

The function block has the following inputs:

Interface

```
VAR_INPUT
bExecute        : BOOL;
sPathName       : STRING;
tTimeOut        : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy           : BOOL;
bErr            : BOOL;
nErrId          : UDINT;
END_VAR
```

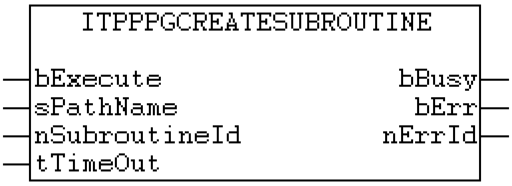
Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPathName	STRING	Name of the parts program including path name
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrId'. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Prerequisites

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.8.0 Build >746 TwinCAT v2.9.0 Build >947	PC (i386)	TcNciUtilities.lib

5.3.7 ItpPpgCreateSubroutine



The function block ItpPpgCreateSubroutine generates a new file, which can later be processed as subroutine. If the file does not yet exist, it is created, otherwise it is overwritten.

The function block has the following inputs:

Interface

```
VAR_INPUT
bExecute       : BOOL;
sPathName      : STRING;
nSubroutineId: UDINT;
tTimeOut       : TIME;
END_VAR
```

```
VAR_OUTPUT
bBusy          : BOOL;
bErr           : BOOL;
nErrId         : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
sPathName	STRING	Name of the subroutine including path name
nSubroutineId	UDINT	Number of the subroutine
tTimeOut	TIME	ADS Timeout-Delay

Output	Data type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.
bErr	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’. Is reset to FALSE by the execution of a command at the inputs.
nErrId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS

Output	Data type	Description
		error documentation [▸ 253] or in the NC error documentation (error codes above 0x4000).

Prerequisites

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.8.0 Build >746 TwinCAT v2.9.0 Build >947	PC (i386)	TcNciUtilities.lib

5.4 PLC Library: TcPlcInterpolation

The TcPlcInterpolation library offers an alternative to the application of G-Code (DIN 66025). This library can be used to execute interpolated motion commands directly from the PLC, without using G-Code.

In a first step a table of different movement commands and additional functions is written. To this end structures such as ST_NciGeoLine are transferred to the FB NciFeedTablePreparation. This appends the movement command to the table. Once the table is full or all required entries have been added, NciFeedTable is called in order to transfer the table content to the NC kernel. The data transfer directly starts the execution.

Function blocks required for grouping of axes or channel control (channel override) are located in the libraries ‚TcNcCfg.lib' and ‚TcNci.lib'.

Function Block	Description
FB_NciFeedTablePreparation [▸ 224]	Fills a table with NCI movements in the PLC
FB_NciFeedTable [▸ 226]	Transfers a previously written table to the NC kernel and starts the motion

The following structures can be used as input parameters for the function block NciFeedTablePreparation:

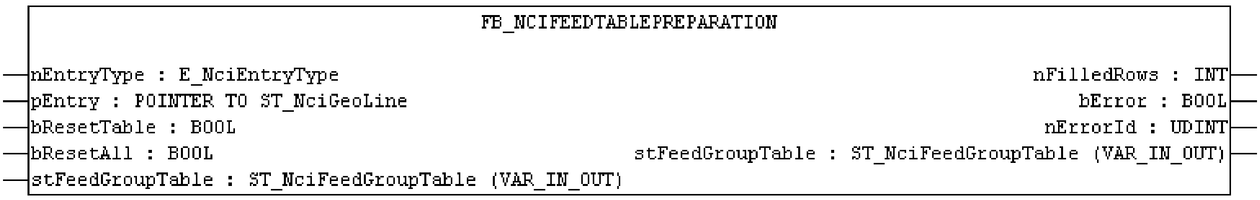
Structures	Enum	Description
	Organization	
	E_NciEntryTypeNone	No function
ST_NciGeoStart [▸ 228]	E_NciEntryTypeGeoStart	Sets the starting position for the first geometry entry
ST_NciEndOfTables [▸ 238]	E_NciEntryTypeEndOfTables	Indicates the end of the geometry table
	Movement commands	
ST_NciGeoLine [▸ 229]	E_NciEntryTypeGeoLine	Describes a straight line
ST_NciGeoCirclePlane [▸ 229]	E_NciEntryTypeGeoCirclePlane	Describes a circle in the main plane (center point programming)
ST_NciGeoCircleCIP [▸ 230]	E_NciEntryTypeGeoCircleCIP	Describes a circle anywhere in the space
ST_NciGeoBezier3 [▸ 231]	E_NciEntryTypeGeoBezier3	Describes a 3rd order Bezier with control points
ST_NciGeoBezier5 [▸ 232]	E_NciEntryTypeGeoBezier5	Describes a 5th order Bezier with control points
ST_NciDwellTime [▸ 237]	E_NciEntryTypeDwellTime	Describes a dwell time
	Path parameters	
ST_NciBaseFrame [▸ 236]	E_NciEntryTypeBaseFrame	Describes a zero shift and rotation
ST_NciVertexSmoothing [▸ 235]	E_NciEntryTypeVertexSmoothing	Activates blending at segment transitions
ST_NciTangentialFollowingDesc [▸ 238]	E_NciEntryTypeTfDesc	Activates tangential following of the tool

Structures	Enum	Description
	Dynamics	
ST_NciDynOvr [▶ 235]	E_NciEntryTypeDynOvr	Modifies the dynamic override
ST_NciAxisDynamics [▶ 237]	E_NciEntryTypeAxisDynamics	Limits the axis dynamics
ST_NciPathDynamics [▶ 236]	E_NciEntryTypePathDynamics	Limits the path dynamics
	Parameter commands	
ST_NciHParam [▶ 234]	E_NciEntryTypeHParam	Sets an H-parameter (DINT)
ST_NciSParam [▶ 234]	E_NciEntryTypeSParam	Sets an S-parameter (WORD)
ST_NciTParam [▶ 235]	E_NciEntryTypeTParam	Sets a T-parameter (WORD)
ST_NciMFuncFast [▶ 233]	E_NciEntryTypeMFuncFast	Parameterizes a fast M-function (no handshake)
ST_NciMFuncHsk [▶ 233]	E_NciEntryTypeMFuncHsk	Parameterizes an M-function with handshake
ST_NciMFuncResetAllFast [▶ 234]	E_NciEntryTypeResetAllFast	Resets all fast M-functions

Prerequisites

Development environment	Target system	PLC libraries to include
TwinCAT v2.10	PC (i386)	TcPlcInterpolation.lib

5.4.1 FB_NciFeedTablePreparation



The function block **FB_NciFeedTablePreparation** appends an entry of a specific type to the feed table (stFeedGropupTable). An appended entry can generate more than one row in the table. If the table has not enough free rows, an error is returned and no entry is added to the table. In this case the entry either has to be added to another table or to the same table, after FB_NciFeedTable was executed. This function block deals with modal functions, such as tangential following. It is therefore important to always use the same instance of this function block. The function block can be called repeatedly in a PLC cycle.

Interface

```
VAR_INPUT
nEntryType       : E_NciEntryType;
pEntry           : POINTER TO UDINT;
bResetTable      : BOOL;
bResetAll        : BOOL;
END_VAR
```

```
VAR_IN_OUT
stFeedGroupTable : ST_NciFeedGroupTable
END_VAR
```

```
VAR_OUTPUT
nFilledRows      : INT;
bError           : BOOL;
nErrorId         : UDINT;
END_VAR
```

Table 8: VAR_INPUT

Input	Data type	Description
nEntryType	E_NciEntryType	Specifies the entry type, e.g. line, circle, tangential following
pEntry	POINTER	Pointer to entry structure – **must match nEntryType**

Input	Data type	Description
bResetTable	BOOL	If bResetTable is TRUE, the table ‚stFeedGroupTable' is set to zero and nFilledRows is also set to zero. If nErrorId = ErrNciFeedTableFull, this error is reset. All modal flags (such as tangential following) remain constant.
bResetAll	BOOL	Like bResetTable. In addition, all modal flags are set to their default values, and all error IDs are reset.

Table 9: VAR_IN_OUT

Input/output	Data type	Description
stFeedGroupTable	ST_NciFeedGroupTable	Table containing the rows for the NC kernel.

Table 10: VAR_OUTPUT

Output	Data type	Description
nFilledRows	INT	Number of filled rows.
bError	BOOL	Becomes TRUE as soon as an error has occurred.
nErrorId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▶ 253] or in the NC error documentation (error codes above 0x4000).

If bResetTable, bResetAll, or bError is true, no further entries are accepted.

The error code 0x4B72 indicates that the table is full and the last entry was not accepted.

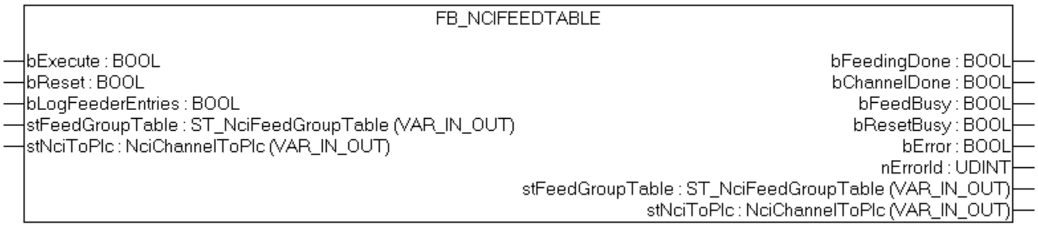
Sample:

```
stGeoLine.nDisplayIndex := 1;
stGeoLine.fEndPosX := 0;
stGeoLine.fEndPosY := 400;
stGeoLine.fEndPosZ := 100;
stGeoLine.fEndPosQ1 :=-90;
stGeoLine.fVelo := 1000; (*mm per sec*)

fbFeedTablePrep(
nEntryType := E_NciEntryTypeGeoLine,
pEntry := ADR(stGeoLine),
bResetTable:= FALSE,
stFeedGroupTable:= stNciFeedGroupTable,
nFilledRows=> nFilledRows,
bError => bError,
nErrorId => nErrorId);
```

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcPlcInterpolation.lib

5.4.2 FB_NciFeedTable



The function block **FB_NciFeedTable** transfers a given table to the NC kernel. If the override is set and the approvals are enabled, execution is started immediately. bFeedingDone becomes TRUE when the transfer is complete. This signal can be used for overwriting the table with NciFeedTablePreparation [▶ 224]. In NciFeedTablePreparation the table first has to be reset.

bChannelDone indicates complete execution of the tables in the NC kernel. The identifier ST_NciEndOfTables [▶ 227] must therefore be placed at the end of the last table.

Interface

```
VAR_INPUT
bExecute          : BOOL;
bReset            : BOOL;
bLogFeederEntries : BOOL;
END_VAR
```

```
VAR_IN_OUT
stFeedGroupTable  : ST_NciFeedGroupTable;
stNciToPlc        : NciChannelToPlc;
END_VAR
```

```
VAR_OUTPUT
bFeedingDone      : BOOL;
bChannelDone      : BOOL;
bFeedBusy:        : BOOL;
bResetBusy:       : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
END_VAR
```

Input	Data type	Description
bExecute	BOOL	The command is triggered by a rising edge at this input.
bReset	BOOL	Triggers a channel reset and also resets the function block
bLogFeederEntries	BOOL	If TRUE, a log file ‚PlcItpFeed.log' is written in the TwinCAT\CNC folder. It contains all entries that are sent to the NC kernel via ADS. **If bLogFeederEntries = TRUE, more time is required until bFeedingDone becomes TRUE.**

Input/output	Data type	Description
stFeedGroupTable	ST_NciFeedGroupTable	Table containing the rows for the NC kernel.
stNciToPlc	NciChannelToPlc	The structure of the cyclic channel interface between NCI and PLC.

Output	Data type	Description
bFeedingDone	BOOL	Becomes TRUE once all table rows have been sent to the NC kernel.
bChannelDone	BOOL	Becomes TRUE once all entries of the table in the NC kernel were executed and ST_NciEndOfTables was detected.

Output	Data type	Description
bFeedBusy	BOOL	Becomes TRUE when the function block sends entries to the NC kernel.
bResetBusy	BOOL	Becomes TRUE when a reset is executed.
bError	BOOL	Becomes TRUE as soon as an error has occurred.
nErrorId	UDINT	Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. The error numbers in ErrId can be looked up in the ADS error documentation [▸ 253] or in the NC error documentation.

Requirements

Development Environment	Target System	PLC Libraries to include
TwinCAT v2.10.0	PC (i386)	TcPlcInterpolation.lib

5.4.3 Types and Enums

E_NciEntryType

```
TYPE E_NciEntryType :
(
E_NciEntryTypeNone := 0,
E_NciEntryTypeGeoStart := 1,
E_NciEntryTypeGeoLine := 2,
E_NciEntryTypeGeoCirclePlane := 3,
E_NciEntryTypeGeoCircleCIP := 4,
E_NciEntryTypeGeoBezier3 := 10,
E_NciEntryTypeGeoBezier5 := 11,
E_NciEntryTypeMFuncHsk := 20,
E_NciEntryTypeMFuncFast := 21,
E_NciEntryTypeMFuncResetAllFast := 23,
E_NciEntryTypeHParam := 24,
E_NciEntryTypeSParam := 25,
E_NciEntryTypeTParam := 26,
E_NciEntryTypeDynOvr := 50,
E_NciEntryTypeVertexSmoothing := 51,
E_NciEntryTypeBaseFrame := 52,
E_NciEntryTypePathDynamics := 53,
E_NciEntryTypeAxisDynamics := 55,
E_NciEntryTypeDwellTime := 56,
E_NciEntryTypeTfDesc := 100,
E_NciEntryTypeEndOfTables := 1000
);
END_TYPE
```

Structures	Enum	Description
	Organization	
	E_NciEntryTypeNone	No function
ST_NciGeoStart [▸ 228]	E_NciEntryTypeGeoStart	Sets the starting position for the first geometry entry
ST_NciEndOfTables [▸ 238]	E_NciEntryTypeEndOfTables	Indicates the end of the geometry table
	Movement commands	
ST_NciGeoLine [▸ 229]	E_NciEntryTypeGeoLine	Describes a straight line
ST_NciGeoCirclePlane [▸ 229]	E_NciEntryTypeGeoCirclePlane	Describes a circle in the main plane (center point programming)
ST_NciGeoCircleCIP [▸ 230]	E_NciEntryTypeGeoCircleCIP	Describes a circle anywhere in the space
ST_NciGeoBezier3 [▸ 231]	E_NciEntryTypeGeoBezier3	Describes a 3rd order Bezier with control points

Structures	Enum	Description
ST_NciGeoBezier5 [▶ 232]	E_NciEntryTypeGeoBezier5	Describes a 5th order Bezier with control points
ST_NciDwellTime [▶ 237]	E_NciEntryTypeDwellTime	Describes a dwell time
	Path parameters	
ST_NciBaseFrame [▶ 236]	E_NciEntryTypeBaseFrame	Describes a zero shift and rotation
ST_NciVertexSmoothing [▶ 235]	E_NciEntryTypeVertexSmoothing	Activates blending at segment transitions
ST_NciTangentialFollowingDesc [▶ 238]	E_NciEntryTypeTfDesc	Activates tangential following of the tool
	Dynamics	
ST_NciDynOvr [▶ 235]	E_NciEntryTypeDynOvr	Modifies the dynamic override
ST_NciAxisDynamics [▶ 237]	E_NciEntryTypeAxisDynamics	Limits the axis dynamics
ST_NciPathDynamics [▶ 236]	E_NciEntryTypePathDynamics	Limits the path dynamics
	Parameter commands	
ST_NciHParam [▶ 234]	E_NciEntryTypeHParam	Sets an H-parameter (DINT)
ST_NciSParam [▶ 234]	E_NciEntryTypeSParam	Sets an S-parameter (WORD)
ST_NciTParam [▶ 235]	E_NciEntryTypeTParam	Sets a T-parameter (WORD)
ST_NciMFuncFast [▶ 233]	E_NciEntryTypeMFuncFast	Parameterizes a fast M-function (no handshake)
ST_NciMFuncHsk [▶ 233]	E_NciEntryTypeMFuncHsk	Parameterizes an M-function with handshake
ST_NciMFuncResetAllFast [▶ 234]	E_NciEntryTypeResetAllFast	Resets all fast M-functions

ST_NciGeoStart

```
TYPE ST_NciGeoStart :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeGeoStart; (*do not override this parameter *)
fPosX: LREAL;
fPosY: LREAL;
fPosZ: LREAL;
fPosQ1: LREAL;
fPosQ2: LREAL;
fPosQ3: LREAL;
fPosQ4: LREAL;
fPosQ5: LREAL;
END_STRUCT
END_TYPE
```

Sets the start position for the first geometry entry. This is necessary, if the first geometry entry is a circle or if tangential following in the first segment is ON. This structure can optionally be written at each start of the first table.

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
fPosX	LREAL	Start position X
fPosY	LREAL	Start position Y
fPosZ	LREAL	Start position Z
fPosQ1	LREAL	Start position Q1
fPosQ2	LREAL	Start position Q2
fPosQ3	LREAL	Start position Q3
fPosQ4	LREAL	Start position Q4
fPosQ5	LREAL	Start position Q5

ST_NciGeoLine

```
TYPE ST_NciGeoLine :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeGeoLine; (*do not override this parameter *)
nDisplayIndex: UDINT;
fEndPosX: LREAL;
fEndPosY: LREAL;
fEndPosZ: LREAL;
fEndPosQ1: LREAL;
fEndPosQ2: LREAL;
fEndPosQ3: LREAL;
fEndPosQ4: LREAL;
fEndPosQ5: LREAL;
fVelo: LREAL;
bRapidTraverse: BOOL;
bAccurateStop: BOOL; (* VeloEnd := 0 *)
END_STRUCT
END_TYPE
```

Describes a straight line with specified velocity.

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fEndPosX	LREAL	Target position X
fEndPosY	LREAL	Target position Y
fEndPosZ	LREAL	Target position Z
fEndPosQ1	LREAL	Target position Q1
fEndPosQ2	LREAL	Target position Q2
fEndPosQ3	LREAL	Target position Q3
fEndPosQ4	LREAL	Target position Q4
fEndPosQ5	LREAL	Target position Q5
fVelo	LREAL	Target path velocity, like F in G-Code, but in basic units per second (e.g. mm/s)
bRapidTraverse	BOOL	TRUE has the same effect as G0 FALSE treats this entry like G01
bAccurateStop	BOOL	Accurate stop [▸ 45] (TRUE has the same effect as G09)

ST_NciGeoCirclePlane

```
TYPE ST_NciGeoCirclePlane :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeGeoCirclePlane; (*do not override this parameter *)
nDisplayIndex: UDINT;
fEndPosX: LREAL;
fEndPosY: LREAL;
fEndPosZ: LREAL;
fCenterX: LREAL;
fCenterY: LREAL;
fCenterZ: LREAL;
fEndPosQ1: LREAL;
fEndPosQ2: LREAL;
fEndPosQ3: LREAL;
fEndPosQ4: LREAL;
fEndPosQ5: LREAL;
fVelo: LREAL;
bClockwise: BOOL;
bAccurateStop: BOOL; (* VeloEnd := 0 *)
nPlane: E_NciGeoPlane := E_NciGeoPlaneXY;
END_STRUCT
END_TYPE
```

Describes a circle in the principal plane. The center point is specified in absolute coordinates.

From TC2.11 R3 B2243 (TcPlcInterpolation.lib V1.1.39)

The orthogonal component at the center is assigned internally. If a circle is programmed in the XY plane, for example, ‚fCenterZ' is assigned internally. If the user has assigned the value explicitly, the value is nevertheless overwritten by the FB.

A helix can be described by programming the height. If helix is programmed in the XY plane, for example, the lifting height of the helix is specified absolutely with ‚fEndPosZ'.

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fEndPosX	LREAL	Target position X
fEndPosY	LREAL	Target position Y
fEndPosZ	LREAL	Target position Z
fCenterX	LREAL	Center position X in absolute coordinates
fCenterY	LREAL	Center position Y in absolute coordinates
fCenterZ	LREAL	Center position Z in absolute coordinates
fEndPosQ1	LREAL	Target position Q1
fEndPosQ2	LREAL	Target position Q2
fEndPosQ3	LREAL	Target position Q3
fEndPosQ4	LREAL	Target position Q4
fEndPosQ5	LREAL	Target position Q5
fVelo	LREAL	Target path velocity in basic units per second (e.g. mm/s), like F in G-Code
bClockwise	BOOL	If TRUE, the circle is drawn clockwise, otherwise counter-clockwise (similar to G02, G03)
bAccurateStop	BOOL	Accurate stop [▸ 45] (TRUE has the same effect as G09)
nPlane	E_NciGeoPlane [▸ 230]	Specifies the layer: XY, YZ, or ZX (similar to G17..G19)

Circle segment as start segment

If the first geometry segment is a circle, the start position has to be set with ST_NciGeoStart [▸ 228].

E_NciGeoPlane

```
TYPE E_NciGeoPlane :
(
E_NciGeoPlaneXY := 17,
E_NciGeoPlaneZX := 18,
E_NciGeoPlaneYZ := 19
);
END_TYPE
```

ST_NciGeoCircleCIP

from library version 1.1.34

```
TYPE ST_NciGeoCircleCIP :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeGeoCircleCIP; (* do not overwrite this parameter *)
nDisplayIndex: UDINT;
fEndPosX:      LREAL;
fEndPosY:      LREAL;
fEndPosZ:      LREAL;
fCIPPosX:      LREAL;
fCIPPosY:      LREAL;
fCIPPosZ:      LREAL;
fEndPosQ1:     LREAL;
fEndPosQ2:     LREAL;
fEndPosQ3:     LREAL;
```

```
fEndPosQ4:      LREAL;
fEndPosQ5:      LREAL;
fVelo:          LREAL;
bAccurateStop: BOOL; (* VeloEnd := 0 *)
END_STRUCT
END_TYPE
```

The CIP circle can be used to describe a circle anywhere in space. It does not have to be in the main plane. In order for the circle to be described unambiguously, not all 3 points (the starting point is specified implicitly) may lie on straight line. It is thus not possible to program a complete circle in this way.

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fCIPPosX	LREAL	X position in absolute coordinates (point on circular path)
fCIPPosY	LREAL	Y position in absolute coordinates (point on circular path)
fCIPPosZ	LREAL	Z position in absolute coordinates (point on circular path)
fEndPosX	LREAL	Target position X
fEndPosY	LREAL	Target position Y
fEndPosZ	LREAL	Target position Z
fEndPosQ1	LREAL	Target position Q1
fEndPosQ2	LREAL	Target position Q2
fEndPosQ3	LREAL	Target position Q3
fEndPosQ4	LREAL	Target position Q4
fEndPosQ5	LREAL	Target position Q5
fVelo	LREAL	Target path velocity in basic units per second (e.g. mm/s), like F in G-Code
bAccurateStop	BOOL	Accurate stop [▶ 45] (TRUE has the same effect as G09)

Circle segment as start segment

If the first geometry segment is a circle, the starting position has to be set with ST_NciGeoStart [▶ 228].

ST_NciGeoBezier3

```
TYPE ST_NciGeoBezier3:
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeGeoBezier3; (*do not override this parameter *)
nDisplayIndex: UDINT;
fControlPoint1X: LREAL;
fControlPoint1Y: LREAL;
fControlPoint1Z: LREAL;
fControlPoint2X: LREAL;
fControlPoint2Y: LREAL;
fControlPoint2Z: LREAL;
fEndPosX: LREAL;
fEndPosY: LREAL;
fEndPosZ: LREAL;
fEndPosQ1: LREAL;
fEndPosQ2: LREAL;
fEndPosQ3: LREAL;
fEndPosQ4: LREAL;
fEndPosQ5: LREAL;
fVelo: LREAL;
bAccurateStop: BOOL; (* VeloEnd := 0 *)
END_STRUCT
END_TYPE
```

Describes a third order Bezier with the aid of control points. The start position results from the previous segment. The third control point is determined by the target position.

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fControlPoint1X	LREAL	X component control point 1
fControlPoint1Y	LREAL	Y component control point 1
...		
fControlPoint2Z	LREAL	Z component control point 2
fEndPosX	LREAL	Target position X
fEndPosY	LREAL	Target position Y
fEndPosZ	LREAL	Target position Z
fEndPosQ1	LREAL	Target position Q1
fEndPosQ2	LREAL	Target position Q2
fEndPosQ3	LREAL	Target position Q3
fEndPosQ4	LREAL	Target position Q4
fEndPosQ5	LREAL	Target position Q5
fVelo	LREAL	Target path velocity in basic units per second (e.g. mm/s), like F in G-Code
bAccurateStop	BOOL	Accurate stop [▶ 45] (TRUE has the same effect as G09)

ST_NciGeoBezier5

from library version 1.1.31

```
TYPE ST_NciGeoBezier5:
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeGeoBezier5; (*do not override this parameter *)
nDisplayIndex: UDINT;
fControlPoint1X: LREAL;
fControlPoint1Y: LREAL;
fControlPoint1Z: LREAL;
fControlPoint2X: LREAL;
fControlPoint2Y: LREAL;
fControlPoint2Z: LREAL;
fControlPoint3X: LREAL;
fControlPoint3Y: LREAL;
fControlPoint3Z: LREAL;
fControlPoint4X: LREAL;
fControlPoint4Y: LREAL;
fControlPoint4Z: LREAL;
fEndPosX: LREAL;
fEndPosY: LREAL;
fEndPosZ: LREAL;
fEndPosQ1: LREAL;
fEndPosQ2: LREAL;
fEndPosQ3: LREAL;
fEndPosQ4: LREAL;
fEndPosQ5: LREAL;
fVelo: LREAL;
bAccurateStop: BOOL; (* VeloEnd := 0 *)
END_STRUCT
END_TYPE
```

Describes a 5^{th} order Bezier with the aid of control points. The starting position results from the previous segment. The fifth control point is determined by the target position.

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fControlPoint1X	LREAL	X component control point 1
fControlPoint1Y	LREAL	Y component control point 1
...		

Input	Data type	Description
fControlPoint4Z	LREAL	Z component control point 4
fEndPosX	LREAL	Target position X
fEndPosY	LREAL	Target position Y
fEndPosZ	LREAL	Target position Z
fEndPosQ1	LREAL	Target position Q1
fEndPosQ2	LREAL	Target position Q2
fEndPosQ3	LREAL	Target position Q3
fEndPosQ4	LREAL	Target position Q4
fEndPosQ5	LREAL	Target position Q5
fVelo	LREAL	Target path velocity in basic units per second (e.g. mm/s), like F in G-Code
bAccurateStop	BOOL	Accurate stop [▶ 45] (TRUE has the same effect as G09)

ST_NciMFuncHsk

```
TYPE ST_NciMFuncHsk :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeMFuncHsk; (*do not override this parameter *)
nDisplayIndex: UDINT;
nMFunc: INT;
END_STRUCT
END_TYPE
```

Describes an M-function [▶ 67] of type handshake. The M-function number is between 0 and 159

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
nMFunc	INT	M-function number (0..159)

M-functions in the PlcInterpolation library

If M-functions are used in the PlcInterpolation library, they do not have to be entered in the user interface of the System Manager. An M-function always takes effect at the programmed location.

ST_NciMFuncFast

```
TYPE ST_NciMFuncFast :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeMFuncFast; (*do not override this parameter *)
nDisplayIndex: UDINT;
nMFuncIn0: INT;
nMFuncIn1: INT;
nMFuncIn2: INT;
nMFuncIn3: INT;
nMFuncIn4: INT;
nMFuncIn5: INT;
nMFuncIn6: INT;
nMFuncIn7: INT;
END_STRUCT
END_TYPE
```

Parameterizes up to 8 fast M-functions [▶ 67]. The first M-function must be assigned nMFuncIn0, the second nMFuncIn1 etc. -1 indicates the end of the assignments.

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
nMFuncIn0	INT	Fast M-function number (0..159)

Input	Data type	Description
nMFuncIn1	INT	Fast M-function number (0..159) -1 indicates the end of the list.
nMFuncIn2	INT	Fast M-function number (0..159) -1 indicates the end of the list.
nMFuncIn3	INT	Fast M-function number (0..159) -1 indicates the end of the list.
nMFuncIn4	INT	Fast M-function number (0..159) -1 indicates the end of the list.
nMFuncIn5	INT	Fast M-function number (0..159) -1 indicates the end of the list.
nMFuncIn6	INT	Fast M-function number (0..159) -1 indicates the end of the list.
nMFuncIn7	INT	Fast M-function number (0..159) -1 indicates the end of the list.

M-functions in the PlcInterpolation library

If M-functions are used in the PlcInterpolation library, they do not have to be entered in the user interface of the System Manager. An M-function always takes effect at the programmed location.

ST_NciMFuncResetAllFast

```
TYPE ST_NciMFuncResetAllFast :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeMFuncResetAllFast; (*do not override this parameter *)
nDisplayIndex: UDINT;
END_STRUCT
END_TYPE
```

Resets all fast M-functions [▶ 67].

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code

ST_NciHParam

```
TYPE ST_NciHParam :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeHParam; (*do not override this parameter *)
nDisplayIndex: UDINT;
nHParam: UDINT;
END_STRUCT
END_TYPE
```

Sets an H-parameter [▶ 71] in the cyclic channel interface.

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
nHParam	UDINT	H-parameter from NC to PLC

ST_NciSParam

```
TYPE ST_NciSParam :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeSParam; (*do not override this parameter *)
nDisplayIndex: UDINT;
nSParam: UINT;
END_STRUCT
END_TYPE
```

Sets an S-parameter [▸ 71] in the cyclic channel interface.

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
nSParam	UINT	S-parameter from NC to PLC

ST_NciTParam

```
TYPE ST_NciTParam :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeTParam; (*do not override this parameter *)
nDisplayIndex: UDINT;
nTParam: UINT;
END_STRUCT
END_TYPE
```

Sets an T-parameter [▸ 71] in the cyclic channel interface.

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
nTParam	UINT	T-parameter from NC to PLC

ST_NciDynOvr

```
TYPE ST_NciDynOvr :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeDynOvr; (*do not override this parameter*)
nDisplayIndex: UDINT;
fDynOvr: LREAL;
END_STRUCT
END_TYPE
```

Modal functions for changing the path dynamics.

See DynOvr [▸ 77] in the interpreter documentation.

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fDynOvr	LREAL	Value for dynamic override (0.01 < fDynOvr <= 1)

ST_NciVertexSmoothing

since library Build 11

```
TYPE ST_NciVertexSmoothing :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeVertexSmoothing; (*do not override this parameter *)
nDisplayIndex: UDINT;
nType: UDINT; (*type of smoothing, e.g. parabola, bi-quad *)
nSubtype: UDINT; (*e.g. adaptive, constant radius *)
fRadius: LREAL; (*max. radius for tolerance ball *)
END_STRUCT
END_TYPE
```

Modal function for activating blending at the segment transition. Blending is active until it is cancelled by setting the radius to 0.

A more detailed description of the parameter can be found in the interpreter documentation (paramVertexSmoothing [▸ 56]).

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
nType	UDINT	Blending type: 2: Parabola 3: Bi-quadratic 4: 3rd order Bezier 5: 5th order Bezier
nSubtype	UDINT	1: Constant tolerance radius 2: Distance between intersection and vertex 3: Adaptive tolerance radius
fRadius	LREAL	Radius of the blending sphere in basic units (e.g. mm)

ST_NciBaseFrame

since library Build 21

```
TYPE ST_NciBaseFrame:
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeBaseFrame; (*Do not override this parameter *)
nDisplayIndex: UDINT;
fShfitX: LREAL;
fShiftY: LREAL;
fShiftZ: LREAL;
fRotX: LREAL;
fRotY: LREAL;
fRotZ: LREAL;
fShiftQ1: LREAL;
fShiftQ2: LREAL;
fShiftQ3: LREAL;
fShiftQ4: LREAL;
fShiftQ5: LREAL;
END_STRUCT
END_TYPE
```

The structure ST_NciBaseFrame describes a modal zero shift and rotation. The operating principle is the same as for zero shift and rotation in the interpreter, i.e. the point of rotation is the current origin (see rotation [▸ 51] in the interpreter documentation)

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fShiftX	LREAL	Zero shift in X direction
fShiftY	LREAL	Zero shift in Y direction
fShiftZ	LREAL	Zero shift in Z direction
fRotX	LREAL	Rotation of the X axis
fRotY	LREAL	Rotation of the Y axis
fRotZ	LREAL	Rotation of the Z axis
fShiftQ1	LREAL	Offset of the Q1 axis
fShfitQ2	LREAL	Offset of the Q2 axis
fShiftQ3	LREAL	Offset of the Q3 axis
fShiftQ4	LREAL	Offset of the Q4 axis
fShiftQ5	LREAL	Offset of the Q5 axis

ST_NciPathDynamics

Since library Build 22

```
TYPE ST_NciPathDynamics:
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypePathDynamics; (*do not override this parameter *)
```

```
nDisplayIndex: UDINT;
fAcc: LREAL;
fDec: LREAL;
fJerk: LREAL;
END_STRUCT
END_TYPE
```

The structure *ST_NciPathDynamics* sets the path dynamics (acceleration, deceleration, jerk). The operating principle is the same as for *paramPathDynamics* in the interpreter (see paramPathDynamics [▶ 78] in the interpreter documentation)

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fAcc	LREAL	Maximum permitted path acceleration
fDec	LREAL	Maximum permitted path deceleration
fJerk	LREAL	Maximum permitted path jerk

ST_NciAxisDynamics

Since library Build 30

```
TYPE ST_NciAxisDynamics:
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeAxisDynamics; (*Do not override this parameter*)
nDisplayIndex: UDINT;
nAxis: UDINT;
fAcc: LREAL;
fDec: LREAL;
fJerk: LREAL;
END_STRUCT
END_TYPE
```

The structure *ST_NciAxisDynamics* sets the path axis dynamics (acceleration, deceleration, jerk). The operating principle is the same as for *paramAxisDynamics* in the interpreter (see paramAxisDynamics [▶ 78] in the interpreter documentation)

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	>Do not override this parameter
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
nAxis	UDINT	Axis in interpolation group X:0 Y:1 Z:2 Q1:3 ... Q5:7
fAcc	LREAL	Maximum permitted axis acceleration
fDec	LREAL	Maximum permitted axis deceleration
fJerk	LREAL	Maximum permitted axis jerk

ST_NciDwellTime

Since library Build 30

```
TYPE ST_NciDwellTime:
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeDwellTime; (*Do not override this parameter *)
nDisplayIndex: UDINT;
fDwellTime: LREAL;
END_STRUCT
END_TYPE
```

The structure *ST_NciDwellTime* is used to activate a dwell time in seconds (see dwell time [▶ 44] in the interpreter documentation)

Input	Data type	Description
nEntryType	E_NciEntryType [▶ 227]	Do not override this parameter

Input	Data type	Description
nDisplayIndex	UDINT	For display purposes, such as block number in G-Code
fDwellTime	LREAL	Dwell time in seconds

ST_NciTangentialFollowingDesc

```
TYPE ST_NciTangentialFollowingDesc :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeTfDesc; (*do not override this parameter *)
bTangOn: BOOL;
nTangAxis: E_NciAxesInGroup; (*axis used for tangential following *)
nPathAxis1: E_NciAxesInGroup; (*describing the plane e.g. x*)
nPathAxis2: E_NciAxesInGroup; (*e.g. y ==> g17, xy plane*)
fOffset: LREAL; (*geo tangent is 0 degree, counting is mathmatical positive *)
fCriticalAngle1: LREAL;
nTfBehavior: E_TangentialFollowingBehavior; (*what to do if angle becomes bigger than critical angle
1 *)
END_STRUCT
END_TYPE
```

This is a modal command for switching tangential following on or off.

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter
bTangOn	BOOL	If TRUE, tangential following is switched on
nTangAxis	E_NciAxesInGroup [▸ 238]	Axis (Q1..Q5) used as tangential axis
nPathAxis1	E_NciAxesInGroup	First path axis describing the plane and orientation for calculating the tangent
nPathAxis2	E_NciAxesInGroup	Second path axis describing the plane and orientation for calculating the tangent
fOffset	LREAL	Offset of the tangential axis
fCritialAngle1	LREAL	Critical angle 1. The response in cases where the angle between two segments is greater than fCriticalAngle1 is specified with nTfBehavior
nTfBehavior	E_TangentialFollowingBehavior [▸ 238]	see fCriticalAngle1

E_NciAxesInGroup

```
TYPE E_NciAxesInGroup :
(
NoneAxis := 0,
XAxis,
YAxis,
ZAxis,
Q1Axis,
Q2Axis,
Q3Axis,
Q4Axis,
Q5Axis
);
END_TYPE
```

E_TangentialFollowingBehavior

```
TYPE E_TangentialFollowingBehavior :
(
E_TfIngoreAll, (*ignore critical angle *)
E_TfErrorOnCritical1 (*if angle becomes bigger than critical angle 1 ==> error *)
);
END_TYPE
```

ST_NciEndOfTables

```
TYPE ST_NciEndOfTables :
STRUCT
nEntryType: E_NciEntryType := E_NciEntryTypeEndOfTables; (*do not override this parameter *)
END_STRUCT
END_TYPE
```

Indicates the last entry of the last table. Is used for signaling the bChannelDone flag in FB_NciFeedTable [▸ 226].

Input	Data type	Description
nEntryType	E_NciEntryType [▸ 227]	Do not override this parameter

6 Sample: FirstNciSample

Overview

The example *FirstNciSample* shows how an NC program is loaded from the PLC and processing is triggered.

Amongst other activities, this includes

- the creation of an interpolation group
- loading an NC program (parts program)
- starting the NC program
- acknowledging M-functions
- dissolving the interpolation group

Further information on the function blocks used can be found in sections PLC Library: NC Configuration [▶ 207] and PLC Library: NCI Interpreter [▶ 101].

Installing the sample program

The file *FirstNciSample.exe* is self-extracting and can be copied to any directory.

After successful unpacking of the data, the directory contains the following files:

- FirstNciSample.pro (PLC program)
- FirstNciSample.tsm (configuration file for the TwinCAT System Manager)
- ScopeFirstNciSample.scp (configuration file for TwinCAT Scope)
- TestIt.nc (sample NC program)

Starting the sample program

1. Copy TestIt.nc into the TwinCAT\CNC directory (otherwise the parts program will not be found during loading). Alternatively adjust the path in the PLC program.
2. Compile the PLC project *FirstNciSample.pro*
3. Open FirstNciSample.tsm and activate the configuration
4. Set PLC to run mode

⇨ The NC program will now be executed once. The scope configuration *ScopeFirstNciSample.scp* can be used to log the positions and velocities.

To execute the NC program again, use the variable *bExec* in the main part of the PLC to trigger the execution.

Download

https://infosys.beckhoff.com/content/1033/tcnci/Resources/471329035/.exe FirstNciSample

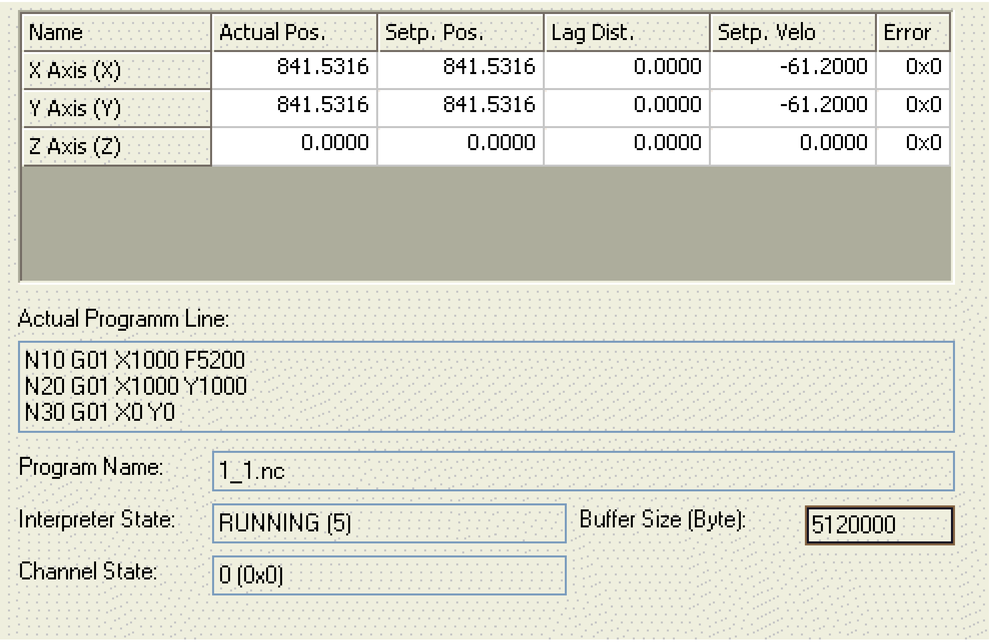
7 Appendix

7.1 Display of the parts program

Reading of the current NC line via ADS

This ADS Read command returns a maximum of 3 lines of the current parts program, i.e. the current line of code and perhaps two previously processed lines.

Function	ADS-Read
Port	500 (dec)
Index Group	0x2300 + channel ID
Index Offset	0x2000 0001
Data	string (30 bytes min.)



Reading of the current program name

This ADS Read command returns the program name of the current main NC program (in this case 1_1.nc).

Function	ADS-Read
Port	500 (dec)
Index Group	0x2100 + channel ID
Index Offset	0x7
Data	string, 100 characters max.

Reading of the current file information

from TwinCAT V2.10, Build 1243

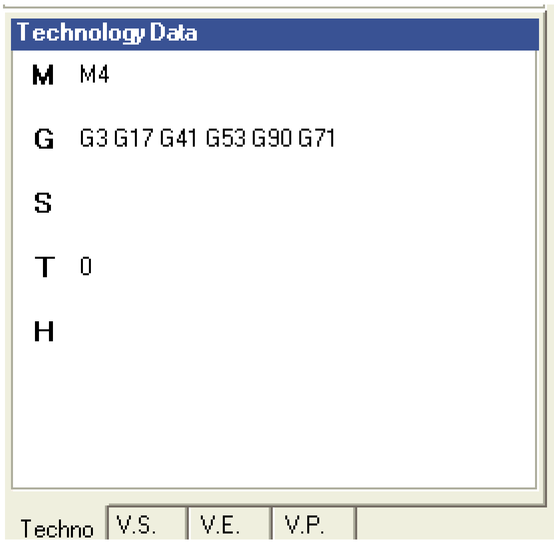
In contrast to the 'Reading the current NC line' function, in this case not the line itself is read, but associated line information. The return value is the current program name (e.g. file name of the subroutine) and a file offset. Based on this information, the user interface can open the associated file and highlight the respective line. The display is no longer limited to 3 rows, i.e. any number of lines can be displayed.

In the event of an NCI load or run-time error, information about the associated line of code can be obtained via this route.

Function	ADS-Read	
Port	500 (dec)	
Index Group	0x2100 + channel ID	
Index Offset	0x12	
Data	UINT32	Current display of 1: SAF- 2: Interpreter 3: Error offset
	UINT32	File offset
	char[260]	path + program name

7.2 Display of technology data

From TwinCAT V2.10 Build 1303



The current active technology data like G functions, zero offset shifts and rotation can be read via ADS.

Activation for reading the technology data

In order to read the above-mentioned parameters, activation via ADS is required first.

The function must be activated before the start of the NC program, or earlier. It remains active until either a TwinCAT restart is performed or the function is reset explicitly.

Function	ADS Write
Port	500 (dec)
Index Group	0x2000 + channel ID
Index Offset	0x0053
Data	DWORD 0: disable (default) 1: enable

Reading the currently active zero shift

This command reads the active zero shift of the segment currently in block execution (SAF). If no zero shift is active (G53), the structure for the individual components contains a zero vector. These data can be used for switching the display between machine coordinates and programming coordinates, for example.

The data, which are read with the function block 'ItpReadZeroShift', for example, may differ from these values, since the interpreter data are read with the FB, which may already take into account new offsets.

Function	ADS-Read	
Port	500 (dec)	
Index Group	0x2100 + channel ID	
Index Offset	0x0014	
Data	{	
	UINT32	block counter
	UINT32	dummy
	LREAL[3]	zero shift G54..G57
	LREAL[3]	zero shift G58
	LREAL[3]	zero shift G59
	}	

Reading the currently active rotation

This command reads the active rotation of the segment currently in block execution (SAF).

Function	ADS-Read	
Port	500 (dec)	
Index Group	0x2100 + channel ID	
Index Offset	0x0015	
Data	{	
	UINT32	block counter
	UINT32	dummy
	LREAL[3]	rotation of X, Y & Z in degrees
	}	

Reading the currently active G-Code

The G-Code is subdivided into groups. For example, the geometries types with modal effect (G01, G02...) and the plane selection (G17..G19) form separate groups. When the G-Code information is read, the enumerator for the groups is also read. These can then be displayed in an application-specific manner.

Since the read command comes with a parameter to be read, not all groups have to be read. The memory provided is always filled by group 1. If, for example, the transferred memory size is 3x8 bytes, the data for the block counter, group 1 and 2 are returned.

Function	ADS-Read	
Port	500 (dec)	
Index Group	0x2100 + channel ID	
Index Offset	0x0013	
Data	{	
	UINT32	block counter
	UINT32	Group 1: ModalGeoTypes
	UINT32	Group 2: BlockwiseGeoTypes
	UINT32	Group 3: ModalPlaneSelection
	UINT32	Group 4: ModalToolCompensation
	UINT32	Group 5: ModalToolFeedDirection
	UINT32	Group 6: ModalZeroShift
	UINT32	Group 7: ModalAccurateStop
	UINT32	Group 8: BlockwiseAccurateStop
	UINT32	Group 9: ModalDesignationAbsInc
	UINT32	Group 10: ModalDesignationInchMetric
	UINT32	Group 11: ModalFeedRateInCurve

	UINT32	Group 12: ModalCenterpointCorr
	UINT32	Group 13: ModalCircleCpAbsInc
	UINT32	Group 14: ModalCollisionDetection
	UINT32	Group 15: ModalRotation
	UINT32	Group 16: ModalCalcExRot
	UINT32	Group 17: ModalDiam
	UINT32	Group 18: ModalFeedrateIpol
	UINT32	Group 19: ModalMirror
	}	

```
#define GCodeOffset 0x1000
#define CommonIdentOffset 0x2000 // used for non-g-code commands, like rot, cfc...
```

Group 1: ModalGeoTypes

```
enum GCodeGroup_ModalGeoTypes
{
ModalGeoTypeUndefined = 0,
ModalGeoTypeG0 = 0 + GCodeOffset, // line - rapid traverse
ModalGeoTypeG01 = 1 + GCodeOffset, // straight line
ModalGeoTypeG02 = 2 + GCodeOffset, // circle clockwise
ModalGeoTypeG03 = 3 + GCodeOffset // circle anticlockwise
};
```

Group 2: BlockwiseGeoTypes

```
enum GCodeGroup_BlockwiseGeoTypes
{
BlockwiseGeoTypeNone = 0,
BlockwiseGeoTypeG04 = 4 + GCodeOffset, // dwell time
BlockwiseGeoTypeG74 = 74 + GCodeOffset, // homing
BlockwiseGeoTypeCip = 1 + CommonIdentOffset // circle parametrized with 3 points
};
```

Group 3: ModalPlaneSelection

```
enum GCodeGroup_ModalPlaneSelection
{
ModalPlaneSelectUndefined = 0,
ModalPlaneSelectG17 = 17 + GCodeOffset, // xy-plane
ModalPlaneSelectG18 = 18 + GCodeOffset, // zx-plane
ModalPlaneSelectG19 = 19 + GCodeOffset // yz-plane
};
```

Group 4: ModalToolCompensation

```
enum GCodeGroup_ModalToolCompensation
{
ModalToolCompUndefined = 0,
ModalToolCompG40 = 40 + GCodeOffset, // tool compensation off
ModalToolCompG41 = 41 + GCodeOffset, // tool compensation left
ModalToolCompG42 = 42 + GCodeOffset // tool compensation right
};
```

Group 5: ModalToolFeedDirection

```
enum GCodeGroup_ModalToolFeedDirection
{
ModalToolFeedDirUndefined = 0,
ModalToolFeedDirPos = 2 + CommonIdentOffset, // tool feed direction positive
ModalToolFeedDirNeg = 3 + CommonIdentOffset // tool feed direction negative
};
```

Group 6: ModalZeroShift

```
enum GCodeGroup_ModalZeroShift
{
ModalZeroShiftUndefined = 0,
ModalZeroShiftG53 = 53 + GCodeOffset, // zero shift off
ModalZeroShiftG54G58G59 = 54 + GCodeOffset, // zero shift G54 + G58+ G59
ModalZeroShiftG55G58G59 = 55 + GCodeOffset, // zero shift G55 + G58+ G59
ModalZeroShiftG56G58G59 = 56 + GCodeOffset, // zero shift G56 + G58+ G59
ModalZeroShiftG57G58G59 = 57 + GCodeOffset // zero shift G57 + G58+ G59
};
```

Group 7: ModalAccurateStop

```
enum GCodeGroup_ModalAccurateStop
{
ModalAccurateStopNone = 0,
ModalAccurateStopG60 = 60 + GCodeOffset // modal accurate stop
};
```

Group 8: BlockwiseAccurateStop

```
enum GCodeGroup_BlockwiseAccurateStop
{
BlockwiseAccurateStopNone = 0,
BlockwiseAccurateStopG09 = 9 + GCodeOffset, // common accurate stop
BlockwiseAccurateStopTpm = 4 + CommonIdentOffset // target position monitoring
};
```

Group 9: ModalDesignationAbsInc

```
enum GCodeGroup_ModalDesignationAbsInc
{
ModalDesignAbsIncUndefined = 0,
ModalDesignAbsIncG90 = 90 + GCodeOffset, // absolute designation
ModalDesignAbsIncG91 = 91 + GCodeOffset // incremental designation
};
```

Group 10: ModalDesignationInchMetric

```
enum
GCodeGroup_ModalDesignationInchMetric
{
ModalDesignInchMetricUndefined = 0,
ModalDesignInchMetricG70 = 70 + GCodeOffset, // designation inch
ModalDesignInchMetricG71 = 71 + GCodeOffset, // designation metric
ModalDesignInchMetricG700 = 700 + GCodeOffset, // designation inch & feedrate recalculated
ModalDesignInchMetricG710 = 710 + GCodeOffset // designation metric & feedrate recalculated
};
```

Group 11: ModalFeedRateInCurve

```
enum GCodeGroup_ModalFeedRateInCurve
{
ModalFeedRateInCurveUndefined = 0,
ModalFeedRateInCurveCfc = 5 + CommonIdentOffset, // constant feed contour
ModalFeedRateInCurveCfin = 6 + CommonIdentOffset, // constant feed inner contour
ModalFeedRateInCurveCftcp = 7 + CommonIdentOffset // constant feed tool center point
};
```

Group 12: ModalCenterpointCorr

```
enum GCodeGroup_ModalCenterpointCorr
{
ModalCenterpointCorrUndefined = 0,
ModalCenterpointCorrOn = 8 + CommonIdentOffset, // circle centerpoint correction on
ModalCenterpointCorrOff = 9 + CommonIdentOffset // circle centerpoint correction off
};
```

Group 13: ModalCircleCpAbsInc

```
enum GCodeGroup_ModalCircleCpAbsInc
{
ModalCircleCpUndefined = 0,
ModalCircleCpIncremental = 10 + CommonIdentOffset, // circle centerpoint incremental to start point
ModalCircleCpAbsolute = 11 + CommonIdentOffset // circle centerpoint absolute
};
```

Group 14: ModalCollisionDetection

```
enum GCodeGroup_ModalCollisionDetection
{
ModalCollisionDetectionUndefined = 0,
ModalCollisionDetectionOn = 12 + CommonIdentOffset, //collision detection on
ModalCollisionDetectionOff = 13 + CommonIdentOffset //collision detection off
};
```

Group 15: ModalRotation

```
enum GCodeGroup_ModalRotation
{
ModalRotationUndefined = 0,
```

```
ModalRotationOn = 14 + CommonIdentOffset, // rotation is turned on
ModalRotationOff = 15 + CommonIdentOffset // rotation is turned off
};
```

Group 16: ModalCalcExRot

```
enum GCodeGroup_ModalCalcExRot
{
ModalCalcExRotUndefined = 0,
ModalCalcExRotOn = 16 + CommonIdentOffset, // extended calculation for rotation turned on
ModalCalcExRotOff = 17 + CommonIdentOffset // extended calculation for rotation turned off
};
```

Group 17: ModalDiam

```
enum GCodeGroup_ModalDiam
{
ModalDiamUndefined = 0,
ModalDiamOn = 18 + CommonIdentOffset, // diameter programming on
ModalDiamOff = 19 + CommonIdentOffset // diameter programming off
};
```

Group 18: ModalFeedrateIpol

```
enum GCodeGroup_ModalFeedrateIpol
{
ModalFeedrateIpolUndefined = 0,
ModalFeedrateIpolConst = 20 + CommonIdentOffset, // federate interpolation constant (default)
ModalFeedrateIpolLinear = 21 + CommonIdentOffset // federate interpoaltion linear to remaining path
};
```

Group 19: ModalMirror

```
enum GCodeGroup_ModalMirror
{
// value - (32+CommonIdentOffset) shows the bitmask for mirrored axes
// that's why the sequence seems to be strange...
//
ModalMirrorUndefined = 0,
ModalMirrorOff = 32 + CommonIdentOffset,
ModalMirrorX = 33 + CommonIdentOffset,
ModalMirrorY = 34 + CommonIdentOffset,
ModalMirrorXY = 35 + CommonIdentOffset,
ModalMirrorZ = 36 + CommonIdentOffset,
ModalMirrorZX = 37 + CommonIdentOffset,
ModalMirrorYZ = 38 + CommonIdentOffset,
ModalMirrorXYZ = 39 + CommonIdentOffset
};
```

7.3 Displaying the remaining path length

from TwinCAT 2.9 B945

If calculation of the remaining path length is switched active, it is calculated up to as far as the next accurate stop, or as far as the last geometric segment in memory (block preparation). An accurate stop is, for instance, generated by G09 or by G60. However, M-functions of type handshake, decoder stops and G04 implicitly generate an accurate stop.

Activation:

Index group: 0x3000 + Group ID
Index offset: 0x0508

see index offset specification for group parameters [▶ 267]

Reading the remaining path length:
Reading is again implemented through ADS, and can also be recorded with TwinCAT Scope.

Index group: 0x3100 + Group ID
Index offset: 0x0522

The remaining path length can be transferred with the cyclic channel interface to the PLC via ItpSetCyclicLrealOffsets [▶ 145].
see index offset specification for group state [▶ 272]

7.4 Parameterisation

The parameterization of the NCI comprises the standard dynamic parameters (acceleration, deceleration, jerk) and their online changes, along with the minimum velocity and the parameters for the reduction of the path velocity including online change.

General characteristics at segment transitions

- Velocity: The segment set velocity VS changes at the segment transition from VS_in to VS_out. At the segment transition the velocity is always reduced to the lower of the two values.
- Acceleration: The current path acceleration is always returned to $a = 0$ at segment transition.
- Jerk: The jerk unit *J* changes according to the geometry at the segment transition. This can cause a significant step change in dynamics.
- It is possible to smooth segment transitions [▸ 32].

Table 11: NCI group parameters

Parameter	Meaning and boundary conditions
Curve velocity reduction mode [▸ 248]	Coulomb, cosine or VELOJUMP
Minimum velocity [▸ 247]	Path velocity which may not be less than this value (except peaks with movement reversal): $V\_min \geq 0.0$
Reduction method for C1 transitions [▸ 248]	Reduction factor for *C1* transitions: $C1 \geq 0.0$
VELOJUMP: C0 reduction factors *C0X*, C0Y, C0Z	Reduction factors for *C0* transitions for *X*, Y, Z axis: *C0X ≥ 0.0, C0Y ≥ 0.0, C0Z ≥ 0.0 (axis parameters, online change in* interpreter [▸ 79] possible).
DEVIATIONANGLE: -Reduction factor *C0 C0*	Path reduction factor for *C0* transitions: $1.0 \geq C0 \geq 0.0$
DEVIATIONANGLE: Critical angle (deep) φ_l	Angle from which a velocity reduction is applied at the segment transition: $0 \leq \varphi\_l < \varphi\_h \leq \pi$
DEVIATIONANGLE: Critical angle (high) φ_h	Angle from which the velocity at the segment transition(*v_link) is reduced to 0.0*: $0 \leq \varphi\_l < \varphi\_h \leq \pi$
Tolerance sphere radius [▸ 59] *TBR*	Radius of tolerance spheres: 1000.0 mm≥*TBR*≥ *0.1 mm*
C2 reduction factor [▸ 79] *C2*	Reduction factor for smoothed transitions: $C2 \geq 0.0$
Global software limit positions for the path [▸ 249]	Switches monitoring of the global software end positions for the path axes

Minimum velocity

Each NCI group has a minimum path velocity $V\_min \geq 0.0$. The actual velocity should always exceed this value. User-specified exceptions are: programmed stop at segment transition, path end and override requests which lead to a velocity below the minimum value. A systemic exception is a motion reversal. With the reduction method DEVIATIONANGLE the deflection angle is $\varphi \geq \varphi\_h$, in which case the minimum velocity is ignored. *V_min* must be less than the set value for the path velocity (F word) of each segment.

The minimum velocity can be set to a new value $V\_min \geq 0.0$ in the NC program at any time. The unit is *mm/sec*.

Classification of the segment transitions

In general, the transition from one segment to the next is not indefinitely smooth. Therefore, it is necessary to reduce the velocity at the transition point in order to avoid dynamic instability. For this purpose, the transitions are geometrically classified and the effective transition velocity - V_link - is determined in three steps.

Segments - as geographical objects - are defined here as curves in terms of differential geometry and are parameterized by the arc length.
A segment transition from a segment *S_in* to a segment *S_out* is classified in geometrical terms as type *Ck*, where *k* is a natural number (including *0*), if each segment has *k* continuous arc length differentials and the k^{th} derivatives at the transition point correspond.

C0 transitions have a knee-point at the transition point.

C1 transitions appear smooth, but are not smooth in dynamic terms. One example is the straight line-semi circle transition in the stadium: at the transition point there is a step change in acceleration.

C2 transitions (and of course *Ck* transitions with *k* > *2*) are dynamically smooth (jerk restricted).

Reduction method for *C2* transitions

As at all transitions, at *C2* transitions *V_link* is set to equal the minimum of both set segment velocities: *V_link* = *min(V_in,V_out)*. There is no further reduction.

Reduction method for C1 transitions

First, *V_link* is set to the lower of the two segment target velocities: *V_link* = min(*V_in,V_out*). The geometrically induced absolute step change in acceleration *AccJump* in the segment transition is calculated depending on the geometry types *G_in* and *G_out*, and the plane selection *G_in* and *G_out* of the segments to be connected, at velocity *V_link*. If this is greater than *C1* times the path acceleration/(absolute) deceleration *AccPathReduced* permissible for the geometries and planes, the velocity *V_link* is reduced until the resulting step change in acceleration is equal to *AccPathReduced*. If this value is less than *V_min*, then *V_min* takes priority.

Notice When changing the dynamic parameters, the permissible path acceleration for the geometries and planes and thereby the reaction of the reduction changes automatically.

Interface: System Manager [▶ 22] and Interpreter [▶ 79]

Reduction modes for C0 transitions

Several reduction methods are available for *C0* transitions. The reduction method VELOJUMP reduces the velocity after permitted step changes in velocity for each axis. The reduction method DEVIATIONANGLE reduces the velocity depending on the deflection angle φ (angle between the normalized end tangent *T_in* of the incoming segment *S_in* and the normalized start tangent *T_out* of the outgoing segment *S_out*). The cosine reduction method is a purely geometrical method (see curve velocity reduction method [▶ 23]).

The VELOJUMP method is recommended for mechanically independent axes, while for mechanically coupled axes (the *Y* axis is attached to the *X* axis, for example) the DEVIATIONANGLE method is usually recommended.

Reduction method for C0 transitions: VELOJUMP

If *V_link* = *min(V_in,V_out)*, and for each axis *V_jump[i]* = *C0[i]* * *min(A+[i],-A-[i])* * *T* is the permitted absolute step change in velocity for the axis *[i]*, wherein *C0[i]* is the reduction factor and *A+[i], A-[i]* are the acceleration/deceleration limits for the axis *[i]*, and *T* is the cycle time. The VELOJUMP reduction method ensures that the path velocity is reduced at the segment transition *V_link* until the absolute step change in the set axis velocity of axis *[i]* is at most *V_jump[i]*. *V_min* nevertheless has priority: if *V_link* is less than *V_min*, *V_link* is set to *V_min*. In the case of movement reversal with no programmed stop, there will be a step change in axis velocity.

Notice When changing the dynamic parameters, the maximum permissible step changes in axis velocity automatically change at the same time.

Reduction method for *C0* transitions: DEVIATIONANGLE

Notice When changing the dynamic parameters, the reduction factors do not automatically change at the same time.

Changing the parameters for *C0* transitions: DEVIATIONANGLE

Table 12: Parameter

Parameter	Meaning and boundary conditions
DEVIATIONANGLE: Reduction factor *C0 C0*	Path reduction factor for *C0* transitions: $1.0 \geq C0 \geq 0.0$
DEVIATIONANGLE: Critical angle (low) φ_l	Angle from which reduction takes effect: $0 \leq \varphi\_l < \varphi\_h \leq \pi$
DEVIATIONANGLE: Critical angle (high) φ_h	Angle from which reduction to $v\_link = 0.0$ takes effect: $0 \leq \varphi\_l < \varphi\_h \leq \pi$

Interface: Interpreter [▸ 79]

Cosine reduction method

See here [▸ 23].

Tolerance sphere radius and C2 reduction factor

These parameters are described under the heading "Smoothing of segment transitions".

Global software limit positions for the path

from TwinCAT V2.9 B946

The *'Global software limit position monitoring for the path'* offers two different ways of monitoring the limit position.

Limit position monitoring by the SAF task

This type of limit position monitoring is always active if the limit position for the axis has been switched to active (axis parameter). The monitoring is carried out component for component by the SAF task. This means that if the limit position is exceeded, the path velocity is instantly set to 0, and the entire interpolation group has an error.

This type of monitoring is activated through the axes parameters, and **not** by means of the group parameters described here.

Software limit positions on the path

To prevent the path velocity being set to 0 immediately when a violation of the software end positions is encountered, the function *'Global software end position monitoring of the path'* must be enabled. If this is active, the movement stops at the NC block in which the end positions were violated. The velocity is reduced via a ramp.

- So that the monitoring is only executed for the desired path axes, the software limit positions for the axis components must be selected (axis parameters)
- The monitoring is carried out for the standard geometry segments. These include

    Straight line

    Circle

    Helix

Auxiliary axes are monitored from TwinCAT V2.10 B1258

- Curves with splines are not monitored. The set values associated with the splines are always within the tolerance sphere. Otherwise the limit position monitoring will make use of the SAF task.
- Because meaningful and generally applicable monitoring of the limit positions can only be carried out at the NC program's run-time (before lookahead) it is possible that the path axes will move as far as (but not including) the NC block in which the limit positions are exceeded.
- If for some reason the axes are located outside the software limit positions it is possible to move back into the correct region in a straight line.

Parameterization:

System Manager: Group parameters [▶ 22]

7.4.1 Path override (interpreter override types)

The path override is a velocity override. This means that changing the override creates a new velocity, but does not affect the ramps (acceleration or jerk). The used override types only differ in terms of reference velocity.

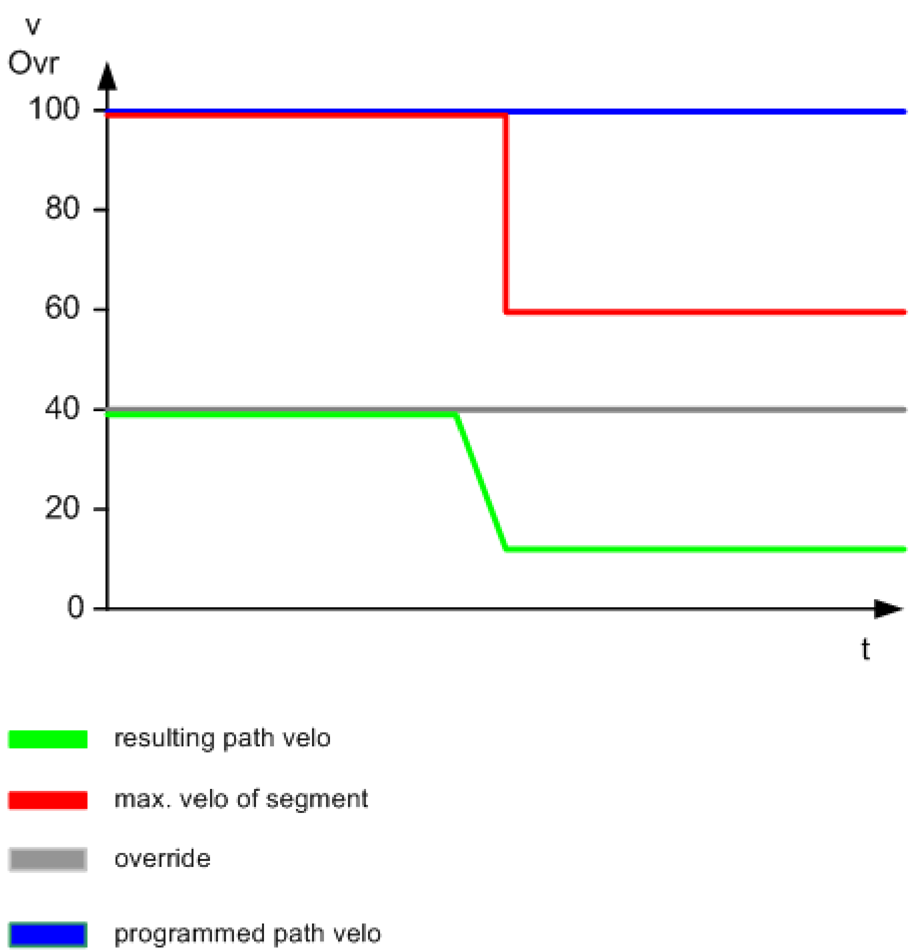
The parameterization takes place in the interpolation channel under the group parameters [▶ 23].

Option 'Reduced' - based on the reduced velocity (default)

Because of the relevant dynamic parameters (braking distance, acceleration etc.) it is not possible for the programmed velocity (the blue line) be achieved in every segment. For this reason a velocity, possibly reduced, (the red line) is calculated for each geometric segment. In the standard case, the override is made with reference to this segment velocity.

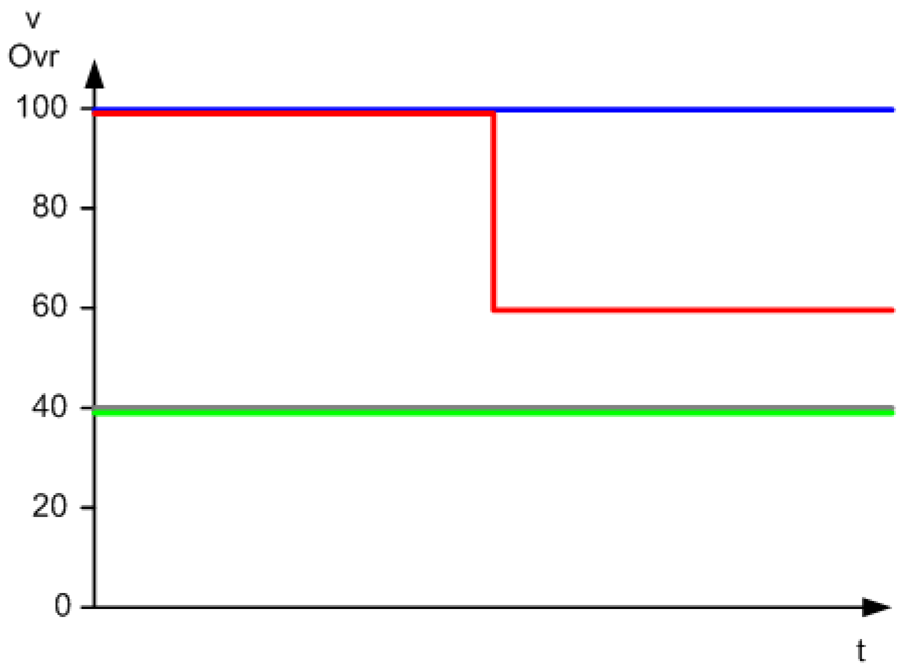
The advantage of this override type is that if override values are small the machine operates with an approximately linear reduction in velocity, and this is therefore the correct setting for most applications.

v_res = v_max * Override



Option 'Original' - based on the programmed path velocity

The override value is based on the velocity programmed by the user. The maximum segment velocity only has a limiting effect.



Option 'Reduced [0 ... >100%]' - based on internally reduced velocity with the option to specify a value greater than 100%

from TwinCAT V2.10, Build 1329

The override type behaves like 'Reduced'. With this override type it is possible to travel along the path more quickly than programmed in the G-Code. There is no limitation to 120%, for example. The maximum possible path velocity is limited by the maximum velocities of the axis components (G0 velocity) and their dynamics.

If limitation to a particular value, e.g. 120%, is required, this can be set in the PLC project.

7.5 Cyclical Channel Interface

The channel interface is responsible for the cyclic data exchange between the PLC and the NCI.

The NC input/output variable names can sometimes differ from the PLC input/output variable names. However, this is not really significant. This description relates to the naming in the PLC.

From the NCI to the PLC (150 bytes)

```
TYPE NciChannelToPlc :
STRUCT
nJobNo              : DWORD;
nFastMFuncMask      : ARRAY[1..5] OF DWORD; (* Mask to evaluate fast M-unctions *)
nHskMFuncNo         : WORD; (* evaluate M-function with handshake *)
nHskMFuncReq        : WORD;
nHFuncValue         : DINT;
nSpindleRpm         : WORD;
nTool               : WORD;
nReserved1          : ARRAY[37..132] OF BYTE;
nLoadedProg         : DWORD; (* loaded program number if exist *)
nItpMode            : WORD; (* Interpreter mode *)
nItpState           : WORD; (* Interpreter status *)
nItpErrCode         : WORD; (* Interpreter-Channel Error Code *)
nReserved2          : ARRAY[143..150] OF BYTE;
END_STRUCT
END_TYPE
```

Variable name	Data type	Description
nJobNo	DWORD	Job number

Variable name	Data type	Description
nFastMFuncMask	ARRAY OF DWORD	Bit mask for evaluation of the fast M-functions [▸ 67]
nHskMFuncNo	WORD	Number of synchronous M-function present (M-function with handshake)
nHskMFuncReq	WORD	Flag indicating that a synchronous M-function is present 0: no synchronous M-function is present 1: a synchronous M-function is present
nHFuncValue	DINT	Value of the auxiliary function
nSpindleRpm	WORD	Spindle rotation speed
nTool	WORD	Tool number
nLoadedProg	DWORD	Name of the NC program presently being executed. If the name is not a DWORD, this value is 0.
nItpOpMode	WORD	Bit mask that indicates execution in interpreter mode.
nItpState	WORD	Interpreter state [▸ 16]
nItpErrorCode	WORD	Error code of the interpreter channel

The above structure contains some reserved areas. Some of these areas are used in newer TwinCAT versions. On the PLC side the structure was not renamed, so that relevant data can be read with the PLC function.

NCI to PLC (structure from NC perspective)

Variable name	Data type	Description
mJobNr		see table above nJobNo
mDWORD		see table above nFastMFuncMask
sHandshake.nFunc		see table above nHskMFuncNo
sHandshake.nRequested		see table above nHskMFuncReq
sHandshake.nHFuncValue		see table above nHFuncValue
sHandshake.nSpindelRPM		see table above nSpindleRpm
sHandshake.nTool		see table above nTool
nChnState	DWORD	
nChnState.bIsEStopRequested	Bit 8 (zero based)	Indicates that an ItpEStop was called, without checking whether the axes are already at standstill.
nChnState.bIsFeedFromBackupList	Bit 10 (zero based)	For retracing the current entries from the interpreter backup list are sent
nChnState.bIsMovingBackward	Bit 11 (zero based)	Indicates that the current motion is a backward movement (from TC 2.11 Build 1550)
nParams	Array of DWORD	Data of the freely configurable channel interface (see ItpSetCyclicUDintOffsets [▸ 147])
fParams	Array of LREAL	Data of the freely configurable channel interface (see ItpSetCyclicLrealOffsets [▸ 145])
mProgNr		see table above nLoadedProg

Variable name	Data type	Description
mItpOpMode		see table above nItpOpMode
mItpOpState		see table above nItpState
mErrorCode		see table above nItpErrorCode
nChnId	WORD	Channel ID
nGrpId	WORD	Group ID
nItfVersion	WORD	Version of this cyclic channel interface

From PLC to NCI (150 bytes)

```
TYPE NciChannelFromPlc :
STRUCT
nSkipLine           : WORD; (* Mask to skip lines *)
nItpMode            : WORD;
nMFuncGranted       : WORD; (* granted signal of the M-function *)
nReserved1          : ARRAY[7..14] OF BYTE;
nChaAxesOvr         : DWORD; (* Channel override in percent * 100 *)
nChaSpindleOvr      : DWORD;
nReserved2          : ARRAY[23..150] OF BYTE;
END_STRUCT
END_TYPE
```

Variable name	Data type	Description
nSkipLine	WORD	Bit mask with which block skipping [▸ 30] of the NCI is parameterized from the PLC.
nItpMode	WORD	Bit mask with which the interpreter execution mode can be altered. This is, for instance, required if the interpreter is to operate in single block [▸ 36] mode.
nMFuncGranted	WORD	Flag with which an M-function of type 'Handshake' is acknowledged 0: not acknowledged 1: acknowledgement
nChaAxesOvr	DWORD	Channel override for the axes from 0...1000000 (corresponds to 0 - 100%)
nChaSpindleOvr	DWORD	Channel override for the spindle from 0...1000000 (corresponds to 0 - 100%)

7.6 ADS Return Codes

Grouping of error codes:
Global error codes: ADS Return Codes [▸ 253]... (0x9811_0000 ...)
Router error codes: ADS Return Codes [▸ 254]... (0x9811_0500 ...)
General ADS errors: ADS Return Codes [▸ 254]... (0x9811_0700 ...)
RTime error codes: ADS Return Codes [▸ 256]... (0x9811_1000 ...)

Global error codes

Hex	Dec	HRESULT	Name	Description
0x0	0	0x98110000	ERR_NOERROR	No error.
0x1	1	0x98110001	ERR_INTERNAL	Internal error.
0x2	2	0x98110002	ERR_NORTIME	No real time.
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Allocation locked – memory error.
0x4	4	0x98110004	ERR_INSERTMAILBOX	Mailbox full – the ADS message could not be sent. Reducing the number of ADS messages per cycle will help.

Hex	Dec	HRESULT	Name	Description
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Wrong HMSG.
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Target port not found – ADS server is not started or is not reachable.
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Target computer not found – AMS route was not found.
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unknown command ID.
0x9	9	0x98110009	ERR_BADTASKID	Invalid task ID.
0xA	10	0x9811000A	ERR_NOIO	No IO.
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unknown AMS command.
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 error.
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port not connected.
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	Invalid AMS length.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Invalid AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installation level is too low –TwinCAT 2 license error.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	No debugging available.
0x12	18	0x98110012	ERR_PORTDISABLED	Port disabled – TwinCAT system service not started.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port already connected.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 error.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync error.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	No index map for AMS Sync available.
0x18	24	0x98110018	ERR_INVALIDAMSPORT	Invalid AMS port.
0x19	25	0x98110019	ERR_NOMEMORY	No memory.
0x1A	26	0x9811001A	ERR_TCPSEND	TCP send error.
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host unreachable.
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	Invalid AMS fragment.
0x1D	29	0x9811001D	ERR_TLSSEND	TLS send error – secure ADS connection failed.
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Access denied – secure ADS access denied.

Router error codes

Hex	Dec	HRESULT	Name	Description
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Locked memory cannot be allocated.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	The router memory size could not be changed.
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages.
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	The Debug mailbox has reached the maximum number of possible messages.
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown.
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	The router is not initialized.
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	The port number is already assigned.
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	The port is not registered.
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	The maximum number of ports has been reached.
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	The port is invalid.
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	The router is not active.
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	The mailbox has reached the maximum number for fragmented messages.
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	A fragment timeout has occurred.
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	The port is removed.

General ADS error codes

Hex	Dec	HRESULT	Name	Description
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	General device error.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service is not supported by the server.
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	Invalid index group.
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Invalid index offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Reading or writing not permitted.

Hex	Dec	HRESULT	Name	Description
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parameter size not correct.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Invalid data values.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Device is not ready to operate.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Device is busy.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Invalid operating system context. This can result from use of ADS blocks in different tasks. It may be possible to resolve this through multitasking synchronization in the PLC.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Insufficient memory.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Invalid parameter values.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Not found (files, ...).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax error in file or command.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objects do not match.
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Object already exists.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol not found.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Invalid symbol version. This can occur due to an online change. Create a new handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Device (server) is in invalid state.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode not supported.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification handle is invalid.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification client not registered.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	No further handle available.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Notification size too large.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Device not initialized.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Device has a timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface query failed.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Wrong interface requested.
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID is invalid.
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID is invalid.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Request pending.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Request is aborted.
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal warning.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Invalid array index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol not active.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Access denied.
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Missing license.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	License expired.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	License exceeded.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Invalid license.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	License problem: System ID is invalid.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	License not limited in time.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Licensing problem: time in the future.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	License period too long.
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception at system startup.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	License file read twice.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Invalid signature.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Invalid certificate.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public key not known from OEM.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	License not valid for this system ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo license prohibited.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Invalid function ID.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Outside the valid range.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Invalid alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Invalid platform level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Context – forward to passive level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Context – forward to dispatch level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Context – forward to real time.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Client error.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	Service contains an invalid parameter.

Hex	Dec	HRESULT	Name	Description
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling list is empty.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var connection already in use.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	The called ID is already in use.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	Timeout has occurred – the remote terminal is not responding in the specified ADS timeout. The route setting of the remote terminal may be configured incorrectly.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Error in Win32 subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Invalid client timeout value.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port not open.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	No AMS address.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Internal error in Ads sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Hash table overflow.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Key not found in the table.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESYM	No symbols in the cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Invalid response received.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port is locked.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was cancelled.

RTime error codes

Hex	Dec	HRESULT	Name	Description
0x1000	4096	0x98111000	RTERR_INTERNAL	Internal error in the real-time system.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer value is not valid.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task pointer has the invalid value 0 (zero).
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack pointer has the invalid value 0 (zero).
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	The request task priority is already assigned.
0x1005	4101	0x98111005	RTERR_NOMORETCB	No free TCB (Task Control Block) available. The maximum number of TCBs is 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	No free semaphores available. The maximum number of semaphores is 64.
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	No free space available in the queue. The maximum number of positions in the queue is 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	An external synchronization interrupt is already applied.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	No external sync interrupt applied.
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Application of the external synchronization interrupt has failed.
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x extension is not supported.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel VT-x extension is not enabled in the BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Missing function in Intel VT-x extension.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Activation of Intel VT-x fails.

Specific positive HRESULT Return Codes:

HRESULT	Name	Description
0x0000_0000	S_OK	No error.
0x0000_0001	S_FALSE	No error. Example: successful processing, but with a negative or incomplete result.
0x0000_0203	S_PENDING	No error. Example: successful processing, but no result is available yet.
0x0000_0256	S_WATCHDOG_TIMEOUT	No error. Example: successful processing, but a timeout occurred.

TCP Winsock error codes

Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not respond.

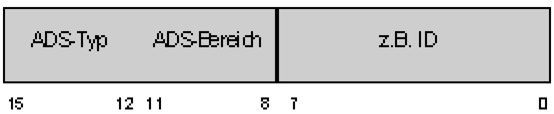
Hex	Dec	Name	Description
0x274D	10061	WSAECONNREFUSED	Connection refused - no connection could be established because the target computer has explicitly rejected it. This error usually results from an attempt to connect to a service that is inactive on the external host, that is, a service for which no server application is running.
0x2751	10065	WSAEHOSTUNREACH	No route to host - a socket operation referred to an unavailable host.
More Winsock error codes: Win32 error codes			

7.7 Specification "Index group" for NC (ID [0x01...0xFF])

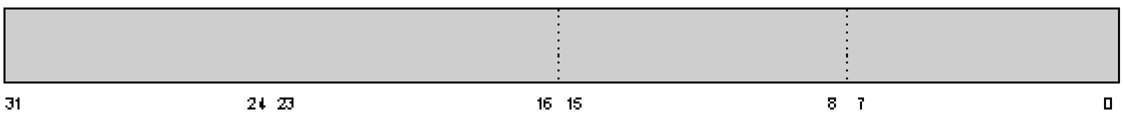
Index group (Hex)	Description		Remarks
0x1000	Ring-0-Manager: Parameter [▸ 258]		Optional !
0x1100	Ring-0-Manager: State [▸ 259]		Optional !
0x1200	Ring-0-Manager: Functions [▸ 260]		Optional !
0x1300	Ring-0-Manager: Cyclic process data		Not implemented !
0x2000 + ID	Channel with corr. ID: Parameter [▸ 260]		
0x2100 + ID	Channel with corr. ID: State [▸ 263]		
0x2200 + ID	Channel with corr. ID: Functions [▸ 264]		
0x2300 + ID	Channel with corr. ID: Cyclic process data [▸ 267]		
0x3000 + ID	Group with corr. ID: Parameter [▸ 267]		Optional!
0x3100 + ID	Group with corr. ID: State [▸ 272]		Optional!
0x3200 + ID	Group with corr. ID: Functions [▸ 276]		Optional!
0x3300 + ID	Group with corr. ID: Cyclic process data		Not implemented!
0x4000 + ID	Axis with corr. ID: Parameter [▸ 282]		
0x4100 + ID	Axis with corr. ID: State [▸ 296]		
0x4200 + ID	Axis with corr. ID: Functions [▸ 304]		
0x4300 + ID	Axis with corr. ID: Cyclic process data [▸ 321]		
0x5000 + ID	Encoder with corr. ID: Parameter [▸ 325]		Optional!
0x5100 + ID	Encoder with corr. ID: State [▸ 329]		Optional!
0x5200 + ID	Encoder with corr. ID: Functions [▸ 333]		Optional!
0x5300 + ID	Encoder with corr. ID: Cyclic process data [▸ 334]		Optional!
0x6000 + ID	Controller with corr. ID: Parameter [▸ 336]		Optional!
0x6100 + ID	Controller with corr. ID: State [▸ 340]		Optional!
0x6200 + ID	Controller with corr. ID: Functions [▸ 343]		Optional!
0x6300 + ID	Controller with corr. ID: Cyclic process data		Not implemented!
0x7000 + ID	Drive with corr. ID: Parameter [▸ 343]		Optional!
0x7100 + ID	Drive with corr. ID: State [▸ 346]		Optional!
0x7200 + ID	Drive with corr. ID: Functions [▸ 348]		Optional!
0x7300 + ID	Drive with corr. ID: Cyclic process data [▸ 348]		Optional!
0xA000 + ID	table (n x m) with corr. ID: Parameter [▸ 349]		
0xA100 + ID	table (n x m) with corr. ID: State [▸ 354]		
0xA200 + ID	table (n x m) with corr. ID: Functions [▸ 354]		
0xA300 + ID	table (n x m) with corr. ID: Cyclic process data		Not implemented!
0xF000 ... 0xFFFF	reserved area (TwinCAT system area)		
IndexGroup:	IndexOffset:		
0xF081	0x00000000 ... 0xFFFFFFFF (n elements)	ADSIGRP_SUMUP_WRITE The *Read-Write-command* contains a list in the Write-data of multiple separate *ADS-Write-commands* (like a	

Index group (Hex)	Description		Remarks
		group request). Structure of the Write-Data: *[IdxGrp(1), IdxOff(1), WriteLen(1), ..., IdxGrp(n), IdxOff(n), WriteLen(n),* *WriteData(1), ..., WriteData(n)]* Structure of the Read-Data: *[Error(1), ..., Error(n)]*	
0xF082	0x00000000 ... 0xFFFFFFFF (n elements)	ADSIGRP_SUMUP_READWRITE The *Read-Write-command* contains a list in the Write-data of multiple separate *ADS-Read-Write-commands* (like a group request). Structure of the Write-Data: *[IdxGrp(1), IdxOff(1),ReadLen(1), WriteLen(1), ..., IdxGrp(n), IdxGrp(n), ReadLen(n), WriteLen(n),* *WriteData(1), ..., WriteData(n)]* Structure of the Read-Data: *[Error(1), ReadLen(1), ..., Error(n), ReadLen(n),* *ReadData(1), ..., ReadData(n)]*	
0xF084	0x00000000 ... 0xFFFFFFFF (n elements)	ADSIGRP_SUMUP_READ (READEX2) The *Read-Write-command* contains a list in the Write-data of multiple separate *ADS-Read-commands* (like a group request). Structure of the Write-Data: *[IdxGrp(1), IdxOff(1), ReadLen(1), ..., IdxGrp(n), IdxGrp(n), ReadLen(n)]* Structure of the Read-Data: *[Error(1), ReadLen(1), ..., Error(n), ReadLen(n),* *ReadData(1), ..., ReadData(n)]*	

Index-Group:



Index-Offset:



7.7.1 Specification Ring-0-Manager

7.7.1.1 "Index offset" specification for Ring-0 parameter (Index group 0x1000)

Index offset (Hex)	Access	Ring-0-Man-ager	Data type	Phys. unit	Definition range	Description	Note
0x00000010	Read	every	UINT32	100 ns		Cycle time SEC task	
0x00000012	Read	every	UINT32	100 ns		Cycle time SPP task	
0x00000014/	Read	every	INT32	ns		Global Time Compensation Shift (SEC task)	from TC 2.11R3 Build 2234
0x00000020	Read/Write	every	UINT16	1	0/1	Cyclic monitoring and correction of	from TC 2.11 B1550

Index offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Note
						NC setpoints for data consistency	

7.7.1.2 "Index offset" specification for Ring-0 state (Index group 0x1100)

Index offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1	0, 1...255	Number of channels	
0x00000002	Read	every	UINT32	1	0, 1...255	Number of groups	
0x00000003	Read	every	UINT32	1	0, 1...255	Number of axes	
0x00000004	Read	every	UINT32	1	0, 1...255	Number of encoders	
0x00000005	Read	every	UINT32	1	0, 1...255	Number of controllers	
0x00000006	Read	every	UINT32	1	0, 1...255	Number of drives	
0x0000000A	Read	every	UINT32	1	0, 1...255	Number of tables (n x m)	
0x00000010	Read	every	UINT32	1		cycle time error counter SEC task (not scopeable)	Reserved!
0x00000014/	Read	every	UINT32	1		I/O cycle time error counter SEC task (not scopeable)	Reserved!
0x00000020	Read	every	UINT32	µs		computing time SEC task (not scopeable)	Reserved!
0x00000031	Read	every	UINT32[number]	1	0, 1...255	supplies the Channel IDs for all Channels in the system	
0x00000032	Read	every	UINT32[number]	1	0, 1...255	supplies the group IDs for all groups in the system	
0x00000033	Read	every	UINT32[number]	1	0, 1...255	supplies the axis IDs for all axes in the system	
0x00000034	Read	every	UINT32[number]	1	0, 1...255	supplies the encoder IDs for all encoders in the system	
0x00000035	Read	every	UINT32[number]	1	0, 1...255	supplies the controller IDs for all controllers in the system	
0x00000036	Read	every	UINT32[number]	1	0, 1...255	supplies the Drive IDs for all Drives in the system	
0x0000003A	Read	every	UINT32[number]	1	0, 1...255	supplies the table IDs for all tables in the system	

Index offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Note
0x000001nn	Read	every	UINT32	1	0, 1...255	supplies for the encoder ID the appropriate axis ID nn = Encoder ID	Reserved!
0x000002nn	Read	every	UINT32	1	0, 1...255	supplies for the Controller ID the appropriate axis ID nn = Controller ID	Reserved!
0x000003nn	Read	every	UINT32	1	0, 1...255	supplies for the Drive ID the appropriate axis nn = Drive-ID	Reserved!

7.7.1.3 "Index offset" specification for Ring-0 functions (Index group 0x1200)

Index offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Note
0x00000020	Write	every	VOID	1		Clear cycle time error counter SEC & SPP	Reserved!

7.7.2 Specification Channels

7.7.2.1 "Index offset" specification for channel parameter (Index group 0x2000 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1		Channel ID	
0x00000002	Read	every	UINT8[30+1]	1		Channel name	
0x00000003	Read	every	UINT32	1	ENUM	Channel type [▸ 355]	
0x00000004	Read	every	UINT32	1	ENUM	Interpreter type [▸ 355]	
0x00000005	Read	every	UINT32	1		program load buffer size in bytes	
0x00000006	Read	every	UINT32	1		Program no. according to job list	
0x00000007	Read/Write	every	UINT32	1	ENUM	Set Load log mode [▸ 355]	
0x00000008	Read/Write	every	UINT32	1	ENUM	Set Trace mode [▸ 355]	
0x00000009	Read/Write	every	UINT32	1		RESERVED	
0x0000000A	Read/Write	every	UINT32	1	0/1	Records all feeder entries in a log file named "TcNci.log"	
0x0000000B	Read/Write	every	UINT32	1	0/1	Channel specific level for NC logger messages 0: errors only 1: all NC messages	From TwinCAT V2.8 B747 V2.9 B948

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
0x00000010	ReadWrite	every	**Write**				
			{				
			UINT32	1	0..159	Start index of the M-function.	
			UINT32	1	1..160	Number of M-functions to be read	
			}				
			Read[number]				
			{				
			UINT8	1	0..159	Rule bit mask of the M-function	
			INT32[10]	1	-1..159	Number of M-functions to be cleared	
			}				
0x00000011	Write	Interpolation				Write M-function description	From TC V2.9 B803 Internal use only!
0x00000012	Read/Write	Interpolation	LREAL64	1		Factor for G70	from TC V2.9 Build 803
0x00000013	Read/Write	Interpolation	LREAL64	1		Factor for G71	from TC V2.9 Build 803
0x00000014/	Write	Interpolation	{			axes user symbols	Not yet released
			char[32]			user symbol (null-terminated)	
			char[10]			system symbol (null-terminated)	
			}				
0x00000015	Read/Write	Interpolation	UINT16 resp. UINT32	1	0/1 Default: FALSE	Activation of default G-code	From TC 2.11R3 B2241
0x00000021	Read	every	UINT32	1		Goup ID (only explicit for 3D and FIFO channel)	
0x00000031	Read/Write	Interpolation	UINT16	1		Standard Output Port of the Interpreter	Reserved function, no standard!
0x00000032	Read/Write	Interpolation	UINT16	1	0/1	Cartesian tool offset entry	Reserved function, no standard!
0x00000040	Read/Write	Interpolation	{			Target address of interpreter hooks	Reserved function, no standard!
			char[6]			Ams Net ID	
			UINT16			Port	
			UINT32			Index group	
			UINT32			Index offset	
			}				
0x00000050	Read/Write	Interpolation	UINT32	1	ENUM	Reaction if at the radius compensation a bottle neck is recognized	From TwinCAT Version 2.8

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
						0: Error and abort 1: Note & correction 2: Only note, without outline modulation	
0x00000051	Read/Write	Interpolation	UINT32	1	1..24	Look ahead for bottleneck detection	From TwinCAT Version 2.8
0x00000052	Read/Write	Interpolation	UINT32	1	0/1	Chamfer on/off	Reserved function, no standard!
0x00000053	Read/Write	Interpolation	UINT32	1		Activation for reading the currently effective interpolation rules, zero offset shifts and rotation 0: off 1: on	From TC V2.10 B1303
0x00000054	Read/Write	Interpolation	UINT32	1	0/1	Retrace on/off	Reserved function, no standard!
0x00000055	Read/Write	Interpolation	UINT32[4]	1		Configuration of the cyclic channel interface for UINT32 Up to 4 index offsets can be configured.	From TC V2.10 B1320
0x00000056	Read/Write	Interpolation	UINT32[4]	1		Configuration of the cyclic channel interface for LREAL Up to 4 index offsets can be configured.	From TC V2.10 B1320
0x00010K0L	Read/Write	every	REAL64	e.g. mm	±MAX REAL64	Value for zero offset shift (NPV)	
					[1..3]	axis index: K=1 → X K=2 → Y K=3 → Z	
					[1..0xA]	L=1 → G54F L=2 → G54G L=3 → G55F ...	
0x0002ww00	Read/Write	every	UINT16			Tool number: Values for tool compensation	
0x0003ww00	Read/Write	every	UINT16		[1...50]	Tool type: ww = Tool 1...50	
0x0004wwnn	Read/Write	every	REAL64		[1...14]	Parameter: nn = Index 1...14	
0x000500gg	Read/Write	every	REAL64	e.g. mm	≥ 0 (value) [1...9] (g)	Radius of the tolerance sphere	

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
						gg = group of the Channel (default: 1)	

7.7.2.2 "Index offset" specification for channel state (Index group 0x2100 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	INT32	1	ENUM	Error code channel	
0x00000002	Read	every	UINT32	1		number of groups in the Channel	
0x00000003	Read	every	UINT32	1	ENUM	Interpreter state [▸ 356]	Cannot be traced by oscilloscope!
0x00000004	Read	every	UINT32	1		Interpreter operation mode (interpreter/ channel operation mode)	
0x00000005	Read	every	UINT32	1		currently loaded program	
0x00000007	Read	every	UINT8[...]	1		Program name of the currently loaded program (100 characters, null-terminated)	Max. 100 characters, null-terminated
0x00000008	Read	Interpreter	UINT32	1	[0,1]	Interpreter simulation mode 0: off (default) 1: on	from V2.9 B946 Cannot be traced by oscilloscope!
0x00000010	Read	Interpreter	UINT32	1		Text index If the interpreter is in the aborted state, the current text index can be read out here	Cannot be traced by oscilloscope!
0x00000011	ReadWrite	Interpreter	**Write**				Cannot be traced by oscilloscope!
			UINT32	1		Textindex	
			Read				
			UINT8[..]	1		Line of the NC part program from the text index	
0x00000012	Read	Interpreter	{				From TC V2.10 B1243
			UINT32	1		Current display for 1: SEC 2: Interpreter 3: Error offset	
			UINT32	1		Fileoffset	
			UINT8[260]	1		path + program name	
			}				
0x00000013	Read	Interpreter	UINT32[18]			Display for currently effective G-code	from TC V2.10 B1303

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
0x00000014/	Read	Interpreter	{			Determines the currently effective zero offset shift	from TC V2.10 B1303
			UINT32	1		block counter	
			UINT32			Dummy	
			LREAL[3]	1		Zero offset shift G54..G57	
			LREAL[3]	1		Zero offset shift G58	
			LREAL[3]	1		Zero offset shift G59	
			}				
0x00000015	Read	Interpreter	{			Determines the currently effective rotation	from TC V2.10 B1303
			UINT32	1		block counter	
			UINT32	1		Dummy	
			LREAL[3]	1		rotation of X, Y & Z in degrees	
			}				
0x00000016	Read	Interpreter	UINT32	1	[0,1]	Feeder info	Internal usage no standard
0x00000100	Read	every	UINT32[number]	1	[0, 1...255]	Returns the respective axis ID in the channel	Number: [1...255] Axis IDs: [0, 1...255] Cannot be traced by oscilloscope!

7.7.2.3 "Index offset" specification for channel functions (Index group 0x2200 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Write	every	UINT32	1		load NC program by program number	
0x00000002	Write	every	VOID			start Interpreter	
0x00000003	Write	every	VOID			RESERVED	
0x00000004	Write	every	UINT8[...]			load NC program by name. The standard NC path does not also have to be given, although it may. Other paths are also permitted.	
0x00000005	Write	every	UINT16	ENUM	see appendix Interpreter operation modes [▸ 355]	Set interpreter operation mode (interpreter/ channel operation mode)	From TwinCAT V2.9 Buld 901
0x00000006	Write	Interpreter	UINT8[...]			set path for subroutines	from TwinCAT V2.9 Build 1001
0x00000008	Write	Interpreter	UINT32	1		Interpreter simulation mode:	Not yet released

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
						0: off (default) 1: on	
0x0000000F	Write	every	VOID			RESERVED	
0x00000010	Write	every	VOID			"Reset" Channel	
0x00000011	Write	every	VOID			"Stop" Channel	
0x00000012	Write	every	VOID			"Retry" Channel (restart Channel)	Not implemented!
0x00000013	Write	every	VOID			"Skip" Channel (skip task/ block)	Not implemented!
0x00000014/ 0x00000015	Write	every	{			"Enable Retrace" / "Disable Retrace"	Reserved function, no standard!
			UINT32	1	>0	Feeder processing direction: 1: forward 2: backward	
			UINT32	1	≥ 0	Entry index	
			REAL64[3]	mm	±∞	Pos. of the main axes X, Y, Z	
			REAL64[5]	mm	±∞	Pos. of the auxiliary axes Q1, ..., Q5	
			}				
0x00000018	Write	Interpreter	_ST_ItpBlockSearchParams			Enable Blocksearch	
0x00000019	Write	Interpreter	VOID			StepOnAfterBlockSearch	
0x00000020	Write	every	VOID			"Save" zero offset shift (NPV)	
0x00000021	Write	every	VOID			"Load" zero offset shift (NPV)	
0x00000022	Write	every	VOID			"Save" tool compensations	
0x00000023	Write	every	VOID			"Load" tool compensations	
0x00000024	Write	Interpolation	{			Saves a snapshot of the interpreter in a specified file	From TwinCAT V2.9 Build 1002
			char[32]			filename in the TwinCAT\CNC directory	
			UINT32	1	0..1	Mask: 0x1: R parameters 0x2: Zero offset shifts (from Build 1235) 0x4: Tool descriptions (from Build 1235)	
			}				

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
0x00000025	Write	Interpolation	{			Reads snapshot from a specified file into the interpreter	From TwinCAT V2.9 Build 1002
			char[32]			filename in the TwinCAT\CNC directory	
			UINT32	1	0..1	Mask: 0x1: R parameters 0x2: Zero offset shifts (from Build 1235) 0x4: Tool description (from build 1235)	
			}				
0x00000026	Write	Interpolation	VOID			set all tool parameters (incl. type and number) to null	From TwinCAT V2.9 Build 1031
0x00000027	Write	Interpolation	VOID			Set all zero offset shifts to null	From TwinCAT V2.9 Build 1031
0x00000030	Write	every	VOID			restart (Go Ahead) of the Interpreter after programmed Interpreter stop	
0x00000040	Write	every	VOID			Triggerevent for deletion of any remaining travel in the NCI	
0x00000041	Write	every				RESERVED for measuring event	
0x00000050	Write	Interpolation	VOID	1		Set ExecIdleInfo in the interpreter	Reserved function, no standard!
0x00000051	Write	Interpolation	UINT32	1		Set block skipping mask in the interpreter Parameter: SkippingMask	Reserved function, no standard!
0x00000052	Write	Interpolation	UINT32	1		Set ItpOperationMode in the interpreter Parameter: Mask of the operation mode	Reserved function, no standard!
0x00000053	Write	Interpolation	VOID			Set ScanningFlag in the NC device	Reserved function, no standard!
0x00000054	Write	Interpolation				*ScanPosition*	Reserved function, no standard!
			double[8]			Position	
0x00000055	Write	Interpolation				Reserved	
0x00000056	Write	Interpolation	VOID			Set interpreter in aborted state	Reserved function, no standard!

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Note
0x00000060	Write	Interpolation	UINT16	1	0..159	Manual reset of a fast M-function	

7.7.2.4 "Index offset" specification for cyclic channel process data (Index group 0x2300 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000000	Read	every (PLC→NC)	{150 Byte}		STRUCT s. Channel interface [▶ 251]	CHANNEL STRUCTURE (PLC→NC)	The associated PLC structure is: NciChannelFromPlc Old structure: PLCTONC_CHANNELSTRUCT
0x00000001	Read	every	UINT8[...] min. 30 Byte	1		Interpreter program display	Cannot be traced by oscilloscope!
0x00000002	Read/Write	every (PLC→NC)	UINT32	%	[0...1000000]	Speed override Channel (axes in the Channel)	1000000 = 100%
0x00000003	Read/Write	every (PLC→NC)	UINT32	%	[0...1000000]	Speed override spindle	1000000 = 100%
0x00000080	Read	every (NC→PLC)	{150 Byte}		STRUCT s. Channel interface [▶ 251]	CHANNEL STRUCTURE (NC→PLC)	The current associated PLC structure is: NciChannelToPlc old structure: NCTOPLC_CHANNELSTRUCT
0x10000000 + Register index	Read/Write	every	REAL64	1	[0...999]	R-parameter of the interpreter	Cannot be traced by oscilloscope!
0x20000001	Read	every	UINT8[...] min. 30 Byte	1	[1...9]	Program display of the group processing (SAF)	Cannot be traced by oscilloscope!

7.7.3 Specification Groups

7.7.3.1 "Index offset" specification for group parameter (Index group 0x3000 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1		Group ID	
0x00000002	Read	every	UINT8[30+1]	1		Group name	
0x00000003	Read	every	UINT32	1		Group type	
0x00000004	Read	every	UINT32	µs		SEC-cycle time group	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000005	Read	every	UINT32	µs		SVB-cycle time group	
0x00000006	Read/Write	every	UINT16	1	0/1	Single block operation mode?	
0x0000000B	Read	every	UINT32	1		size of the SVB table (max. number of SVB entries	
0x0000000C	Read	every	UINT32	1		size of the SEC table (max. number of SEC entries	
0x00000010	Read/Write	every	UINT32	1	[1,2...32]	Internal SEC cycle time divisor (divides the internal SEC cycle time by this factor)	Default: 1
0x00000021	Read	Channel: every	UINT32	1		Channel ID	
0x00000022	Read	Channel: every	UINT8[30+1]	1		Channel name	
0x00000023	Read	Channel: every	UINT32	1		Channel type	
0x00000024	Read	Channel: every	UINT32	1	>0	Number in the Channel	
0x00000500	Read/Write	DXD group	INT32	ENUM	[0, 1]	Curve velocity reduction method [▸ 23] 0: Coulomb scattering 1: Cosine law 2: VeloJump	
0x00000501	Read/Write	DXD group	REAL64	1	[0.0...1.0]	Velocity reduction factor C0 transition (continual course, but neither once nor twice differentiable)	
0x00000502	Read/Write	DXD group	REAL64	1	[0.0...1.0]	Velocity reduction factor C1 transition (continual course and once differentiable)	
0x00000503	Read/Write	DXD group	REAL64	Degrees	[0.0...180.0]	Critical angle on the segment transition "Low" (must genuinely be smaller than or equal to the velocity reduction angle C0)	
0x00000504	Read/Write	DXD group	REAL64	Degrees	[0.0...180.0]	Critical angle on the segment transition "High" (must genuinely be smaller than or equal to the	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
						velocity reduction angle C0)	
0x00000505	Read/Write	DXD group	REAL64	mm/s	≥ 0	Minimum velocity that must be maintained at segment transitions in spite of any possible velocity reduction.	
0x00000506	Read/Write	DXD group	REAL64	e.g. mm	[0.0...1000.0]	Radius of the tolerance sphere for smoothing	Not implemented!
0x00000507	Read/Write	DXD group	REAL64	1		Velocity reduction factor C2 transition	
0x00000508	Read/Write	DXD group	UINT32	1	0/1	Enables calculation of the total remaining path length	
0x00000509	Read/Write	DXD group	UINT16	1	0/1	General activation of the software limit position monitoring for the main axes (X, Y, Z) (see encoder parameters)	From TwinCAT V2.9 B959
0x0000050A	Read/Write	DXD group	UINT32	1	0/1	NCI Override type 0: Related to internal reduced velocity (without iteration) 1: Related to original external (programmed) velocity	From TwinCAT V2.9 B948
0x0000050B	Read/Write	DXD group	UINT16	1	0/1	NCI override 0: Override limited to 100% 1: Override >100% possible	From TwinCAT V2.10 B1226
0x00000510	Read/Write	DXD group	REAL64	1	≥ 0	For reduction method VeloJump [▸ 247] Reduction factor for C0 transitions: X axis	Not implemented!
0x00000511	Read/Write	DXD group	REAL64	1	≥ 0	For reduction method VeloJump Reduction factor for C0 transitions: Y axis	Not implemented!

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000512	Read/Write	DXD group	REAL64	1	≥ 0	For reduction method VeloJump Reduction factor for C0 transitions: Z axis	Not implemented!
0x00000513	Read/Write	DXD group	LREAL64	1	]0.0..1.0[	Blending for auxiliary axes: If the resulting path velocity is smaller than the programmed one multiplied with this factor, an accurate stop is inserted	From TwinCAT V2.11 B1552
0x00000604	Read/Write	Encoder group	REAL64	e.g. mm/s	[0.0...1000.0]	Velocity window resp. standstill window	Base Unit / s
0x00000605	Read/Write	Encoder group	REAL64	s	[0.0...60.0]	Filter time for standstill window in seconds	
0x00000606	Read/Write	Encoder group	REAL64	s	[0.0...60.0]	Dead time compensation master/slave coupling ("angle pre-control")	
0x00000701	Read	FIFO group	UINT32	1	[1...8] or [1...16]	FIFO dimension (m = number of axes) (from TC 2.11 Build 1547 the FIFO dimension has been increased from 8 to 16)	(n x m)-FIFO boot data!
0x00000702	Read	FIFO group	UINT32	1	[1...10000]	FIFO size (length) (n = number of FIFO entries)	(n x m)-FIFO boot data!
0x00000703	Read	FIFO group	UINT32	1	[0, 1, 4]	Interpolation type for FIFO setpoint generator 0: INTERPOLATIONTYPE_LINEAR (Default) 1: INTERPOLATIONTYPE_4POINT 4: INTERPOLATIONTYPE_CUBICSPLINE (with 6 points)	From TwinCAT 211R3 Build 2257
0x00000704	Read/Write	FIFO group	UINT32	1	[1, 2]	Override type for FIFO setpoint generator	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
						Type 1: OVERRIDETYPE_INSTANTANEOUS (Default) Type 2: OVERRIDETYPE_PT2	
0x00000705	Read/Write	FIFO group	REAL64	s	> 0.0	P-T2-time for override change (T1=T2=T0)	
0x00000706	Read/Write	FIFO group	REAL64	s	≥ 0.0	Time delta for two sequenced FIFO entries (FIFO entry timebase)	
0x00000801	ReadWrite	Kinematic group	**Write**			Calculation of the kinematic forward transformation for the positions (ACS -> MCS)	
			{				
			REAL64[8]	e.g. degrees	±∞	Positions of the ACS axes (Axis Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				
			Read				
			{				
			REAL64[8]	e.g. mm	±∞	Positions of the MCS axes (Machine Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				
0x00000802	ReadWrite	Kinematic group	**Write**			Calculation of the kinematic inverse transformation for the positions (MCS -> ACS)	
			{				
			REAL64[8]	e.g. mm	±∞	Positions of the MCS axes (Machine Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				
			Read				
			{				

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
			REAL64[8]	e.g. degrees	±∞	Positions of the ACS axes (Axis Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				

7.7.3.2 "Index offset" specification for group state (Index group 0x3100 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	INT32	1	ENUM	Error code group	
0x00000002	Read	every	UINT32	1		Number of master axes	
0x00000003	Read	every	UINT32	1		Number of slave axes	
0x00000004	Read	every	UINT32	1	s. ENUM	SVB group state (state)	
0x00000005	Read	every	UINT32	1	s. ENUM	SEC group state (main state)	
0x00000006	Read	every	UINT32	1	s. ENUM	Moving state (state)	
0x00000007	Read	every	UINT32	1	s. ENUM	SEC sub-group state (sub state)	
0x00000008	Read	every	UINT32	1	s. ENUM	Calibration state (state)	
0x00000009	read	every	UINT32	1	s. ENUM	Coupling state (state)	Cannot be traced by oscilloscope!
0x0000000A	read	every	UINT32	1	≥0	Coupling table index	Cannot be traced by oscilloscope!
0x0000000B	read	every	UINT32	1	≥0	current number of SVB entries/ tasks	*Symbolic access:* *'SvbEntries' (DXD)*
0x0000000C	read	every	UINT32	1	≥0	Current number of SEC entries/ tasks	*Symbolic access:* *'SafEntries' (DXD)*
0x0000000D	read	every	UINT32	1		Current block number (active only for interpolation group)	*Symbolic access:* *'BlockNumber' (DXD)*
0x0000000E	read	every	UINT32	1	≥0	current number of free SVB entries/tasks	From TwinCAT V2.9 B903 Not oscilloscopeabl e!
0x0000000F	read	every	UINT32	1	≥0	Current number of free SEC entries/tasks	From TwinCAT V2.9 B903 Not oscilloscopeabl e!
0x00000011	read	every	UINT16	1	0/1	Emergency Stop (E-Stop) active?	Cannot be traced by oscilloscope!

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000110	read	PTP group	{			Internal NC information (resolutions)	Reserved
			REAL64	e.g. mm	± ∞	ExternalEndPosition	
			REAL64	e.g. mm/s	>0	ExternalTargetVelocity	
			REAL64	e.g. mm/s^2	>0	ExternalAcceleration	
			REAL64	e.g. mm/s^2	>0	ExternalDeceleration	
			REAL64	e.g. mm/s^3	>0	ExternalJerk	
			UINT32	1	>0	ExternalOverrideType	
			REAL64	e.g. mm	± ∞	InternalEndPosition	
			REAL64	e.g. mm/s	>0	InternalTargetVelocity (refers to 100 %)	
			REAL64	%	[0 ... 100]	InternalActualOverride	
			REAL64	e.g. mm/s^2	>0	InternalAcceleration	
			REAL64	e.g. mm/s^2	>0	InternalDeceleration	
			REAL64	e.g. mm/s^3	>0	InternalJerk	
			REAL64	e.g. mm	>0	PositionResolution	
			REAL64	e.g. mm/s	≥0	VelocityResolution	
			REAL64	e.g. mm/s^2	≥0	AccelerationResolution	
			REAL64	e.g. mm/s	≥0	VelocityResolutionAtAccelerationZero	
			}				
0x00000500	read	DXD group	REAL64	e.g. mm	≥ 0	Path rest way (remaining arc length) on the current path segment	*Symbolic access:* *'SetPathRemLength'*
0x00000501	read	DXD group	REAL64	e.g. mm	≥ 0	Racked out arc length on the current path segment	*Symbolic access:* *'SetPathLength'*
0x00000502	read	DXD group	REAL64	e.g. mm/s	≥ 0	Current path set velocity	*Symbolic access:* *'SetPathVelo'*
0x00000503	read	DXD group	REAL64	e.g. mm/s^2	± ∞	Current path set acceleration	*Symbolic access:* *'SetPathAcc'*
0x00000504	read	DXD group	REAL64	e.g. mm/s^2	≥ 0	amount of the current vectorial set acceleration	*Symbolic access:* *'SetPathAbsAcc'*
0x00000505	read	DXD group	REAL64	e.g. mm/s	≥ 0	Maximum segment end path set velocity	*Symbolic access:* *'SetPathVeloEnd'*
0x00000506	read	DXD group	REAL64	e.g. mm/s	≥ 0	Segment maximum path set velocity	*Symbolic access:* *'SetPathVeloMax'*

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000507	read	DXD group	REAL64	e.g. mm	≥ 0	Current relative braking distance based on the current arc length	*Symbolic access:* *'SetPathStopDist'*
0x00000508	read	DXD group	REAL64	e.g. mm	± ∞	Safety distance = segment arc length - current arc length - relative braking distance	*Symbolic access:* *'SetPathSecurityDist'*
0x00000509	read	DXD group	REAL64	1	0/1	Segment transition	*Symbolic access:* *'SetPathSegmentChange'*
0x0000050A	read	DXD group	REAL64	%	[0 ... 100]	Path velocity override	*Symbolic access:* *'SetPathOverride'*
0x00000511	read	DXD group	REAL64	e.g. mm/s	≥ 0	Amount of the path actual velocity	*Symbolic access:* *'ActPathAbsVelo'*
0x00000512	read	DXD group	REAL64	e.g. mm/s^2	± ∞	Path actual acceleration on the current segment	*Symbolic access:* *'ActPathAcc'*
0x00000513	read	DXD group	REAL64	e.g. mm/s^2	≥ 0	Amount of the path actual acceleration on the current segment	*Symbolic access:* *'ActPathAbsAcc'*
0x00000514	read	DXD group	REAL64	e.g. mm	± ∞	Position error on the path in a tangential direction (signed for lead and lag)	*Symbolic access:* *'PathDiffTangential'*
0x00000515	read	DXD group	REAL64	e.g. mm	≥ 0	Position error on the path in orthogonal direction	Symbolic access: *'PathDiffOrthogonal'*
0x00000520	read	DXD-group	REAL64	1	≥ 0	Covered arc length of the current segment, normalized to 1.0	
0x00000521	read	DXD-group	REAL64	1	0/1	Change of partial segment (radius of tolerance sphere)	
0x00000522	read	DXD group	REAL64	1	≥ 0	Total remaining path length to the last geometry entry or the next accurate stop. Refers to group parameter 0x508.	
0x00000523	read	DXD group	REAL64	1	≥ 0	Programmed velocity of the current segment	From TC V2.9 B1031

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000530	read	DXD group	{			Current or last target position of the main axes X, Y and Z	
			REAL64	e.g. mm	± ∞	Target position X-axis	
			REAL64	e.g. mm	± ∞	Target position Y-axis	
			REAL64	e.g. mm	± ∞	Target position Z-axis	
			}				
0x00000531	read	DXD group	{			Current or last target position of the auxiliary axes Q1 to Q5	
			REAL64[5]	e.g. mm	± ∞	Target position of axis Q1 to Q5	
			}				
0x00000532	read	DXD group	{			Reads path length, H parameter and Entry ID of the next 11 segments in relation to the current DC time	From TC 2.11 B2226
			UINT32			DC Time	
			UINT32			Reserved	
			PreViewTab[11]			11*24 Bytes	
			}				
			PreViewTab				
			{				
			REAL64	e.g. mm		Segment length	
			UINT32	1		block number	
			UINT32	1		H-Parmeter	
			UINT32	1		Entry ID	
			UINT32	1		Reserved	
			}				
0x0000054n	read	DXD group	REAL64	1	0/1	Within the tolerance sphere of the auxiliary axis n = 1..5 Number of the auxiliary axis (not axis ID)	From TC V2.9 B932
0x00000550	read	DXD group	{			Reads the axis IDs within a 3D group:	
			UINT32	1	[0, 1...255]	X axis ID	
			UINT32	1	[0, 1...255]	Y axis ID	
			UINT32	1	[0, 1...255]	Z axis ID	
			}				

0x00000552	read	DXD group
		FIFO group
		Kinematic group
			{
			UINT32[m]
			}
				1	[0, 1...255]	Axis allocation of the group:
						1st axis ID – mth axis ID
						m: Dimension of the 3D group with main and auxiliary axes	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
						(X, Y, Z, Q1, Q2, Q3, Q4, Q5) or the FIFO group or the ACS axes of the kinematic group	
0x00000553	read	Kinematic group	{			Reading the axis allocation (ID's) inside the kinematic group:	
			UINT32[8]	1	[0, 1...255]	MCS axis IDs (machine coordinate system)	
			UINT32[8]	1	[0, 1...255]	ACS axis IDs (axis coordinate system)	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve (NEW)	
			}				
0x00000556	read	DXD group	ST_ItpBlockSearchData			Reading the block search data	
0x0000056n	read	DXD group	REAL64	1	± ∞	Current position error of the auxiliary axis within the tolerance sphere (set value side only) Only for auxiliary axes n = 1..5 Number of the auxiliary axis (not axis ID)	From TC V2.9 B932

7.7.3.3 "Index offset" specification for group functions (Index group 0x3200 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Write	every	VOID			Reset group	
0x00000002	Write	every	VOID			Stop group	
0x00000003	Write	every	VOID			Clear group (buffer/task)	
0x00000004	Write	PTP group, 3D group	{			Emergency Stop (E-Stop) (Emergency stop with regulated ramp)	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration (must be greater than or equal to the original deceleration)	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
						(must be greater than or equal to the original jerk)	
			}				
0x00000005	Write	PTP group	{			Parametrized stop (with regulated ramp)	Reserved function, no standard!
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			}				
0x00000006	Write	PTP group, 3D group	VOID			"Step on" after Emergency Stop (E-Stop)	
0x00000050	Write	PTP group 3D group	{			Axis allocation of the group:	
			UINT32	1	[0, 1...255]	X axis ID	
			UINT32	1	[0, 1...255]	Y axis ID	
			UINT32	1	[0, 1...255]	Z axis ID	
			}				
0x00000051	Write	PTP group 3D group FIFO group	{			Axis allocation of the group:	
			UINT32	1	[1...255]	Axis ID	
			UINT32	1	[0 ... (m-1)]	Place index of the axis in the group m: Group dimension (PTP: 1;DXD: 3, FIFO: 8 or 16) (from TC 2.11 Build 1547 the FIFO dimension has been increased from 8 to 16)	
			}				

0x00000052	Write	3D group
		FIFO group
			{
			UINT32[m]
			}
				1	[0, 1...255]	Axis allocation of the group:
						1st axis ID – mth axis ID
						m: Dimension of the 3D group (X, Y, Z, Q1, Q2, Q3, Q4, Q5) or FIFO group	

0x00000053	Write	3D group FIFO group Kinematics group	VOID			Delete the 3D axis allocation, FIFO axis allocation or Kinematic axis allocation and return of the axes to their own PTP groups	
0x00000054	Write	Kinematic group	{			axis allocation of the kinematic group:	
			UINT32[8]	1	[0, 1...255]	MCS axis IDs (machine coordinate system)	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
			UINT32[8]	1	[0, 1...255]	ACS-axis ID's (Axis Coordinate System)	
			UINT32	1	≥ 0	Reserved	
			UINT32	1	≥ 0	Reserved (NEW)	
			}				
0x00000060	ReadWrite	3D group		1		internal "feed group" command ("Feeder")	Execute command!
0x00000061	ReadWrite	3D group		1		internal "feed group" command ("Feeder")	Execute command!
0x00000110	Write	1D group	VOID			Reference 1D group ("calibration")	
0x00000111	Write	1D group	{			New end position 1D group	
			UINT32	ENUM	s. appendix	End position type (see appendix)	
			REAL64	e.g. mm	±∞	new end position (target position)	
			}				
0x0000011A	Write	1D group	{			set actual position 1D group	**Caution when using!** Always to SEC Port 501!
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	actual position for axis	
			}				
0x0000011B	Write	1D group	UINT32	1	0/1	Set reference flag ("calibrate flag")	**Caution when using!**
0x00000120	Write	1D group	{			start 1D group (standard start):	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	mm/s	≥ 0.0	Required velocity	
			}				
0x00000121	Write	1D group (SERVO)	{			Start 1D group (Advanced start):	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	mm/s	≥0.0	Required velocity	
			UINT32	1	0/1	Standard acceleration?	
			REAL64	mm/s^2	≥ 0.0	Acceleration	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
			UINT32	1	0/1	Standard deceleration?	
			REAL64	mm/s^2	≥ 0.0	Deceleration	
			UINT32	1	0/1	Standard jerk?	
			REAL64	mm/s^3	≥ 0.0	Jerk	
			}				
0x00000122	Write	1D group (MW-SERVO)	{			Start 1D group (special start):	Reserved start function, no standard!
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	mm/s	≥0.0	required start velocity	
			REAL64	e.g. mm	±∞	Position for a new velocity level	
			REAL64	mm/s	≥0.0	new end velocity level	
			UINT32	1	0/1	Standard acceleration?	
			REAL64	mm/s^2	≥0.0	Acceleration	
			UINT32	1	0/1	Standard deceleration?	
			REAL64	mm/s^2	≥0.0	Deceleration	
			UINT32	1	0/1	Standard jerk?	
			REAL64	mm/s^3	≥0.0	Jerk	
			}				
0x00000126	Write	1D group	{			Start drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000127	Write	1D group	VOID			Stop drive output	
0x00000128	Write	1D group	{			Change the drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000130	Write	1D group (SERVO)	{			1D section compensation (SERVO):	
			UINT32	ENUM	s. appendix	Compensation type (see appendix)	
			REAL64	mm/s/s	≥ 0.0	Max. acceleration increase	
			REAL64	mm/s/s	≥ 0.0	Max. deceleration increase	
			REAL64	mm/s	≥ 0.0	Max. increase velocity	
			REAL64	mm/s	≥ 0.0	Base velocity for the process	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	e.g. mm	±∞	Path difference to be compensated	
			REAL64	e.g. mm	≥ 0.0	Path distance for compensation	
			}				
0x00000131	Write	1D group SERVO	VOID			Stop section compensation (SERVO)	
0x00000140 (0x00n00140)	Write	Master/Slave coupling: 1D group (SERVO)	{			Master/Slave coupling (SERVO):	Extension for "flying saw"! Angle > 0.0 and £ 90.0 degrees (Parallel saw: 90.0 degrees)
			UINT32	ENUM	s. appendix	Slave type/ coupling type (see appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis/ group	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Gear ratio FlySawVelo: Reserve FlySaw: Abs. synchron position master [mm]	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve FlySawVelo: Reserve FlySawPos: Abs. synchron position slave [mm]	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve FlySawVelo: Angle of inclination in [DEGREES] FlySawPos: Angle of inclination in [DEGREES]	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve FlySawVelo: Gear ratio FlySawPos: Gear ratio	
			}				
0x00000141	Write	Master/slave decoupling: 1D group	VOID			Master/Slave decoupling (SERVO)	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
		(SERVO)					
0x00000142	Write	Master/slave parameter 1D group (SERVO)	{			Change of the coupling parameters (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Gear ratio	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
			}				
0x00000144	Write	Slave stop 1D group (SERVO)	VOID			Stop the "flying saw" (SERVO)	Only for "flying saw"
0x00000149	Write	Slave tables 1D group (SERVO)	REAL64	1	±∞	set the slave table scaling of a solo table coupling (SERVO)	Only for Solo table slave
0x00000150	Write	1D group	VOID			Deactivate complete 1D group/axis (disable)	
0x00000151	Write	1D group	VOID			Activate complete 1D group / axis (enable)	
0x00000160	Write	1D group	VOID			Deactivate drive output of the 1D group (disable)	
0x00000161	Write	1D group	VOID			Activate drive output of the 1D group (enable)	
0x00000362	Write	High/low speed group	UINT16	1	0/1	Release parking brake? 0: automatic activation (default) 1: mandatorily always released!	
0x00000701	Write	FIFO group	VOID			Start FIFO group (FIFO table must first have been filled)	(n*m)-FIFO
0x00000710	Write	FIFO group	{ REAL64[x*m] }	e.g. mm	±∞	Write x FIFO entries (lines): (x*m)-values (one or more lines) n: FIFO length (number of lines)	Only possible on a line-by-line basis! (integer multiple)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
						m: FIFO dimension (number of columns) Value range x: [1 ... n]	

0x00000711	Write	FIFO group	{
			REAL64[x*m]
			}
				e.g. mm	±∞	Overwrite the last x FIFO entries (lines):
						(x*m)-values (one or more lines)
						n: FIFO length (number of lines)
						m: FIFO dimension (number of columns)
						Value range x: [1 ... n]
							Only possible on a line-by-line basis!
							(integer multiple)

0x00000801	Write	Kinematic group	VOID			Start Kinematic group	Reserved function, no standard!

7.7.4 Specification Axes

7.7.4.1 "Index offset" specification for axis parameter (Index group 0x4000 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n00000	read	every (Structure for all axis parameters)	{			General AXIS PARAMETER STRUCTURE (NC/CNC), also contains the sub-elements such as encoder, controller and drive (s. MC_ReadParameterSet in TcMc.lib)	
			UINT32	1		Axis ID	
			UINT8[30+1+1]	e.g. mm		Axis name	
			UINT32	1		Axis type	
			. . .	. . .	. . .	. . .	
			}			512 bytes	
0x00000001	read	every	UINT32	1		Axis ID	
0x00000002	read	every	UINT8[30+1]	1		Axis name	
0x00000003	read	every	UINT32	ENUM		Axis type	
0x00000004	read	every	UINT32	µs		Cycle time axis (SEC)	
0x00000005	read	every	UINT8[10+1]	1		physical unit	
0x00000006	Read/Write	every	REAL64	e.g. mm/s		ref. velocity in cam direction	
0x00000007	Read/Write	every	REAL64	e.g. mm/s		ref. velocity in sync direction	
0x00000008	Read/Write	every	REAL64	e.g. mm/s		velocity hand slow	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000009	Read/Write	every	REAL64	e.g. mm/s		velocity hand fast	
0x0000000A	Read/Write	every	REAL64	e.g. mm/s		velocity rapid traverse	
0x0000000F	Read/Write	every	UINT16	1	0/1	position range monitoring?	
0x00000010	Read/Write	every	REAL64	e.g. mm		position range window	
0x00000011	Read/Write	every	UINT16	1	0/1	motion monitoring?	
0x00000012	Read/Write	every	REAL64	s		motion monitoring time	
0x00000013	Read/Write	every	UINT16	1	0/1	loop?	
0x00000014	Read/Write	every	REAL64	e.g. mm		Looping distance (±)	
0x00000015	Read/Write	every	UINT16	1	0/1	target position monitoring?	
0x00000016	Read/Write	every	REAL64	e.g. mm		target position window	
0x00000017	Read/Write	every	REAL64	s		target position monitoring time	
0x00000018	Read/Write	every	REAL64	e.g. mm		pulse way in pos. direction	
0x00000019	Read/Write	every	REAL64	e.g. mm		pulse way in neg. direction	
0x0000001A	Read/Write	every	UINT32	ENUM (≥0)		Error reaction mode: 0: instantaneous (default) 1: delayed (e.g. for Master/ Slave coupling)	From TC 2.11
0x0000001B	Read/Write	every	REAL64	s	[0...1000]	Error delay time (if delayed error reaction is selected)	From TC 2.11
0x0000001C	Read/Write	every	UINT16	1	0/1	Couple slaves via actual values if not ready to operate?	From TC 2.11
0x0000001D	Read/Write	every	REAL64	e.g. mm/s^2	[0, 0.01...1.0E10]	Acceleration for transition profile for switching from SET to ACTUAL values (fading of position): Default: 0 (the minimum of the axis accelerations is used here, i.e. MIN(Acc, Dec))	From TC 2.11 R2
0x0000001E	Read/Write	every	UINT32	ENUM (≥0)		Fast Axis Stop Signal Type: Signal type selection to force a Fast Axis Stop (s. Bit 7 from Drive->nStatus4)*"0 (SignalType_OFF)","1 (SignalType_RisingEdge)","2 (SignalType_Fa*	from TC 2.11 R3

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						llingEdge)","3 (SignalType_BothEdges)","4 (SignalType_HighActive)","5 (SignalType_LowActive)"	
0x00000020	Read/Write	every	UINT16	1	0/1	Allow motion commands for slave axis? Default: FALSE	From TC 2.11
0x00000021	Read/Write	every	UINT16	1	0/1	Allow motion commands for axes with active external setpoint generation? Default: FALSE	from TC 2.11 R2
0x00000026	Read/Write	every	UINT32	1		interpretation of the units (position, velocity, time) Bit 0: Velocity in x/min instead of x/s Bit 1: Position in thousandths of the base unit Bit 2: Modulo position display	See encoder! bit array
0x00000027	Read/Write	every	REAL64	e.g. mm/s		max. allowed velocity	
0x00000028	Read/Write	every	REAL64	e.g. mm		motion monitoring window	
0x00000029	Read/Write	every	UINT16	1	0/1	PEH time monitoring?	Posi. end and accurate stop
0x0000002A	Read/Write	every	REAL64	s		PEH monitoring time	
0x0000002B	Read/Write	every	UINT16	1	0/1	Backlash compensation?	
0x0000002C	Read/Write	every	REAL64	e.g. mm		Backlash	
0x00000030	read	every	UINT16	1	[0,1]	Persistent data e.g. for actual position and reference state of the encoder?	boot parameter
0x00000031	read	every	{ UINT8[6] UINT16 }	AmsAddr	1	Read the hardware AMS address (AMS Net ID and device port)	ALT!
0x00000031	read	every	{ UINT8[6] UINT16 UINT16 }	AmsAddr ChannelNo	1	Read the hardware AMS address (AMS Net ID and device port) and the channel number (0=Channel A, 1=Channel B)	
0x00000033	read	every	{ UINT16 ApplRequestBit UINT16 ApplRequestType	1 1 1 1	0/1 ≥0 >0 ≥0	General APPLICATION REQUEST STRUCTURE (NC/NCI),	From TC 2.11 R2

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			UINT32 ApplCmdNo UINT32 ApplCmdVersion . . . } 1024 bytes			e.g. for ApplicationHoming request (see *MC_ReadApplicationRequest* in *TcMc2.lib*)	
0x00000051	read	Channel: every	UINT32			Channel ID	
0x00000052	read	Channel: every	UINT8[30+1]			Channel name	
0x00000053	read	Channel: every	UINT32			Channel type	
0x00000054	read	Group: every	UINT32			Group ID	
0x00000055	read	Group: every	UINT8[30+1]			Group name	
0x00000056	read	Group: every	UINT32			Group type	
0x00000057	read	every	UINT32			Number of encoders	
0x00000058	read	every	UINT32			Number of controllers	
0x00000059	read	every	UINT32			Number of drives	
0x0000005A	read	every	{			read all sub-elements of an axis:	
			UINT32[9]	1	[0, 1...255]	Axis encoder IDs	
			UINT32[9]	1	[0, 1...255]	Axis controller IDs	
			UINT32[9]	1	[0, 1...255]	Axis drive IDs	
			} 108 bytes				
0x00000101	Read/Write	Servo	REAL64	e.g. mm/s^2		Acceleration	
0x00000102	Read/Write	Servo	REAL64	e.g. mm/s^2		Deceleration	
0x00000103	Read/Write	Servo	REAL64	e.g. mm/s^3		Jerk	
0x00000104	Read/Write	Servo	REAL64	s	[0.0 ... 1.0]	Deceleration time between velocity and position values of the setpoint generator in seconds	Default value: 0.0 s
0x00000105	Read/Write	Servo	UINT32	ENUM		Override type for velocity: 1: Related to internal reduced velocity (without iteration) 2: Related to original external start velocity (without iteration) 3: Related to internal reduced velocity (optimization by means of iteration) 4: Related to original external start velocity (optimization by means of iteration)	Default value: type 1

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000106	Read/Write	Servo	REAL64	1	[0.0 ... 1.0E6]	Maximum allowed velocity jump for dynamic reduction *D V = factor * min(A+, A-) * DT*	Default value: 0.0
0x00000107	Read/Write	Servo	UINT16	1	[0,1]	Activates acceleration and jerk limitation for the auxiliary axis (Q1 to Q5)	Default value: 1
0x00000108	Read/Write	Servo	REAL64	e.g. mm	[0.0..1000.0]	Radius of the tolerance sphere for the auxiliary axes	From TC V2.9 B932
0x00000109	Read/Write	Servo	REAL64	e.g. mm	[0.0..10000.0]	Maximum allowed position deviation if the tolerance sphere is reduced Only for auxiliary axes	From TC V2.9 B1013
0x0000010A	Read/Write	Servo	REAL64	e.g. mm/s^2	[0.01 ... 1.0E20]	Fast Axis Stop: Acceleration (s.a. Fast Axis Stop Signal Type)	From TC 2.11 R3
0x0000010B	Read/Write	Servo	REAL64	e.g. mm/s^2	[0.01 ... 1.0E20]	Fast Axis Stop: Deceleration (s.a. Fast Axis Stop Signal Type)	From TC 2.11 R3
0x0000010C	Read/Write	Servo	REAL64	e.g. mm/s^3	[0.1 ... 1.0E30]	Fast Axis Stop: Jerk (s.a. Fast Axis Stop Signal Type)	From TC 2.11 R3
0x00000201	Read/Write	Stepper motor	UINT32	ENUM		operation mode stepper motor	
0x00000202	Read/Write	Stepper motor	REAL64	e.g. mm/STEP	[1.0E-6 ... 1000.0]	Distance scaling of a motor step	
0x00000203	Read/Write	Stepper motor	REAL64	e.g. mm/s	[0.0 ... 1000.0]	Minimum velocity for velocity profile	
0x00000204	Read/Write	Stepper motor	UINT32	1	[0 ... 100]	Number of steps per frequency/ velocity step	
0x00000205	Read/Write	Stepper motor	UINT32	1		Motor mask as sync pulse	Not implemented!
0x00000301	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Creep distance in pos. direction	
0x00000302	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Creep distance in neg. direction	
0x00000303	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Braking distance in pos. direction	
0x00000304	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Braking distance in neg. direction	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000305	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Braking deceleration in pos. direction	
0x00000306	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Braking deceleration in neg. direction	
0x00000307	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Switching time from high to low speed	
0x00000308	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Creep distance stop	
0x00000309	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Delay time to release brake	
0x0000030A	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Pulse time in pos. direction	
0x0000030B	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Pulse time in neg. direction	
ENCODER							
0x00n10001	read	Encoder: every	UINT32	1	[1 ... 255]	Encoder ID n = 0: Standard encoder for the axis n > 0: n-th encoder for the axis (optional)	
0x00n10002	read	Encoder: every	UINT8[30+1]	1	30 characters	Encoder name	
0x00n10003	read	Encoder: every	UINT32	1	s. ENUM (>0)	Encoder type [▶ 360]	
0x00n10004	Read/Write	Encoder: every	UINT32	1	Byteoffset	Input address offset (I/O-Input-Image)	change I/O address
0x00n10005	Read/Write	Encoder: every	UINT32	1	Byteoffset	Output address offset (I/O-Output-Image)	change I/O address
0x00n10006	Read/Write	Encoder: every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	resulting scaling factor (numerator / denominator) Note: from TC 3.0 the scaling factor consists of two components – numerator and denominator (default: 1.0).	
0x00n10007	Read/Write	Encoder: every	REAL64	e.g. mm	[±1.0E+9]	Position offset	
0x00n10008	Read/Write	Encoder: every	UINT16	1	[0,1]	Encoder count direction	
0x00n10009	Read/Write	Encoder: every	REAL64	e.g. mm	[0.001 ... 1.0E+9]	modulo factor	
0x00n1000A	Read/Write	Encoder: every	UINT32	1	s. ENUM (>0)	Encoder mode [▶ 361]	
0x00n1000B	Read/Write	Encoder: every	UINT16	1	0/1	soft end min. monitoring?	
0x00n1000C	Read/Write	Encoder: every	UINT16	1	0/1	Soft end max. monitoring	
0x00n1000D	Read/Write	Encoder: every	REAL64	mm		Soft end position min.	
0x00n1000E	Read/Write	Encoder: every	REAL64	mm		Soft end position max.	
0x00n1000F	Read/Write	Encoder: every	UINT32	1	s. ENUM (≥0)	Encoder evaluation direction	s. appendix

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						[▶ 361] (enable for log. counting direction)	
0x00n10010	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for position actual value in seconds (P-T1)	
0x00n10011	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for velocity actual value in seconds (P-T1)	
0x00n10012	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for acceleration actual value in seconds (P-T1)	
0x00n10013	Read/Write	Encoder: every	UINT8[10+1]	1		physical unit	Not implemented!
0x00n10014	Read/Write	Encoder: every	UINT32	1		interpretation of the units (position, velocity, time) Bit 0: Velocity in x/min instead of x/s Bit 1: Position in thousandths of the base unit	Not implemented! bit array
0x00n10015	read	Encoder: every	UINT32	INC	[0x0... 0xFFFFFFFF]	Encoder mask (maximum value of the encoder actual value in increments) Note: From TwinCAT 2.11 R2 the encoder mask may be any numerical value (e.g. 3600000) and does not have to correspond to a continuous sequence of binary ones (2^n-1).	Read-only parameter see also "Encoder Sub Mask" parameter
0x00n10016	Read/Write	Encoder: every	UINT16	1	0/1	Actual position correction (measurement system error correction)?	
0x00n10017	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for actual position correction in seconds (P-T1)	
0x00n10019	Read/Write	Encoder: every	UINT32	1	s. ENUM (≥0)	Encoder absolute dimensioning system [▶ 362]	s. appendix
0x00n1001A	Read/Write	Encoder: every	UINT32	1	s. ENUM (≥0)	Encoder position initialization [▶ 362]	Not implemented!
0x00n1001B	Read/Write	Encoder: every	REAL64	e.g. mm	[≥0, modulo factor/2]	Tolerance window for modulo-start	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n1001C	read	Encoder: every	UINT32	1	s. ENUM (≥0)	Encoder sign interpretation (data type) [▶ 362]	
0x00n1001D	read	Encoder: every	UINT16	1	0/1	Incremental or absolute encoder? 0: Incremental encoder type 1: Absolute encoder type	
0x00n10023	Read/Write	Encoder: every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: numerator (=> scaling factor numerator / scaling factor denominator)	From TC 3.0
0x00n10024	Read/Write	Encoder: every	REAL64	1	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: denominator (=> scaling factor numerator / scaling factor denominator) Default: 1.0	From TC 3.0

0x00n10025	Read/Write	Encoder: every	{
			REAL64
			REAL64
			}
				e.g. mm/INC
				1
					[1.0E-12 ... 1.0E+30]
					[1.0E-12 ... 1.0E+30]
						Component of the scaling factor: numerator
						Component of the scaling factor: denominator
						(=> scaling factor numerator / scaling factor denominator)
							From TC 3.0

0x00n10030	Read/Write	Encoder: every	UINT32	1		Internal Encoder Control DWORD for specifying the operation modes and properties	From 211R3 B2227
0x00n10101	Read/Write	E: INC	UINT16	1	[0,1]	inverse search direction for ref.cam?	
0x00n10102	Read/Write	E: INC	UINT16	1	[0,1]	inverse search direction for sync pulse?	
0x00n10103	Read/Write	E: INC	REAL64	e.g. mm	[±1000000.0]	Reference position	
0x00n10104	Read/Write	E: INC	UINT16	1	[0,1]	distance monitoring between Ref. cams and sync pulse active?	Not implemented!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n10105	Read/Write	E: INC	UINT32	INC	[0 ... 65536]	minimum gap between Ref. cams and sync pulse in increments	Not implemented!
0x00n10106	Read/Write	E: INC	UINT16	1	[0,1]	external sync pulse?	
0x00n10107	Read/Write	E: INC	UINT32	1	s. ENUM (>0)	Reference mode	s. appendix
0x00n10108	Read/Write	E: INC	UINT32	1	[0x0000000F... 0xFFFFFFFF] Binary mask: (2^n - 1)	Encoder Sub Mask (maximum value of the absolute range of the encoder actual value in increments) Used, for example, as a reference mark for the referencing mode "Software Sync" and for the NC Retain Data "ABSOLUTE (MODULO)", "INCREMENTAL (SINGLETURN ABSOLUTE)") Note 1: The Encoder Sub Mask must be smaller than or equal to the Encoder Mask. Note 2: The Encoder Mask must be an integer multiple of the Encoder Sub Mask. Note 3: The Encoder Sub Mask must be a continuous sequence of binary ones (2^n-1), e.g. 0x000FFFFF.	NEW see also "Encoder Mask"
0x00n10110	Read/Write	E: INC (encoder simulation)	REAL64	1	[0.0 ... 1000000.0]	scaling/weight of the noise part for the simulation encoder	
CONTROLLER:							
0x00n20001	Read	Controller: every	UINT32	1	[1 ... 255]	Controller ID n = 0: Standard axis controller n > 0: n-th controller of the axis (optional)	
0x00n20002	Read	Controller: every	UINT8[30+1]	1	30 characters	Controller name	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n20003	Read	Controller: every	UINT32	1	s. ENUM (>0)	Controller type [▸ 359]	
0x00n2000A	Read/Write	Controller: every		1	s. ENUM (>0)	Controller mode	
0x00n2000B	Read/Write	Controller: every	REAL64	%	[0.0 ... 1.0]	Weighting of the velocity pre-control (default value: 1.0 = 100 %)	
0x00n20010	Read/Write	Controller: every	UINT16	1	0/1	Lag error monitoring Pos.?	
0x00n20011	Read/Write	Controller: every	UINT16	1	0/1	Lag error monitoring Velocity?	
0x00n20012	Read/Write	Controller: every	REAL64	e.g. mm		max. lag error position	
0x00n20013	Read/Write	Controller: every	REAL64	s		Max. lag error filter time position	
0x00n20014	Read/Write	Controller: every	REAL64	e.g. mm/s		max. lag error velocity	
0x00n20015	Read/Write	Controller: every	REAL64	s		Max. lag error filter time velocity	
0x00n20100	Read/Write	P/PID (Pos., (velocity)	REAL64	1	[0.0...1.0]	Maximum output limitation (±) for controller total output	(Default value: 0.5 == 50%)
0x00n20102	Read/Write	P/PID (pos.)	REAL64	e.g. mm/s/ mm	[0.0...1000.0]	Proportional gain kp or kv Unit: Base Unit / s / Base Unit	Position control
0x00n20103	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Integral action time Tn	Position control
0x00n20104	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Derivative action time Tv	Position control
0x00n20105	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Damping time Td	Position control
0x00n20106	Read/Write	PP (Pos.)	REAL64	e.g. mm/s/ mm	[0.0...1000.0]	Additional proportional gain, kp or kv respectively, that applies above a limiting velocity in percent. Unit: Base Unit / s / Base Unit	Position control
0x00n20107	Read/Write	PP (Pos.)	REAL64	%	[0.0...1.0]	Threshold velocity in percent above which the additional proportional gain, kp or kv, applies	
0x00n20108	Read/Write	P/PID (Acc.)	REAL64	s	[0.0 ... 100.0]	Proportional gain ka	Acceleration pre-control
0x00n2010D	Read/Write	P/PID	REAL64	mm	[0.0 ... 10000.0]	"dead band" for position error (control deviation)	reserved function

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						(for P/PID controllers with velocity or torque interface)	
0x00n2010F	Read/Write	P/PP/PID (pos.) Slave control	REAL64	(mm/s) / mm	[0.0...1000.0]	Slave coupling difference control: Proportional gain k_{cp}	Slave coupling difference control
0x00n20110	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: active/passive	
0x00n20111	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: hold mode	
0x00n20112	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: Fading mode	
0x00n20114	Read/Write	P (Pos.)	REAL64	%	[0.0 ... 1.0]	Automatic offset calibration: pre-control limit	
0x00n20115	Read/Write	P (Pos.)	REAL64	s	[0.1 ... 60.0]	Automatic offset calibration: time constant	
0x00n20116	Read/Write	PID (pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I part in percent (default setting: 0.1 = 10%)	
0x00n20117	Read/Write	PID (pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D part in percent (default setting: 0.1 = 10%)	
0x00n20118	Read/Write	PID (pos.)	UINT16	1	0/1	Deactivation of the I part during an active positioning process (assuming I part active)? (Default setting: 0 = FALSE)	
0x00n20120	Read/Write	P/PID (pos.)	REAL64	s	≥0	PT-1 filter value for position error (pos. control deviation)	Reserved function, no standard!
0x00n20202	Read/Write	P/PID (velocity)	REAL64	1	[0.0...1000.0]	Proportional gain kp or kv	Velocity control
0x00n20203	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Integral action time Tn	Velocity control
0x00n20204	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Derivative action time Tv	Velocity control
0x00n20205	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Damping time Td	Velocity control
0x00n20206	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I part in	Velocity control

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						percent (default setting: 0.1 = 10%)	
0x00n20207	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D part in percent (default setting: 0.1 = 10%)	Velocity control
0x00n2020D	Read/Write	P/PID (velocity)	REAL64	mm/s	[0.0 ... 10000.0]	"dead band" for velocity error (control deviation) (for P/PID controllers with velocity or torque interface)	reserved function
0x00n20220	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-2 filter value for velocity error (vel. control deviation)	Velocity control, not standard!
0x00n20221	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-1 filter value for velocity error (vel. control deviation)	Reserved function, no standard!
0x00n20250	Read/Write	P/PI (observer)	UINT32	1	s. ENUM (≥0)	Observer mode [▸ 360] for control in the torque interface 0: OFF (default) 1: LUENBERGER	From TC 2.10 Build 1320
0x00n20251	Read/Write	P/PI (observer)	REAL64	Nm / A	>0.0	Motor: Torque constant K_T	
0x00n20252	Read/Write	P/PI (observer)	REAL64	kg m^2	>0.0	Motor: Moment of inertia J_M	
0x00n20253	Read/Write	P/PI (observer)	REAL64	Hz	[100.0 ... 2000.0] Default: 500	Bandwidth f_0	
0x00n20254	Read/Write	P/PI (observer)	REAL64	1	[0.0 ... 2.0] Default: 1.0	Correction factor k_c	
0x00n20255	Read/Write	P/PI (observer)	REAL64	s	[0.0 ... 0.01] Default: 0.001	Velocity filter (1st order): time constant T	
0x00n20A03	Read/Write	P/PID (MW)	REAL64	cm^2	[0.0 ... 1000000]	Cylinder area A_A of the A side in cm^2	Reserved parameters!
0x00n20A04	Read/Write	P/PID (MW)	REAL64	cm^2	[0.0 ... 1000000]	Cylinder area AB of the B side in cm^2	Reserved parameters!
0x00n20A05	Read/Write	P/PID (MW)	REAL64	cm^3/s	[0.0 ... 1000000]	Nominal volume flow Q_{nom} in cm^3/s	Reserved parameters!
0x00n20A06	Read/Write	P/PID (MW)	REAL64	bar	[0.0 ... 1000000]	Nominal pressure or valve pressure drop P_{nom} in bar	Reserved parameters!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n20A07	Read/Write	P/PID (MW)	UINT32	1	[1 ... 255]	Axis ID for the system pressure Po	Reserved parameters!
DRIVE:							
0x00n30001	Read	Drive: every	UINT32	1	[1 ... 255]	Drive ID	
0x00n30002	Read	Drive: every	UINT8[30+1]	1	30 characters	Drive name	
0x00n30003	Read	Drive: every	UINT32	1	s. ENUM (>0)	Drive type [▸ 364]	
0x00n30004	Read/Write	Drive: every	UINT32	1	Byteoffset	Input address offset (I/O-Input-Image)	change I/O address
0x00n30005	Read/Write	Drive: every	UINT32	1	Byteoffset	Output address offset (I/O-Output-Image)	change I/O address
0x00n30006	Read/Write	Drive: every	UINT16	1	[0,1]	motor polarity	
0x00n3000A	Read/Write	Drive: every	UINT32	1	s. ENUM (>0)	drive mode	
0x00n3000B	Read/Write	Drive: every	REAL64	%	[-1.0 ... 1.0]	Minimum output limit (output limitation) (Default setting: -1.0 = -100%)	
0x00n3000C	Read/Write	Drive: every	REAL64	%	[-1.0 ... 1.0]	Maximum output limit (output limitation) (default setting: 1.0 = 100%)	
0x00n3000D	Read	Drive: every	UINT32	INC		Maximum number of output increments (output mask)	
0x00n30010	Read/Write	Drive: every	UINT32	1		Internal Drive Control DWord to determine the drive operation modes	Reserved!
0x00n30011	Read/Write	every	UINT32	1	≥ 5	Internal drive reset counter (time in NC cycles for enable and reset)	Reserved!
0x00n30101	Read/Write	D: Servo	REAL64	e.g. mm/s	>0.0	Reference velocity at reference output (velocity pre-control)	
0x00n30102	Read/Write	D: Servo	REAL64	%	[0.0 ... 5.0]	reference output in percent (Default setting: 1.0 = 100%)	
0x00n30103	Read	D: Servo	REAL64	e.g. mm/s	>0.0	resulting velocity at 100% output	
0x00n30104	Read/Write	D: Servo	REAL64	e.g. mm/s	±∞	velocity offset (DAC offset) for drift calibration (offset calibration) of the axis	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n30105	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	velocity scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x00n30106	Read/Write	D: Profi Drive DSC	UINT32	0.001 * 1/s	≥ 0	Profibus/Profi Drive DSC: Position control gain Kpc	Only for Profi Drive DSC
0x00n30107	Read/Write	D: Profi Drive DSC	REAL64	1	≥ 0.0	Profibus/Profi Drive DSC: Scaling for calculation of 'XERR' (default: 1.0)	Only for Profi Drive DSC
0x00n30109	Read/Write	D: Servo (Sercos, CANopen)	REAL64	1	[0.0 ... 100000000.0]	position scaling (scaling factor to react to the weight in the drive)	For Sercos, CANopen
0x00n3010A	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	acceleration scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x00n30120	Read/Write	D: Servo/ hydraulics	UINT32	1	≥ 0	Table ID (0: no table)	Only for KL4xxx, M2400, Universal
0x00n30121	Read/Write	D: Servo/ hydraulics	UINT32	1	≥ 0	Interpolation type 0: linear 2: Spline	Only for KL4xxx, M2400, Universal
0x00n30122	Read/Write	Servo/ hydraulics	REAL64	%	[-1.0 ... 1.0]	Output offset in percent **Acts according to the characteristic evaluation!**	Only for KL4xxx, M2400, Universal
0x00n30151	Read/Write	D: Servo / non-linear	REAL64	1	[0.0 ... 100.0]	Quadrant equalizing factor (relation between quadrants I and III)	
0x00n30152	Read/Write	D: Servo / non-linear	REAL64	1	[0.01 ... 1.0]	velocity reference point in percent (1.0 = 100 %)	
0x00n30153	Read/Write	D: Servo / non-linear	REAL64	1	[0.01 ... 1.0]	Output reference point in percent (1.0 = 100 %)	
0x00030301	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 1	
0x00030302	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 2	
0x00030303	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 3	
0x00030304	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 4	
0x00030305	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 5	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00030306	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 6	
0x00030307	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 7	
0x00030308	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Cycle 8	
0x00030310	Read/Write	D: Stepper motor	UINT8	1		Bit mask: Holding current	

7.7.4.2 "Index offset" specification for axis state (Index group 0x4100 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n00000	Read	every (online structure for axis data)	{			AXIS ONLINE STRUCTURE (NC/CNC)	Cannot be traced by oscilloscope! (NCAXISSTATE_ONLINESTRUCT)
			INT32	1		Error state	
			REAL64	e.g. mm		Actual position	
			REAL64	e.g. degrees		Modulo actual position	
			REAL64	e.g. mm		Set position	
			REAL64	e.g. degrees		Modulo set position	
			REAL64	e.g. mm/s		Optional: Actual velocity	
			REAL64	e.g. mm/s		Set velocity	
			UINT32	%	0...1000000	Velocity override (1000000 == 100%)	
			REAL64	e.g. mm		Lag error position	
			REAL64	e.g. mm		PeakHold value for max. neg. position lag (pos.)	
			REAL64	e.g. mm		PeakHold value for max. pos. lag error (pos.)	
			REAL64	%		Controller output in percent	
			REAL64	%		Total output in percent	
			UINT32	1	≥ 0	Axis-Status-DWord	
			UINT32	1	≥ 0	Axis-Control-DWord	
			UINT32	1	≥ 0	Slave coupling state (state)	
			UINT32	1	0; 1,2,3...	Axle control loop index	
			}			112 bytes	
0x00000001	Read	every	UINT32	1		Axis state error code	*Symbolic access: 'ErrState"*
0x00n00009	Read	every	UINT32	1	≥ 0	Set cycle counter (SEC-Timestamp)	
0x00n0000A	Read	every	REAL64	e.g. mm		Set position	*Symbolic access:*

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
							'SetPos"
0x00n0000B	Read	every	REAL64	e.g. DEGREES		Modulo set position	*Symbolic access:* *'SetPosModulo"*
0x00n0000C	Read	every	INT32	1		Modulo set rotation	
0x00n0000D	Read	every	REAL64	1	[-1.0, 0.0, 1.0]	Set travel direction	
0x00n0000E	Read	every	REAL64	e.g. mm/s		Set velocity	*Symbolic access:* *'SetVelo"*
0x00n0000F	Read	every	REAL64	e.g. mm/s^2		Set acceleration	Symbolic access: *'SetAcc"*
0x00n00010	Read	every	REAL64	e.g. mm/s^3		Set jerk (time derivative of the set acceleration)	
0x00n00011	Read	every	REAL64	Nm resp. N		Set torque (rot. motor) or set force (linear motor)	
0x00n00012	Read	every	REAL64	1		Set coupling factor (set gear ratio)	
0x00n00013	Read	every	REAL64	e.g. mm		Expected target position	
0x00n00014	Read	Servo	{			Remaining travel time and distance (SERVO):	Always to SEC Port 501!
			REAL64	s	≥ 0	Remaining travel time	
			REAL64	e.g. mm	≥ 0	Remaining distance	
			}				
0x00n00015	Read	every	UINT32	1	≥ 0	Set command number	
0x00n00016	Read	Servo	REAL64	s	≥ 0	Positioning time of the last motion command (Start → target position window)	
0x00000018	ReadWrite	Servo	**Write**			Reading the "Stop information" (stop distance, stop time)	From TC 2.11 R2 Only port 500!
			REAL64	e.g. mm/s^2	≥ 0	Deceleration for axis stop	
			REAL64	e.g. mm/s^3	≥ 0	Jerk for axis stop	
			Read				
			REAL64	e.g. mm	≥ 0	Stop distance	
			REAL64	s	≥ 0	Stop time	
0x00n0001A	Read	every	REAL64	e.g. mm		Uncorrected set position	
0x00n0001D	Read	every	REAL64	1	[-1.0, 0.0, 1.0]	Uncorrected set travel direction	
0x00n0001E	Read	every	REAL64	e.g. mm/s		Uncorrected set velocity	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n0001F	Read	every	REAL64	e.g. mm/s^2		Uncorrected set acceleration	
0x00000020	Read	every	UINT32	1	s. ENUM	Coupling state (state)	
0x00000021	Read	every	UINT32	1	≥ 0	Coupling table index	
0x00000022	Read	Servo Master/Slave coupling Type: LINEAR, (&SPECIAL)	{			Reading the coupling parameters (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Gear ratio	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
			}				
0x00000023	Read	Servo Master/Slave coupling Type: LINEAR, (&SPECIAL)	REAL64	1	[±1000000.0]	Reading the gear ratio (SERVO) Type: LINEAR,	
0x00000024	Read	Servo	UINT32	1	≥ 0	Number / index of the active axis control loop (triple of encoder, controller and axis interfaces)	
0x00000025	Read	Servo	UINT16	1	0/1	External setpoint specification via axis interface PCLtoNC active?	
0x00000026	Read	Servo Master/Slave coupling Type: SYNCHRONIZING	REAL64 [64]	1	±∞	Reading of the characteristic values of the slave synchronization profile Type: SYNCHRONIZING	
0x00000027	ReadWrite	Servo Master/Slave coupling Type: TABULAR, MF	**Write**			Read the "Tabular coupling information"	Only port 500!
			VOID or REAL64	e.g. mm	±∞	- No data for the "current information" - optional for a certain "master axis position"	
			Read				
			REAL64 [32]		±∞	Reading the structure for the table coupling information	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000028	ReadWrite	Servo Master/Slave coupling Type: MULTICAM (CamAddition)	**Write**			Reading the "multi tabular coupling information" (CamAddition)	Only port 500!
			UINT32	1	≥ 0	Table ID to which the query relates	
			Read				
			96 bytes			Reading the structure for the multi tabular coupling information	
0x00000029	Read	Servo	UINT32	1		Delayed error code (error pre-warning) in case of a delayed error reaction (see bit *ErrorPropagationDelayed*)	From TC 2.11 R3 B2229
0x0000002A	Read	Servo	REAL64	e.g. mm	±∞	Position difference while fading from set position to actual position (fading part)	From TC 2.11 R2
0x0000002B	Read	Servo	REAL64	e.g. mm/s	±∞	Relative velocity while fading from set position to actual position (fading part)	From TC 2.11 R2
0x0000002C	Read	Servo	REAL64	e.g. mm/s ^2	±∞	Relative acceleration while fading from set position to actual position (fading part)	From TC 2.11 R2
0x0000002D	Read	Servo	UINT32	1	≥ 0	Counter for initialization command (InitializeCommandCounter)	
0x0000002E	Read	Servo	UINT32	1	≥ 0	Counter for reset command (ResetCommandCounter)	
0x00000050	Read	every	UINT32	1		Set travel phase (SWGenerator)	Cannot be traced by oscilloscope!
0x00000051	Read	every	UINT16	1		Is the axis disabled?	Cannot be traced by oscilloscope!
0x00n00060	Read/Write	every (online setpoint structure)	{			AXIS SETPOINT STRUCTURE (NC/CNC)	Cannot be traced by oscilloscope!
			REAL64	e.g. mm		Set position	
			REAL64	e.g. mm/s		Set velocity	
			REAL64	e.g. mm/s^2		Set acceleration / deceleration	
			REAL64	1	[-1.0, 0.0, 1.0]	Set travel direction	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	e.g. mm/s^3		Set jerk	
			REAL64	Nm resp. N		Set torque or set force (new from TC V2.11 B1514)	
			}				
0x00n00061	Read/Write	every (online dynamic setpoint structure)	{			AXIS DYNAMIC SETPOINT STRUCTURE (NC/CNC)	
			REAL64	e.g. mm/s		Set velocity	
			REAL64	e.g. mm/s^2		Set acceleration / deceleration	
			REAL64	1	[-1.0, 0.0, 1.0]	Set travel direction	
			REAL64	e.g. mm/s^3		Set jerk	
			REAL64	Nm resp. N		Set torque or set force (new from TC V2.11 B1514)	
			}				
0x00n10002	Read	every (Encoder)	REAL64	e.g. mm		Actual position (charge with actual position compensation value) n = 0: Standard encoder for the axis n > 0: n-th encoder for the axis (optional)	*Symbolic access:* *'ActPos"*
0x00n10003	Read	every (Encoder)	REAL64	e.g. DEGREES		Modulo actual position	*Symbolic access:* *'ActPosModulo'*
0x00n10004	Read	every (Encoder)	INT32	1		Modulo actual rotation	
0x00n10005	Read	every (Encoder)	REAL64	e.g. mm/s		Optional: Actual velocity	*Symbolic access:* *'ActVelo"*
0x00n10006	Read	every (Encoder)	REAL64	e.g. mm/s^2		Optional: Actual acceleration	*Symbolic access:* *'ActAcc"*
0x00n10007	Read	every (Encoder)	INT32	INC		Encoder actual increments	
0x00n10008	Read	every (Encoder)	INT64	INC		Software - actual increment counter	
0x00n10009	Read	every (Encoder)	UINT16	1	0/1	Reference flag ("calibrate flag")	
0x00n1000A	Read	every (Encoder)	REAL64	e.g. mm		Actual position correction value (measurement system error correction)	
0x00n1000B	Read	every (Encoder)	REAL64	e.g. mm		Actual position without actual position compensation value	Cannot be traced by oscilloscope!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n10010	Read	every (Encoder)	REAL64	e.g. mm/s		Actual velocity without actual position compensation value	
0x00n10012	Read	every (Encoder)	REAL64	e.g. mm		Unfiltered actual position (charge with actual position compensation value)	
0x00n10015	Read	every (Encoder)	REAL64	e.g. mm/s		Optional: Unfiltered actual velocity	Base Unit / s
0x00n10101	Read	INC (Encoder)	REAL64	e.g. mm		Read back of the position difference between activation of the internal hardware latch and the time when it becomes valid	Cannot be traced by oscilloscope!
0x00n20001	Read	R: every	INT32	1		Error state of the controller n = 0: Standard axis controller n > 0: n-th controller of the axis (optional)	
0x00n20002	Read	R: every	REAL64	e.g. mm/s		Controller output in absolute units	*Symbolic access:* *'CtrlOutput"*
0x00n20003	Read	R: every	REAL64	%		Controller output in percent	Cannot be traced by oscilloscope!
0x00n20004	Read	R: every	REAL64	V		Controller output in volts	Cannot be traced by oscilloscope!
0x00n2000D	Read	R: every	REAL64	e.g. mm		Lag error position (without dead time compensation)	Base Unit
0x00n2000F	Read	R: every	REAL64	e.g. mm		Lag error position (with dead time compensation)	*Symbolic access:* *'PosDiff"*
0x00n20010	Read	R: every	REAL64	e.g. mm		Peak hold value for maximum negative lag error of the position	
0x00n20011	Read	R: every	REAL64	e.g. mm		Peak hold value for minimum positive lag error of the position	
0x00n20012	Read	R: every	REAL64	e.g. mm/s		Lag error velocity	Not implemented!
0x00n20021	Read	R: every	REAL64	e.g. mm		Difference (deviation) between the lag	*Symbolic access:*

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						error from master and slave axis (master error minus slave error)	*'PosDiffCouple'*
0x00n20022	Read	R: every	REAL64	e.g. mm		PeakHold value for the maximum negative difference between master and slave axis lag error of the position	Base Unit
0x00n20023	Read	R: every	REAL64	e.g. mm		PeakHold value for the maximum positive difference between master and slave axis lag error of the position	Base Unit
0x00n20101	Read	R: P/PID (Pos.)	REAL64	e.g. mm/s		P part of the controller in absolute units	
0x00n20102	Read	R: PID (Pos.)	REAL64	e.g. mm/s		I part of the controller in absolute units	
0x00n20103	Read	R: PID (Pos.)	REAL64	e.g. mm/s		D part of the controller in absolute units	
0x00n20104	Read	R: PID (Pos.)	UINT16	1	0/1	Limitation of the I part active?	
0x00n20105	Read	R: PID (Pos.)	UINT16	1	0/1	Limitation of the D part active?	
0x00n20106	Read	R: PID (Pos.)	UINT16	1	0/1	ARW measures for the I part active? ARW: Anti Reset Windup	Not implemented!
0x00n20110	Read	R: PID (Pos.)	REAL64	e.g. mm/s		Acceleration pre-control Yacc of the controller in absolute units **Function depends on controller type!**	Acceleration pre-control
0x00n20111	Read	R: PP (Pos.)	REAL64	mm/s/ mm	≥0	Internal interpolated proportional gain kp or kv	PP controller
0x00n20201	Read	R: P,PID (velocity)	REAL64	e.g. mm/s		Velocity part of the controller	Base Unit / s
0x00n20202	Read	R: P,PID (velocity)	REAL64	%		Velocity part of the controller in percent	Cannot be traced by oscilloscope!
0x00n20203	Read	R: P,PID (velocity)	REAL64	V		Velocity part of the controller in volts	Cannot be traced by oscilloscope!
0x00n20201	Read	R: P,PID (velocity)	REAL64	e.g. mm/s		P part of the controller in absolute units	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n20202	Read	R: P,PID (velocity)	REAL64	e.g. mm/s		I part of the controller in absolute units	
0x00n20203	Read	R: P,PID (velocity)	REAL64	e.g. mm/s		D part of the controller in absolute units	
0x00n20204	Read	R: P,PID (velocity)	UINT16	1	0/1	Limitation of the I part active?	
0x00n20205	Read	R: P,PID (velocity)	UINT16	1	0/1	Limitation of the D part active?	
0x00n20206	Read	R: P,PID (velocity)	UINT16	1	0/1	ARW measures for the I part active?	ARW: Anti Reset Windup
0x00n2020A	Read	R: P,PID (velocity)	REAL64	e.g. mm/s		Total input size of the velocity controller	
0x00n20A00	Read	R: PID (MW)	REAL64	%	[-1.0...1.0]	Offsetting of the set velocity (pre-control)	Reserved parameters!
0x00n20A01	Read	R: PID (MW)	REAL64	e.g. mm/s		P part of the controller in absolute units or percent (according to output weight)	Reserved parameters!
0x00n20A02	Read	R: PID (MW)	REAL64	e.g. mm/s		I part of the controller in absolute units or percent (according to output weight)	Reserved parameters!
0x00n20A03	Read	R: PID (MW)	REAL64	e.g. mm/s		D part of the controller in absolute units or percent (according to output weight)	Reserved parameters!
0x00n20A04	Read	R: PID (MW)	UINT16	1	0/1	Limitation of the I part active?	Reserved parameters!
0x00n20A05	Read	R: PID (MW)	UINT16	1	0/1	Limitation of the D part active?	Reserved parameters!
0x00n20A06	Read	R: PID (MW)	UINT16	1	0/1	ARW measures for the I part active? ARW: Anti Reset Windup	Reserved parameters!
0x00n20A10	Read	R: PID (MW)	REAL64	e.g. mm/s		Acceleration pre-control Yacc of the controller in absolute units	Reserved parameters!
0x00n30001	Read	D: every	INT32	1		Error state of the drive	
0x00n30002	Read	D: every	REAL64	e.g. mm/s		Total output in absolute units	*Symbolic access:* *'DriveOutput'*
0x00n30003	Read	D: every	REAL64	%		Total output in percent	
0x00n30004	Read	D: every	REAL64	V		Total output in volts	Cannot be traced by oscilloscope!
0x00n30005	Read	D: every	REAL64	e.g. mm/s		PeakHold value for maximum negative total output	Base Unit / s

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n30006	Read	D: every	REAL64	e.g. mm/s		PeakHold value for maximum positive total output	Base Unit / s
0x00n30013	Read	D: every	REAL64	%		Total output in percent (according to non-linear characteristic curve!)	
0x00n30014	Read	D: every	REAL64	V		Total output in volts (according to non-linear characteristic curve!)	Cannot be traced by oscilloscope!
0x00n3011A	Read	D: Servo (Sercos, CANopen)	REAL64	e.g. mm		Optional output filtering: Filtered set position	NEW For Sercos, CANopen
0x00n3011E	Read	D: Servo (Sercos, CANopen)	REAL64	e.g. mm/s		Optional output filtering: Filtered set velocity	NEW For Sercos, CANopen
0x00n3011F	Read	D: Servo (Sercos, CANopen)	REAL64	e.g. mm/s^2		Optional output filtering: Filtered set acceleration / set deceleration	NEW For Sercos, CANopen

7.7.4.3 "Index offset" specification for axis functions (Index group 0x4200 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Write	every	VOID			Reset axis	For FIFO axes too!
0x00000002	Write	every	VOID			Stop axis	For FIFO axes too!
0x00000003	Write	every	VOID			Clear axis (task)	For FIFO axes too!
0x00000004	Write	every	{			Emergency Stop (Emergency stop with regulated ramp)	Only for PTP axes!
			REAL64	e.g. mm/s^2	> 0.0	Deceleration (must be greater than or equal to the original deceleration)	
			REAL64	e.g. mm/s^3	> 0.0	Jerk (must be greater than or equal to the original jerk)	
			}				
0x00000005	Write	PTP axis	{			Parametrized stop (with regulated ramp)	Only for PTP axes!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	e.g. mm/s^2	> 0.0	Deceleration	Reserved function, no standard!
			REAL64	e.g. mm/s^3	> 0.0	Jerk	
			}				
0x00000009	Write	PTP axis	{			Oriented stop (oriented end position)	Only for PTP axes!
			REAL64	e.g. degrees	≥ 0.0	Modulo end position (modulo target position)	
			REAL64	e.g. mm/s^2	> 0.0	Deceleration (not yet implemented)	
			REAL64	e.g. mm/s^3	> 0.0	Jerk (not yet implemented)	
			}				
0x00000010	Write	every	VOID			Reference axis ("calibration")	
0x00000011	Write	every	{			New end position axis	
			UINT32	ENUM	s. appendix	End position type (see appendix)	
			REAL64	e.g. mm	±∞	new end position (target position)	
			}				
0x00000012	Write	every	{			New end position and new velocity axis	
			UINT32	ENUM	s. appendix	Command type (s. appendix)	
			UINT32	ENUM	s. appendix	End position type (see appendix)	
			REAL64	e.g. mm	±∞	new end position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	New end velocity (requested traveling velocity)	
			REAL64	e.g. mm	±∞	Optional: Switching position from which the new travel profile is activated	
			}				
0x00000015	Write	every	{			New dynamic parameters for active positioning	
			REAL64	e.g. mm/s^2	> 0.0	Acceleration	
			REAL64	e.g. mm/s^2	> 0.0	Deceleration	
			REAL64	e.g. mm/s^3	> 0.0	Optional: Jerk (not yet implemented)	
			}				
0x00000016	**ReadWrite**	every SERVO	**Write** (76 bytes)			Universal axis start (UAS):	Always to SEC Port 501!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						Merge of single commands, such as axis start, and online changes in combination with "Buffer Mode" (see TcMc2.lib)	
			{				
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			UINT32	1	≥ 0	Bit mask for checks and operation modes (default value: 0)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	Required velocity *Vrequ*	
			REAL64	e.g. mm/s^2	≥ 0.0	Optional: Acceleration	
			REAL64	e.g. mm/s^2	≥ 0.0	Optional: Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Optional: Jerk	
			UINT32	ENUM	s. appendix	Buffer mode (command buffer) [▶ 357]	
			REAL64	e.g. mm	±∞	Optional: blending position (command blending position)	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment start velocity *Vi* *(0 ≤ Vi ≤ Vrequ)*	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment end velocity *Vf* *(0 ≤ Vf ≤ Vrequ)*	
			}				
			Read				
			{				
			UINT16	1	≥ 0	Command number (job number)	
			UINT16	1	≥ 0	Command status	
			}				
0x00000017	ReadWrite	SERVO	Write (76 bytes)			"Master/Slave decoupling" and "Universal Axis Start (UAS)": Merge of the decoupling command of a slave axis (IdxOffset: 0x00000041) and subsequent Universal Axis	Not yet released!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						Start (UAS) (IdxOffset: 0x00000016)	
			{				
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			UINT32	1	≥ 0	Bit mask for checks and operation modes (default value: 0)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	Required velocity Vrequ	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			UINT32	ENUM	s. appendix	Buffer mode (command buffer) [▸ 357]	
			REAL64	e.g. mm	±∞	Optional: blending position (command blending position)	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment start velocity Vi (0 ≤ Vi ≤ Vrequ)	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment end velocity Vf (0 ≤ Vf ≤ Vrequ)	
			}				
			Read				
			{				
			UINT16	1	≥ 0	Command number (job number)	
			UINT16	1	≥ 0	Command state	
			}				
0x00000018	Write	every	VOID			Release axis lock for motion commands (TcMc2)	
0x00000019	Write	every	UINT32	1	> 0	Set external axis error (runtime error)	Caution by using!
0x00n0001A	Write	every	{			Set actual axis position	**Caution when using!** For FIFO axes too! Always to SEC Port 501!
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	actual position for axis n = 0: Standard encoder for the axis n > 0: n-th encoder for the axis (optional)	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			}				
0x00n0001B	Write	every	UINT32	1	0/1	Set reference flag ("calibrate flag") n = 0: Standard encoder for the axis n > 0: n-th encoder for the axis (optional)	**Caution when using!** For FIFO axes too!
0x00n0001C	Write	SERVO	{			Set only actual axis position without manipulating the set position (also for slave and with active process)	**Caution when using!**
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	actual position for axis n = 0: Standard encoder for the axis n > 0: n-th encoder for the axis (optional) **Caution when using!**	
			}				
0x00n0001D	Write	every	{			Set actual value of the axis on the drive side (Position interface and encoder offset of null required!) n = 0: Standard encoder for the axis n > 0: n-th encoder for the axis (optional)	**Caution when using!** Only for CANopen
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	actual position for axis	
			}				
0x00n0001E	Write	every	{			Set a new encoder scaling factor on the fly (in motion of the axis)	Caution by using! Always to SEC Port 501!
			UINT32	ENUM	1	Encoder scaling factor type 1: Absolute 2: Relative	
			REAL64	e.g. mm/INC	[1.0E-8 ... 100.0]	New encoder scaling factor n = 0: Standard encoder for the axis	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						n > 0: n-th encoder for the axis (optional)	
			}				
0x00n0001F	Write	every	{			Set actual axis position on the fly (in motion of the axis)	**Caution when using!** Always to SEC Port 501!
			UINT32	ENUM		Position type for setting actual value on the fly 1: Absolute 2: Relative	
			UINT32	1		Control DWord, e.g. for "clearing the lag"	
			REAL64			Reserve	
			REAL64	e.g. mm	±∞	New actual axis position	
			UINT32			Reserve	
			UINT32			Reserve	
			}				
0x00000020	Write	every 1D-Start	{			Standard axis start	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥0.0	Required velocity	
			}				
0x00000021	Write	every 1D-Start	{			Extended axis start (SERVO):	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	Required velocity	
			UINT32	0/1	0/1	Standard acceleration?	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			UINT32	0/1	0/1	Standard deceleration?	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			UINT32	0/1	0/1	Standard jerk?	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			}				
0x00000022	Write	SERVO(MW)	{			Special axis start (SERVO):	Reserved start function, no standard!
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	mm/s	≥ 0.0	required start velocity	
			REAL64	e.g. mm	±∞	Position for a new velocity level	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	e.g. mm/s	≥ 0.0	new end velocity level	
			UINT32	0/1	0/1	Standard acceleration?	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			UINT32	0/1	0/1	Standard deceleration?	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			UINT32	0/1	0/1	Standard jerk?	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			}				
0x00000023	Write	SERVO	{			Start external setpoint specification (setting by cyclic axis interface PLCtoNC)	
			UINT32	ENUM	1: Absolute 2: Relative	Start type	
			REAL64	e.g. mm	±∞	New end position (target position) optional!	
			}				
0x00000024	Write	SERVO	VOID			Stop/disable external set value setting (cycl. axis interface PLCtoNC)	
0x00000025	Write	SERVO	{			Start reversing operation for positioning (SERVO):	
			UINT32	ENUM	1	Start type (default: 1)	
			REAL64	e.g. mm	±∞	End position 1 (target position)	
			REAL64	e.g. mm	±∞	End position 2 (target position)	
			REAL64	0/1	0/1	Required velocity	
			REAL64	s	≥ 0.0	Idle time	
			}				
0x00000026	Write	every	{			Start drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000027	Write	every	VOID			Stop drive output	
0x00000028	Write	every	{			Change the drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000029	Write	every	VOID			Instantaneously adopt current override value	Reserved function, no standard!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						and freeze until next override change!	
0x0000002A	Write	every	{ 32 bytes }			Calculate and set encoder offset	Reserved function, no standard!
0x0000002B	ReadWrite	every	WriteData: s. 'UAS' ReadData: s. 'UAS'			Stop external setpoint generator and continuous endless motion ('UAS': Universal axis start)	Reserved function, no standard!
0x00000030	Write	SERVO	{			Start section compensation (SERVO)	
			UINT32	ENUM	s. appendix	Compensation type (see appendix)	
			REAL64	e.g. mm/s^2	≥ 0.0	Max. acceleration increase	
			REAL64	e.g. mm/s^2	≥ 0.0	Max. deceleration increase	
			REAL64	e.g. mm/s	> 0.0	Max. increase velocity	
			REAL64	e.g. mm/s	> 0.0	Base velocity for the process	
			REAL64	e.g. mm	±∞	Path difference to be compensated	
			REAL64	e.g. mm	> 0.0	Path distance for compensation	
			}				
0x00000030	ReadWrite	SERVO returns the real active values	{ **READ+WRITE:**			Start section compensation (SERVO) **Contained only in "TcMc2.lib"**	
			UINT32	ENUM	s. appendix	Compensation type (see appendix)	
			REAL64	e.g. mm/s^2	≥ 0.0	=> Max. acceleration increase <= Returns the implemented acceleration increase (new in "TcMc2.lib")	
			REAL64	e.g. mm/s^2	≥ 0.0	=> Max. deceleration increase <= Returns the implemented deceleration increase (new in "TcMc2.lib")	
			REAL64	e.g. mm/s	> 0.0	=> Requested max. increase velocity	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						<= Returns the implemented increase velocity	
			REAL64	e.g. mm/s	> 0.0	Base velocity for the process	
			REAL64	e.g. mm	±∞	=> Requested path difference to be compensated <= Returns the implemented path difference	
			REAL64	e.g. mm	> 0.0	=> Requested max. distance for compensation <= Returns implemented distance	
			UINT32	1	≥ 0	<= Returns Warning ID (e.g. 0x4243)	
			}				
0x00000031	Write	SERVO	VOID			Stop section compensation (SERVO)	
0x00000032	Write	SERVO	{			Start reversing operation with velocity jumps (SERVO) (can be used to determine the velocity step response)	
			UINT32	ENUM	1	Start type (default: 1)	
			REAL64	e.g. mm/s	±∞	Required velocity 1 (negative values also allowed)	
			REAL64	e.g. mm/s	±∞	Required velocity 2 (negative values also allowed)	
			REAL64	s	> 0.0	Travel time for velocity 1 and 2	
			REAL64	s	≥ 0.0	Idle time	
			UINT32	1	0, 1,2,3...	Optional: Number of repetitions Default "0": unlimited in time	
			}				
0x00000033	Write	SERVO	{			Sine oscillation sequence - used as single sinus oscillation (sinus generator) - used as sinus oscillation sequence (e.g. for bode plot)	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			UINT32	ENUM	1	Start type (fixed to start type 1 yet)	
			REAL64	e.g. mm/s	> 0.0	Base amplitude (e.g. 2.5 mm/s)	
			REAL64	Hz	[0.0 .... 10.0]	Base frequency (e.g. 1.953125 Hz)	
			REAL64	e.g. mm/s	≥ 0.0	Start amplitude at begin (e.g. 0.0 mm/s)	
			REAL64	e.g. mm/REV	> 0.0	Feed constant motor (per motor turn) (e.g. 10.0 mm/REV)	
			REAL64	Hz	≥ 1.0	Frequency range: start frequency (e.g. 20.0 Hz)	
			REAL64	Hz	≤ 1/(2*dT)	Frequency range: stop frequency (e.g. 500.0 Hz)	
			REAL64	s	> 0.0	Step duration (e.g. 2.048s)	
			UINT32	1	[1 ... 200]	Number of measurements (step cycles) (e.g. 20)	
			UINT32	1		Number of parallel measurements (e.g. 1) not used yet!	
			}				
0x00000034	Write	SERVO	{			Phasing - Start Phasing - Stop Phasing	
			UINT32	ENUM	1	Phasing Type: 1: ABSOLUTE 2: RELATIVE 4096: STOP	
			UINT32	1	≥ 0	Control Mask	
			UINT32	1	≥ 0	Master axis ID (multi master)	
			UINT32			Reserve	
			REAL64	e.g. mm	> 0.0	Phase shift	
			REAL64	e.g. mm/s	> 0.0	Velocity	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			REAL64[4]			Reserve	
			UINT32			Reserve	
			UINT32	1	ENUM	Buffer mode (NOT IMPLEMENTED)	
			REAL64	e.g. mm	±∞	Blending position (NOT IMPLEMENTED)	
			}				

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000040 (0x00n00040)	Write	Master/Slave coupling: (SERVO)	{			Master/Slave coupling (SERVO):	Extension for "flying saw"!
			UINT32	ENUM	s. appendix	Slave type/ coupling type (see appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis/ group	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Gear ratio FlySawVelo: Reserve FlySaw: Abs. synchron position master [mm]	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve FlySawVelo: Reserve FlySawPos: Abs. synchron position slave [mm]	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve FlySawVelo: Angle of inclination in [DEGREES] FlySawPos: Angle of inclination in [DEGREES]	Angle > 0.0 and £ 90.0 degrees (Parallel saw: 90.0 degrees)
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve FlySawVelo: Gear factor FlySawPos: Gear ratio	
			}				
0x00000041	Write	Master/Slave decoupling (SERVO)	VOID			Master/Slave decoupling (SERVO)	
0x00000041	Write	Master/Slave decoupling with configurable follow-up function (SERVO)	{			Master/slave decoupling with configurable follow-up function (e.g. new end position, new velocity, stop, E-stop) (SERVO)	Not yet released!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			UINT32	ENUM	see appendix [▸ 359]	Decoupling type (see appendix)	
			REAL64	e.g. mm	±∞	Optional: New end position	
			REAL64	e.g. mm/s	> 0.0	Optional: New requested velocity	
			REAL64	e.g. mm/s^2	≥ 0.0 (0: Default)	Optional: Acceleration for new end position, new velocity and emergency stop (E-stop)	
			REAL64	e.g. mm/s^2	≥ 0.0 (0: Default)	Optional: Deceleration for new end position, new velocity and emergency stop (E-stop)	
			REAL64	e.g. mm/s^3	≥ 0.0 (0: Default)	Optional: Jerk for new end position, new velocity and emergency stop (E-stop)	
			}				
0x00000042	Write	Master/Slave coupling Type: LINEAR, (&SPECIAL)	{			Change of the coupling parameters (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Gear ratio	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
			}				
0x00000043	Write	Master/slave table coupling Type: TABULAR	{			Change of the table coupling parameters (SERVO)	
			REAL64	mm	±∞	Slave position offset	
			REAL64	mm	±∞	Master position offset	
			}				
0x00000043	Write	Master/slave table coupling Type: TABULAR and "Motion Function"	{			Change of the table coupling parameters (SERVO):	Also for "Motion Function"
			REAL64	mm	±∞	Slave position offset	
			REAL64	mm	±∞	Master position offset	
			REAL64	1	±∞ (<> 0.0)	Slave position scaling	
			REAL64	1	±∞ (<> 0.0)	Master position scaling	
			}				

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000043	Write	Master/slave table coupling Type: TABULAR	{			Change of the table coupling parameters (SERVO):	
			REAL64	mm	±∞	Slave position offset	
			REAL64	mm	±∞	Master position offset	
			REAL64	1	±∞ (<> 0.0)	Slave position scaling	
			REAL64	1	±∞ (<> 0.0)	Master position scaling	
			REAL64	e.g. mm	±∞	Absolute master activation position	
			}				
0x00000044	Write	Slave-Stop (SERVO)	VOID			Stop the "flying saw" (SERVO)	Only for "flying saw"
0x00000045 (0x00n00045)	Write	Master/Slave table coupling (SERVO)	{			Master/Slave table coupling (SERVO):	
			UINT32	ENUM	s. appendix	Slave type/ coupling type (see appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	
						SOLO TABLE SECTION	
			REAL64	mm	±∞	Slave position offset (type: TABULAR)	
			REAL64	mm	±∞	Master position offset (type: TABULAR)	
			UINT32	1	[0,1]	Slave positions absolute (type: TABULAR)	
			UINT32	1	[0,1]	Master positions absolute (type: TABULAR)	
			UINT32	1	[1...255]	Table ID of the coupling table (type: TABULAR)	
						MULTI TABLE SECTION	
			UINT16	1	[0...8]	Number of tables (type: MULTITAB) **Misused as interpolation type for solo tables**	
			UNIT16	1	[0...8]	Number of profile tables (type: MULTITAB)	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			UNIT32[8]	1	[1...255]	Tables IDs of the coupling tables (type: MULTITAB)	
			}				
0x00000046	Write	Master/slave multi-tables	UINT32	1	[1...255]	Activation of correction table correction table ID	
0x00000046	Write	Master/slave multi-tables	{			Activation of correction table	
			UINT32	1	[1...255]	Correction table ID	
			REAL64	e.g. mm	±∞	Absolute master activation position	
			}				
0x00000047	Write	Master/Slave Multi-tables	UINT32	1	[1..255]	Deactivation of profile table at end of cycle Table ID of the current monocyclic profile table	
0x00000048	ReadWrite	Master/Slave Multi-tables	Write: UINT32	1	[1..255]	Read the last correction offset: Table ID of the correction table	
			Read: REAL32	e.g. mm	±∞	Offset by departing the correction table with the according table ID	
0x00000049	Write	Master/slave table coupling Type: TABULAR	REAL64	1	±∞	Change the slave table scaling Scaling factor of the slave table column (default value: 1.0)	
0x0000004A (0x00n0004A)	Write	Master/Slave Universal Table Coupling(SERVO)	{			Master/Slave Solo Table Coupling (SERVO):	
			UINT32	ENUM	s. appendix	Slave type/ coupling type (see appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	
			UINT32	1	1...255]	Table ID of the coupling table (type: TABULAR)	
			UINT32	1		Table interpolation type	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	mm	±∞	Slave position offset (type: TABULAR)	
			REAL64	mm	±∞	Master position offset (type: TABULAR)	
			REAL64	mm	±∞	Slave position scaling (type: TABULAR)	
			REAL64	mm	±∞	Master position scaling (type: TABULAR)	
			UINT32	1	[0,1]	Slave positions absolute ? (type: TABULAR)	
			UINT32	1	[0,1]	Master positions absolute ? (type: TABULAR)	
			UINT32	ENUM	s. appendix	Activation type of the change (NEW) 0: 'instantaneous' (default) 1: 'at master cam position' 2: 'at master axis position' 3: 'next cycle'	
			REAL64	mm	±∞	Activation position (NEW)	
			UINT32	ENUM	s. appendix	Master scaling type (NEW) 0: user defined (default) 1: scaling with auto offset 2: off	
			UINT32	ENUM	s. appendix	Slave scaling type (NEW) 0: user defined (default) 1: scaling with auto offset 2: off	
			}				
0x0000004B (0x00n0004B)	Write	Master/slave universal flying saw (SERVO)	{			Master/slave synchronizing coupling (SERVO):	
			UINT32	ENUM	s. appendix	Slave type/ coupling type (see appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	1	±∞ (<> 0.0)	Gear ratio	
			REAL64	mm	±∞	Master synchron position	
			REAL64	mm	±∞	Slave synchron position	
			REAL64	mm/s	≥ 0.0	Slave velocity (optional)	
			REAL64	mm/s^2	≥ 0.0	Slave acceleration (optional)	
			REAL64	mm/s^2	≥ 0.0	Slave deceleration (optional)	
			REAL64	mm/s^3	≥ 0.0	Slave jerk (optional)	
			UINT32	1	≥ 0	Bit mask for checks and operation modes (default value: 0)	
			}				
0x0000004D (0x00n0004D)	Write	Master/slave table coupling Type: TABULAR and MF	{			Change of the table scaling (SERVO):	Also for MF Reserved function, no standard!
			UINT32	ENUM	s. appendix	Activation type of the change 0: 'instantaneous' (default) 1: 'at master cam position' 2: 'at master axis position' 3: 'next cycle'	
			REAL64	e.g. mm	±∞	Activation position	
			UINT32	ENUM	s. appendix	Master scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			UINT32	ENUM	s. appendix	Slave scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			REAL64	e.g. mm	±∞	Master position offset	
			REAL64	e.g. mm	±∞	Slave position offset	
			REAL64	1	±∞ (<> 0.0)	Master position scaling	
			REAL64	1	±∞	Slave position scaling	
			}				

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000050	Write	every	VOID			Deactivate complete axis (disable)	
0x00000051	Write	every	VOID			Activate complete axis (enable)	
0x00000052	Write	SERVO	{			Change of the active axis control loop (triple from encoder, controller and axis interfaces) with/without external setpoint specification:	
			UINT32	1	≥ 0	Number/index of the axis control loop (default -value: 0)	
			UINT32	ENUM	see appendix (>0)	Switching type for synchronization behavior 1: 'Standard'	
			REAL64	1	±∞	Synchronization value for switching (optional)	
			UINT32	0/ 1	0/1	External setpoint specification by means of axis interface ? **Not used so far!**	
			}				
0x00000060	Write	every	VOID			Deactivate drive output (disable)	
0x00000061	Write	every	VOID			Activate drive output (enable)	
0x00000062	Write	high/low	UINT16	1	0/1	Release parking brake? 0: automatic activation (default) 1: mandatorily always released! **Reset to '0' when resetting the axis!**	
0x00000070	Write	every	VOID			Return of the axis from, e.g. a 3D group to its own PTP group	

7.7.4.4 "Index offset" specification for cyclic axis process data (Index group 0x4300 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n00000	Read/Write	every (PLC→NC)	{ 128 bytes }		STRUCT see axis interface	AXIS STRUCTURE (PLC→NC) n = 0: Standard axis interface n > 0: n-th axis interface (optional)	Write command only optional! Consider safety aspects!
0x00n00001	Read/Write	every (PLC→NC)	UINT32	1	>0	Control double word	Write command only optional!
0x00n00002	Read/Write	every (PLC→NC)	UINT16	1	0/1	Controller enable	Cannot be traced by oscilloscope!
0x00n00003	Read/Write	every (PLC→NC)	UINT16	1	0/1	Feed enable plus	Cannot be traced by oscilloscope!
0x00n00004	Read/Write	every (PLC→NC)	UINT16	1	0/1	Feed enable minus	Cannot be traced by oscilloscope!
0x00n00007	Read/Write	every (PLC→NC)	UINT16	1	0/1	Referencing cam	Cannot be traced by oscilloscope!
0x00n00021	Read/Write	every (PLC→NC)	UINT32	%	0...1000000	Velocity override (1000000 == 100%)	Write command only optional!
0x00n00022	Read/Write	every (PLC→NC)	UINT32	1	ENUM	operation mode axis	Write command only optional!
0x00n00025	Read/Write	every (PLC→NC)	REAL64	e.g. mm		Actual position correction value (measurement system error correction)	Write command only optional!
0x00n00026	Read/Write	every (PLC→NC)	REAL64	e.g. mm/s		External controller part (position controller part)	Write command only optional!
0x00n00027	Read/Write	every (PLC→NC)	{			External setpoint generation	Write command only optional!
			REAL64	e.g. mm	±∞	External set position	
			REAL64	e.g. mm/s	±∞	External set velocity	
			REAL64	e.g. mm/s^2	±∞	External set acceleration	
			INT32	1	+1, 0, -1	External set travel direction	
			}				
0x00n00080	Read	every (PLC→NC)	{ 128 bytes }		STRUCT see axis interface	AXIS STRUCTURE (NC→PLC) n = 0: Standard axis interface n > 0: n-th axis interface (optional)	NCTOPLC_AXLESTRUCT

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n00071	Read	every (PLC→NC)	UINT8	1	>0	State double word: byte 1	
0x00n00072	Read	every (PLC→NC)	UINT8	1	>0	State double word: byte 2	
0x00n00073	Read	every (PLC→NC)	UINT8	1	>0	State double word: byte 3	
0x00n00074	Read	every (PLC→NC)	UINT8	1	>0	State double word: byte 4	
0x00n00081	Read	every (PLC→NC)	UINT32	1	>0	State double word (complete)	
0x00n00082	Read	every (PLC→NC)	UINT16	1	0/1	Axis is ready for operation	Cannot be traced by oscilloscope!
0x00n00083	Read	every (PLC→NC)	UINT16	1	0/1	Axis has been referenced	Cannot be traced by oscilloscope!
0x00n00084	Read	every (PLC→NC)	UINT16	1	0/1	Axis in protected operation mode (e.g. slave axis)	Cannot be traced by oscilloscope!
0x00n00085	Read	every (PLC→NC)	UINT16	1	0/1	Axis is in rapid mode	Cannot be traced by oscilloscope!
0x00n00088	Read	every (PLC→NC)	UINT16	1	0/1	Axis has invalid I/O data	Cannot be traced by oscilloscope!
0x00n00089	Read	every (PLC→NC)	UINT16	1	0/1	Axis is in an error state	Cannot be traced by oscilloscope!
0x00n0008A	Read	every (PLC→NC)	UINT16	1	0/1	Axis moving to larger values	Cannot be traced by oscilloscope!
0x00n0008B	Read	every (PLC→NC)	UINT16	1	0/1	Axis moving to smaller values	Cannot be traced by oscilloscope!
0x00n0008C	Read	every (PLC→NC)	UINT16	1	0/1	Axis is at a logical standstill (only setpoints are considered) (position controller?)	Cannot be traced by oscilloscope!
0x00n0008D	Read	every (PLC→NC)	UINT16	1	0/1	Axis is being referenced	Cannot be traced by oscilloscope!
0x00n0008E	Read	every (PLC→NC)	UINT16	1	0/1	Axis is in position window	Cannot be traced by oscilloscope!
0x00n0008F	Read	every (PLC→NC)	UINT16	1	0/1	Axis is at target position (target position reached)	Cannot be traced by oscilloscope!
0x00n00090	Read	every (PLC→NC)	UINT16	1	0/1	Axis has constant velocity or rotary speed	Cannot be traced by oscilloscope!
0x00n0009A	Read	every (PLC→NC)	UINT16	1	0/1	Operation mode not executed (busy)	Cannot be traced by oscilloscope!
0x00n0009B	Read	every (PLC→NC)	UINT16	1	0/1	Axis has instructions, is carrying instructions out	Cannot be traced by oscilloscope!
0x00n000B1	Read	every (PLC→NC)	UINT32	1	≥0	Axis error code	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n000B2	Read	every (PLC→NC)	UINT32	1	ENUM	Present state of the axis movement	
0x00n000B3	Read	every (PLC→NC)	UINT32	1	ENUM	Operation mode of the axis (rev. NC)	
0x00n000B4	Read	every (PLC→NC)	UINT32	1	ENUM	Axis calibration state	
0x00n000B5	Read	every (PLC→NC)	UINT32	1	ENUM	Axis coupling state	
0x00n000B6	Read	every (PLC→NC)	UINT32	1	≥0	SVB entries/ tasks of the axis (PRE table)	
0x00n000B7	Read	every (PLC→NC)	UINT32	1	≥0	SEC entries/ tasks of the axis (EXE table)	
0x00n000B8	Read	every (PLC→NC)	UINT32	1	≥0	Axis ID	
0x00n000B9	Read	every (PLC→NC)	UINT32	1	≥0	Operation modes state double word: Bit 0: Position range monitoring active? Bit 1: Target position window monitoring active? Bit 2: Looping distance active? Bit 3: Physical motion monitoring active? Bit 4: PEH time monitoring active? Bit 5: Backlash compensation active? Bit 6: NEW: Delayed error reaction active? Bit 7: NEW: Modulo operation mode active (modulo axis)? Bit 16: Lag error monitoring position active? Bit 17: Lag error monitoring vel. active? Bit 18: End position monitoring min. active? Bit 19: End position monitoring max. active?	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						Bit 20: Actual position correction active?	
0x00n000BA	Read	every (PLC→NC)	REAL64	e.g. mm		Actual position (calculated absolute value)	
0x00n000BB	Read	every (PLC→NC)	REAL64	e.g. mm		Modulo actual position	
0x00n000BC	Read	every (PLC→NC)	INT32	1		Modulo rotations	
0x00n000BD	Read	every (PLC→NC)	REAL64	e.g. mm/s		Actual velocity (optional)	
0x00n000BE	Read	every (PLC→NC)	REAL64	e.g. mm		Lag error position	
0x00n000BF	Read	every (PLC→NC)	REAL64	e.g. mm		Set position	
0x00n000C0	Read	every (PLC→NC)	REAL64	e.g. mm/s		Set velocity	
0x00n000C1	Read	every (PLC→NC)	REAL64	e.g. mm/s^2		Set acceleration	
0x00n000C2	Read	every (PLC→NC)	REAL64			reserve 2	
0x00n000C3	Read	every (PLC→NC)	REAL64			reserve 3	
0x00n000C4	Read	every (PLC→NC)	REAL64			reserve 4	
0x00n10000	Read/Write	Encoder: every (NC→IO)	{ 12 bytes }		STRUCT s. encoder interface	ENCODER-OUTPUT-STRUCTURE (NC→IO, 12 bytes)	Write command only optional! Consider safety aspects!
0x00n10000	Read/Write	Encoder: every (NC→IO)	{ 40 bytes }		STRUCT s. new encoder interface	ENCODER-EXTENDED-OUTPUT-STRUCTURE (new) (NC→IO, 40 bytes)	Write command only optional! Consider safety aspects!
0x00n10080	Read	Encoder: every (IO→NC)	{ 12 bytes }		STRUCT s. encoder interface	ENCODER-INPUT-STRUCTURE (IO→NC, 12 bytes)	
0x00n10080	Read	Encoder: every (IO→NC)	{ 40 bytes }		STRUCT s. new encoder interface	ENCODER-EXTENDED-INPUT-STRUCTURE (new) (IO→NC, 40 bytes)	
0x00n30000	Read/Write	Drive: every (NC→IO)	{ 12 bytes }		STRUCT see drive interface	DRIVE-OUTPUT-STRUCTURE (NC→IO, 12 bytes)	Write command only optional! Consider safety aspects!
0x00n30000	Read/Write	Drive: every (NC→IO)	{ 40 bytes }		STRUCT See new drive interface	DRIVE-EXTENDED-OUTPUT-STRUCTURE (new) (NC→IO, 40 bytes)	Write command only optional! Consider safety aspects!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n30080	Read	Drive: every (IO→NC)	{ 12 bytes }		STRUCT see drive interface	DRIVE-INPUT-STRUCTURE (NC→IO, 12 bytes)	
0x00n30080	Read	Drive: every (IO→NC)	{ 40 bytes }		STRUCT See new drive interface	DRIVE-EXTENDED-INPUT-STRUCTURE (new) (NC→IO, 40 bytes)	

7.7.5 Specification Encoder

7.7.5.1 "Index offset" specification for encoder parameter (Index group 0x5000 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1	[1 ... 255]	Encoder ID	
0x00000002	Read	every	UINT8[30+1]	1	30 characters	Encoder name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Encoder type [▸ 360]	
0x00000004	Read/Write	every	UINT32	1	Byteoffset	Input address offset (I/O-Input-Image)	change I/O address
0x00000005	Read/Write	every	UINT32	1	Byteoffset	Output address offset (I/O-Output-Image)	change I/O address
0x00000006	Read/Write	every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	resulting scaling factor (numerator / denominator) Note: from TC 3.0 the scaling factor consists of two components – numerator and denominator (default: 1.0).	
0x00000007	Read/Write	every	REAL64	e.g. mm	[±1.0E+9]	Position offset	
0x00000008	Read/Write	every	UINT16	1	[0,1]	Encoder count direction	
0x00000009	Read/Write	every	REAL64	e.g. mm	[0.001 ... 1.0E+9]	modulo factor	
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0)	Encoder mode [▸ 361]	s. appendix
0x0000000B	Read/Write	every	UINT16	1	0/1	soft end min. monitoring?	
0x0000000C	Read/Write	every	UINT16	1	0/1	Soft end max. monitoring?	
0x0000000D	Read/Write	every	REAL64	mm		Soft end position min.	
0x0000000E	Read/Write	every	REAL64	mm		Soft end position max.	
0x0000000F	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder evaluation direction [▸ 361] (enable for log. counting direction)	s. appendix

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000010	Read/Write	every	REAL64	s	[0.0...60.0]	Filter time for actual position value in seconds (P-T1)	
0x00000011	Read/Write	every	REAL64	s	[0.0...60.0]	Filter time for actual velocity value in seconds (P-T1)	
0x00000012	Read/Write	every	REAL64	s	[0.0...60.0]	filter time for actual acceleration value in seconds (P-T1)	
0x00000013	Read/Write	every	UINT8[10+1]	1		physical unit	Not implemented!
0x00000014/	Read/Write	every	UINT32	1		interpretation of the units (position, velocity, time) Bit 0: Velocity in x/min instead of x/s Bit 1: Position in thousandths of the base unit	Not implemented! bit array
0x00000015	Read	every	UINT32	INC	[0x0... 0xFFFFFFFF]	Encoder mask (maximum value of the encoder actual value in increments) Note: From TwinCAT 2.11 R2 the encoder mask may be any numerical value (e.g. 3600000) and does not have to correspond to a continuous sequence of binary ones (2^n-1).	Read-only parameter see also "Encoder Sub Mask" parameter
0x00000016	Read/Write	every	UINT16	1	0/1	Actual position correction (measurement system error correction)?	
0x00000017	Read/Write	every	REAL64	s	[0.0...60.0]	Filter time for actual position correction in seconds (P-T1)	
0x00000018	Read/Write	every	UINT32	1	[0x0... 0xFFFFFFFF]	Filter mask for raw incremental value (0x0: full passage)	
0x00000019	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder absolute dimensioning system [▶ 362]	s. appendix
0x0000001A	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder position initialization [▶ 362]	Not implemented!
0x0000001B	Read/Write	every	REAL64	e.g. mm	[≥0, modulo factor/2]	Tolerance window for modulo-start	

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x0000001C	Read	every	UINT32	1	s. ENUM (≥0)	Encoder sign interpretation (data type) [▸ 362]	
0x0000001D	Read	every	UINT16	1	0/1	Incremental or absolute encoder ? 0: Incremental encoder type 1: Absolute encoder type	
0x00000020	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder dead time compensation mode 0: Off (Default) 1: On (with velocity) 2: On (with velocity and acceleration)	
0x00000021	Read/Write	every	UINT32	1		Control double word (32 bits) for the encoder dead time compensation: Bit 0 = 0: relative I/O times (default) Bit 0 = 1: absolute I/O times	
0x00000022	Read/Write	every	INT32	ns	[±1.0E+9]	Sum of the parameterized time shifts for the encoder dead time compensation (typically positive numerical values)	
0x00000023	Read/Write	every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: numerator (=> scaling factor numerator / scaling factor denominator)	From TC 3.0
0x00000024	Read/Write	every	REAL64	1	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: denominator (=> scaling factor numerator / scaling factor denominator) Default: 1.0	From TC 3.0
0x00000025	Read/Write	every	{ REAL64 REAL64 } 16 bytes	e.g. mm/INC 1	[1.0E-12 ... 1.0E+30] [1.0E-12 ... 1.0E+30]	Component of the scaling factor: numerator	From TC 3.0

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
						Component of the scaling factor: denominator (=> scaling factor numerator / scaling factor denominator)	
0x00000030	Read/Write	every	UINT32	1		Internal encoder control double word for specifying the operation modes and properties	From 211R3 B2227
0x00000101	Read/Write	INC	UINT16	1	[0,1]	inverse search direction for ref.cam?	
0x00000102	Read/Write	INC		1	[0,1]	inverse search direction for sync pulse?	
0x00000103	Read/Write	INC	REAL64	e.g. mm	[±1.0E+9]	Reference position	
0x00000104	Read/Write	INC	UINT16	1	[0,1]	distance monitoring between Ref. cams and sync pulse active?	Not implemented!
0x00000105	Read/Write	INC	UINT32	INC	[0 ...65536]	minimum gap between Ref. cams and sync pulse in increments	Not implemented!
0x00000106	Read/Write	INC	UINT16	1	[0,1]	external sync pulse?	
0x00000107	Read/Write	INC	UINT32	1	s. ENUM (>0)	Reference mode [▸ 363]	s. appendix
0x00000108	Read/Write	INC	UINT32	1	[0x0000000F... 0xFFFFFFFF] Binary mask: (2^n - 1)	Encoder Sub Mask (maximum value of the absolute range of the encoder actual value in increments) Used, for example, as a reference mark for the referencing mode "Software Sync" and for the NC Retain Data "ABSOLUTE (MODULO)", "INCREMENTAL (SINGLETURN ABSOLUTE)"). Note 1: The Encoder Sub Mask must be smaller than or equal to the Encoder Mask.	see also "Encoder Mask"

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
						Note 2: The Encoder Mask must be an integer multiple of the Encoder Sub Mask. Note 3: The Encoder Sub Mask must be a continuous sequence of binary ones (2^n-1), e.g. 0x000FFFFF.	
0x00000110	Read/Write	INC (encoder simulation)	REAL64	1	[0.0 ... 1000000.0]	scaling/weight of the noise part for the simulation encoder	

7.7.5.2 "Index offset" specification for encoder state (Index group 0x5100 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	INT32			Error state encoder	
0x00000002	Read	every	REAL64			Actual position (charge with actual position compensation value)	Symbolic access possible! *'fPosIst'*
0x00000003	Read	every	REAL64			Modulo actual position	Symbolic access possible! *'FModuloPosIst'*
0x00000004	Read	every	INT32			Modulo actual rotation	Symbolic access possible! *'nModuloTurns'*
0x00000005	Read	every	REAL64			Optional: Actual velocity	Base Unit / s *Symbolic access possible!* *'fVeloIst'*
0x00000006	Read	every	REAL64			Optional: Actual acceleration	Base Unit / s^2 *Symbolic access possible!* *'fAccIst'*
0x00000007	Read	every	INT32			Encoder actual increments	Symbolic access possible! *'nHardIncs'*
0x00000008	Read	every	INT64			Software - actual increment counter	Symbolic access possible! *'nSoftIncs'*
0x00000009	Read/Write	every	UINT16			Reference flag ("calibrate flag")	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x0000000A	Read	every	REAL64			Actual position correction value (measurement system error correction)	
0x0000000B	Read	every	REAL64			Actual position without actual position compensation value	
0x0000000C	Read	every	REAL64	e.g. mm		Actual position compensation value due to the dead time compensation	
0x0000000D	Read	every	REAL64	s		Sum of the time shifts for encoder dead time compensation (parameterized and variable dead time) Note: A dead time is specified in the system as a positive value.	
0x0000000E	Read	every	REAL64	e.g. mm		Internal position offset as a correction value for a value reduction to the base period (modulo range)	
0x00000010	Read	every	REAL64	e.g. mm/s		Actual velocity without actual position compensation value	
0x00000012	Read	every	REAL64	e.g. mm		Unfiltered actual position (charge with actual position compensation value)	
0x00000015	Read	every	REAL64	e.g. mm/s		Optional: Unfiltered actual velocity	Base Unit / s
0x00000016	Read	every	**READ** (16 bytes * N)			Read the actual position buffer	from TC 2.11 R3
			{				
			UINT32	ns	≥0	DcTimeStamp with 32 bits	
			UINT32			Reserve	
			REAL64	e.g. mm	±∞	Actual position for the associated DC timestamp	
			} [N]				
0x00000101	Read	INC	REAL64	e.g. mm		Read back the position difference between the hardware latch being activated and becoming valid	Cannot be traced by oscilloscope!

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
0x00000200	**Read Write**	**Function group "TouchProbeV2":** **- SERCOS (SoE),** **- EtherCAT (CANopen DS402)** **- SoftDrive (TCom)**	**WRITE** (24 bytes)			Read "Touch Probe" state (state of external latch)	TC 2.11 Build 1547 only for SEC-port 501
			{				
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32[5]			Reserved	
			}				
			READ (64 bytes)				
			{				
			UINT32	1	[0/1]	Touch probe rising edge active?	
			UINT32	1	[0/1]	Touch probe rising edge became valid?	
			REAL64	e.g. mm		Touch probe rising edge position value	
			UINT32	1	≥0	Touch probe rising edge counter (continuous mode)	
			UINT32			Reserved	
			UINT32	1	[0/1]	Touch probe falling edge active?	
			UINT32	1	[0/1]	Touch probe falling edge became valid?	
			REAL64	e.g. mm		Touch probe falling edge position value	
			UINT32	1	≥0	Touch probe falling edge counter (continuous mode)	
			UINT32[5]			Reserved	
			}				
0x00000201	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT16	1	[0,1]	"External latch function" active? or "Touch probe function" active ? *(edge-independent)*	Cannot be traced by oscilloscope!
0x00000201	Read	**CANopen**	UINT32[4]	1	[0,1]	"External latch functions 1 to 4" active? or "Touch probe functions 1 to 4" active ?	Cannot be traced by oscilloscope!
0x00000202	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT16	1	[0,1]	External latch value became valid? or	cf. Axis interface NcToPlc (state double word)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
						touch probe latched? *(edge-independent)*	
0x00000202	Read	**CANopen**	UINT32[4]	1	[0,1]	External latch values 1 to 4 became valid? or touch probes 1 to 4 latched?	cf. Axis interface NcToPlc (state double word)
0x00000203	Read	**KL5101, SERCOS, AX2xxx, ProiDrive**	UINT32	INC		External / touch probe hardware incremental latch value	
0x00000204	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT64	INC		External / touch probe software incremental latch value	
0x00000205	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	REAL64	e.g. mm		External / touch probe position latch value	Base Unit
0x00000205	Read	**CANopen**	REAL64[4]	e.g. mm		External touch probe values / position latch values	Base Unit
0x00000206	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT32	INC		Difference hardware incremental latch values (NewLatch - LastLatch)	Cannot be traced by oscilloscope!
0x00000207	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT64	INC		Difference software incremental latch values (NewLatch - LastLatch)	Cannot be traced by oscilloscope!
0x00000208	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	REAL64	e.g. mm		Difference position latch values (NewLatch - LastLatch)	Cannot be traced by oscilloscope! Base Unit
0x00000210	Read	**KL5101, AX2xxx, ProviDrive**	UINT16	1	[0,1]	"External latch function" for *rising edge* active? or "Touch probe function" for *rising edge* active?	**Extension for KL5101 (3E), AX2xxx (3.51) and ProviDrive (3.1)** Cannot be traced by oscilloscope!
0x00000210	Read	**CANopen**	UINT16[4]	1	[0,1]	"External latch function" for *rising edge* active? or "Touch probe function" for *rising edge* active?	Cannot be traced by oscilloscope!
0x00000211	Read	**KL5101, AX2xxx, ProviDrive**	UINT16	1	[0,1]	"External latch function" for *falling edge* active? or	**Extension for KL5101 (3E), AX2xxx (3.51) and ProviDrive (3.1)**

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Note
						"Touch probe function" for *falling edge* active?	Cannot be traced by oscilloscope!
0x00000211	Read	**CANopen**	UINT16[4]	1	[0,1]	"External latch function" for *falling edge* active? or "Touch probe function" for *falling edge* active?	Cannot be traced by oscilloscope!

7.7.5.3 "Index offset" specification for encoder functions (Index group 0x5200 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x0000001A	Write	every	{			Set actual position encoder/axis	Base Unit
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	mm	±∞	Actual position for encoder/axis **caution by using!**	
			}				
0x0000001B	Write	every	void			Re-initialization of the actual encoder position Note: Only takes effect for reference system “ABSOLUTE (with single overflow)”	From TC 2.11R3 Build 2261
0x00000200	Write	Function group "TouchProbeV2": - SERCOS (SoE), - EtherCAT (CANopen DS402) - SoftDrive (TCom)	{			Activate "Touch Probe" (external latch)	From TC 2.11 Build 154 Only for SAF-port 501
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32	1	[0,1]	Signal edge (0=rising edge, 1=falling edge)	
			UINT32	1	[1,2]	Probe mode (1=single, 2=continuous, ...)	
			UINT32			Reserved (e.g. signal source)	
			UINT32			Reserved	
			UINT32			Reserved	
			} 24 bytes				
0x00000201	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	VOID			Activate "External latch" or	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
						Activate "Measuring sensor function" *(typically rising edge)*	
0x00000201	Write	CANopen	UINT32[4]			Activate "External Latch" 1 to 4 or Activate "Measuring sensor function" 1 to 4 *(typically rising edge)*	
0x00000202	Write	KL5101, AX2xxx, PROFIDrive	VOID			Activate "External Latch" or Activate "Measuring sensor function" *(falling edge)*	
0x00000202	Write	CANopen	UINT32[4]			Activate "External Latch" 1 to 4 or Activate "Measuring sensor function" 1 to 4 *(falling edge)*	
0x00000205	Write	Function group "TouchProbeV2": - SERCOS (SoE), - EtherCAT (CANopen DS402) - SoftDrive (TCom)	{			Deactivate "Touch Probe" (external latch)	From TC 2.11 Build 1547 only for SAF-port 501
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32	1	[0,1]	Signal edge (0=rising edge, 1=falling edge)	
			UINT32			Reserved	
			UINT32			Reserved	
			UINT32			Reserved	
			UINT32			Reserved	
			} 24 byte				
0x00000205	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	VOID			Deactivate "External Latch" or Deactivate "Measuring sensor function"	
0x00000205	Write	CANopen	UINT32[4]			Deactivate "External Latch" or Deactivate "Measuring sensor function"	
0x00000210	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	REAL64	e.g. mm	±∞	Set "External latch event" and "External latch position"	Only for EtherCAT:

7.7.5.4 "Index offset" specification for cyclic encoder process data (Index group 0x5300 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000000	Read/Write	every (NC→IO)	{		STRUCT	ENCODER-OUTPUT-STRUCTURE	Write command only optional!

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
					see encoder interface or see extended encoder interface	(NC→IO, 12 bytes) or ENCODER-OUTPUT-STRUCTURE (NC→IO, 40 bytes)	Consider safety aspects!
			INT32	INC	≥ 0	nOutData1	
			INT32	INC	≥ 0	nOutData2	
			UINT8	1	≥ 0	nControl1	
			UINT8	1	≥ 0	nControl2	
			UINT8	1	≥ 0	nControl3	
			UINT8	1	≥ 0	nControl4	
			Optional:				
			INT32	INC	≥ 0	nOutData3	
			INT32	INC	≥ 0	nOutData4	
			INT32	INC	≥ 0	nOutData5	
			INT32	INC	≥ 0	nOutData6	
			UINT8	1	≥ 0	nControl5	
			UINT8	1	≥ 0	nControl6	
			UINT8	1	≥ 0	nControl7	
			UINT8	1	≥ 0	nControl8	
			INT32		≥ 0	reserved	
			INT32		≥ 0	reserved	
			}				
0x00000001	Write	every (IO→NC)	{		STRUCT s. encoder interface	Bitwise access to ENCODER-OUTPUT-STRUCT (NC→IO, 40 Byte) NCENCSTRUCT_OUT2	Write command only optional! Consider safety aspects! From TC2.11 R3 B2303
			UINT32	1	[0 … 39]	ByteOffset Relative address offset [0..39] in output structure. E.G.: To write "nControl1" the ByteOffset must be 8.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	BitSelectMask (BSM) The mask defines write enabled bits in a DWORD. Zero bits are protected and remain unaffected.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	Value Only those bits in value are overwritten where BSM equals 1.	
			}				
0x00000080	Write	every (IO→NC)	{		STRUCT	ENCODER-INPUT-STRUCTURE (IO→NC, 12	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
					see encoder interface or see extended encoder interface	bytes) or optional ENCODER-INPUT-STRUCTURE (IO→NC, 40 bytes)	
			INT32	INC	≥ 0	nInData1	
			INT32	INC	≥ 0	nInData2	
			UINT8	1	≥ 0	nStatus1	
			UINT8	1	≥ 0	nStatus2	
			UINT8	1	≥ 0	nStatus3	
			UINT8	1	≥ 0	nStatus4	
			Optional:				
			INT32	INC	≥ 0	nInData3	
			INT32	INC	≥ 0	nInData4	
			INT32	INC	≥ 0	nInData5	
			INT32	INC	≥ 0	nInData6	
			UINT8	1	≥ 0	nStatus5	
			UINT8	1	≥ 0	nStatus6	
			UINT8	1	≥ 0	nStatus7	
			UINT8	1	≥ 0	nStatus8	
			INT32		≥ 0	Reserved	
			INT32		≥ 0	Reserved	
			}				

7.7.6 Specification Controller

7.7.6.1 "Index offset" specification for controller parameter (Index group 0x6000 + ID)

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1	[1 ... 255]	Controller ID	
0x00000002	Read	every	UINT8[30+1]	1	30 characters	Controller name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Controller type	
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0)	Controller mode	
0x0000000B	Read/Write	every	REAL64	%	[0.0 ... 1.0]	Weighting of the velocity pre-control (default value: 1.0 == 100 %)	
0x00000010	Read/Write	every	UINT16	1	0/1	Lag error monitoring Pos.?	
0x00000011	Read/Write	every	UINT16	1	0/1	Lag error monitoring Velocity?	
0x00000012	Read/Write	every	REAL64	mm		max. lag error position	
0x00000013	Read/Write	every	REAL64	s		Max. lag error filter time position	
0x00000014/	Read/Write	every	REAL64	mm/s		max. lag error velocity	
0x00000015	Read/Write	every	REAL64	s		Max. lag error filter time velocity	

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Note
0x00000021	Read/Write	every	REAL64	1	[0.0...1000000.0]	Scaling factor (multiplier) for the difference between the lag error of the master and that of the slave (Conversion in the same coordinate system of the master)	Reserved function, no standard!
0x00000100	Read/Write	P/PID (Pos., (velocity)	REAL64	1	[0.0...1.0]	Maximum output limitation (±) for controller total output	(default value: 0.5 == 50%)
0x00000102	Read/Write	P/PID (pos.)	REAL64	(mm/s) / mm	[0.0...1000.0]	Proportional gain kp or kv	base unit / s / base unit Position control
0x00000103	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Integral action time T_n	Position control
0x00000104	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Derivative action time T_v	Position control
0x00000105	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Damping time T_d	Position control
0x00000106	Read/Write	PP (Pos.)	REAL64	(mm/s) / mm	[0.0...1000.0]	Additional proportional gain, k_p or k_v, that applies above a limiting velocity in percent.	base unit / s / base unit Position control
0x00000107	Read/Write	PP (Pos.)	REAL64	%	[0.0...1.0]	Threshold velocity in percent above which the additional proportional gain, k_p or k_v, applies	(default value: 0.01 == 1%)
0x00000108	Read/Write	P/PID (Acc.)	REAL64	s	[0.0 ... 100.0]	Proportional gain k_a	Acceleration pre-control
0x0000010A	Read/Write	every	UINT32	1	ENUM	Filter for maximum increase in the set speed (acceleration-limited): 0: Off, 1: Velo, 2: Pos+Velo	Reserved function, no standard!
0x0000010B	Read/Write	every	REAL64	mm/s^2		Filter for maximum increase in the set speed (maximum acceleration)	Reserved function, no standard!
0x0000010D	Read/Write	P/PID	REAL64	mm	[0.0 ... 10000.0]	"dead band" for position error (control deviation) (for P/PID controllers with velocity or torque interface)	reserved function
0x0000010F	Read/Write	P/PP/PID (pos.) Slave control	REAL64	(mm/s) / mm	[0.0...1000.0]	Slave coupling difference control:	Slave coupling difference control

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Note
						Proportional gain k_{cp}	
0x00000110	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: active/passive	
0x00000111	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: Hold mode	
0x00000112	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: Fading mode	
0x00000114	Read/Write	P (Pos.)	REAL64	%	[0.0 ... 1.0]	Automatic offset calibration: Pre-control limit	(default value: 0.05 == 5%)
0x00000115	Read/Write	P (Pos.)	REAL64	s	[0.1 ... 60.0]	Automatic offset calibration: Time constant	
0x00000116	Read/Write	PID (pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I part in percent (default setting: 0.1 == 10%)	
0x00000117	Read/Write	PID (pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D part in percent (default setting: 0.1 == 10%)	
0x00000118	Read/Write	PID (pos.)	UINT16	1	0/1	Deactivation of the I part during an active positioning process (assuming I part active)? (default setting: 0 = FALSE)	
0x00000120	Read/Write	PID (pos.)	REAL64	s	≥0	PT-1 filter value for position error (pos. control deviation)	Reserved function, no standard!
0x00000202	Read/Write	P/PID (velocity)	REAL64	1	[0.0...1000.0]	Proportional gain k_p or k_v	Velocity control
0x00000203	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Integral action time T_n	Velocity control
0x00000204	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Derivative action time T_v	Velocity control
0x00000205	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Damping time T_d	Velocity control
0x00000206	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I part in percent (default setting: 0.1 == 10%)	Velocity control
0x00000207	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D part in	Velocity control

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Note
						percent (default setting: 0.1 == 10%)	
0x0000020D	Read/Write	P/PID (velocity)	REAL64	mm/s	[0.0 ... 10000.0]	"dead band" for velocity error (control deviation) (for P/PID controllers with velocity or torque interface)	reserved function
0x00000220	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-2 filter value for velocity error (vel. control deviation)	Velocity control, not standard!
0x00000221	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-1 filter value for velocity error (vel. control deviation)	Reserved function, no standard!
0x00000250	Read/Write	P/PI (observer)	UINT32	1	s. ENUM (≥0)	Observer mode [▶ 360] for control in the torque interface 0: OFF (default) 1: LUENBERGER	From TC 2.10 Build 1320
0x00000251	Read/Write	P/PI (observer)	REAL64	Nm / A	>0.0	Motor: Torque constant K_T	
0x00000252	Read/Write	P/PI (observer)	REAL64	kg m^2	>0.0	Motor: Moment of inertia J_M	
0x00000253	Read/Write	P/PI (observer)	REAL64	Hz	[100.0 ... 2000.0] Default: 500	Bandwidth f_0	
0x00000254	Read/Write	P/PI (observer)	REAL64	1	[0.0 ... 2.0] Default: 1.0	Correction factor k_c	
0x00000255	Read/Write	P/PI (observer)	REAL64	s	[0.0 ... 0.01] Default: 0.001	Velocity filter (1st order): time constant T	
0x00000A03	Read/Write	PID (MW)	REAL64	cm^2	[0.0 ...1000000]	Cylinder area A_A of the A side in cm^2	
0x00000A04	Read/Write	PID (MW)	REAL64	cm^2	[0.0 ...1000000]	Cylinder area A_B of the B side in cm^2	
0x00000A05	Read/Write	PID (MW)	REAL64	cm^3/s	[0.0 ...1000000]	Nominal volume flow Q_{nom} in cm^3/s	
0x00000A06	Read/Write	PID (MW)	REAL64	bar	[0.0 ...1000000]	Nominal pressure or valve pressure drop, P_{nom} in bar	
0x00000A07	Read/Write	PID (MW)	UINT32	1	[1 ... 255]	Axis ID for the system pressure P_0	

7.7.6.2 "Index offset" specification for controller state (Index group 0x6100 + ID)

Access	Controller type	Data type	Phys. unit	Definition range	Description	Note	Access
0x00000001	Read	every	INT32			Error state controller	Symbolic access possible! *'nErrState'*
0x00000002	Read	every	REAL64	e.g. mm/s		Controller output in absolute units	Base Unit / s *Symbolic access possible!* *'fOutput'*
0x00000003	Read	every	REAL64	%		Controller output in percent	Cannot be traced by oscilloscope!
0x00000004	Read	every	REAL64	V		Controller output in volts	Cannot be traced by oscilloscope!
0x0000000D	Read	every	REAL64	mm		Lag error position (without dead time compensation)	Base Unit
0x0000000E	Read	every	REAL64	mm		Lag error position (without set position correction)	Base Unit
0x0000000F	Read	every	REAL64	mm		Lag error position (with set position correction and with dead time compensation)	Base Unit *Symbolic access possible!* *'fPosDiff'*
0x00000010	Read	every	REAL64	mm		Peak hold value for maximum negative lag error of the position	Base Unit
0x00000011	Read	every	REAL64	mm		Peak hold value for minimum positive lag error of the position	Base Unit
0x00000012	Read	every	REAL64	mm/s		Lag error velocity	Base Unit / s
0x00000021	Read	every	REAL64	mm		Difference (deviation) between the lag error from master and slave axis (master error minus slave error)	Base Unit *Symbolic access possible via axis!* *'fPosDiffCouple'*
0x00000022	Read	every	REAL64	mm		PeakHold value for the maximum negative difference between master and slave axis lag error of the position	Base Unit

Access	Controller type	Data type	Phys. unit	Definition range	Description	Note	Access
0x00000023	Read	every	REAL64	mm		PeakHold value for the maximum positive difference between master and slave axis lag error of the position	Base Unit
0x00000101	Read	P/PID (pos.)	REAL64	e.g. mm/s		P part of the controller in absolute units	
0x00000102	Read	PID (pos.)	REAL64	e.g. mm/s		I part of the controller in absolute units	
0x00000103	Read	PID (pos.)	REAL64	e.g. mm/s		D part of the controller in absolute units	
0x00000104	Read	PID (pos.)	UINT16	1	0/1	Limitation of the I part active?	
0x00000105	Read	PID (pos.)	UINT16	1	0/1	Limitation of the D part active?	
0x00000106	Read	PID (pos.)	UINT16	1	0/1	ARW measure for the I part active?	ARW: Anti Reset Windup
0x00000110	Read	PID (pos.)	REAL64	e.g. mm/s		Acceleration pre-control Y_{acc} of the controller in absolute units **Function depends on controller type!**	Acceleration pre-control
0x00000111	Read	PP (Pos.)	REAL64	mm/s/ mm	≥0	Internal interpolated proportional gain kp or kv	PP controller
0x0000011A 0x0000011B 0x0000011C 0x0000011D 0x0000011E 0x0000011F 0x00000120 0x00000121 0x00000122 0x00000123 0x00000124	Read	P (Pos.)	 UINT32 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64	 1 mm mm/s mm/s mm/s^2 mm mm mm/s mm/s^2 mm/s mm/s^2		Set velocity filter: InternalPhase InternalPosSoll Error! TestVeloSoll InternalLimited VeloSoll InternalAccSoll Rel InternalPosSoll Rel PosSollCorrect ed! VeloSollCorrect ed! AccSollCorrect ed! TestVeloSollCo rrected TestAccSollCor rected	Reserved function, no standard!
0x00000201	Read	P,PID (velocity)	REAL64	e.g. mm/s		Velocity part of the controller	Base Unit / s
0x00000202	Read	P,PID (velocity)	REAL64	%		Velocity part of the controller in percent	Cannot be traced by oscilloscope!

Access	Controller type	Data type	Phys. unit	Definition range	Description	Note	Access
0x00000203	Read	P,PID (velocity)	REAL64	V		Velocity part of the controller in volts	Cannot be traced by oscilloscope!
0x00000201	Read	P/PID (velocity)	REAL64	e.g. mm/s		P part of the controller in absolute units	
0x00000202	Read	P/PID (velocity)	REAL64	e.g. mm/s		I part of the controller in absolute units	
0x00000203	Read	P/PID (velocity)	REAL64	e.g. mm/s		D part of the controller in absolute units	
0x00000204	Read	P/PID (velocity)	UINT16	1	0/1	Limitation of the I part active?	
0x00000205	Read	P/PID (velocity)	UINT16	1	0/1	Limitation of the D part active?	
0x00000206	Read	P/PID (velocity)	UINT16	1	0/1	ARW measures for the I part active?	ARW: Anti Reset Windup
0x0000020A	Read	P/PID (velocity)	REAL64	e.g. mm/s		Total input size of the velocity controller	
0x00000250	Read	P/PI (observer)	REAL64	e.g. mm		Observer: Position difference (Actual position - Observer position	
0x00000251	Read	P/PI (observer)	REAL64	e.g. mm		Observer: Position	
0x00000252	Read	P/PI (observer)	REAL64	e.g. mm/s		Observer: Velocity 2 (for P part)	
0x00000253	Read	P/PI (observer)	REAL64	e.g. mm/s		Observer: Velocity 1 (for I part)	
0x00000254	Read	P/PI (observer)	REAL64	e.g. mm/s^2		Observer: Acceleration	
0x00000255	Read	P/PI (observer)	REAL64	A		Observer: Actual motor current	
0x00000256	Read	P/PI (observer)	UINT16	1	0/1	Observer: Limitation of the I part active?	
0x00000A00	Read	PID (MW)	REAL64	%	[-1.0...1.0]	Calculation of the set velocity (pre-control) in percent	
0x00000A01	Read	PID (MW)	REAL64	e.g. mm/s		P part of the controller in absolute units or percent (according to output weight)	
0x00000A02	Read	PID (MW)	REAL64	e.g. mm/s		I part of the controller in absolute units or percent (according to output weight)	
0x00000A03	Read	PID (MW)	REAL64	e.g. mm/s		D part of the controller in absolute units	

Access	Controller type	Data type	Phys. unit	Definition range	Description	Note	Access
						or percent (according to output weight)	
0x00000A04	Read	PID (MW)	UINT16	1	0/1	Limitation of the I part active?	
0x00000A05	Read	PID (MW)	UINT16	1	0/1	Limitation of the D part active?	
0x00000A10	Read	PID (pos.)	REAL64	e.g. mm/s		Acceleration pre-control Y_{acc} of the controller in absolute units	Acceleration pre-control

7.7.6.3 "Index offset" specification for controller functions (Index group 0x6200 + ID)

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks

7.7.7 Specification Drive

7.7.7.1 "Index offset" specification for drive parameter (Index group 0x7000 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1	[1 ... 255]	Drive ID	
0x00000002	Read	every	UINT8[30+1]	1	30 characters	Drive name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Drive type [▶ 364]	
0x00000004	Read/Write	every	UINT32	1	Byteoffset	Input address offset (I/O-Input-Image)	change I/O address
0x00000005	Read/Write	every	UINT32	1	Byteoffset	Output address offset (I/O-Output-Image)	change I/O address
0x00000006	Read/Write	every	UINT16	1	[0,1]	motor polarity	
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0)	drive mode	
0x0000000B	Read/Write	every	REAL64	%	[-1.0 ... 1.0]	Minimum output limit (output limitation) (Default setting: -1.0 == -100%)	
0x0000000C	Read/Write	every	REAL64	%	[-1.0 ... 1.0]	Maximum output limit (output limitation) (Default setting: 1.0 == 100%)	
0x0000000D	Read	every	UINT32	INC		Maximum number of output increments (output mask)	
0x00000010	Read/Write	every	UINT32	1		Internal Drive Control double word to determine the drive operation modes	Reserved!

Index offset (Hex)	Access	Drive type	Data type	Phys. unit	Definition range	Description	Note
0x00000011	Read/Write	every	UINT32	1	≥ 5	Internal drive reset counter (time in NC cycles for enable and reset)	Reserved!
0x00000020	Read/Write	every	UINT32	1	s. ENUM (≥0)	Drive dead time compensation mode 0: Off (Default) 1: On (with velocity) 2: On (with velocity and acceleration)	s. appendix
0x00000021	Read/Write	every	UINT32	1		Control double word (32 bits) for the drive dead time compensation: Bit 0 = 0: relative I/O times (default) Bit 0 = 1: absolute I/O times	
0x00000022	Read/Write	every	INT32	ns	[±1.0E+9]	Sum of the parameterized time shifts for the drive dead time compensation (typically positive numerical values)	
0x00000101	Read/Write	Servo	REAL64	e.g. mm/s	>0.0	Reference velocity at reference output (velocity pre-control)	Base Unit / s
0x00000102	Read/Write	Servo	REAL64	%	[0.0 ... 5.0]	reference output in percent	
0x00000103	Read	Servo	REAL64	e.g. mm/s	>0.0	resulting velocity at 100% output	Base Unit / s
0x00000104	Read/Write	Servo	REAL64	e.g. mm/s	±∞	velocity offset (DAC offset) for drift calibration (offset calibration) of the axis	Base Unit / s
0x00000105	Read/Write	Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	velocity scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x00000106	Read/Write	Profi Drive DSC	UINT32	0.001 * 1/s	≥ 0	Profibus/Profi Drive DSC: Position control gain Kpc	Only for Profi Drive DSC
0x00000107	Read/Write	Profi Drive DSC	REAL64	1	≥ 0.0	Profibus/Profi Drive DSC: Scaling for calculation of 'XERR' (default: 1.0)	Only for Profi Drive DSC

Index offset (Hex)	Access	Drive type	Data type	Phys. unit	Definition range	Description	Note
0x00000109	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	position scaling (scaling factor to react to the weight in the drive)	For Sercos, CANopen
0x0000010A	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	acceleration scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x0000010B	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Torque scaling (rotary motor) or force scaling (linear motor) (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x0000010D	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Damping time for drive velocity output	For Sercos, CANopen
0x0000010E	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Damping time for drive acceleration output	For Sercos, CANopen
0x0000010F	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Damping time for drive torque output or force output	For Sercos, CANopen
0x00000110	Read/Write	Servo (Sercos, CANopen)	UINT32	1	s. ENUM (≥0)	Optional output filtering of the set position value: Filter type 0: OFF (default) 1: Moving Average 2: P-Tn	For Sercos, CANopen
0x00000111	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Optional output filtering of the position setpoint: The maximum filter time depends on the NC cycle time and is limited **to the following maximum value:** 1 ms => 64 ms 2 ms => 128 ms 3 ms => 192 ms	For Sercos, CANopen
0x00000112	Read/Write	Servo (Sercos, CANopen)	UINT32	1	[0 ... 10]	Optional output filtering of the position setpoint: Filter order 'n' (only for P-Tn type)	For Sercos, CANopen
0x00000120	Read/Write	Servo/ hydraulics/	UINT32	1	≥ 0	Table ID (0: no table)	Only for KL4xxx, M2400, Universal

Index offset (Hex)	Access	Drive type	Data type	Phys. unit	Definition range	Description	Note
0x00000121	Read/Write	Servo/ hydraulics	UINT32	1	≥ 0	Interpolation type 0: linear 2: Spline	Only for KL4xxx, M2400, Universal
0x00000122	Read/Write	Servo/ hydraulics	REAL64	%	[-1.0 ... 1.0]	Output offset in percent **Acts according to the characteristic evaluation!**	Only for KL4xxx, M2400, Universal
0x00000151	Read/Write	Servo/non-linear	REAL64	1	[0.0 ... 100.0]	Quadrant equalizing factor (relation between quadrants I and III)	
0x00000152	Read/Write	Servo / non-linear	REAL64	1	[0.01 ... 1.0]	velocity reference point in percent (1.0 == 100 %)	
0x00000153	Read/Write	Servo / non-linear	REAL64	1	[0.01 ... 1.0]	Output reference point in percent (1.0 == 100%)	
0x00000301	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 1	
0x00000302	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 2	
0x00000303	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 3	
0x00000304	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 4	
0x00000305	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 5	
0x00000306	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 6	
0x00000307	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 7	
0x00000308	Read/Write	Stepper motor	UINT8			Bit mask: Cycle 8	
0x00000310	Read/Write	Stepper motor	UINT8			Bit mask: Holding current	

7.7.7.2 "Index offset" specification for drive state (Index group 0x7100 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Read	every	INT32			Error state drive	Symbolic access possible! *'nErrState'*
0x00000002	Read	every	REAL64	e.g. mm/s		Total output in absolute units	Base Unit / s *Symbolic access possible!* *'fOutput'*
0x00000003	read	every	REAL64	%		Total output in percent	

Index offset (Hex)	Access	Drive type	Data type	Phys. unit	Definition range	Description	Note
0x00000004	read	every	REAL64	V		Total output in volts	Cannot be traced by oscilloscope!
0x00000005	read	every	REAL64	e.g. mm/s		PeakHold value for maximum negative total output	Base Unit / s
0x00000006	read	every	REAL64	e.g. mm/s		PeakHold value for maximum positive total output	Base Unit / s
0x0000000C	read	every	REAL64	e.g. mm		Set position correction value for drive output due to dead time compensation	
0x0000000D	read	every	REAL64	s		Sum of time shift for drive dead time compensation (parameterized and variable dead time) Note: A dead time is specified in the system as a positive value.	
0x00000013	read	every	REAL64	%		Total output in percent (according to non-linear characteristic curve!)	
0x00000014	read	every	REAL64	V		Total output in volts (according to non-linear characteristic curve!)	Cannot be traced by oscilloscope!
0x0000011A	read	Servo (Sercos, CANopen)	REAL64	e.g. mm		Optional output filtering: Filtered set position	NEW For Sercos, CANopen
0x0000011E	read	Servo (Sercos, CANopen)	REAL64	e.g. mm/s		Optional output filtering: Filtered set velocity	NEW For Sercos, CANopen
0x0000011F	read	Servo (Servo, CANopen)	REAL64	e.g. mm/s^2		Optional output filtering: Filtered set acceleration / set deceleration	NEW For Sercos, CANopen

7.7.7.3 "Index offset" specification for drive functions (Index group 0x7200 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. unit	Definition range	Description	Note
0x00000102	Write	SERVO	{			Remove and delete the characteristic drive table	Only for SEC-Port 501!
			ULONG	1	>0	see Axis function with index offset 0x00000012	
			}				

7.7.7.4 "Index offset" specification for cyclic drive process data (Index group 0x7300 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x00000000	Read/Write	every (NC→IO)	{		STRUCT see drive interface or see extended drive interface	DRIVE-OUTPUT-STRUCTURE (NC→IO, 12 bytes) or DRIVE-OUTPUT-STRUCTURE (NC→IO, 40 bytes)	Write command only optional! Consider safety aspects!
			INT32	INC	≥ 0	nOutData1	
			INT32	INC	±2^31	nOutData2	
			UINT8	1	≥ 0	nControl1	
			UINT8	1	≥ 0	nControl2	
			UINT8	1	≥ 0	nControl3	
			UINT8	1	≥ 0	nControl4	
			Optional:				
			INT32	INC	≥ 0	nOutData3	
			INT32	INC	≥ 0	nOutData4	
			INT32	INC	≥ 0	nOutData5	
			INT32	INC	≥ 0	nOutData6	
			UINT8	1	≥ 0	nControl5	
			UINT8	1	≥ 0	nControl6	
			UINT8	1	≥ 0	nControl7	
			UINT8	1	≥ 0	nControl8	
			INT32	1	≥ 0	reserved	
			INT32	1	≥ 0	reserved	
			}				
0x00000001	Write	every (IO→NC)	{		STRUCT s. drive interface	Bitwise access to DRIVE-OUTPUT-STRUKTUR (NC→IO, 40 Byte) *NCDRIVESTRUCT_OUT2*	Write command only optional! Consider safety aspects! From TC 2.11 R3 B2303
			UINT32	1	[0 … 39]	ByteOffset Relative address offset [0..39] in output structure.	

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
						E.G.: To write “nControl1” the ByteOffset must be 8.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	BitSelectMask (BSM) The mask defines write enabled bits in a DWORD. Zero bits are protected and remain unaffected.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	Value Only those bits in value are overwritten where BSM equals 1.	
			}				
0x00000080	Write	every (IO→NC)	{		STRUCT see drive interface or see extended drive interface	DRIVE-INPUT-STRUCTURE (IO→NC, 12 bytes) or DRIVE-INPUT-STRUCTURE (IO→NC, 40 bytes)	
			INT32	INC	≥ 0	nInData1	
			INT32	INC	±2^31	nInData2	
			UINT8	1	≥ 0	nStatus1	
			UINT8	1	≥ 0	nStatus2	
			UINT8	1	≥ 0	nStatus3	
			UINT8	1	≥ 0	nStatus4	
			Optional:				
			INT32	INC	≥ 0	nInData3	
			INT32	INC	≥ 0	nInData4	
			INT32	INC	≥ 0	nInData5	
			INT32	INC	≥ 0	nInData6	
			UINT8	1	≥ 0	nStatus5	
			UINT8	1	≥ 0	nStatus6	
			UINT8	1	≥ 0	nStatus7	
			UINT8	1	≥ 0	nStatus8	
			INT32	1	≥ 0	Reserved	
			INT32	1	≥ 0	Reserved	
			}				

7.7.8 Specification Tables

7.7.8.1 "Index offset" specification for table parameter (Index group 0xA000 + ID)

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	read	every	UINT32	1	[1 ... 255]	Table ID	
0x00000002	read	every	UINT8[30+1]	1	30 characters	Table name	

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
0x00000003	read	every	UINT32	1	s. ENUM (>0)	Table sub types [▸ 366]	
0x00000004	read	every	UINT32	1	s. ENUM (>0)	Table main types [▸ 366]	
0x00000010	read	every	UINT32	1	[0... 16777216]	Number of lines (n)	
0x00000011	read	every	UINT32	1	[0... 16777216]	Number of columns (m)	
0x00000012	read	every	UINT32	1	≥0	Number of total elements (n*m)	
0x00000013	read	equidistant table	REAL64	e.g. mm	≥0.0	Step size (position delta) (equidistant tables)	Base Unit
0x00000014	read	cyclical table	REAL64	e.g. degrees	≥0.0	Master period (cyclical tables)	Base Unit
0x00000015	read	cyclical table	REAL64	e.g. degrees	≥0.0	Slave difference per master period (cyclic tables)	Base Unit
0x0000001A	Read/Write	"Motion Function" (motion laws)	{			Activation type for online changes of table data (MF only)	
			UINT32	ENUM	s. appendix	Activation mode 0: 'instantaneous' (default) 1: 'master cam pos.' 2: 'master' axis pos.' 3: 'next cycle' 4: 'next cycle once' 5: 'as soon as possible' 6: 'off' 7: 'delete queued data'	
			REAL64	e.g. mm	±∞	Activation position	
			UINT32	ENUM	s. appendix	Master scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			UINT32	ENUM	s. appendix	Slave scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			}				
0x00000020	Read/Write	every	{			Write single value [n,m]:	

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
			UINT32	1	[0 ... 16777216]	n-th line	
			UINT32	1	[0 ... 16777216]	m-th column	
			REAL64	e.g. mm	±∞	Single value	
			}				
0x00000021	ReadWrite	every	*REAL64	e.g. mm	±∞	Read slave position for the specified master position (related to the "raw values" in the table)	
0x00000022	ReadWrite	"Motion Function" (motion laws)	**Write**			Read the "Motion Function" as a "point cloud"	Only possible on a line-by-line basis! (integer multiple)
			{				
			UINT16	1	0/1	Prompt consistent data adoption?	
			UINT16	1	Bitmask (≥0)	Select bit mask (number of columns m is master position plus number of bits): Bit 0: Pos (Slave) Bit 1: Velo (Slave) Bit 2: Acc (Slave) Bit 3: Jerk (Slave)	
			REAL64	e.g. mm	±∞	Startposition (Master)	
			REAL64	e.g. mm	> 0.0	Step size	
			}				
			Read				
			{				
			REAL64[x*m]	e.g. mm	±∞	Read x rows starting from the master start position: (x*m)-values (one or more lines)	
			}				
0x00000023	ReadWrite	every	**Write**			Read slave values for the specified master position (related to the "raw values" in the table)	
			REAL64	e.g. mm	±∞	Master position	
			Read				
			{				
			REAL64	e.g. mm	±∞	Slave position	
			REAL64	mm/s	±∞	Slave velocity	
			REAL64	mm/s^2	±∞	Slave acceleration	
			}				

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
0x00000050	Read/Write	every	REAL64 [64]	1	±∞	Characteristic values in the table	
0x00000050	ReadWrite	every	**Write**			Read the characteristic values in a table in relation to the nominal master velocity	
			REAL64 [64]	...	±∞	Optional nominal master reference velocity "fMasterVeloNom" (normalized => 1.0 mm/s), the remaining elements are not evaluated	
			Read				
			REAL64 [64]	...	±∞	Read the characteristic values in a table	
0x00000115	Write	monotonic linear monotonic cycl.	{			Set/change the table scaling:	
			REAL64	1	[±1000000.0]	Original weighting of the table	
			REAL64	e.g. mm	[±1000000.0]	Master column position offset	
			REAL64	1	[±1000000.0]	Scaling of the master column	
			REAL64	e.g. mm	[±1000000.0]	Position offset of the slave column	
			REAL64	1	[±1000000.0]	Scaling of the slave column	
			REAL64	e.g. mm	[±1000000.0]	Lower range limit (starting position)	
			REAL64	e.g. mm	[±1000000.0]	Upper range limit (end position)	
			}				

0x01000000 +n-th start line	Read/ Write[<=16777216]	every	{
			REAL64[x*m]
			}
				e.g. mm	±∞	Read/write x lines starting from the n-th line:
						(x*m)-values (one or more lines)
						Value range n: [0 ... 16777216]
							Only possible on a line-by-line basis!
							(integer multiple)

0x02000000 +m-th start column	Read/ Write[<=16777216]	every	{ REAL64[x*n] }	e.g. mm	±∞	Read/write x columns starting from the m-th column: (x*n)-values (one or more lines)	Only possible on a column-by-column basis! (integer multiple)

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
						Value range m: [0 ... 16777216]	
0x05000000 +n-th start line	Read/ Write[<=16777216]	"Motion Function" (Motion laws) Data: STRUCT[x*m]	{			Read/write x lines starting from the n-th line: (x*m)-structures (one or more lines) Value range n: [0 ... 16777216]	Only possible line by line! (integer multiple)
			UINT32	1		Abs. point index (not evaluated)	
			UINT16	ENUM		Function type 1: Polynomial 1 15: Polynomial 5	
			UINT16	ENUM		Point type 0: default 1: ignore	
			INT32	1		Rel. address index at end point (Default: 1)	
			REAL64	mm		Master position	
			REAL64	mm		Slave position	
			REAL64	mm/s		Slave velocity	
			REAL64	mm/s^2		Slave acceleration	
			REAL64	mm/s^3		Slave jerk	
			}				
0x06000000 +m-th start column	Read/ Write[<=16777216]	"Motion Function" (Motion laws) Data: STRUCT[x*n]	{			Read/write x columns starting from the m-th column: (x*n)-structures (one or more lines) Value range m: [0 ... 16777216]	Only possible on a column-by-column basis! (integer multiple)
			UINT32	1		Abs. point index (not evaluated)	
			UINT16	ENUM		Function type 1: Polynomial 1 15: Polynomial 5	
			UINT16	ENUM		Point type 0: default 1: ignore	
			INT32	1		Rel. address index at end point (Default: 1)	
			REAL64	mm		Master position	
			REAL64	mm		Slave position	
			REAL64	mm/s		Slave velocity	
			REAL64	mm/s^2		Slave acceleration.	

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	mm/s^3		Slave jerk	
			}				

7.7.8.2 "Index offset" specification for table state (Index group 0xA100 + ID)

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
0x0000000A	Read	every	INT32	1	≥ 0	'User Counter' (Number of users of this table)	Cannot be traced by oscilloscope!

7.7.8.3 "Index offset" specification for table functions (Index group 0xA200 + ID)

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
0x00010000	Write	Cam plate	{			Generates a cam plate table with dimension (n*m):	Table types: 1,2,3,4 Dimension: at least 2x1
			UINT32	1	s. ENUM (>0)	Table type (see appendix)	
			UINT32	1	[2...16777216]	Number of lines	
			UINT32	1	[1...16777216]	Number of columns	
			}				
0x00010001	Write	Characteristic curve	{			Generates characteristic table with dimensions (n*m):	Table types: 1.3 Dimension: at least 2x1
			UINT32	1	s. ENUM (>0)	Table type (see appendix)	
			UINT32	1	[2...16777216]	Number of lines	
			UINT32	1	[1...16777216]	Number of columns	
			}				
0x00010010	Write	"Motion Function" (motion laws)	{			Generates "Motion Function" table with dimension (n*m):	Table types: 3.4 Dimension: at least 2x1
			UINT32	1	s. ENUM (>0)	Table type (see appendix)	
			UINT32	1	[2...16777216]	Number of lines	
			UINT32	1	[1...16777216]	Number of columns	
			}				
0x00020000	Write	every	VOID			Deletes table with dimension (n*m)	Table types: 1,2,3,4
0x00030000	Write	every	VOID			Initializes table Initialization is no longer needed, because now it happens	

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Note
						automatically in the following cases: a) when coupling by means of a table b) when reading out the slave position (see table para.)	

7.7.9 Appendix

Enum Channel types

Define	Channel types
1	Standard
2	Interpreter
3	FIFO
4	Kinematic transformation

Enum Interpreter types

Define	Interpreter types
0	NOT DEFINED
1	reserved
2	DIN 66025 (Siemens dialect)

Enum Interpreter Operation modes

Define	interpreter/channel operation mode
0x0	Default (deactivates the other modes)
0x1	Single block mode in the NC core (Block execution task/SAF)
0x1000	reserved
0x2000	reserved
0x4000	Single block mode in the interpreter

Enum Interpolation load log mode

Define	Load log mode
0	Loader log off
1	Source only
2	Source & Compiled

Enum Interpolation Trace mode

Define	Trace mode
0	Trace off
1	Trace line numbers
2	Trace Source

Enum Interpreter state

See Interpreter state [▶ 16]

Enum Group types

Define	Group types
0	NOT DEFINED
1	PTP-Group + x Slave
2	1D-Group + x Slave
3	2D-Group + x Slave
4	3D-Group + x Slave
5	High/low speed + x Slave
6	Low cost stepper motor (dig. IO) + x Slave
7	Table Group + x Slave
9	Encoder Group + x Slave
11	FIFO Group + x Slave
12	Kinematic Transformation Group + x Slave

Enum Curve velocity reduction method

See Curve velocity reduction method [▶ 23]

Enum Axis types

Define	Axis types
0	NOT DEFINED
1	Continuous axis (Servo)
2	Discrete axis (high/low speed)
3	Continuous axis (stepper motor)

Enum Stepper motor operation mode

Define	Stepper motor operation mode
0	NOT DEFINED
1	2-phase excitation (4 cycles)
2	1-2-phase excitation (6 cycles)
3	Power section

Enum Override types for PTP axes (velocity override)

Define	Override types
1	Reduced Old variant, replaced by "(3) Reduced (iterated)"
2	Original Old variant, replaced by "(4) Original (iterated)"
3	Reduced (iterated) Default value: the override value is related to the velocity which is internally reduced in a special case. This results in a directly proportional velocity (=> linear relationship) for the entire override range from 0 to 100%.
4	Original (iterated)

Define	Override types
	The override value is always referred to the velocity programmed by the user. If this velocity cannot be driven, however, then a maximum override value results from which no higher velocity can be reached (=> limitation).

Enum Group/axis start types

Define	Group/axis start types
0	NOT DEFINED
1	Absolute start
2	Relative start
3	Continuous start positive
4	Continuous start negative
5	Modulo start (OLD)
261	Modulo start on the shortest distance
517	Modulo start in positive direction (with modulo tolerance window)
773	Modulo start in negative direction (with modulo tolerance window)
4096	Stop and lock (axis locked for motion commands)
8192	Halt (without motion lock)

Enum Command buffer types (buffer mode) for universal axis start (UAS)

Define	Buffer mode
0	ABORTING (default) (instantaneous, aborts current movement and deletes any buffered commands)
1	BUFFERED (stored in command buffer to be executed after an active movement)
18	BLENDING LOW (buffered, no stop, runs through intermediate target position at the lowest velocity of two commands)
19	BLENDING PREVIOUS (buffered, no stop, runs through intermediate target position at the velocity of the active command)
20	BLENDING NEXT (buffered, no stop, runs through intermediate target position at the velocity of the buffered command)
21	BLENDING HIGH (buffered, no stop, runs through intermediate target position at the highest velocity of two commands)

Enum End position types (new end position)

Define	End position types
0	NOT DEFINED
1	Absolute position
2	Relative position
3	Continuous position positive
4	Continuous position negative
5	Modulo position

Enum Command types for new end position with new velocity (new end position and/or new velocity)

Define	Command types for new end position with new velocity
0	NOT DEFINED
1	Position (instantaneous)
2	Velocity (instantaneous)
3	Position and velocity (instantaneous)
9	Position (switching position)
10	Velocity (switching position)
11	Position and velocity (switching position)

Enum Actual position types (set actual position)

Define	Actual position types
0	NOT DEFINED
1	Absolute position
2	Relative position
5	Modulo position

Enum Compensation types (path compensation or superimposed)

Define	Compensation types
0	NOT DEFINED
1	VELOREDUCTION_ADDITIVEMOTION The max. velocity VelocityDiff is reduced. The path over which the compensation trip is effective consists of length + distance.
2	VELOREDUCTION_LIMITEDMOTION The max. velocity VelocityDiff is reduced. The path over which the compensation trip is effective is defined by the Length parameter.
3	LENGTHREDUCTION_ADDITIVEMOTION The max. available path is reduced and consists of length + distance. The system tries to use the max. velocity VelocityDiff.
4	LENGTHREDUCTION_LIMITEDMOTION The max. available path is reduced and is limited by the Length parameter. The system tries to use the max. velocity VelocityDiff.

Enum Slave types

Define	Slave types
0	NOT DEFINED
1	Linear
2	Flying saw (velocity, jerk restricted profile)
3	Flying saw (position and velocity, jerk restricted profile)
5	Synchronization generator (velocity, jerk restricted profile)
6	Synchronization generator (position and velocity, jerk restricted profile)
10	Tabular

Define	Slave types
11	Multi-tabular
13	'Motion Function' (MF)
15	Linear with cyclic gearing factor change (ramp filter for acceleration limits)
100	Specific

Enum Slave decoupling types (for subsequent axis command)

Define	Slave decoupling types (for subsequent axis command)
0	Stop, E-stop or P-stop (default) (STOP)
1	Oriented stop (O-stop) (ORIENTEDSTOP)
2	Reduce any acceleration to 0 (force-free) and continue to endless target position (ENDLESS)
3	Continue to endless target position at new requested velocity (ENDLESS_NEWVELO)
4	New end position (NEWPOS)
5	New end position and new requested velocity (NEWPOSANDVELO)
6	Logical decoupling and stopping of axis immediately without velocity ramp (INSTANTANEOUSSTOP)

Enum Controller types

Define	Controller types
0	NOT DEFINED
1	P-controller (standard) (Position)
2	PP-controller (with ka) (Position)
3	PID-controller (with ka) (Position)
5	P-controller (Velocity)
6	PI-controller (Velocity)
7	High/low speed controller (Position)
8	Stepper motor controller (Position)
9	SERCOS controller (Position in the drive)
10	RESERVED
11	RESERVED

Define	Controller types
12	RESERVED
13	RESERVED
14	TCom Controller (Soft Drive) (Position in the drive)

Enum Controller Observer mode

Define	Controller observer mode
0	No observer active (default)
1	"Luenberger" observer (classic observer design)

Requires control loop with torque interface

Enum Encoder types

Define	Encoder types
0	NOT DEFINED
1	Simulation Encoder (Incremental)
2	M3000 Encoder (Multi/Single-Turn) (Absolute)
3	M31x0 / M2000 Encoder (Incremental)
4	MDP 511 Encoder: EL7041, EL7342, EL5101, EL5151, EL2521, EL5021, IP5101 (Incremental)
5	MDP 500/501 Enc.: EL5001, IP5009, KL5001 (SSI) (Absolute)
6	MDP 510 Encoder: KL5051, KL2502-30K Encoder (BiSSI) (Incremental)
7	KL30xx Encoder (Analog) (Absolute)
8	SERCOS and EtherCAT SoE (Position) (Incremental)
9	SERCOS and EtherCAT SoE (Position and velocity) (Incremental)
10	Binary encoder (0/1) (Incremental)
11	M2510 Encoder (Absolute)
12	FOX50 Encoder (Absolute)
14	AX2000 (Lightbus) (Incremental)
15	Provi-Drive MC (Simodrive 611U) (Incremental)

Define	Encoder types
16	Universal encoder (variable bit mask) (Incremental)
17	NC back plane (Incremental)
18	Special CANopen type (e.g. Lenze Drive 9300) (Incremental)
19	MDP 513 (DS402): CANopen and EtherCAT CoE (AX2xx-B1x0/B510, EL7201) (Incremental)
20	AX2xx-B900 (Ethernet) (Incremental)
21	KL5151 Encoder (Incremental)
24	IP5209 Encoder (Incremental)
25	KL2531/KL2541 Encoder (Stepper Motor) (Incremental)
26	KL2532/KL2542 Encoder (DC motor), KL2535/ KL2545 (PWM current terminal) (Incremental)
27	Time base encoder (Time Base Generator) (Incremental)
28	TCom Encoder (Soft Drive) (Incremental)

Enum Encoder mode

Define	Encoder mode
0	NOT DEFINED
1	Determination of position
2	Determination of position and velocity
3	Determination of position, velocity and acceleration

Enum Encoder evaluation direction (log. counting direction)

Define	Encoder evaluation direction (log. counting direction)
0	Evaluation in positive and negative counting direction (default configuration, i.e. compatible with the previous state)
1	Evaluation only in positive counting direction
2	Evaluation only in negative counting direction
3	Evaluation neither in positive nor in negative counting direction (evaluation blocked)

Not for all encoder types; only for KL5101, KL5151, KL2531, KL2541, IP5209, Universal encoder, etc.

	Encoder types		
Encoder evaluation direction (Log. counting direction)	KL5101, ...	Universal Encoder	other types
0: positive and negative	√	√	—
1: only positive	√	√	—
2: only negative	√	√	—
3: disabled	√	√	—

Enum Encoder sign interpretation (data type)

Define	Sign interpretation (data type) of the encoder actual increments
0	NOT DEFINED (default configuration, i.e. compatible with the previous state)
1	UNSIGNED: unsigned interpretation of the encoder actual increments
2	SIGNED: signed interpretation of the encoder actual increments

For KL30xx/KL31xx only for the time being

Enum Encoder absolute dimensioning system

Define	Encoder absolute dimensioning system
0	INC: Incremental absolute dimension system with underflow and overflow offset (default, i.e. compatible with the previous state)
1	ABS: Absolute dimension system without underflow and overflow offset (no underflow or overflow of the encoder allowed)
2	ABS MODULO: Conditionally absolute dimension system, since it has underflow and overflow offset (absolute value that modulo (endless) continues)

Not for all encoder types; only for Profi Drive MC, M3000, KL5001/EL5001, IP5009, SERCOS, UNIVERSAL, etc.

Enum Encoder position initialization

Define	Encoder position initialization
0	Direct adoption of the position increments without further logic (default configuration, i.e. compatible with the previous state)
1	With underflow and overflow offset logic (direct adoption, or underflow or overflow offset)

For SERCOS only for the time being

Enum Reference mode for incremental encoder

Define	Reference mode for incremental encoder
0	NOT DEFINED (default configuration, i.e. compatible with the previous state)
1	Latch event: shutdown of the PLC cam (falling edge)
2	Latch event: Hardware sync pulse (zero track)
3	Latch event: External hardware latch with rising edge (measuring sensor or, respectively, measurement on the fly with rising edge)
4	Latch event: External hardware latch with falling edge (measuring sensor or, respectively, measurement on the fly with falling edge)
5	Latch event: Synthetically emulated software sync pulse (software zero track); REQUIREMENT: absolute per motor revolution, e.g. resolver!
6	Latch event: Hardware latch event defined in the drive with rising edge (e.g. for SoftDrive) (NEW)
7	Latch event: Hardware latch event defined in the drive with falling edge (e.g. for SoftDrive) (NEW)
20	Application defined Homing sequence (PLC code): Application defined Homing request is signaled to the PLC with the ApplicationRequest-Bit (NEW)

	Referencing mode: Latch event					
Encoder types	**0: not defined**	**1: PLC cam (falling edge)**	**2: Hardware Sync pulse (zero-/C-track)**	**3: External hardware Latch with rising edge**	**4: External hardware Latch with falling edge**	**5: Software Sync pulse (Software zero track)**
AX2xxx-B200 (Lightbus)	—	√	√	√	√	√ (only resolver)
AX2xxx-B510 (CANopen)	—	√	—	—	—	√ (only resolver) (see "Reference mask" parameter)
AX2xxx-B1x0 (EtherCAT)	—	√	√	√	√	√ (only resolver) (fixed 20-bit)
AX2xxx-B900 (Ethernet)	—	√	√	√	√	√ (only resolver)
Sercos	—	√	√	√ (AX5xxx specific implemented)	√	√ (see "Reference mask" parameter)
Profi Drive	—	√	√	√	√	√
KL5101 IP5109	—	√	√	√	√	√
KL5111	—	√	√	—	—	√
KL5151	—	√	√	√	√	√

	Referencing mode: Latch event					
Encoder types	**0: not defined**	**1: PLC cam (falling edge)**	**2: Hardware Sync pulse (zero-/C-track)**	**3: External hardware Latch with rising edge**	**4: External hardware Latch with falling edge**	**5: Software Sync pulse (Software zero track)**
						(not useful)
IP5209	—	√	√	—	—	√ (not useful)
CANopen (e.g. Lenze)	—	√	—	√ (Input In1)	√ (Input In2)	√ (only resolver) (fixed 16-bit)
others Types	—	—	—	—	—	—

Enum Drive types

Define	Drive types
0	NOT DEFINED
1	Analog Servo Drive: M2400 DAC 1 (Analog)
2	Analog Servo Drive: M2400 DAC 2 (Analog)
3	Analog Servo Drive: M2400 DAC 3 (Analog)
4	Analog Servo Drive: M2400 DAC 4 (Analog)
5	MDP 252 Drive: Analog Servo Drive: KL4xxx, KL2502-30K (Analog)
6	MDP 252 Drive: Analog Servo Drive (non-linear): KL4xxx, KL2502-30K (Analog)
7	High/low speed drive (Digital)
8	Stepper motor drive (Digital)
9	SERCOS-Drive (Digital)
10	MDP 510 Drive: KL5051 (BiSSI-Interface) (Digital)
11	AX2000 (Lightbus) (Digital)
12	Provi-Drive MC (Simodrive 611U) (Digital)
13	Universal Drive (Analog)
14	NC back plane (Analog)
15	Special CANopen type (e.g. Lenze Drive 9300)

Define	Drive types
	(Digital)
16	MDP 742 (DS402): CANopen and EtherCAT CoE (AX2xx-B1x0/B510) (Digital)
17	AX2xx-B900 Drive (Ethernet) (Digital)
20	KL2531/KL2541 Encoder (Stepper Motor) (Digital)
21	KL2532/KL2542 Encoder (DC motor), KL2535/ KL2545 Encoder (PWM current terminal) (Digital)
22	TCom Drive (Soft Drive) (Digital)
23	MDP 733 Drive: Profile MDP 733 (EL7332, EL7342, EP7342) (Digital)
24	MDP 703 Drive: Profile MDP 703 (EL7031, EL7041, EP7041) (Digital)

Enum Drive-Output-Start types

Define	Enum Drive-Output-Start types
0	NOT DEFINED
1	Output value in percent
2	Output as velocity, e.g. m/min

Enum Moving phases / Movement state for master axes

Define	Moving phases / Movement state (distinction between internal and external setpoint generation)
Internal setpoint generation	
0	Setpoint generator not active (INACTIVE)
1	Setpoint generator active (RUNNING)
2	Velocity override is zero (OVERRIDE_ZERO)
3	Constant velocity (PHASE_VELOCONST)
4	Acceleration phase (PHASE_ACCPOS)
5	Deceleration phase (PHASE_ACCNEG)
External setpoint generation:	
41	External setpoint generation active (EXTSETGEN_MODE1)
42	Internal and external setpoint generation active (EXTSETGEN_MODE2)

Enum Moving phases / Movement state for slave axes

Define	Moving phases / Movement state
0	Slave generator not active (INACTIVE)
11	Slave is in a movement pre-phase (PRE-PHASE)
12	Slave is synchronizing (SYNCHRONIZING)
13	Slave is synchronized and moves synchronously (SYNCHRON)

Only for slaves of the type synchronization generator for the time being

Enum Table main types

Define	Table main types
1	(n*m) Cam plate tables (Camming)
10	(n*m) Characteristic curves tables (Characteristics) (e.g. hydraulic valve characteristics) Only non-cyclic table sub-types (1, 3) are supported!
16	(n*m) "Motion Function" tables (MF) Only non-equidistant table sub-types (3, 4) are supported!

Enum Table sub-types

Define	Table sub types
1	(n*m) Table with equidistant master positions and no cyclic continuation of the master profile (equidistant linear)
2	(n*m) Table with equidistant master positions and cyclic continuation of the master profile (equidistant cyclic)
3	(n*m) Table with non-equidistant, but strictly monotonously increasing master positions and a non-cyclic continuation of the master profile (monotonously linear)
4	(n*m) Table with non-equidistant, but strictly monotonously increasing master positions and a cyclic continuation of the master profile (monotonously cyclic)

Enum Table interpolation types

Define	Table interpolation types between the reference points
0	Linear interpolation (NC_INTERPOLATIONTYPE_LINEAR) (Standard)
1	4-point interpolation (NC_INTERPOLATIONTYPE_4POINT) (for equidistant table types only)
2	Cubic spline interpolation of all reference points ("global spline") (NC_INTERPOLATIONTYPE_SPLINE
3	sliding cubic spline interpolation of n reference points ("local spline") (NC_INTERPOLATIONTYPE_SLIDINGSPLINE)(from TC V2.11 B1514)

Structure of the tables (CAM) coupling information

Tables		(CAM) coupling information
nTableID;	1.	cam table ID
nTableMainType;	2.	e.g. CAMMING, CHARACTERISTIC, MOTIONFUNCTION

Tables		(CAM) coupling information
nTableSubType;	3.	e.g. EQUIDIST_LINEAR, EQUIDIST_CYCLE, NONEQUIDIST_LINEAR, NONEQUIDIST_CYCLE
nInterpolationType;	4.	e.g. LINEAR, 4POINT, SPLINE
nNumberOfRows;	5.	number of rows/elements
nNumberOfColumns;	6.	number of columns
fRawMasterPeriod;	7.	master period/cycle (raw value, not scaled)
fRawSlaveStroke;	8.	slave difference per master period/ cycle (raw value, not scaled)
fMasterOffset;	9.	total master offset
fSlaveOffset;	10.	total slave offset
fMasterScaling;	11.	total master scaling
fSlaveScaling;	12.	total slave scaling
fActualMasterAxisPos;	13.	actual master axis setpos (absolute)
fActualSlaveAxisPos;	14.	actual slave axis setpos (absolute)
fActualMasterCamPos;	15.	actual master cam setpos
fActualSlaveCamPos;	16.	actual master cam setpos

Structure of the characteristic values

Characteristic values		
fMasterVeloNom;	1.	master nominal velocity (normed: => 1.0)
fMasterPosStart;	2.	master start position
fSlavePosStart;	3.	slave start position
fSlaveVeloStart;	4.	slave start velocity
fSlaveAccStart;	5.	slave start acceleration
fSlaveJerkStart;	6.	slave start jerk
fMasterPosEnd;	7.	master end position
fSlavePosEnd;	8.	slave end position
fSlaveVeloEnd;	9.	slave end velocity
fSlaveAccEnd;	10.	slave end acceleration
fSlaveJerkEnd;	11.	slave end jerk
fMPosAtSPosMin;	12.	master pos. at slave min. position
fSlavePosMin;	13.	slave minimum position
fMPosAtSVeloMin;	14.	master pos. at slave min. velocity
fSlaveVeloMin;	15.	slave minimum velocity
fMPosAtSAccMin;	16.	master pos. at slave min. acceleration
fSlaveAccMin;	17.	slave minimum acceleration
fSVeloAtSAccMin;	18.	slave velocity at slave min. acceleration
fSlaveJerkMin;	19.	slave minimum jerk
fSlaveDynMomMin;	20.	slave minimum dynamic momentum (NOT SUPPORTED YET!)
fMPosAtSPosMax;	21.	master pos. at slave max. position
fSlavePosMax;	22.	slave maximum position

Characteristic values		
fMPosAtSVeloMax;	23.	master pos. at slave max. velocity
fSlaveVeloMax;	24.	slave maximum velocity
fMPosAtSAccMax;	25.	master pos. at slave max. acceleration
fSlaveAccMax;	26.	slave maximum acceleration
fSVeloAtSAccMax;	27.	slave velocity at slave max. acceleration
fSlaveJerkMax;	28.	slave maximum jerk
fSlaveDynMomMax;	29.	slave maximum dynamic momentum (NOT SUPPORTED YET!)
fSlaveVeloMean;	30.	slave mean absolute velocity
fSlaveAccEff;	31.	slave effective acceleration

Enum Axis control loop switch types

Define	Axis control loop switch types
0	NOT DEFINED
1	Simple switching (similar to an axis reset) (STANDARD)
2	Switching/synchronization by means of I/D-part of the controller to an internal initial value (jerk-free/smooth)
3	Switching/synchronization by means of I/D-part of the controller to a parameterisable initial value

7.8 Overview of NC Errors

Error code (hex)	Description
0x4000 - 0x4FFF: NC error code range	
0x40nn	General errors [▶ 368]
0x41nn	Channel Errors [▶ 371]
0x42nn	Group Errors [▶ 374]
0x43nn	Axis Errors [▶ 396]
0x44nn	Encoder Errors [▶ 404]
0x45nn	Controller Errors [▶ 414]
0x46nn	Drive Errors [▶ 420]
0x4Ann	Table Errors [▶ 427]
0x4Bnn	NC PLC errors [▶ 429]
0x4Cnn	Kinematic Transformation [▶ 437]
0x8000 … 0x8FFF: New extended NC error code range	
0x81nn - 0x811F	Bode plot (diagnosis)
0x8120 - 0x8FFF	further errors [▶ 439]

7.8.1 General NC Errors

Error(Hex)	Error(Dec)	Error type	Description
4000	**16384**	Internal	**Internal error** Internal system error in the NC on ring 0, no further details.

Error(Hex)	Error(Dec)	Error type	Description
4001	**16385**	Memory	**Memory error** The ring-0 memory management is not providing the required memory. This is usually a result of another error, as a result of which the controller will halt normal operation (now if not before).
4002	**16386**	Internal	**NC retain data error (persistent data)** An error occurred when loading the NC Retain data, so that the affected axes are no longer referenced (status bit "Homed" is FALSE). This error can have the following reasons: - no NC Retain data were found - only old NC Retain data were found (old backup data set) - the NC Retain data are corrupt or inconsistent
4003	**16387**	Parameter	**Parameter for monitoring the NC setpoint output is invalid** The parameter for activating or deactivating the function "cyclic monitoring of NC setpoint output for steadiness and consistency" is invalid. (Special function)
4004	**16388**	Internal	**External error** This error code can be set by an external module (e.g. third-party module) or can be set if an external module has an error.
4010	**16400**	Parameter	**Channel identifier not allowed** Either an unacceptable value (not 1...255) has been used, or a channel that does not exist in the system has been named.
4011	**16401**	Parameter	**Group identifier not allowed** Either an unacceptable value (not 1...255) has been used, or a group that does not exist in the system has been named.
4012	**16402**	Parameter	**Axis identifier not allowed** Either an unacceptable value (not 1...255) has been used, or an axis that does not exist in the system has been named.
4013	**16403**	Parameter	**Encoder identifier not allowed** Either an unacceptable value (not 1...255) has been used, or a encoder that does not exist in the system has been named.
4014	**16404**	Parameter	**Controller identifier not allowed** Either an unacceptable value (not 1...255) has been used, or a controller that does not exist in the system has been named.
4015	**16405**	Parameter	**Drive identifier not allowed** Either an unacceptable value (not 1...255) has been used, or a drive that does not exist in the system has been named.
4016	**16406**	Parameter	**Table identifier not allowed** Either an unacceptable value (not 1...255) has been used, or a table that does not exist in the system has been named.
4020	**16416**	Internal	**Missing process image** There is no PLC-axis interface when creating an axis.

Error(Hex)	Error(Dec)	Error type	Description
4021	**16417**	Internal	**Missing process image** There is no axis-PLC interface when creating an axis.
4022	**16418**	Internal	**Missing process image** There is no encoder-I/O interface when creating an axis.
4023	**16419**	Internal	**Missing process image** There is no I/O-encoder interface when creating an axis.
4024	**16420**	Internal	**Missing process image** There is no drive-I/O interface when creating an axis.
4025	**16421**	Internal	**Missing process image** There is no I/O-drive interface when creating an axis.
4030	**16432**	Internal	**Coupling type not allowed** The master/slave coupling type is not allowed.
4031	**16433**	Internal	**Axis type not allowed** The type specification in the axis generation is inadmissible.
4032	**16434**	Parameter	**Unknown channel type** The NC channel type is unknown. Known types are e.g. an NCI channel, a FIFO channel, etc.
4040	**16448**	Internal	**Axis is incompatible** The axis is not suitable for the intended purpose. A rapid/slow traverse axis, for example, cannot function as a slave in an axis coupling.
4050	**16464**	Internal	**Channel not ready to operate** The channel is not complete, and is therefore not ready for operation. This is usually a consequence of problems at system start-up.
4051	**16465♠**	Internal	**Group not ready to operate** The group is not complete, and is therefore not ready for operation. This is usually a consequence of problems at system start-up.
4052	**16466**	Internal	**Axis not ready to operate** The axis is not complete, and is therefore not ready for operation. This is usually a consequence of problems at system start-up.
4060	**16480**	Internal	**Channel exists** The channel that is to be created already exists.
4061	**16481**	Internal	**Group exists** The group that is to be created already exists.
4062	**16482**	Internal	**Axis exists** The axis that is to be created already exists.
4063	**16483**	Internal	**Table exists** The table that is to be created already exists, resp. it is tried internally to use an already existing table id (e.g. for the universal flying saw).
4070	**16496**	Internal	**Axis index not allowed** The location within the channel specified for an axis is not allowed.
4071	**16497**	Internal	**Axis index not allowed**

Error(Hex)	Error(Dec)	Error type	Description
			The location within the group specified for an axis is not allowed.

7.8.2 Channel Errors

Error(Hex)	Error(Dec)	Error type	Description
4101	**16641**	Parameter	**Group index not allowed** The location within the channel specified for a group is not allowed.
4102	**16642**	Address	**Null pointer** The pointer to the group is invalid. This is usually a consequence of an error at system start-up.
4103	**16643**	Internal	**Missing process image** It is not possible to exchange data with the PLC. Possible causes: 1. The channel has no interface (no interpreter available). 2. The connection to the PLC is faulty.
4104	**16644**	Parameter	**M-function index not allowed** Unacceptable M-function (not 0...159) detected at the execution level.
4105	**16645**	Memory	**No memory** No more system memory is available. This is usually the result of another error.
4106	**16646**	Function	**Not ready** The function is not presently available, because a similar function is already being processed. Usually this is a result of access conflicts: more than one instance wants to issue commands to the channel. This can, for example, be the consequence of an incorrect PLC program.
4107	**16647**	Function	**Function/command not supported** A requested function or command is not supported by the channel.
4108	**16648**	Parameter	**Invalid parameter while starting** Parameters to start the channel (TwinCAT-Start) are invalid. Typically there is an invalid memory size or channel type requested.
4109	**16649**	Function	**Channel function/command is not executable** A channel function e.g. interpreter start is not executable because the channel is already busy, no program is loaded or in an error state.
410A	**16650**	Function	**ItpGoAhead not executable** The requested ItpGoAhead command is not executable, because the interpreter is not executing a decoder stop.
4110	**16656**	Parameter	**Error opening a file** The specified file does not exist. Example: NC program unknown.
4111	**16657**	NC programming	**Syntax error during loading** The NC has found a syntax error when loading an NC program.

Error(Hex)	Error(Dec)	Error type	Description
4112	**16658**	NC programming	**Syntax error during interpretation** The NC has found a syntax error when executing an NC program.
4113	**16659**	NC programming	**Missing subroutine** The NC has found a missing subroutine while loading.
4114	**16660**	Memory	**Loading buffer of interpreter is too small** The capacity of the interpreter loading buffer has been exceeded.
4115	**16661**	Internal	**Symbolism** Reserved, currently not used.
4116	**16662**	Internal	**Symbolism** Reserved, currently not used.
4117	**16663**	NC programming	**Subroutine incomplete** The header of the subroutine is missing.
4118	**16664**	NC programming	**Error while loading the NC program** The maximum number of loadable NC programs has been reached. Possible cause: Too many subprograms have been loaded from a main program.
4119	**16665**	NC programming	**Error while loading the NC program** The program name is too long.
4120	**16672**	NC programming	**Divide by Zero** The NC encountered a computation error during execution: division by 0.
4121	**16673**	NC programming	**Invalid circle parameterization** The NC has detected a calculation error during processing: the specified circle is not calculable.
4122	**16674**	NC programming	**Invalid FPU-Operation** The NC encountered an invalid FPU-Operation during execution. This error occurs e.g. by calculating the square root of a negative number
4130	**16688**	NC programming	**Stack overflow: Subroutines** The NC has detected a stack overflow while processing: Too many subroutine levels.
4131	**16689**	NC programming	**Stack underflow: Subroutines** The NC has detected a stack underflow while processing: Too many subroutine return commands. **A main program must not be terminated with a return command.**
4132	**16690**	NC programming	**Stack overflow: Arithmetic unit** The NC has detected a stack overflow during processing: the calculation is too complex or is not written correctly.
4133	**16691**	NC programming	**Stack underflow: Arithmetic unit** The NC has detected a stack underflow during processing: The calculation is too complex or is not written correctly.
4140	**16704**	Parameter	**Register index not allowed**

Error(Hex)	Error(Dec)	Error type	Description
			The NC has detected an invalid register index during processing: The program contains an invalid specification (not R0...R999) or a pointer register contains an invalid value.
4141	**16705**	NC programming	**G-function index not allowed** The NC has encountered an unacceptable G-function (not 0...159) during execution.
4142	**16706**	NC programming	**M-function index not allowed** The NC has encountered an unacceptable M-function (not 0...159) during execution.
4143	**16707**	NC programming	**Extended address specification not allowed** The NC has encountered an unacceptable extended address (not 1...9) during execution.
4144	**16708**	NC programming	**Index to the internal H-function not allowed** The NC has encountered an unacceptable internal H-function in the course of processing. This is usually a consequence of an error during loading.
4145	**16709**	Parameter	**Machine data value not allowed** While processing instructions the NC has detected an impermissible value for the machine data (MDB) (not 0…7).
4150	**16720**	Parameter	**Tool compensation parameters cannot be changed here** The NC has encountered an unacceptable change of parameters for the tool compensation during execution. This can be, for example, a changed tool radius while a circle was programmed.
4151	**16721**	Parameter	**Cannot calculate tool compensation** The NC has encountered an error by the calculation of the tool compensation.
4152	**16722**	NC programming	**Tool compensation** The plane for the tool compensation cannot be changed here. This error occurred for instance by changing the plane when the tool radius compensation is turned on or active.
4153	**16723**	NC programming	**Tool compensation** The D-Word is missing or invalid by turning on the tool compensation.
4154	**16724**	NC programming	**Tool compensation** The specified tool radius is invalid because the value is less or equal zero.
4155	**16725**	NC programming	**Tool compensation** The tool radius cannot be changed here
4156	**16726**	Internal	**Tool compensation** The Collision Detection Table is full.
4157	**16727**	Internal	**Tool compensation** An internal error has occurred when switching on the bottleneck detection
4158	**16728**	Internal	**Tool compensation** An internal error occurred in the bottleneck detection: Update Reversed Geo failed.

Error(Hex)	Error(Dec)	Error type	Description
4159	**16729**	NC programming	**Tool compensation** An unexpected combination of geometry types with bottleneck detection turned on was detected.
415A	**16730**	NC programming	**Tool compensation** The programmed inner circle is smaller than the cutter radius.
415B	**16731**	NC programming	**Tool compensation** The bottleneck detection recognized a contour violation.
415C	**16732**	Memory	**No memory** The table for corrected entries is full.
415D	**16733**	Memory	**No memory** The input table for tangential following is full.
415E	**16734**	Memory	**No memory** The executing table for tangential following is full.
415F	**16735**	Internal	**Geometry calculation** The geometric entry for the tangential following cannot be calculated.
4160	**16736**	Internal	**Reserved** Reserved, currently not used.
4161	**16737**	Internal	**Reserved** Reserved, currently not used.
4162	**16738**	Parameter	**Interpolation rules can not be determined** The actual active interpolation rules (G-Code), zero offset shifts or rotation cannot be detected.
4170	**16752**	NC programming	**Error loading: invalid parameter** The NC has found an invalid parameter while loading an NC program.
4171	**16753**	Internal	**Invalid contour start position** The NC has detected a calculation error during processing: The specified contour cannot be calculated because the start position is not on the contour.
4172	**16754**	Internal	**Reverse: invalid table index** The NC encountered an invalid internal entry index during execution of the retrace function.
4173	**16755**	NC programming	**Invalid G code** Invalid default G code. Wrong expression/syntax in the default G code.
4174	**16756**	NC programming	**Error opening the G code file** Error opening the default G code file.

7.8.3 Group Errors

Error(Hex)	Error(Dec)	Error type	Description	
4200	**16896**	Parameter	**Group ID not allowed** The value for the group ID is not allowed, e.g. because it has already been assigned, is less than or equal to zero, or is greater than 255.	
			Value range: [1 ... 255]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description	
4201	**16897**	Parameter	**Group type not allowed** The value for the group type is unacceptable because it is not defined. Type 1: PTP group with slaves (servo) Type 4: DxD group with slaves (3D group) Type 5: fast/creep group Type 6: stepper motor group Type 9: encoder group with slaves (servo) …	
			Value range: [1 ... 12]	Unit: 1
4202	**16898**	Initialization	**Master axis index not allowed** The value for the master axis index in an interpolating 3D group is not allowed, because, for instance, it has gone outside the value range. Index 0: x-axis (first master axis) Index 1: Y axis (second master axis) Index 2: Z axis (third master axis)	
			Value range: [0, 1, 2]	Unit: 1
4203	**16899**	Initialization	**Slave axis index not allowed (internal error)** The value for the slave axis index in a group is not allowed, because, for instance, it has passed outside the value range, the slave location to be used when inserting a new slave connection is already occupied, or because no slave is present when such a connection is being removed. Index 0: first slave axis Index 1: second slave axis Index 2: etc.	
			Value range: [0 ... 7]	Unit: 1
4204	**16900**	Initialization	**Internal error** An unexpected internal error has occurred. The following situations could be the cause: Not enough TC router memory or Windows memory available to build the internal NC objects, internal NC structures and links (pointers between the NC objects) are faulty or missing, a fatal internal error has occurred in the calculation for a stop command, internal checks of the NC own logic and algorithms (self-monitoring software), unexpected modes and cases that are not regularly foreseen but are detected as erroneous. **Often in such an error situation an additional error message is output in the logger (Windows Event Viewer), which can be helpful for more detailed analysis by Beckhoff or the user.**	
4205	**16901**	Parameter	**Cycle time for set execution task (SEC) not allowed** The value of the cycle time for the NC set execution task (SEC 1/2) is not allowed, because it has passed outside the value range.	
			Value range: [0.001 ... 0.1]	Unit: s
4206	**16902**	Initialization	"GROUPERR_RANGE_MAXELEMENTSINAXIS "	
4207	**16903**	Parameter	**Cycle time for the set preparation task (SPP) not allowed**	

Error(Hex)	Error(Dec)	Error type	Description
			The value of the cycle time for the NC set preparation task (SPP 1/2) is not allowed, because it has passed outside the value range. Value range: [0.001 ... 1.0] — Unit: s
4208	**16904**	Parameter	**Single step mode not allowed** The flag for the activation or deactivation of single step mode is not allowed. Value 0: passive (buffer mode) Value 1: active (single step mode) Value range: [0, 1] — Unit: 1
4209	**16905**	Parameter	**Group (de)activation invalid (internal error)** The flag for (de)activating the full group is invalid. Value 0: group active Value 1: group passive Value range: [0, 1] — Unit: 1
420A	**16906**	Initialization	**Set execution state (SEC state) not allowed (internal error)** The value for the state of the set execution state machine (SEC state) is not allowed. This error occurs on passing outside the range of values, or if the state machine enters an error state. Value range: [0 ... 5] — Unit: 1
420B	**16907**	Address	**Channel address** The group does not have a channel, or the channel address has not been initialized.
420C	**16908**	Address	**Axis address (master axis)** The group does not have a master axis (or axes) or the axis address(es) has (have) not been initialized.
420D	**16909**	Address	**Master axis address** A master/slave coupling is to be inserted into the group, but there is no valid address for the leading master axis.
420E	**16910**	Address	**Slave axis address** A master/slave coupling is to be inserted into the group, but there is no valid address for the slave axis.
420F	**16911**	Address	**Slave setpoint generator address** A master/slave coupling is to be inserted into the group, but there is no valid address for the slave setpoint generator.
4210	**16912**	Address	**Encoder address** An axis in the group does not have an encoder, or the encoder address has not been initialized.
4211	**16913**	Address	**Controller address** An axis in the group does not have a controller, or the controller address has not been initialized.
4212	**16914**	Address	**Drive address** An axis in the group does not have a drive, or the drive address has not been initialized.
4213	**16915**	Address	**Master setpoint generator address** A group (e.g. FIFO group) does not have a master setpoint generator or the setpoint generator address has not been initialized. There may not be enough memory available.
4214	**16916**	Address	**Axis interface NC to PLC address**

Error(Hex)	Error(Dec)	Error type	Description
			Group/axis does not have an axis interface from the NC to the PLC, or the axis interface address has not been initialized.
4215	**16917**	Address	**Slave axis address** An existing master/slave coupling is to be removed from the group, but there is no valid address for the slave axis.
4216	**16918**	Address	**Table unknown** The table, respectively the table ID, is unknown. This table is used for the master/slave coupling or for the characteristic curve.
4217	**16919**	Address	**NcControl address** The NcControl address has not been initialized.
4218	**16920**	Initialization	**Command lock because persistent NC data is waiting to be transferred** Axis is blocked for commands while waiting for valid I/O data to accept the queued persistent NC data.
4219	**16921**	Function	**The scaling mode MASTER-AUTOOFFSET is invalid because no reference table was found** The scaling mode MASTER-AUTOOFFSET used is invalid in this context because no reference to an existing reference table can be established. This error can occur, for example, when tables are added if no unique reference to an existing reference table can be established (e.g. because the reference is not unique, etc.).
421A	**16922**	Parameter	**The master axis start position does not permit synchronization** When a slave axis is being coupled on, the position of the master axis does not permit synchronization at the given synchronization positions.
421B	**16923**	Parameter	**Slave coupling factor (gear ratio) of 0.0 is not allowed** A master/slave coupling with a gear ratio of 0.0 is being created. This value is not allowed, since it does not correspond to any possible coupling, and division will generate an FPU exception.
421C	**16924**	Function	**Insertion of master axis into group not allowed** A master axis is to be inserted into a group at a location that is already occupied by another master axis. Maybe the reconfiguration cannot be done, because this axis has got an existing slave coupling. This master/slave coupling must be revoked before.
421D	**16925**	Function	**Deletion of master axis from group not allowed (internal error)** A master axis is to be removed from a location in a group that is not in fact occupied by master axis.
421E	**16926**	Function	**Function/feature is not supported from the setpoint generator** A function or feature is not supported from the setpoint generation (e.g. PTP master setpoint generator). This can be in general or only in a special situation.
421F	**16927**	Initialization	**Group initialization** The group has not been initialized. Although the group has been created, the rest of the initialization has not been performed (1. initialization group I/O, 2. initialization group, 3. reset group).

Error(Hex)	Error(Dec)	Error type	Description
4220	**16928**	Monitoring	**Group not ready / group not ready for new task** The group is being given a new task while it is still in the process of executing an existing task. This request is not allowed because it would interrupt the execution of the previous task. The new task could, for instance, be a positioning command, or the "set actual position" function. Precisely the converse relationships apply for the "set new end position" function. In that case, the group/axis must still be actively moving in order to be able to cause a change in the end position.
4221	**16929**	Monitoring	**Requested target velocity is not allowed** The value requested for the target velocity of a positioning task is less than or equal to zero, larger than the "maximum velocity" (see axis parameters), or, in the case of servo-drives, is larger than the "reference velocity" of the axis (see drive parameters).
4222	**16930**	Monitoring	**Requested target position is not allowed (master axis)** The requested value for the target position of a positioning task is not within the software end positions. In other words, it is either less than the minimum software end position or larger than the maximum software end position. This check is only carried out if the relevant end position monitoring is active.
4223	**16931**	Monitoring	**No enable for controller and/or feed (master axis)** The axis enables for the master axis needed for positioning are not present. This can involve the controller enable and/or the relevant, direction-dependent feed enable (see axis interface PlcToNc).
4224	**16932**	Monitoring	**Travel path smaller than one encoder increment (internal error)** The travel path that a group/axis is supposed to move is smaller than the physical significance of one encoder increment. In other words the movement is smaller than the scaling factor of the axis. The reaction to this is that the axis is reported as having logically finished without having actively moved. This means that an external error is not generated for the user. This error is also issued for rapid/slow traverse axes if a looping distance with nonzero parameters is smaller than the sum of the creeping and braking distances. In such a case it is not meaningful to either exceed or to fail to reach the target position.
4225	**16933**	Monitoring	**Drive hardware not ready to operate at axis start** During an axis start it is ascertained that the drive hardware is not ready to operate. This can be caused by the following reasons: - the drive is in error state (hardware error) - the drive is in the start-up phase (e.g. after an axis reset preceded by a hardware error) - the drive lacks the controller enable (ENABLE) **The time required for the "start-up" of a drive after a hardware error can be in the range of several seconds.**
4226	**16934**	Monitoring	**The parameters of the emergency stop are invalid.** Either, both, the deceleration and the jerk are less than zero or one of the parameters is weaker than the corresponding parameter of the start data.
4227	**16935**	Function	**Setpoint generator not active**

Error(Hex)	Error(Dec)	Error type	Description
			The setpoint generator is inactive such that no instructions are accepted.
4228	**16936**	Monitoring	**Requested travel path/looping distance is not allowed** The requested travel path or looping distance is smaller than the braking distance of the rapid/slow traverse axis.
4229	**16937**	Monitoring	**Requested target position is not allowed (slave axis)** The value for the target position of a positioning task when calculated for the slave axis is not within the software end positions. In other words, it is either less than the minimum software end position or larger than the maximum software end position. This check is only carried out if the relevant end position monitoring is active.
422A	**16938**	Monitoring	**No enable for controller and/or feed (slave axis)** The axis enables for one or more coupled slave axes needed for positioning are not present. This can involve the controller enable and/or the relevant, direction-dependent feed enable (see axis interface PlcToNc).
422B	**16939**	Parameter	**The activation position (position threshold) is out of range of the actual positioning** The activation position (position threshold) of a new axis command (e.g. "new velocity activated at a position") is out of range. E.g. the activation position is before the actual position or behind the target position.
422C	**16940**	Parameter	**The start or activation data of the external setpoint generation are not valid** This may be caused through: 1. The external setpoint generation is active and a new activation with a start type (1: absolute, 2: relative) unequal to the current one is send. 2. The internal setpoint generation is active (e.g. PTP) and the external one is activated with the start type absolute (2 setpoint generators of the type absolute are not possible).
422D	**16941**	Parameter	**Velocity is not constant** For changing the dynamic parameter 'acceleration' und 'deceleration' the axis has to be in dynamic state without acceleration and deceleration (that means constant velocity).
422E	**16942**	Parameter	**Jerk less than or equal to 0.0 is not allowed** A value less than or equal to 0.0 for the jerk (PTP and CNC) is not allowed, since the jerk is by definition positive, and with a jerk of 0.0, division will generate an FPU exception.
422F	**16943**	Parameter	**Acceleration less than or equal to 0.0 is not allowed** A value less than or equal to 0.0 for the acceleration (PTP and CNC) is not allowed, since the acceleration is positive by definition, and an acceleration of 0.0 will not allow a motion to be generated.
4230	**16944**	Parameter	**Absolute deceleration value less than or equal to 0.0 is not allowed** A value less than or equal to 0.0 for the absolute value of the deceleration (PTP and CNC) is not allowed, since the absolute value of the deceleration is positive by definition, and an absolute value of the deceleration of 0.0 will not allow a motion to be generated
4231	**16945**	Parameter	**Target velocity less than or equal to 0.0 is not allowed**

Error(Hex)	Error(Dec)	Error type	Description
			A value less than or equal to 0.0 or outside the range from 10^{-3} up to 10^{+10} for the target velocity (PTP and CNC) is not allowed, since the target velocity is by definition strictly positive, and with a target velocity of 0.0, division will generate an FPU exception.
4232	**16946**	Monitoring	**Loss of resolution accuracy for requested positioning** The positioning is so long in space or time that decimal places become irrelevant and inaccuracies may occur during positioning (LOSS_OF_PRECISION).
4233	**16947**	Parameter	**Cycle time less than or equal to 0.0 is not allowed** A value less than or equal to 0.0 for the cycle time (PTP and CNC) is not allowed, since the cycle time is by definition strictly positive, and with a cycle time of 0.0, division will generate an FPU exception.
4234	**16948**	Internal	**PTP data type <intasdouble> range exceeded** Such extreme parameters have been supplied for the start task, the override or the new target position that the internal data type loses its precision.
4235	**16949**	Function	**PTP LHL velocity profile cannot be generated (internal error)** Such extreme parameters have been supplied for the start task, the override or the new target position that it is not possible to generate a velocity profile of the type LHL (Low-High-Low).
4236	**16950**	Function	**PTP HML velocity profile cannot be generated (internal error)** Such extreme parameters have been supplied for the override or the new target position that it is not possible to generate a velocity profile of the type HML (High-Middle-Low).
4237	**16951**	Address	**Start data address is invalid** The address of the start data is invalid.
4238	**16952**	Parameter	**Velocity override (start override) is not allowed** The value for the velocity override is not allowed, because it is less than 0.0% or more than 100.0% (see axis interface PlcToNc). Here, 100.0 % corresponds to the integral value 1000000 in the axis interface. Value range: [0 ... 1000000]
4239	**16953**	Parameter	**Start type not allowed** The start type supplied does not exist.
423A	**16954**	Monitoring	**Velocity overshoot** The new dynamics with the parameterized jerk is so weak that a velocity overshoot is imminent under given boundary conditions. The command is therefore not supported.
423B	**16955**	Parameter	**Start parameter for the axis structure is invalid** External or internal parameters for the start structure for a positioning task are invalid. Thus, for instance, the scaling factor, the SEC cycle time or the requested velocity may be less than or equal to zero, which is not allowed.
423C	**16956**	Parameter	**Override generator initialization parameter invalid** One of the override generator (re)initialization parameters is invalid
423D	**16957**	Monitoring	**Slave axis has not setpoint generator (internal error)**

Error(Hex)	Error(Dec)	Error type	Description
			It is found that a slave axis within a group does not have a valid slave generator (setpoint generator). A slave axis and a slave setpoint generator must always be present as a pair. This is an internal error.
423E	**16958**	Function	**Table is empty** Either the SPP table or the SEC table does not contain any entries.
423F	**16959**	Function	**Table is full** The SPP table or the SEC table has no more free lines.
4240	**16960**	Memory	**No memory available** The SPP memory allocation for the dynamic entry in SEC table has failed.
4241	**16961**	Function	**Table already contains an entry (internal error)** The SEC table entry was canceled because an entry already exists by mistake.
4242	**16962**	Function	**Stop is already active** The stop instruction is not forwarded, because it has already been activated.
4243	**16963**	Function	**Compensation has not been carried out over the full compensation section** The compensations start parameters do not permit compensation over the full section to be compensated. For this reason the compensation will be carried out over a smaller section.
4244	**16964**	Parameter	**Internal parameters for the compensation are invalid (internal error)** Invalid internal parameters or start parameters of the lower-level generator.
4245	**16965**	Function	**Compensation active** The start of the compensation was refused, because the compensation is already active or the master/slave axis is not moved actively at all, which makes an execution of the compensation impossible.
4246	**16966**	Function	**Compensation not active** The stop of the compensation was denied because the compensation is not active.
4247	**16967**	Function	**Compensation type invalid** The type supplied for the section compensation is invalid. At the present time only compensation type 1 (trapezoidal velocity profile) is allowed.
4248	**16968**	Function	**Axis address for compensation invalid (internal error)** The address of the master of slave axis on which the section compensation is to act is invalid. This is an internal error.
4249	**16969**	Address	**Invalid slave address (internal error)** The specified slave address for online coupling/uncoupling is invalid.
424A	**16970**	Function	**Coupling velocities not allowed** The velocity of what is to become the master axis is 0, which means that on-line coupling is not possible.
424B	**16971**	Function	**Coupling velocities not constant**

Error(Hex)	Error(Dec)	Error type	Description
			The velocity of what is to become the master axis and the velocity of what is to become the slave axis are not constant, so that on-line coupling is not possible.
424C	**16972**	Parameter	**Cycle time less than or equal to 0.0 is not allowed** A value less than or equal to 0.0 for the cycle time (Slave) is not allowed, since the cycle time is by definition strictly positive, and with a cycle time of 0.0, division will generate an FPU exception.
424D	**16973**	Function	**Decoupling task not allowed** The slave axis is of such a type (e.g. a table slave) or is in such a state (master velocity 0) that on-line decoupling is not possible.
424E	**16974**	Function	**Function not allowed** The function cannot logically be executed, e.g. some commands are not possible and not allowed for slave axes.
424F	**16975**	Parameter	**No valid table weighting has been set** The weighting factor of each table is 0, so that no table can be read.
4250	**16976**	Function	**Axis start type, actual position type or end position type not allowed** The start type for a positioning task in invalid. Valid start types are: ABSOLUTE (1), RELATIVE (2), ENDLESS POSITIVE (3), ENDLESS NEGATIVE (4), MODULO (5), , etc. Furthermore, it is possible that the types for setting a new actual position or moving to a new end position are invalid.
4251	**16977**	Function	**Function is not supported** An NC function has been activated that is currently not released for use, or which is not even implemented. This can be a command which is not possible or not allowed for master axes.
4252	**16978**	Monitoring	**State of state machine invalid (internal error)** The state for one of the internal state machines is invalid. This is an internal error.
4253	**16979**	Monitoring	**PLC reference cam became free too soon** During the referencing process for an axis it is moved in the direction of the PLC referencing cam, and is only stopped again when the cam signal is reached. After the axis has then also physically stopped, the referencing cam must remain occupied until the axis subsequently starts back down from the cam in the normal way.
4254	**16980**	Monitoring	**Distance monitoring between activation of the hardware latch and appearance of the I/O sync pulse** When the distance monitoring is active, a check is kept on whether the number of increments between activation of the hardware latch and occurrence of the sync pulse (zero pulse) has become smaller than a pre-set value. If this case has occurred, this error will be generated. (see parameters for the incremental encoder)
4255	**16981**	Memory	**No memory available**

Error(Hex)	Error(Dec)	Error type	Description
			The dynamic memory allocation for the setpoint generator, the SPP table or the SEC table has failed.
4256	**16982**	Monitoring	**The table slave axis has no active table** Although the table slave axis has tables, none of the tables is designated as active. If this occurs during the runtime the whole master/slave group is stopped by a runtime error.
4257	**16983**	Function	**Function not allowed** The requested function or the requested task is not logically allowed. An example for such an error message would be "set an actual position" for an absolute encoder (M3000, KL5001, etc.).
4258	**16984**	Function	**Stopping compensation not allowed** It is not possible to stop the compensation, since compensation is already in the stopping phase.
4259	**16985**	Function	**Slave table is being used** The slave table cannot be activated, because it is currently being used.
425A	**16986**	Function	**Master or slave axis is processing a job (e.g. positioning command) while coupling is requested** A master/slave coupling of a certain slave type (e.g. linear coupling) cannot be executed, because either the master axis or the future slave axis is not at standstill, but executes a job (e.g. a positioning) at the coupling time. For this couple type this is not allowed.
425B	**16987**	Parameter	**Slave (start) parameter is incorrect** One of the slave start/coupling parameters is not allowed (Coupling factor is zero, the master position scaling of a cam plate is zero, etc.).
425C	**16988**	Parameter	**Slave type is incorrect** The slave type does not match up to the (SPP) start type.
425D	**16989**	Function	**Axis stop is already active** The axis stop/Estop is not initiated, because the stop/estop is already active.
425E	**16990**	Function	**Maximum number of tables per slave generator reached** The maximum number of tables per slave generator is reached (e.g. "MC_MultiCamIn" is limited to 4 tables).
425F	**16991**	Function	**The scaling mode is not allowed** The used scaling is invalid in this context. Either the mode is not defined or not yet implemented or it cannot be implemented in this constellation. For example, the MASTER-AUTOOFFSET mode cannot be used if relative coupling is performed, since there is a contradiction here. Furthermore, the MASTER-AUTOOFFSET mode cannot be used when coupling for the first time, since no reference to an existing reference table coupling (reference table) can be established here.
4260	**16992**	Monitoring	**Controller enable** The controller enable for an axis or for a coupled slave axis is not present (see axis interface PlcToNc). This error occurs if the controller enable is withdrawn while an axis or a group of axes (also a master/slave group) is being actively positioned. The error also occurs if a PTP axis or a coupled slave axis is started without controller enable.

Error(Hex)	Error(Dec)	Error type	Description
4261	**16993**	Function	**Table not found** No table exists with the ID prescribed or the table ID is not unique.
4262	**16994**	Function	**Incorrect table type** The table referred to in the function is of the incorrect type.
4263	**16995**	Function	**Single step mode** This error occurs if single step mode is selected for a group or axis and a new task is requested while one of the individual tasks is still being processed.
4264	**16996**	Function	**Group task unknown (asynchronous table entry)** The group has received a task whose type or sub-type is unknown. Valid tasks can be single or multi-dimensional positioning tasks (Geo 1D, Geo 3D), referencing tasks, etc.
4265	**16997**	Function	**Group function unknown (synchronous function)** The group has received a function whose type is unknown. Valid functions are "Reset", "Stop", "New end position", "Start/stop section compensation", "Set actual position", "Set/reset calibration state" etc.
4266	**16998**	Function	**Group task for slave not allowed** Group tasks are usually only possible for master axes, not for slave axes. A slave axis only moves as an indirect result of a positioning task given to its associated master axis. A slave cannot therefore receive an order directly. Exception: see axis parameter "Allow motion commands for slave axes"
4267	**16999**	Function	**Group function for slave not allowed** Group functions are in principle only possible for master axes, not for slave axes. The only exception is represented by the "Start/stop section compensation" function, which is possible both for masters and for slaves. A slave cannot directly execute any other functions beyond this.
4268	**17000**	Function	**NCI setpoint generator is inactive** An NCI command such as "StopAndKeep" is sent to a logically inactive DXD group or to a group with the channel override state zero. However, it is expected that the NCI group is actively in setpoint generation for the implementation of this command. This error can occur in connection with the functions "delete distance to go" and "measurement event (latch actual position)".
4269	**17001**	Parameter	**Start position = target position** Invalid position parameters.
426A	**17002**	Parameter	**Parameters of the delay-generator are invalid** Invalid external/internal parameters of the delay generator (delay time, cycle time, tics).
426B	**17003**	Parameter	**External parameters of the compensation are invalid** Invalid external parameters of the superimposed functionality (acceleration, deceleration, velocity, process velocity, length).
426C	**17004**	Parameter	**Invalid override type** The selected override type is invalid.
426D	**17005**	Function	**Activation position under/overrun** The requested activation position is located in the past of the master (e.g. when exchanging a cam plate).

Error(Hex)	Error(Dec)	Error type	Description
426E	**17006**	Function	**Activation impossible: Master at standstill** The required activation of the correction is impossible since the master axis is not moving. An accurate synchronization is not possible, because the master axis is at standstill and the slave axis is not yet synchronized.
426F	**17007**	Function	**Activation mode not possible** The requested activation mode is not possible when the slave axis is moving. Otherwise the slave velocity would jump to zero.
4270	**17008**	Parameter	**Start parameter of the compensation invalid** One of the dynamic parameters of the compensation is invalid (necessary condition): acceleration (>0) deceleration (>0) process velocity (>0)
4271	**17009**	Parameter	**Start parameter of the compensation invalid** Velocity overshoot is negative.
4272	**17010**	Parameter	**Start parameter of the compensation invalid** The section on which the compensation is to occur is not positive.
4273	**17011**	Monitoring	**Target position under/overrun" (internal error)** The position (calculated from the modulo target position) where the axis should stand at end of oriented stop has been run over.
4274	**17012**	Monitoring	**Target position will be under/overrun (internal error)** The position designated for the end of the oriented stop (calculated from the modulo target position) is too close and will be overrun.
4275	**17013**	Parameter	**Group parameter is invalid** A group parameter is invalid. This can be, for example, a parameterized velocity, acceleration, deceleration, jerk or `NC` cycle time whose value has been parameterized to be less than or equal to zero.
4276	**17014**	Monitoring	**Group error at the start of the setpoint generation** When starting the setpoint generation for e.g. the flying saw, different parameters or states can lead to this error. For example, dynamic parameters such as acceleration, deceleration and jerk may be invalid (less than or equal to zero) or the NC cycle time or the override value at startup may be outside the range from `0%` to `100%`.
4277	**17015**	Monitoring	**Dynamic parameters not permitted (internal error)** The dynamic parameters resulting from internal calculation like acceleration, deceleration and jerk are not permitted.
4279	**17017**	Monitoring	**New target position is invalid or cannot be reached** A newly commanded target position is invalid, since it has either already been passed over or is passed over during a stop with the momentarily acting dynamics.
427A	**17018**	Monitoring	**New travel velocity or final velocity is invalid** For a newly commanded command, either the required travel velocity or the required final velocity (target velocity in the target position) is invalid. The travel velocity must always be greater than zero and the final velocity must always be greater than or equal to zero (default case is zero).

Error(Hex)	Error(Dec)	Error type	Description
427B	**17019**	Monitoring	**New final velocity or new target position is invalid** For a newly commanded command, either the requested final velocity (target velocity in the target position) or the requested target position is invalid. The final velocity must always be greater than or equal to zero (default case is zero).
427C	**17020**	Monitoring	**New travel velocity is invalid** A newly commanded travel velocity is invalid because it is either less than or equal to zero or other reasons do not allow this velocity.
427D	**17021**	Monitoring	**Internal start mode is invalid** For a newly commanded command the start mode is invalid or is not allowed in this travel situation. A user cannot directly influence the start mode.
427E	**17022**	Monitoring	**A requested movement command could not be realized (BISECTION)** A requested movement command could not be realized using the requested parameters. The movement command has been executed best possible and this message is therefore to be understood just as a warning. Examples: An axis start is requested in motion in an unfavorable dynamic situation (acceleration phase) in which the travel distance is too short or the velocity is significantly too high. Another possibility is a slave axis, which is decoupled in motion in a unfavorable dynamic situation and is afterwards given a motion as in the previous case.
427F	**17023**	Monitoring	**The new target position either has been overrun or will be overrun** The new target position either has been overrun or will be overrun, since until there it is impossible to stop. An internal stop command is commended.
4280	**17024**	Monitoring	**Group not ready / group not ready for new task" (internal error / information)** The group is being given a new task while it is still in the process of executing an existing task. This request is not allowed because it would interrupt the execution of the previous task. The new task could, for instance, be a positioning command, or the "set actual position" function. Precisely the converse relationships apply for the "set new end position" function. In that case, the group/axis must still be actively moving in order to be able to cause a change in the end position.
4281	**17025**	Parameter	**Parameters of the oriented stop (O-Stop) are not allowed** The modulo target position must not be lower than zero and not greater than or equal to the encoder modulo period (e.g. in the interval [0.0,360.0)). **Even in case of an error, the axis is stopped safely, but then it is not at the desired oriented position afterwards.**
4282	**17026**	Monitoring	**The modulo target position of the modulo-start is invalid.** The modulo target position is outside of the valid parameter range. So the position value should not be lower than zero and not greater or equal than the encoder modulo-period (e. g. in the interval [0.0,360.0] for the modulo start type "SHORTEST_WAY (261)").

Error(Hex)	Error(Dec)	Error type	Description
4283	**17027**	Parameter	**The activation mode is not allowed** The activation mode may have been used for online modification, scaling, as well as for online modification of the motion function. However, the activation mode used is not valid in this context. Either the mode is not defined or not yet implemented or it cannot be implemented in this constellation (e.g. if linear tables with an illegal cyclic activation mode NEXTCYCLE or NEXTCYCLEONCE are used). In other cases the mode is valid in principle, but the command cannot be implemented because the function already executes a job.
4284	**17028**	Parameter	**The parameterized jerk rate is not allowed** The jerk rate is smaller than the minimum jerk rate. The minimum jerk rate is 1.0 (e.g. mm/s^3).
4285	**17029**	Parameter	**The parameterized acceleration or deceleration is not permitted.** The parameterized acceleration or deceleration is lower than the permitted minimum acceleration. The value for minimum acceleration is calculated from minimum jerk rate and NC cycle time (minimum jerk rate multiplied with NC cycle time). The unit for example is mm/s^2.
4286	**17030**	Parameter	**The parameterized velocity is not permitted.** The parameterized target velocity is lower than the minimum velocity (but the value zero is permitted). The value for minimum velocity is calculated from the minimum jerk rate and the NC cycle time (minimum jerk rate multiplied with the square of the NC cycle time). The unit for example is mm/s.
4287	**17031**	Monitoring	**Activation cannot be executed due to a pending activation** A activation e.g. "CamIn", "CamScaling" or "WriteMotionFunction" cannot be executed due to a pending activation (e.g. "CamIn", "CamScaling", "WriteMotionFunction"). Only activation can be enabled.
4288	**17032**	Monitoring	**Illegal combination of different cycle times within an axis group** Within a logical axis group, different cycle times have been detected for the common setpoint generation or for I/O processing of an axis. This situation can occur with Master/ Slave coupling or configuring 3D- and FIFO-groups (including path, auxiliary, and slave axes).
4289	**17033**	Monitoring	**Invalid axis motion reversal** Due to the current dynamic state (current velocity, acceleration and jerk) a motion reversal would be caused. To avoid this motion reversal the axis command is not performed and the previous system state restored.
428A	**17034**	Monitoring	**Illegal command timing, because another instruction with future activation position is active** The moment for the command is illegal because there is still an old command with activation position active (e.g. "go to new velocity at threshold position" or "reach new velocity at threshold position").
428B	**17035**	Monitoring	**Stop-calculation routine (internal error)** Due to an internal error in the stop-calculation routine the current commando cannot be performed. The previous system state is restored.

Error(Hex)	Error(Dec)	Error type	Description
428C	**17036**	Monitoring	**A command with activation position cannot fully be performed because the remaining path is too short** A command with activation position (threshold) like "reach a new velocity at a position" can be just partially executed because the path from the actual position to the activation position is too short.
428D	**17037**	Monitoring	**Invalid decouple type** The command to release a slave coupling with subsequent restart command has been called with an invalid decoupling or restart type.
428E	**17038**	Monitoring	**Illegal target velocity when decoupling a slave axis** The command to release a slave coupling with subsequent restart command has been called with an impermissible target velocity [1 < V < Vmax].
428F	**17039**	Monitoring	**Activation new dynamic parameters cannot be performed** The command to activate new dynamic parameters such as acceleration, deceleration and jerk cannot be executed, as this would require a new assigned travel velocity. This error situation can occur, for example, if the axis is close to the target position in the accelerated state and the dynamics parameters are reduced.
4290	**17040**	Monitoring	**A command with activation position cannot be performed because the axis is already in the brake phase** A command with activation position (threshold) e.g. "reach new velocity at position" cannot be performed because the axis is already in the brake phase and the remaining path from the actual position to the activation position is too short.
4291	**17041**	Monitoring	**Jerk scaling of the decouple routine when decoupling a slave axis cannot find a valid solution** Internal jerk scaling of decouple routine cannot evaluate a valid solution (decoupling slave axis and transform to master axis). Otherwise, an unexpected velocity overshoot, motion reversal or exceeding of the target position could occur.
4292	**17042**	Monitoring	**Command cannot be executed because the command buffer is full** The command is rejected because the command buffer is full filled.
4293	**17043**	Internal	**Command is rejected due to an internal error in the Look Ahead (internal error)** The command is rejected due to an internal error in the "look ahead".
4294	**17044**	Monitoring	**Command is rejected because the new travel velocity cannot be implemented** The command is rejected, because the new travel velocity (target velocity) *Vrequ* is not realizable and an internal optimizing is impossible.
4295	**17045**	Monitoring	**Successive commands have the same end position** Successive commands have the same end position. So the travel path is zero.

Error(Hex)	Error(Dec)	Error type	Description
4296	**17046**	Monitoring	**Logical direction of travel of the axis is inconsistent with the parameterized direction of travel of the buffer command** In the extended buffer mode, where the actual end position is replaced by the new buffer start position, the logical positioning direction is inconsistent with the direction of the buffer command (=> contradiction). A buffered command (*BufferMode, BlendingLow, BlendingPrevious, BlendingNext, BlendingHigh*) is rejected with error 0x4296 if the command is using the Beckhoff-specific *optional blending position* but the blending position is located beyond the target position of the previous motion command.
4297	**17047**	Monitoring	**Command is rejected because the remaining distance in the current segment is too short** The remaining distance for positioning is not sufficient, therefore the command cannot be executed. This can be the case, for example, in the BufferMode (BlendingMode), if the remaining distance in the current segment is not sufficient to travel without acceleration and to have reached a specified velocity at the segment change (depending on the BufferMode).
429A	**17050**	Function	**Restart failed** There is already a motion command in the PTP command buffer and another new motion command, which should modify the existing command by a restart, has failed.
429B	**17051**	Monitoring	**Group error for invalid start parameters** This error refers to a wrong parameterization of the user (group error). E.g. dynamic parameters like Velo, Acc or Dec could be less than or equal to zero. Or the following error cases: - BaseFrequence < 0.0 - StartFrequence < 1.0 - StepCount < 1, StepCount > 200 - BaseAmplitude <= 0.0 - StepDuration <= 0.0 - StopFrequence >= 1/(2*CycleTime)
429C	**17052**	Monitoring	**PLC referencing cam is not found** During the referencing process for an axis it is moved in the direction of the PLC referencing cam. This referencing cam, however, was not found as expected (=> leads to the abortion of the referencing procedure).
429D	**17053**	Monitoring	**PLC referencing cam has not been released again** During the referencing process for an axis it is moved in the direction of the PLC referencing cam, and is only stopped again when the cam signal is reached. After the axis has also come to a physical standstill, the axis is subsequently started regularly from the cam again. In this case, the reference cam did not become free again as expected when driving down (=> leads to the abortion of the referencing procedure).
429E	**17054**	Monitoring	**I/O sync pulse was not found (only when using hardware latch)** If the hardware latch is activated, a sync pulse (zero pulse) is expected to be found and a sync event triggered following the expiry of a certain time or a certain distance. If this is not the case, the reaction is an error and the abortion of the referencing procedure.

Error(Hex)	Error(Dec)	Error type	Description	
429F	**17055**	Function	**The used buffer mode is unknown or not supported in this context** The buffer mode used for a PTP command (e.g. ABORTING, etc.) is unknown or not supported in this context.	
42A0	**17056**	Internal	**Group/axis consequential error** Consequential error resulting from another causative error related to another axis within the group. Group/axis consequential errors can occur in relation to master/slave couplings or with multiple axis interpolating DXD groups. If, for instance, it is detected that the lag error limit of a master axis has been exceeded, then this consequential error is assigned to all the other master axes and slave axes in this group.	
42A1	**17057**	Parameter	**Velocity reduction factor for C0/C1 transition is not allowed** A C0 transition describes two geometries which, while they are themselves continuous, no not have either continuous first or second differentials. The velocity reduction factor C0 acts on such transitions. **A C1 transition is characterized by the fact that the two geometries have a continuous course, but are continuously differentiable only once. The velocity reduction factor C1 acts on such transitions.**	
			Value range: [0.0 ... 1.0]	Unit: 1
42A2	**17058**	Parameter	**Critical angle at segment transition not allowed** The angle at the segment transition is not allowed.	
			Value range: (0.0 ... 180.0]	Unit: degrees
42A3	**17059**	Parameter	**Radius of the tolerance sphere** The radius of the tolerance sphere is outside the permitted range of values.	
			Value range: [0.0 ... 100.0]	Unit: e.g. mm
42A4	**17060**	Parameter	**Reserved** Reserved, currently not used.	
42A5	**17061**	Parameter	**Start type**	
			Value range: [0.1]	Unit: 1
42A6	**17062**	Parameter	**Reserved** Reserved, currently not used.	
42A7	**17063**	Parameter	**Blending** Blending is not possible with the given parameters.	
42A8	**17064**	Parameter	**Reserved** Reserved, currently not used.	
42A9	**17065**	Parameter	**Curve velocity reduction method not allowed (internal error)** The curve velocity reduction method does not exist.	
42AA	**17066**	Parameter	**Minimum velocity not allowed** The minimum velocity that has been entered is less than 0.0.	
42AB	**17067**	Parameter	**Power function input not allowed (internal error)** The input parameters in the power_() function lead to an FPU exception.	

Error(Hex)	Error(Dec)	Error type	Description
42AC	**17068**	Parameter	**Dynamic change parameter not allowed** A parameter that regulates the change of dynamics is invalid. Parameter: 1.Absolute path dynamics change: all parameters must be strictly positive. 2.Relative reduction c_f: 0.0 < c_f <= 1.0
42AD	**17069**	Memory	**Memory allocation error (internal error)** An error occurred during memory allocation.
42AE	**17070**	Function	**End position (internal error).** The calculated end position differs from the end position in the NC block
42AF	**17071**	Parameter	**Calculate remaining path length** invalid value Value range: [0,1]
42B0	**17072**	Function	**Setpoint generator SPP active** Starting the setpoint generator (SPP, SEC) has been refused, since the SPP task is already active.
42B1	**17073**	Parameter	**SPP parameter not allowed (internal error)** A parameter related to the internal structure of the setpoint generator (SPP) results in logical errors and/or to an FPU exception. Affects these parameters: Minimum velocity (>0.0), TimeMode, ModeDyn, ModeGeo, StartType, DistanceToEnd, TBallRadius.
42B2	**17074**	Parameter	**Velocity reduction factor not allowed** A parameter that regulates the reduction of velocity at segment transitions is invalid. Parameter: 1. Once continuously differentiable transitions: VeloVertexFactorC1 2. Not continuously differentiable transitions: VeloVertexFactorC0 CriticalVertexAngleLow, CriticalVertexAngleHigh.
42B3	**17075**	Parameter	**Helix is a circle** The helix has degenerated to a circle, and should be entered as such.
42B4	**17076**	Parameter	**Helix is a straight line** The helix has degenerated to a straight line, and should be entered as such.
42B5	**17077**	Parameter	**Guider parameter not allowed** One of the guider's parameters leads to logical errors and/or to an FPU exception.
42B6	**17078**	Address	**Invalid segment address (internal error)** The geometry segment does not have a valid geometry structure address or does not have a valid dynamic structure address.
42B7	**17079**	Parameter	**Not parameterized generator" (internal error)** The SPP generator is not yet parameterized and is therefore unable to operate.
42B8	**17080**	Address	**Not parameterized table (internal error)** The table has no information concerning the address of the corresponding dynamic generator.
42BA	**17082**	Internal	**Arc length of the smoothed path (internal error)**

Error(Hex)	Error(Dec)	Error type	Description
			The calculation of the arc length of the smoothed path.
42BB	**17083**	Parameter	**Tolerance sphere** The radius of the tolerance sphere is too small (smaller than 0.1 mm).
42BC	**17084**	Internal	**DXD software end positions (internal error)** An error has occurred in the calculation of the DXD software end positions.
42BD	**17085**	Function	**NC block violates software end positions of the group** At least one path axis with active software end position monitoring has violated the limit switches. Therefore the geometric entry is denied with an error.
42BE	**17086**	Parameter	**A path axis violates the end position** At least one path axis with active position limit monitoring violates the limit switches.
42BF	**17087**	Parameter	**Invalid reference velocity type**
42C0	**17088**	Internal	**Interpolating group contains axes of an incorrect axis type** An interpolating 3D group may only contain continuously guided axes of axis type 1 (SERVO).
42C1	**17089**	Internal	**Scalar product cannot be calculated** The length of one of the given vectors is 0.0.
42C2	**17090**	Internal	**Inverse cosine cannot be calculated** The length of one of the given vectors is 0.0.
42C3	**17091**	Parameter	**Invalid table entry type** The given table entry type is unknown.
42C4	**17092**	Parameter	**Invalid DIN66025 information type (internal error)** The given DIN66025 information type is unknown. Known types: G0, G1, G2, G3, G17, G18, G19.
42C5	**17093**	Parameter	**Invalid dimension (internal error)** The CNC dimension is unknown. Known dimensions: 1, 2, 3. Or: The CNC dimension is invalid for the given geometrical object. For a circle the dimension must be 2 or 3, while for a helix it must be 3.
42C6	**17094**	Parameter	**Geometrical object is not a straight line** The given object, interpreted as a straight line, has a length of 0.0.
42C7	**17095**	Parameter	**Geometrical object is not a circle** Interpreted as a circular arc, the given object has a length of 0.0, or an angle of 0.0 or a radius of 0.0.
42C8	**17096**	Parameter	**Geometrical object is not a helix** Interpreted as a circular arc, the given object has a length of 0.0, or an angle of 0.0, or a radius of 0.0. or a height of 0.0.
42C9	**17097**	Parameter	**Target velocity less than or equal to 0.0 is invalid** A value less than or equal to 0.0 for the target velocity (CNC) is not allowed, since the target velocity is positive by definition, and a target velocity of 0.0 cannot generate any motion.
42CA	**17098**	Address	**Address for look-ahead invalid (internal error)** The address supplied for the look-ahead is invalid.
42CB	**17099**	Function	**Setpoint generator SEC active**

Error(Hex)	Error(Dec)	Error type	Description
			Starting the setpoint generator (SEC) has been refused, since the SEC task is already active.
42CC	**17100**	Function	**CNC setpoint generation not active** The stop or override change was denied because the setpoint generation is not active.
42CD	**17101**	Function	**CNC setpoint generation in the stop phase** The stop or override change was denied because the setpoint generation is in the stop phase.
42CE	**17102**	Parameter	**Override not allowed** An override of less than 0.0 % or more than 100.0 % is invalid.
42CF	**17103**	Address	**Invalid table address (internal error)** The table address given for the initialization of the setpoint generator is invalid, or no valid logger connection (report file) is present.
42D0	**17104**	Parameter	**Invalid table entry type** The given table entry type is unknown.
42D1	**17105**	Memory	**Memory allocation failed** The memory allocation for a table failed.
42D2	**17106**	Memory	**Memory allocation failed** The memory allocation for a filter failed.
42D3	**17107**	Parameter	**Invalid parameter** Filter parameter is not allowed.
42D4	**17108**	Function	**Residual distance deletion not possible.** Delete Distance to go (only interpolation) failed. This error occurred, if e.g. the command 'DelDTG' was not programmed in the actual movement of the NC program.
42D5	**17109**	Internal	**The setpoint generator of the flying saw generates incompatible values (internal error)**
42D6	**17110**	Function	**Axis will be stopped since otherwise it will overrun its target position (old PTP setpoint generator)** If, for example, in case of a slave to master transformation for the new master a target position is commanded that will be overrun because of the actual dynamics the axis will be stopped internally to guarantee that the target position will not be overrun.
42D7	**17111**	Function	**Internal error in the transformation from slave to master**
42D8	**17112**	Function	**Wrong direction in the transformation of slave to master**
42DA	**17114**	Parameter	**Parameters of Motion Function (MF) table incorrect** The parameters of the Motion Function (MF) are invalid. This may refer to the first time created data set or to online changed data.
42DB	**17115**	Parameter	**Parameters of Motion Function (MF) table incorrect** The parameters of the Motion Function (MF) are invalid. This can refer to the data set created for the first time or to data changed online. The cause of the error may be that, for example, an active MF point (i.e. not an IGNORE point) points to a passive MF point (i.e. IGNORE point).

Error(Hex)	Error(Dec)	Error type	Description
42DC	**17116**	Monitoring	**Internal error by using Motion Function (MF)** An internal error occurs by using the Function (MF). This error cannot be solved by the user. Please ask the TwinCAT Support.
42DD	**17117**	Function	**Axis coupling with synchronization generator declined because of incorrect axis dynamic values** The axis coupling with the synchronization generator has been declined, because one of the slave dynamic parameter (machine data) is incorrect. Either the maximum velocity, the acceleration, the deceleration or the jerk is smaller or equal to zero, or the expected synchronous velocity of the slave axis is higher as the maximum allowed slave velocity.
42DE	**17118**	Function	**Coupling conditions of synchronization generator not allowed** If the direction of travel of the master axis is positive, the master synchronous position must be greater than the master coupling position ("i.e. lie in the future"). With negative master travel direction, the master synchronous position must be smaller than the master coupling position.
42DF	**17119**	Monitoring	**Motion profile of synchronization generator declines dynamic limit of slave axis or required characteristic of profile** One of the parameterized checks has detected an exceeding of the dynamic limits (max. velocity, max. acceleration, max. deceleration or max. jerk) of the slave axis or a profile property (e.g. overshoot or undershoot in position or velocity) is not allowed. See also additional/further messages in the Windows Event Viewer and in the message window of the system manager.
42E0	**17120**	Parameter	**Invalid parameter** The encoder generator parameter is not allowed.
42E1	**17121**	Parameter	**Invalid parameter** The external (FIFO) generator parameter is not allowed.
42E2	**17122**	Function	**External generator is active** The external generator cannot be started, as it is already active.
42E3	**17123**	Function	**External generator is not active** The external generator cannot be stopped, as it is not active.
42E4	**17124**	Function	**NC block with auxiliary axis violates software limit switches of the group** At least one auxiliary axis with active software end position monitoring has violated the limit switches. Therefore the geometric entry is denied with an error.
42E5	**17125**	Function	**NC block of Bezier curve (Bezier spline) type contains a singularity** The Bezier curve (Bezier spline) has a peak, i.e. at an inner point both the curvature and the magnitude of the velocity strive towards zero in such a way that the radius of curvature is infinite. **The Bezier spline should be divided at exactly this point according to the "Casteljau algorithm". This preserves the geometry and eliminates the interior singularity.**
42E7	**17127**	Parameter	**Value for dead time compensation not allowed**

Error(Hex)	Error(Dec)	Error type	Description	
			The value for the dead time compensation in seconds for a slave coupling to an encoder axis (virtual axis) is not allowed.	
			Value range: [0.0 ... 60.0]	Unit: s
42E8	**17128**	Parameter	**Internal error** GROUPERR_RANGE_NOMOTIONWINDOW	
			Value range: [0.0 ... 1000.0]	Unit: e.g. mm/s
42E9	**17129**	Parameter	**Internal error** GROUPERR_RANGE_NOMOTIONFILTERTIME	
			Value range: [0.0 ... 60.0]	Unit: s
42EA	**17130**	Parameter	**Internal error** GROUPERR_RANGE_TIMEUNITFIFO	
			Value range: (0.0 ... 1000.0]	Unit: s
42EB	**17131**	Parameter	**Internal error** GROUPERR_RANGE_OVERRIDETYPE	
			Value range: [1, 2]	Unit: 1
42EC	**17132**	Parameter	**Internal error** GROUPERR_RANGE_OVERRIDECHANGETIME	
			Value range: (0.0 ... 1000.0]	Unit: s
42ED	**17133**	Parameter	**Internal error** GROUPERR_FIFO_INVALIDDIMENSIO **The FIFO dimension (number of axes) has been increased from 8 to 16 from TC 2.11 Build 1547.**	
			Value range: [1 ... 8] resp. [1 ... 16]	Unit: 1 (number of axes)
42EE	**17134**	Address	**Internal error** GROUPERR_ADDR_FIFOTABLE	
42EF	**17135**	Monitoring	**Axis is locked for motion commands because a stop command is still active** The axis/group is locked for motion commands because a stop command is still active. This lock can be released by calling the stop command with Execute=FALSE or by an axis reset (see also *MC_Stop* and *MC_Reset* in *TcMC2.Lib*).	
42F0	**17136**	Parameter	**Invalid number of auxiliary axes** The local number of auxiliary axes does not tally with the global number of auxiliary axes.	
42F1	**17137**	Parameter	**Invalid reduction parameter for auxiliary axes** The velocity reduction parameters for the auxiliary axes are inconsistent.	
42F2	**17138**	Parameter	**Invalid dynamic parameter for auxiliary axes** The dynamic parameters for the auxiliary axes are inconsistent.	
42F3	**17139**	Parameter	**Invalid coupling parameter for auxiliary axes** The coupling parameters for the auxiliary axes are inconsistent.	
42F4	**17140**	Parameter	**Invalid auxiliary axis entry** The auxiliary axis entry is empty (no axis motion).	
42F6	**17142**	Parameter	**Invalid parameter**	

Error(Hex)	Error(Dec)	Error type	Description	
			The limit for velocity reduction of the auxiliary axes is invalid. It has to be in the interval 0..1.0.	
42F8	**17144**	Parameter	**BlockSearch - segment not found** The segment specified as parameter could not be found until the end of the NC program. Possible cause: - nBlockId is not specified in the mode described by eBlockSearchMode	
42F9	**17145**	Parameter	**Blocksearch - Invalid remaining segment length** The remaining distance in the fLength parameter is incorrectly parameterized	
42FB	**17147**	Internal	**Internal error in connection with coupled axes (slave axes)** Internal fatal error when using coupled axes (slave axes). Inconsistent internal state. Please contact our Support.	
42FC	**17148**	Parameter	**Parameter for maximum number of jobs (entries) to be transferred is invalid** The parameter describing the maximum number of entries to be transferred from the SPP to the SEC table per NC cycle is invalid.	
			Value range: [1 20]	Unit: 1
42FF	**17151**	Monitoring	**Customer-specific error** This is a customer-specific monitoring function.	

7.8.4 Axis Errors

Error(Hex)	Error(Dec)	Error type	Description	
4300	**17152**	Parameter	**Axis ID not allowed** The value for the axis ID is not allowed, e.g. because it has already been assigned, is less than or equal to zero, is greater than 255, or does not exist in the current configuration.	
			Value range: [1 ... 255]	Unit: 1
4301	**17153**	Parameter	**Axis type not allowed** The value for the axis type is unacceptable because it is not defined. Type 1: servo Type 2: fast/creep Type 3: stepper motor	
			Value range: [1 ... 3]	Unit: 1
4306	**17158**	Parameter	**Slow manual velocity not allowed** The value for the slow manual velocity is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. m/min
4307	**17159**	Parameter	**Fast manual velocity not allowed** The value for the fast manual velocity is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. m/min
4308	**17160**	Parameter	**Rapid traverse velocity not allowed** The value for the rapid traverse velocity is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. m/min
4309	**17161**	Parameter	**Acceleration not allowed** The value for the axis acceleration is not allowed.	
			Value range: [0.0, 1000000.0]	Unit: e.g. m/s/s

Error(Hex)	Error(Dec)	Error type	Description
430A	**17162**	Parameter	**Deceleration not allowed** The value for the axis deceleration is not allowed. Value range: [0.0, 1000000.0] — Unit: e.g. m/s/s
430B	**17163**	Parameter	**Jerk not allowed** The value for the axis jerk is not allowed. Value range: [0.0, 1000000.0] — Unit: e.g. m/s/s/s
430C	**17164**	Parameter	**Delay time between position and velocity is not permissible (dead time compensation)** The value for the delay time between position and velocity ("dead time compensation") is not allowed. Value range: [0, 0.1] — Unit: s
430D	**17165**	Parameter	**Override type not allowed** The value for the velocity override type is not allowed. Type 1: Related to internal reduced velocity (default value) Type 2: Related to original external start velocity Value range: [1 ... 4] — Unit: 1
430E	**17166**	Parameter	**NCI: Velo-Jump-Factor not allowed** The value for the velo-jump-factor ("VeloJumpFactor") is not allowed. This parameter only works for TwinCAT NCI. Value range: [0, 1000000] — Unit: 1
430F	**17167**	Parameter	**NCI: Radius of tolerance sphere for the auxiliary axis is invalid** It was tried to enter an invalid value for the size of the tolerance sphere. This sphere affects only auxiliary axes! Value range: [0, 1000] — Unit: e.g. mm
4310	**17168**	Parameter	**NCI: Value for maximum deviation for the auxiliary axis is invalid** It was tried to enter an invalid value for the maximum allowed deviation. This parameter affects only auxiliary axes! Value range: [0, 10000] — Unit: e.g. mm
4312	**17170**	Parameter	**Referencing velocity in direction of cam not allowed** The value for the referencing velocity in the direction of the referencing cam is not allowed. Value range: [0.0, 10000.0] — Unit: e.g. m/min
4313	**17171**	Parameter	**Referencing velocity in sync direction not allowed** The value for the referencing velocity in the direction of the sync pulse (zero track) is not allowed. Value range: [0.0, 10000.0] — Unit: e.g. m/min
4314	**17172**	Parameter	**Pulse width in positive direction not allowed** The value for the pulse width in the positive direction is not allowed (pulsed operation). The use of the pulse width for positioning is chosen implicitly through the axis start type. Pulsed operation corresponds to positioning with a relative travel path that corresponds precisely to the pulse width. Value range: [0.0, 1000000.0] — Unit: e.g. mm

Error(Hex)	Error(Dec)	Error type	Description	
4315	**17173**	Parameter	**Pulse width in negative direction not allowed** The value for the pulse width in the negative direction is not allowed (pulsed operation). The use of the pulse width for positioning is chosen implicitly through the axis start type. Pulsed operation corresponds to positioning with a relative travel path that corresponds precisely to the pulse width.	
			Value range: [0.0, 1000000.0	Unit: e.g. mm
4316	**17174**	Parameter	**Pulse time in positive direction not allowed** The value for the pulse width in the positive direction is not allowed (pulsed operation).	
			Value range: [0.0, 600.0]	Unit: s
4317	**17175**	Parameter	**Pulse time in negative direction not allowed** The value for the pulse width in the negative direction is not allowed (pulsed operation).	
			Value range: [0.0, 600.0]	Unit: s
4318	**17176**	Parameter	**Creep distance in positive direction not allowed** The value for the creep distance in the positive direction is not allowed.	
			Value range: [0.0, 100000.0]	Unit: e.g. mm
4319	**17177**	Parameter	**Creep distance in negative direction not allowed** The value for the creep distance in the negative direction is not allowed.	
			Value range: [0.0, 100000.0]	Unit: e.g. mm
431A	**17178**	Parameter	**Braking distance in positive direction not allowed** The value for the braking distance in the positive direction is not allowed.	
			Value range: [0.0, 100000.0]	Unit: e.g. mm
431B	**17179**	Parameter	**Braking distance in negative direction not allowed** The value for the braking distance in the negative direction is not allowed.	
			Value range: [0.0, 100000.0]	Unit: e.g. mm
431C	**17180**	Parameter	**Deceleration time in positive direction not allowed** The value for the deceleration time in the positive direction is not allowed.	
			Value range: [0.0, 60.0]	Unit: s
431D	**17181**	Parameter	**Deceleration time in negative direction not allowed** The value for the deceleration time in the negative direction is not allowed.	
			Value range: [0.0, 60.0]	Unit: s
431E	**17182**	Parameter	**Switching time from rapid to slow traverse not allowed** The value for the time to switch from rapid to slow traverse is not allowed.	
			Value range: [0.0, 60.0]	Unit: s
431F	**17183**	Parameter	**Creep distance for stop not allowed** The value for the creep distance for an explicit stop is not allowed.	
			Value range: [0.0, 100000.0]	Unit: e.g. mm

Error(Hex)	Error(Dec)	Error type	Description	
4320	**17184**	Parameter	**Motion monitoring not allowed** The value for the activation of the motion monitoring is not allowed.	
			Value range: [0, 1]	Unit: 1
4321	**17185**	Parameter	**Position window monitoring not allowed** The value for the activation of the position window monitoring is not allowed.	
			Value range: [0, 1]	Unit: 1
4322	**17186**	Parameter	**Target window monitoring not allowed** The value for the activation of target window monitoring is not allowed.	
			Value range: [0, 1]	Unit: 1
4323	**17187**	Parameter	**Loop not allowed** The value for the activation of loop movement is not allowed.	
			Value range: [0, 1]	Unit: 1
4324	**17188**	Parameter	**Motion monitoring time not allowed** The value for the motion monitoring time is not allowed.	
			Value range: [0.0, 600.0]	Unit: s
4325	**17189**	Parameter	**Target window range not allowed** The value for the target window is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm
4326	**17190**	Parameter	**Position window range not allowed** The value for the position window is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm
4327	**17191**	Parameter	**Position window monitoring time not allowed** The value for the position window monitoring time is not allowed.	
			Value range: [0.0, 600.0]	Unit: s
4328	**17192**	Parameter	**Looping distance not allowed** The value for the looping distance is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm
4329	**17193**	Parameter	**Axis cycle time not allowed** The value for the axis cycle time is not allowed.	
			Value range: [0.001, 0.1]	Unit: s
432A	**17194**	Parameter	**Operation mode stepper motor not allowed** The value for the stepper motor operating mode is not allowed.	
			Value range: [1, 2]	Unit: 1
432B	**17195**	Parameter	**Displacement per stepper motor step not allowed** The value for the displacement associated with one step of the stepper motor is not allowed (step scaling).	
			Value range: [0.000001, 1000.0]	Unit: e.g. mm/ STEP
432C	**17196**	Parameter	**Minimum velocity for stepper motor set value profile not allowed** The value for the minimum velocity of the stepper motor velocity profile is not allowed.	
			Value range: [0.0, 1000.0]	Unit: e.g. m/min

Error(Hex)	Error(Dec)	Error type	Description	
432D	**17197**	Parameter	**Stepper motor stages for one velocity level not allowed** The value for the number of steps for each velocity level in the setpoint generation is not allowed.	
			Value range: [0, 100]	Unit: 1
432E	**17198**	Parameter	**DWORD for the interpretation of the axis units not allowed** The value that contains the flags for the interpretation of the position and velocity units is not allowed.	
			Value range: [0, 0xFFFFFFFF]	Unit: 1
432F	**17199**	Parameter	**Maximum velocity not allowed** The value for the maximum permitted velocity is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. m/min
4330	**17200**	Parameter	**Motion monitoring window not allowed** The value for the motion monitoring window is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm
4331	**17201**	Parameter	**PEH time monitoring not allowed** The value for the activation of the PEH time monitoring is not allowed (PEH: positioning end and halt).	
			Value range: [0, 1]	Unit: 1
4332	**17202**	Parameter	**PEH monitoring time not allowed** The value for the PEH monitoring time (timeout) is not allowed (PEH: positioning end and halt). Default value: 5s	
			Value range: [0.0, 600.0]	Unit: s
4333	**17203**	Parameter	**Parameter "Brake Release Delay" is invalid** The parameter for the brake release delay of a rapid/slow traverse axis is invalid.	
			Value range: [0.0, 60.0]	Unit: s
4334	**17204**	Parameter	**Parameter NC Data Persistence is invalid** The boolean parameter NC Data Persistence of an axis is invalid.	
			Value range: [0, 1]	Unit: 1
4335	**17205**	Parameter	**Parameter for the error reaction mode is invalid** The parameter for the error reaction mode of the axis is invalid (instantaneous, delayed).	
			Value range: [0, 1]	Unit: 1
4336	**17206**	Parameter	**Parameter for the error reaction delay is invalid** The parameter for the error reaction delay of the axis is invalid.	
			Value range: [0.0, 1000.0]	Unit: s
4337	**17207**	Parameter	**Parameter "Use actual values in deactivated state" is invalid** The parameter "Use actual values in deactivated state" is invalid.	
			Value range: [0, 1]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description	
4338	17208	Parameter	**Parameter "Allow Motion Commands for Slave Axes" is invalid** The boolean parameter "Allow motion commands for slave axes" is invalid. This parameter determines whether a motion command may be sent to a slave axis or whether this is rejected with an NC error `0x4266` or `0x4267`.	
			Value range: [0, 1]	Unit: 1
4339	17209	Parameter	**Parameter "Allow motion commands for axis in external setpoint generation" is invalid** The boolean parameter "Allow motion commands for axis in external setpoint generation" is invalid. This parameter determines whether a motion command may be sent to an axis in the external setpoint generation state or whether this is rejected with an error `0x4257`.	
			Value range: [0, 1]	Unit: 1
433A	17210	Parameter	**Parameter "Fading Acceleration" is invalid** The Fading Acceleration parameter for the fading profile from SET to ACTUAL values is invalid. This parameter defines how to fade from a setpoint based axis coupling to an actual value based coupling (indirectly results in a time for the fading). **The value 0.0 causes the minimum of the default acceleration and default deceleration to be used internally in the NC as the fading acceleration.**	
			Value range: [0; 0.01 .. 1.0e+20]	Unit: e.g. mm/s^2

Error(Hex)	Error(Dec)	Error type	Description
433B	17211	Parameter	**Fast Axis Stop signal type not allowed** The value for the Signal Type of the 'Fast Axis Stop' is not allowed [0...5].
433C	17212	Parameter	**ADS index offset not allowed** New value for the 'ADS offset (axis state) ' for user-defined parameter named 'UserData' in AxisRef is invalid.
4340	17216	Initialization	**Axis initialization** The axis has not been initialized. Although the axis has been created, the rest of the initialization has not been performed (1. Initialization of axis I/O, 2. Initialization of axis, 3. Reset axis).
4341	17217	Address	**Group address** The axis does not have a group, or the group address has not been initialized (group contains the setpoint generation).
4342	17218	Address	**Encoder address** The axis does not have an encoder, or the encoder address has not been initialized.
4343	17219	Address	**Controller address** **The** axis does not have a controller, or the controller address has not been initialized.
4344	17220	Address	**Drive address** **The** axis does not have a drive, or the drive address has not been initialized.

Error(Hex)	Error(Dec)	Error type	Description
4345	**17221**	Address	**Axis interface PLC to NC address** The axis has no axis interface from the PLC to the NC (PlcToNc) or the axis interface address has not been initialized.
4346	**17222**	Address	**"Axis interface NC to PLC address** The axis has no axis interface from the NC to the PLC (NcToPlc) or the axis interface address has not been initialized.
4347	**17223**	Address	**Size of the axis interface NC to PLC is not allowed (internal error)** The size of the axis interface from NC to PLC (NcToPlc) is not allowed.
4348	**17224**	Address	**Size of the axis interface PLC to NC is not allowed (internal error)** The size of the axis interface from PLC to NC (PlcToNc) is not allowed.
4356	**17238**	Monitoring	**Controller enable** **The** controller enable for the axis is not available (see axis interface PlcToNc). This enable is required, for instance, for an axis positioning task.
4357	**17239**	Monitoring	**Feed enable minus** **A** feed enable for movement in the negative direction is not available (see axis interface PlcToNc). This enable is required, for instance, for an axis positioning task in the negative direction.
4358	**17240**	Monitoring	**Feed enable plus** **A** feed enable for movement in the positive direction is not present (see axis interface PlcToNc). This enable is required, for instance, for an axis positioning task in the positive direction.
4359	**17241**	Monitoring	**Target velocity not allowed** The target velocity requested for a positioning task is not allowed. This can happen if the velocity is less than or equal to zero, larger than the maximum permitted axis velocity, or, in the case of servo drives, is larger than the reference velocity of the axis (see axis and drive parameters).
435A	**17242**	Monitoring	**Movement smaller than one encoder increment (internal error)** The movement required of an axis is, in relation to a positioning task, smaller than one encoder increment (see scaling factor). This information is, however, handled internally in such a way that the positioning is considered to have been completed without an error message being returned.
435B	**17243**	Monitoring	**Set acceleration monitoring (internal error)** The set acceleration has exceeded the maximum permitted acceleration or deceleration parameters of the axis.

Error(Hex)	Error(Dec)	Error type	Description
435C	**17244**	Monitoring	**PEH time monitoring** The PEH time monitoring has detected that, after the PEH monitoring time that follows a positioning has elapsed, the target position window has not been reached. The following points must be checked: Is the PEH monitoring time, in the sense of timeout monitoring, set to a sufficiently large value (e.g. 1-5 s)? The PEH monitoring time must be chosen to be significantly larger than the target position monitoring time. Have the criteria for the target position monitoring (range window and time) been set too strictly? **The PEH time monitoring only functions when target position monitoring is active!**
435D	**17245**	Monitoring	**Motion Monitoring** The actual position of the axis has not changed or has changed only slightly during the motion monitoring time. To avoid an error, the axis must change by more than the parameterized motion monitoring window in at least one NC cycle during the monitoring time. => Check, whether axis is mechanically blocked, or the encoder system failed.
435E	**17246**	Monitoring	**Looping distance smaller than braking distance** The absolute value of the looping distance is less or equal than the positive or negative braking distance. This is not allowed.
435F	**17247**	Monitoring	**Starting velocity not allowed** The required starting velocity for a positioning task is not allowed (normally the starting velocity is zero). This can happen if the velocity is less than or equal to zero, larger than the maximum permitted axis velocity, or, in the case of servo drives, is larger than the reference velocity of the axis (see axis and drive parameters).
4360	**17248**	Monitoring	**Final velocity not allowed** The required final velocity for a positioning task is not allowed (normally the final velocity is zero). This can happen if the velocity is less than or equal to zero, larger than the maximum permitted axis velocity, or, in the case of servo drives, is larger than the reference velocity of the axis (see axis and drive parameters).
4361	**17249**	Monitoring	**Time range exceeded (future)** The calculated position lies too far in the future (e.g. when converting a position value in a DC timestamp).
4362	**17250**	Monitoring	**Time range exceeded (past)** The calculated position lies too far in the past (e.g. when converting a position value in a DC timestamp).
4363	**17251**	Monitoring	**Position cannot be determined** The requested position cannot be determined mathematically because a) it has never been reached in the past or b) it will never be reached in the future (e.g. if the axis velocity is zero or if a reversal of motion occurs due to an acceleration).

Error(Hex)	Error(Dec)	Error type	Description
4364	**17252**	Monitoring	**"Position indeterminable (conflicting direction of travel)"** The direction of travel expected by the caller of the function deviates from the actual direction of travel (conflict between PLC and NC view, for example when converting a position to a DC time).
4370	**17264**	Monitoring	**No slave coupling possible (velocity violation)** A slave coupling to a master axis (e.g. by a universal flying saw) is rejected because otherwise the maximum velocity of the slave axis would be exceeded (a velocity monitoring has been selected).
4371	**17265**	Monitoring	**No slave coupling possible (acceleration violation)** A slave coupling to a master axis (e.g. by a universal flying saw) is rejected, because otherwise the maximum acceleration of the slave axis will be exceeded (an acceleration monitoring is selected).
4372 - 438B	**17266 - 17291**		See TF5055 NC Flying Saw - Error Codes
43A0	**17312**	Internal	**Axis consequential error** Consequential error resulting from another causative error related to another axis. Axis consequential errors can occur in relation to master/slave-couplings or with multiple axis interpolating DXD groups.

7.8.5 Encoder Errors

Error(Hex)	Error(Dec)	Error type	Description	
4400	**17408**	Parameter	**Encoder ID not allowed** The value for the encoder ID is not allowed, e.g. because it has already been assigned, is less than or equal to zero, or is bigger than 255.	
			Value range: [1 ... 255]	Unit: 1
4401	**17409**	Parameter	**Encoder type not allowed** The value for the encoder type is unacceptable because it is not defined. Type 1: Simulation (incremental) Type 2: M3000 (24 bit absolute) Type 3: M31x0 (24 bit incremental) Type 4: KL5101 (16 bit incremental) Type 5: KL5001 (24 bit absolute SSI) Type 6: KL5051 (16 bit BISSI)	
			Value range: [1 ... 6]	Unit: 1
4402	**17410**	Parameter	**Encoder mode** The value for the encoder mode (operation mode) is invalid. Mode 1: Determination of actual position Mode 2: Determination of actual position and actual velocity (filter)	
			Value range: [1, 2]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description	
4403	**17411**	Parameter	**Encoder count direction** The flag for the encoder counting direction is not allowed. Flag 0: Positive encoder count direction Flag 1: Negative encoder count direction	
			Value range: [0, 1]	Unit: 1
4404	**17412**	Initialization	**Calibration state** The flag for the calibration state is not allowed. Flag 0: Axis is not referenced Flag 1: Axis is referenced	
			Value range: [0, 1]	Unit: 1
4405	**17413**	Parameter	**Encoder increments for each physical encoder rotation** The value for the number of encoder increments for each physical rotation of the encoder is not allowed. This value is used by the software for the calculation of encoder overruns and underruns.	
			Value range: [255, 0xFFFFFFFF]	Unit: INC
4406	**17414**	Parameter	**Scaling factor** The value for the scaling factor is not allowed. This scaling factor provides the weighting for the conversion of an encoder increment (INC) to a physical unit such as millimeters or degrees.	
			Value range: [0.000001, 100.0]	Unit: e.g. mm/INC
4407	**17415**	Parameter	**Position offset (zero point offset)** The value for the position offset of the encoder is not allowed. This value is added to the calculated encoder position, and is interpreted in the physical units of the encoder.	
			Value range: [-1000000.0, 1000000.0]	Unit: e.g. mm
4408	**17416**	Parameter	**Modulo factor** The value for the encoder's modulo factor is not allowed.	
			Value range: [1.0, 1.0E+9]	Unit: e.g. mm or degrees
4409	**17417**	Parameter	**Position filter time** The value for the actual position filter time is not allowed (P-T1 filter).	
			Value range: [0.0, 60.0]	Unit: s
440A	**17418**	Parameter	**Velocity filter time** The value for the actual velocity filter time is not allowed (P-T1 filter).	
			Value range: [0.0, 60.0]	Unit: s

Error(Hex)	Error(Dec)	Error type	Description
440B	**17419**	Parameter	**Acceleration filter time** The value for the actual acceleration filter time is not allowed (P-T1 filter). Value range: [0.0, 60.0] Unit: s
440C	**17420**	Initialization	**Cycle time not allowed (internal error)** The value of the SEC cycle time for the calculation of actual values is not allowed (e.g. is less than or equal to zero).
440D	**17421**	Initialization	**Setting of the selected units is invalid** Settings for modulo position, velocity etc. lead to an error.
440E	**17422**	Parameter	**Actual position correction / measurement system error correction** The value for the activation of the actual position correction ("measuring system error correction") is not allowed. Value range: [0, 1]
440F	**17423**	Parameter	**Filter time actual position correction** The value for the actual position correction filter time is not allowed (P-T1 filter). Value range: [0.0, 60.0] Unit: 1
4410	**17424**	Parameter	**Search direction for referencing cam inverted** The value of the search direction of the referencing cam in a referencing procedure is not allowed. Value 0: Positive direction Value 1: Negative direction Value range: [0, 1] Unit: 1
4411	**17425**	Parameter	**Search direction for sync pulse (zero pulse) inverted** The value of the search direction of the sync pulse (zero pulse) in a referencing procedure is not allowed. Value 0: Positive direction Value 1: Negative direction Value range: [0, 1] Unit: 1
4412	**17426**	Parameter	**Reference position** The value of the reference position in a referencing procedure is not allowed. Value range: [-1000000.0, 1000000.0] Unit: e.g. mm

Error(Hex)	Error(Dec)	Error type	Description	
4413	**17427**	Parameter	**Distance monitoring between activation of the hardware latch and occurrence of the sync pulse (obsolete)** The flag for the distance monitoring between activation of the hardware latch and occurrence of the sync/zero pulse ("latch valid") is not allowed. Value 0: Passive Value 1: Active	
			Value range: [0, 1]	Unit: 1
4414	**17428**	Parameter	**Minimum gap between activation of the hardware latch and occurrence of the sync pulse (obsolete)** The value for the minimum gap in increments between activation of the hardware latch and occurrence of the sync/zero pulse ("latch valid") during a referencing procedure is not allowed.	
			Value range: [0, 65536]	Unit: INC
4415	**17429**	Parameter	**External sync pulse (obsolete)** The value of the activation or deactivation of the external sync pulse in a referencing procedure is not allowed. Value 0: Passive Value 1: Active	
			Value range: [0, 1]	Unit: 1
4416	**17430**	Parameter	**Scaling of the noise rate is not allowed** The value of the scaling (weighting) of the synthetic noise rate is not allowed. This parameter exists only in the simulation encoder and serves to produce a realistic simulation.	
			Value range: [0, 1000000]	Unit: 1
4417	**17431**	Parameter	**Tolerance window for modulo-start** The value for the tolerance window for the modulo-axis-start is invalid. The value must be greater or equal than zero and smaller than the half encoder modulo-period (e.g. in the interval [0.0,180.0)).	
			Value range: [0.0, 180.0], Max: 0.5*modulo period	Unit: e.g. mm or degrees
4418	**17432**	Parameter	**Encoder referencing mode** The value for the encoder reference mode is not allowed, resp. is not supported for this encoder type.	
			Value range: [0, 5]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description	
4419	**17433**	Parameter	**Encoder evaluation direction** The value for the encoder evaluation direction (log. counting direction) is not allowed.	
			Value range: [0, 3]	Unit: 1
441A	**17434**	Parameter	**Encoder absolute dimensioning system** The value for the encoder reference system is invalid: 0: INCREMENTAL, 1: ABSOLUTE, 2: ABSOLUTE+MODULO	
			Value range: [0, 2]	Unit: 1
441B	**17435**	Parameter	**Encoder position initialization mode** When starting the TC system the value for the encoder position initialization mode is invalid.	
			Value range: [0, 1]	Unit: 1
441C	**17436**	Parameter	**Encoder sign interpretation (UNSIGNED / SIGNED data type)** The value for the encoder sign interpretation (data type) for the encoder the actual increment calculation is invalid: 0: default/not defined, 1: UNSIGNED, 2: SIGNED.	
			Value range: [0, 2]	Unit: 1
441D	**17437**	Parameter	**Homing Sensor Source** The value for the Encoder Homing Sensor Source is invalid or not supported for this encoder type.	
			Value range: [0, 16]	Unit: 1
4420	**17440**	Parameter	**Software end position monitoring minimum not allowed** The value for the activation of the software end position monitoring minimum is not allowed.	
			Value range: [0, 1]	Unit: 1
4421	**17441**	Parameter	**Software end position monitoring maximum not allowed** The value for the activation of the software end position monitoring maximum is not allowed.	
			Value range: [0, 1]	Unit: 1
4422	**17442**	Function	**Actual value setting is outside the value range** The "set actual value" function cannot be carried out, because the new actual position is outside the expected range of values.	
			Value range: [-1.0E10, 1.0E10]	Unit: e.g. mm

Error(Hex)	Error(Dec)	Error type	Description	
4423	**17443**	Parameter	**Software end position minimum not allowed** The value for the software end position minimum is not allowed.	
			Value range: [-1000000000.0, 1000000000.0]	Unit: e.g. mm
4424	**17444**	Parameter	**Software end position maximum not allowed** The value for the software end position maximum is not allowed.	
			Value range: [-1000000000.0, 1000000000.0]	Unit: e.g. mm
4425	**17445**	Parameter	**Filter mask for the raw value of the encoder not allowed** The value for the filter mask of the encoder raw data in increments is invalid.	
			Value range: [0x0, 0xFFFFFFFF]	Unit: 1
4426	**17446**	Parameter	**Reference mask for the raw value of the encoder not allowed** The value for the reference mask (increments per encoder turn, absolute resolution) for the raw data of the encoder is invalid. E.g. this value is used for axis reference sequence (calibration) with the reference mode "Software Sync".	
			Value range: [0x0000000F, 0xFFFFFFFF]	Unit: 1
4427	**17447**	Parameter	**Parameter "Dead time compensation mode" (encoder) is invalid** The parameter for the dead time compensation mode on the `NC` encoder is invalid (OFF, ON with velocity, ON with velocity and acceleration).	
			Value range: [0, 1, 2]	Unit: 1
4428	**17448**	Parameter	**Parameter 'Control bits of the dead time compensation' (encoder) is invalid** The parameter for the control bits of the dead time compensation at the encoder is invalid (e.g. relative or absolute time interpretation).	
			Value range: [>0]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description
4429	**17449**	Parameter	**Parameter 'time shift of dead time compensation mode' (encoder) is invalid** The parameter for the time shift of the dead time compensation (Time Shift in nanoseconds) at the encoder is invalid. Value range: [-1.0E9 .. 1.0E9] | Unit: ns
4430	**17456**	Function	**Hardware latch activation (encoder)** Activation of the encoder hardware latch was implicitly initiated by the referencing procedure. If this function has already been activated but a latch value has not yet become valid ("latch valid"), another call to the function is refused with this error.
4431	**17457**	Function	**Activation of external hardware latch / Touch probe function (encoder)** The activation of the external hardware latch (only available for KL5101, SERCOS, AX2xxx) is initiated explicitly by an ADS command (called from the PLC program of the Visual Basic interface). If this function has already been activated, but the latch value has not yet been made valid by an external signal ("external latch valid" or "touch probe latched" or "real-time status bit"), another call to the function is refused with this error. Likewise, it is possible that this function cannot be executed because of another simultaneous function, such as referencing an incremental encoder axis.
4432	**17458**	Function	**External hardware latch activation (encoder)** If a referencing procedure has previously been initiated and the hardware still signals a valid latch value ("latch valid"), this function must not be called. In practice, however, this error can almost never occur.

Error(Hex)	Error(Dec)	Error type	Description
4433	**17459**	Function	**Activation of external hardware latch / Touch probe function (encoder)** This function has been activated before and has not been finished since (the internal handshake communication between NC and I/O device is still active). In the meantime, a renewed activation is not allowed and is therefore rejected with an error. (If this function has already been triggered before and the hardware still signals that the external latch value is already valid ("external latch valid" or "touch probe" or "real-time status bit"), a renewed activation must not be executed. In this case the validity of the external hardware latch would be signaled immediately by mistake, but still with an old latch value.)
4434	**17460**	Monitoring	**Encoder function is not supported** An encoder function has been activated that is currently not released for use, or which is not even implemented.
4435	**17461**	Monitoring	**Encoder function is already active** An encoder function cannot been activated because this functionality is already active.
4440	**17472**	Initialization	**Encoder initialization** Encoder has not been initialized. Although the axis has been created, the rest of the initialization has not been performed: 1. Initialization of axis I/O, 2. Initialization of axis, 3. Reset axis.
4441	**17473**	Address	**Axis address** The encoder does not have an axis, or the axis address has not been initialized.
4442	**17474**	Address	**Address I/O input structure** Drive has no valid I/O input address in the process image.
4443	**17475**	Address	**Address I/O output structure** The encoder does not have a valid I/O output address in the process image.
4450	**17488**	Monitoring	**Encoder counter underflow monitoring** The encoder's incremental counter has underflowed.
4451	**17489**	Monitoring	**Encoder counter overflow monitoring** The encoder's incremental counter has overflowed.

Error(Hex)	Error(Dec)	Error type	Description
4460	**17504**	Monitoring	**Minimum software position limit (axis start)** While monitoring for the minimum software position limit is active, an axis start has been performed on a position that is below the minimum software position limit.
4461	**17505**	Monitoring	**Maximum software position limit (axis start)** While monitoring for the maximum software position limit is active, an axis start has been performed towards a position that is above the maximum software position limit.
4462	**17506**	Monitoring	**Minimum software position limit (positioning process)** While monitoring for the minimum software position limit is active, the actual position has fallen below the minimum software position limit. For servo axes - they are continuously guided - this limit is extended by the amount of the parameterized lag error window.
4463	**17507**	Monitoring	**Maximum software position limit (positioning process)** While monitoring for the maximum software position limit is active, the actual position has exceeded the maximum software position limit. For servo axes - they are continuously guided - this limit is extended by the amount of the parameterized lag error window.
4464	**17508**	Monitoring	**Encoder hardware error** The drive resp. the encoder system reports a hardware error of the encoder. An optimal error code is displayed in the message of the event log.
4465	**17509**	Monitoring	**Position initialization error at system startup** At the first initialization of the actual position was this for all initialization trials (without over-/under-flow, with underflow and overflow) out of the final position minimum and maximum.

Error(Hex)	Error(Dec)	Error type	Description
4466	**17510**	Monitoring	**Invalid I/O data for more than n continuous NC cycles (encoder)** The axis (encoder) has detected invalid encoder I/O data for more than `n` continuous NC cycles (NC SEC task) (e.g. `n=3`). Typically, an EtherCAT device is a "Working Counter Error" (WcState), which shows that the data transmission between I/O device and controller is disturbed. If this error is continuously present for a longer period of time, then this can lead to the loss of the referencing of the axis (the "Homed" flag is reset and the encoder gets the state "unreferenced"). Possible reasons for this error: an EtherCAT slave may have left its OP state or there is too high a real-time load on the controller or too high a real-time jitter.
4467	**17511**	Monitoring	**Invalid actual position (encoder)** The I/O device returns an invalid actual position (for CANopen/CoE see bit 13 of encoder state "TxPDO data invalid" or "invalid actual position value").
4468	**17512**	Monitoring	**Invalid I/O input data (error type 1)** The monitoring of the "cyclic I/O input counter" (2 bit counter) has detected an error. The input data has not been updated for at least 3 NC SEC cycles (the 2 bit counter shows a constant value for several NC SEC cycles instead of increasing exactly by 1 from cycle to cycle).
4469	**17513**	Monitoring	**Invalid I/O input data (error type 2)** The monitoring of the "cyclic I/O input counter" (2 bit counter) has detected an error. The quality of the input data, based on the 2 bit counter, is not sufficient (there is a simple statistical evaluation here, which evaluates both GOOD cases and bad cases and leads to an error if a special threshold value is exceeded).
4470	**17520**	Monitoring	**SSI transformation faulty or not finished** The SSI transformation of the FOX 50 module was faulty for some NC-cycles or did not finished respectively.
44A2	**17570**	Monitoring	**Internal error** ENCERR_ADDR_CONTROLLER
44A3	**17571**	Monitoring	**Internal error** ENCERR_INVALID_CONTROLLERTYPE

7.8.6 Controller Errors

Error(Hex)	Error(Dec)	Error type	Description	
4500	**17664**	Parameter	**Controller ID not allowed** The value for the controller ID is not allowed, e.g. because it has already been assigned, is less than or equal to zero, or is greater than 255.	
			Value range: [1 ... 255]	Unit: 1
4501	**17665**	Parameter	**Controller type not allowed** The value for the controller type is unacceptable because it is not defined. Type 1: P-controller (position) Type 7: Fast/creep controller Type 8: Stepper motor controller Type 9: Sercos controller	
			Value range: [1 ... 9]	Unit: 1
4502	**17666**	Parameter	**Operation mode controller not allowed** The value for the controller operating mode is not allowed.	
			Value range: [1]	Unit: 1
4503	**17667**	Parameter	**Weighting of the velocity pre-control not allowed** The value for the percentage weighting of the velocity pre-control is not allowed. The parameter is pre-set to 1.0 (100%) as standard.	
			Value range: [0.0 ... 1.0]	Unit: %
4504	**17668**	Parameter	**Lag error monitor (position) not allowed** The value for the activation of the lag error monitor is not allowed.	
			Value range: [0, 1]	Unit: 1
4505	**17669**	Parameter	**Lag monitoring (velocity) not allowed** The value for the activation of the lag error monitoring (velocity) is not allowed.	
			Value range: [0, 1]	Unit: 1
4506	**17670**	Parameter	**Lag error window (position) not allowed** The value for the lag error window (maximum allowable lag error) is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm
4507	**17671**	Parameter	**Lag error filter time (position) not allowed** The value for the lag error filter time (position) is not allowed.	
			Value range: [0.0, 600.0]	Unit: s

Error(Hex)	Error(Dec)	Error type	Description	
4508	**17672**	Parameter	**Lag error window (velocity) not allowed** The value for the lag error window (velocity) is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. m/min
4509	**17673**	Parameter	**Lag error filter time (velocity) not allowed** The value for the lag error filter time (velocity) is not allowed.	
			Value range: [0.0, 600.0]	Unit: s
450A	**17674**	Parameter	**Controller output limitation (output limitation) not allowed** The value for the output limitation of the controller at the total manipulated variable is invalid. The default setting is 0.5 (50 percent). Typically, this parameter is effective if the velocity interface has been parameterized for the drive unit and the `NC` executes the position control of the position on the controller.	
			Value range: [0.0, 1.0]	Unit: %
4510	**17680**	Parameter	**Proportional gain Kv or Kp (controller) not allowed** *Position* The value for the proportional gain (Kv factor or Kp factor) is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm/s/mm
4511	**17681**	Parameter	**Integral action time Tn (controller) not allowed** *Position* The value for the integral action time is not allowed (I part of the PID T1 controller).	
			Value range: [0.0, 60.0]	Unit: s
4512	**17682**	Parameter	**Rate time Tv (controller) not allowed** *Position* The value for the derivative action time is not allowed (D part of the PID T1 controller).	
			Value range: [0.0, 60.0]	Unit: s
4513	**17683**	Parameter	**Damping time Td (controller) not allowed** *Position* The value for the damping time is not allowed (D part of the PID T1 controller).	
			Value range: [0.0, 60.0]	Unit: s

Error(Hex)	Error(Dec)	Error type	Description	
4514	**17684**	Function	**Activation of the automatic offset calibration not allowed** Activation of the automatic offset calibration is only possible for certain types of controller (with no I component).	
4515	**17685**	Parameter	**Additional proportional gain Kv or Kp (controller) not allowed** *Position* The value for the second term of the proportional gain (Kv factor or Kp factor) is not allowed.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm/s/mm
4516	**17686**	Parameter	**Reference velocity for additional proportional gain Kv or Kp (controller) not allowed** *Position* The value for the reference velocity percentage data entry, to which the additional proportional gain is applied, is not allowed. The standard setting for the parameter is 0.5 (50%).	
			Value range: [0.0 ... 1.0]	Unit: %
4517	**17687**	Parameter	**Proportional gain Pa (proportion) not allowed** *Acceleration* The value for the proportional gain (Pa factor) is not allowed.	
			Value range: [0.0, 1000000.0]	Unit: s
4518	**17688**	Parameter	**Proportional gain Kv (controller) not allowed** *Velocity* The value for the proportional gain (Kv factor) is not allowed.	
			Value range: [0.0, 10000.0]	Unit: 1
4519	**17689**	Parameter	**Integral action time Tn (controller) not allowed** *Velocity* The value for the integral action time is not allowed (I part of the PID T1 controller).	
			Value range: [0.0, 60.0]	Unit: s
451A	**17690**	Parameter	**Reserved** Reserved, currently not used.	
451B	**17691**	Parameter	**Reserved** Reserved, currently not used.	

Error(Hex)	Error(Dec)	Error type	Description	
451C	**17692**	Parameter	**Velocity filter time not allowed** The parameter for the velocity filter time in seconds is invalid (`P-T1` filter). This filter can be used in the `NC` for filtering an actual velocity or a velocity difference (`Geschwindigkeitsfehler = Soll-Geschwindigkeit - Ist-Geschwindigkeit`) in special `NC` controllers (e.g. in the torque interface).	
			Value range: [0.0, 60.0]	Unit: s
451D	**17693**	Parameter	**Dead range not allowed** The value for the dead range (neutral zone) of the position error or the velocity error (control deviation) is not allowed (applies to more complex controllers with velocity or torque interface).	
			Value range: [0.0, 10000.0]	Unit: mm or mm/s
451F	**17695**	Parameter	**Proportional gain Kcp not allowed** The parameter for the proportional gain `Kcp` of the slave coupling difference control is invalid.	
			Value range: [0.0, 10000.0]	Unit: e.g. mm^2/mm
4520	**17696**	Parameter	**Rate time Tv (controller) not allowed** *Velocity* The value for the derivative action time is not allowed (D part of the PID T1 controller).	
			Value range: [0.0, 60.0]	Unit: s
4521	**17697**	Parameter	**Damping time Td (controller) not allowed** *Velocity* The value for the damping time is not allowed (D part of the PID T1 controller). Suggested value: 0.1 * Tv	
			Value range: [0.0.. 60.0]	Unit: s
4522	**17698**	Parameter	**Limitation of the I part not allowed** The parameter for limiting the `I`-part of a `PI`- or `PID`-controller is not allowed. This internal state variable can be limited in percentage (`1.0` corresponds to `100` percent).	
			Value range: [0.0 .. 1.0]	Unit: %

Error(Hex)	Error(Dec)	Error type	Description
4523	**17699**	Parameter	**Limitation of the D part not allowed** The parameter for limiting the `D`-part of a `PI`- or `PID`-controller is not allowed. This internal state variable can be limited in percentage (`1.0` corresponds to `100` percent). Value range: [0.0 .. 1.0] | Unit: %
4524	**17700**	Parameter	**Parameter 'Deactivation of the I part during travel' not allowed** The boolean parameter for switching off the `I`-part during active positioning is invalid. Value range: [0, 1] | Unit: 1
4525	**17701**	Parameter	**Parameter 'Filter time for P-T2 filter' not allowed** The time `T0` in seconds is not permissible as a filter time for the `P-T2` element of the velocity controller. The filter time must be less than twice the NC SEC cycle time. Value range: [0.0, 60.0] | Unit: s
4526	**17702**	Parameter	**Velocity observer: 'Parameterized mode' is not allowed** The parameterized mode (`0=OFF`, `1=LUENBERGER`) for the velocity observer of the special `NC` controller in the torque interface is not permitted. Value range: [0, 1] | Unit: 1
4527	**17703**	Parameter	**Velocity observer: 'Motor torque constant Kt or Kf' is not allowed** The parameter for the motor torque constant `Kt` (rotary motor) or `Kf` (linear motor) of the velocity observer of the special NC controller in the torque interface is invalid. Value range: [0.0 .. 100000.0] | Unit: Nm/A or N/A
4528	**17704**	Parameter	**Velocity observer: 'Motor moment of inertia JM' is not allowed** The parameter for the motor moment of inertia J_M of the velocity observer of the special NC controller in the torque interface is invalid. Value range: [0.0001 .. 100000.0] | Unit: kg cm^2

Error(Hex)	Error(Dec)	Error type	Description	
4529	**17705**	Parameter	**Velocity observer: 'Bandwidth f0' is not allowed** The parameter for the bandwidth f_0 of the velocity observer of the special NC controller in the torque interface is invalid. The bandwidth must be less than the reciprocal of 6 times the NC cycle time (f_0 < 1/(6*T)).	
			Value range: [0.0 .. 10000.0]	Unit: Hz
452A	**17706**	Parameter	**Velocity observer: 'Correction factor kc' is not allowed** The parameter for the correction factor k_c of the velocity observer of the special NC controller in the torque interface is invalid. The correction factor k_c establishes the relationship between current and acceleration or angular acceleration.	
			Value range: [0.0 .. 100.0]	Unit: s
452B	**17707**	Parameter	**Velocity observer: 'Time constant T for 1st order filter' is not allowed** The time constant `T` for the 1st order velocity filter (PID-T_2 or "Lead Lag") of the velocity observer of the special NC controller in the torque interface is not allowed. The correction factor k_c establishes the relationship between current and acceleration or angular acceleration.	
			Value range: [0.0 .. 100.0]	Unit: s
452C	**17708**	Parameter	**Velocity observer: 'Amplitude damping d for 2nd order filter' is not allowed** The high-pass/low-pass amplitude attenuation d_{HP} or d_{TP} for the 2nd order velocity filter ("bi-quad") of the velocity observer of the special NC controller in the torque interface is not allowed.	
			Value range: [0.2 .. 10.0]	Unit: 1
452D	**17709**	Parameter	**Velocity observer: 'Frequency fHP or fTP for 2nd order filter' is not allowed** The high-pass/low-pass frequency f_{HP} or f_{TP} for the 2nd order velocity filter ("bi-quad") of the velocity observer of the special NC controller in the torque interface is not allowed.	
			Value range: [0.0, ... 10000.0]	Unit: Hz

Error(Hex)	Error(Dec)	Error type	Description
4540	**17728**	Initialization	**Controller initialization** The controller has not been initialized. Although the controller has been created, the rest of the initialization has not been performed (1. Initialization of controller, 2. Reset of controller).
4541	**17729**	Address	**Axis address** The controller does not know its axis, or the axis address has not been initialized.
4542	**17730**	Address	**Drive address** The controller does not know its drive, or the drive address has not been initialized.
4550	**17744**	Monitoring	**Lag error monitor (position)** While the lag error monitor was active (position), the lag error was exceeded by an amount greater than the lag error window and which lasting longer than the parameterized lag error filter time.
4551	**17745**	Monitoring	**Lag monitoring (velocity)** With active lag error monitoring (velocity) a velocity lag error exceedance has occurred, whose magnitude is greater than the lag error window, and whose duration is longer than the parameterized lag error filter time.
45A0	**17824**	Monitoring	**Internal error** CONTROLERR_RANGE_AREA_ASIDE
45A1	**17825**	Monitoring	**Internal error** CONTROLERR_RANGE_AREA_BSIDE
45A2	**17826**	Monitoring	**Internal error** CONTROLERR_RANGE_QNENN
45A3	**17827**	Monitoring	**Internal error** CONTROLERR_RANGE_PNENN
45A4	**17828**	Monitoring	**Internal error** CONTROLERR_RANGE_AXISIDPRESP0

7.8.7 Drive error

Error(Hex)	Error(Dec)	Error type	Description	
4600	**17920**	Parameter	**Drive ID not allowed** The value for the drive ID is not allowed, e.g. because it has already been assigned, is less than or equal to zero, or is greater than 255.	
			Value range: [1 ... 255]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description	
4601	**17921**	Parameter	**Drive type not allowed**	
			The value for the drive type is impermissible, since it is not defined.	
			Value range: [1, 20]	Unit: 1
4602	**17922**	Parameter	**Operation mode drive not allowed**	
			The value for the drive operating mode is impermissible (mode 1: standard).	
			Value range: [1]	Unit: 1
4603	**17923**	Parameter	**Motor polarity inverted?**	
			The flag for the motor polarity is invalid. Flag 0: Positive motor polarity Flag 1: Negative motor polarity	
			Value range: [0, 1]	Unit: 1
4604	**17924**	Parameter	**Drift compensation/velocity offset (DAC offset)**	
			The value for the drift compensation (DAC offset) is impermissible.	
			Value range: [-100.0, 100.0]	Unit: e.g. m/min
4605	**17925**	Parameter	**Reference velocity (velocity pre-control)**	
			The value for the reference velocity (also called velocity pre-control) is impermissible.	
			Value range: [0.0, 10000.0]	Unit: e.g. m/min
4606	**17926**	Parameter	**Reference output in percent**	
			The value for the reference output in percent is impermissible. The value 1.0 (100 %) usually corresponds to a voltage of 10.0 V.	
			Value range: [0.0, 5.0]	Unit: %
4607	**17927**	Parameter	**Quadrant compensation factor**	
			The value for the quadrant compensation factor is impermissible.	
			Value range: [0.0, 100.0]	Unit: 1
4608	**17928**	Parameter	**Velocity reference point in percent**	
			The value for the velocity reference point in percent is impermissible. The value 1.0 corresponds to 100 percent.	
			Value range: [0.01, 1.0]	Unit: %
4609	**17929**	Parameter	**Output reference point**	
			The value for the output reference point in percent is impermissible. The value 1.0 corresponds to 100 percent.	
			Value range: [0.01, 1.0]	Unit: %
460A	**17930**	Parameter	**Minimum or maximum output limits (output limitation)**	
			The value for the minimum and/or maximum output limit is impermissible. This will happen if the value range is exceeded, the maximum limit is smaller than the minimum limit, or the distance between the minimum and maximum limits is zero. The minimum limit is initially set to –1.0 (-100 percent) and the maximum limit to 1.0 (100 percent).	
			Value range: [-1.0, 1.0]	Unit: %

Error(Hex)	Error(Dec)	Error type	Description	
460B	**17931**	Parameter	**Parameter 'Maximum value for output' is not allowed** The value for the maximum number of output digits of the drive (maximum output value) is not allowed. Depending on the interface used (e.g. position, velocity or torque/current). For a velocity interface this is often a signed 16 bit output value (± 32767).	
			Value range: [0x000000FF .. 0xFFFFFFFF]	Unit: INC or Digits
460C	**17932**	Parameter	**Parameter 'Internal Drive Control Word' is not allowed** The value as internal Drive Control Word for the NC is not allowed. This contains information from the System Manager to the NC, which is evaluated by the NC at the TC start.	
			Value range: [>0]	Unit: 1
460D	**17933**	Parameter	**Parameter 'Internal timer for RESET behavior Drive' is not allowed** The special parameter that influences the internal time behavior between NC Drive and the I/O Drive (servo drive) is not allowed.	
			Value range: [>5]	Unit: 1 (NC SEC cycles)
460E	**17934**	Parameter	**Parameter 'Master Motion Controller ID' is not allowed** The "Master Motion Controller ID" parameter is not allowed for a further NC Motion Controller in slave mode. An additional NC Motion Controller in slave mode can be used if it is one and the same drive device to which different NC information for different operation modes is connected (e.g. velocity mode and torque mode). **This parameter is not directly accessible by the user, but can only be influenced indirectly by configuring additional NC motion controllers underneath the NC axis.**	
			Value range: [0 .. 255]	Unit: 1
460F	**17935**	Parameter	**Drive torque output scaling is not allowed** The value is shown as Drive torque output scaling (rotary motor) or as force output scaling (linear motor).	
			Value range: [0, 1000000]	Unit: 1
4610	**17936**	Parameter	**Drive velocity output scaling is not allowed** The value for the drive velocity output scaling is not allowed.	
			Value range: [0, 1000000]	Unit: 1
4611	**17937**	Parameter	**Profi Drive DSC proportional gain Kpc (controller) not allowed** *Position* The value for the Profi Drive DSC position control gain (Kpc factor) is impermissible.	
			Value range: [0, 0xFFFFFF]	Unit: 0.001 * 1/s
4612	**17938**	Parameter	**Table ID not allowed** The value for the table ID is impermissible.	
			Value range: [0, 255]	Unit: 1
4613	**17939**	Parameter	**Table interpolation type is not allowed** The value is impermissible as the table interpolation type.	
			Value range: 0 (LINEAR), 2 (SPLINE)	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description	
4614	**17940**	Parameter	**Output offset in percent is not allowed** The value is impermissible as an output offset in percent (+/- 1.0).	
			Value range: [-1.0, 1.0]	Unit: %
4615	**17941**	Parameter	**Profi Drive DSC scaling for calculation of 'Xerr' (controller) not allowed** *Position* The value is impermissible as Profi Drive DSC scaling for the calculation of 'Xerr'.	
			Value range: [0, 1000000]	Unit: 1
4616	**17942**	Parameter	**Drive acceleration output scaling not allowed** The value is impermissible as drive acceleration/ deceleration output scaling.	
			Value range: [0, 1000000]	Unit: 1
4617	**17943**	Parameter	**Drive position output scaling not allowed** The value is impermissible as drive position output scaling.	
			Value range: [0, 1000000]	Unit: 1
4618	**17944**	Parameter	**Parameter 'Dead time compensation mode' (Motion Controller) is invalid** The parameter for the dead time compensation mode of the NC Motion Controller is invalid (OFF, ON with velocity, ON with velocity and acceleration).	
			Value range: [0, 1, 2]	Unit: 1
4619	**17945**	Parameter	**Parameter 'Control bits of the dead time compensation' (Motion Controller) is invalid** The parameter for the "Control bits of the dead time compensation" of the NC Motion Controller is invalid (e.g. relative or absolute time interpretation).	
			Value range: [>0]	Unit: 1
461A	**17946**	Parameter	**Parameter 'time shift of dead time compensation mode' (Motion Controller) is invalid** The parameter for the time shift of the dead time compensation (Time Shift in nanoseconds) of the NC Motion Controller is invalid.	
			Value range: [-1.0E9 .. 1.0E9]	Unit: ns
461B	**17947**	Parameter	**Parameter 'Output delay velocity interface Motion Controller' is invalid** The parameter for an optional output delay in the velocity interface to the Motion Controller is invalid (Delay Generator Velocity). The maximum permitted delay time must be less than 100 times the NC SEC cycle time.	
			Value range: [0.0 .. 0.1]	Unit: s
461C	**17948**	Parameter	**Drive filter type not allowed for command variable filter for the output position** The value is impermissible as a drive filter type for the smoothing of the output position (command variable filter for the setpoint position).	
			Value range: [0, 2]	Unit: 1
461D	**17949**	Parameter	**Drive filter time not allowed for command variable filter for the output position** The value is impermissible as a drive filter time for the smoothing of the output position (command variable filter for the setpoint position).	
			Value range: [0.0, 1.0]	Unit: s

Error(Hex)	Error(Dec)	Error type	Description	
461E	**17950**	Parameter	**Drive filter order not allowed for command variable filter for the output position** The value is impermissible as a drive filter order (P-Tn) for the smoothing of the output position (command variable filter for the setpoint position).	
			Value range: [0, 10]	Unit: 1
4620	**17952**	Parameter	**Bit mask for stepper motor cycle not allowed** A value of the different stepper motor masks is impermissible for the respective cycle.	
			Value range: [0, 255]	Unit: 1
4621	**17953**	Parameter	**Bit mask for stepper motor holding current not allowed** The value for the stepper motor holding mask is impermissible.	
			Value range: [0, 255]	Unit: 1
4622	**17954**	Parameter	**Scaling factor for actual torque (actual current) not allowed** The value is impermissible as a scaling factor for the actual torque (or actual current).	
			Value range: [0, 1E+30]	Unit:
4623	**17955**	Parameter	**Filter time for actual torque is not allowed** The value is impermissible as a filter time for the actual torque (or the actual current) (P-T1 filter).	
			Value range: [0.0, 60.0]	Unit: s
4624	**17956**	Parameter	**Filter time for the time derivative of the actual torque is not allowed** The value as filter time for the time derivative of the actual torque (or the actual current) is not allowed (P-T1 filter).	
			Value range: [0.0, 60.0]	Unit: s
4625	**17957**	Parameter	**Parameter "Drive operation mode" is invalid** The parameter for the drive operation mode (motion controller operation mode: position mode, velocity mode, torque mode, ...) is invalid. It is possible that an NC operation mode changeover has been attempted or that an attempt was made to activate a preconfigured operation mode during the TC system startup. **The generic operation modes defined in the NC are implemented by the NC in a drive-specific manner, i.e. in particular for the SERCOS/ SoE and CANopen/ CoE (DS402) protocols.** Here, protocol-specific, drive-specific or even vendor-specific features must be taken into account (e.g. with SERCOS/ SoE, predefined operation modes can only be activated at runtime in the SERCOS parameters S-0-0032 to S-0-0035). Furthermore, not every generic NC operation mode can be converted into a drive-specific operation mode (there may be gaps in the specification here). The generic NC operation mode 0 is a special case. This value is used as an identifier to activate an NC default mode (if this identifier is known to the NC).	
			Value range: [0, >=1]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description
4626	**17958**	Monitoring	**Motion Controller function is not supported** A Motion Controller functionality has been triggered that is not enabled for use or is not implemented (e.g. writing or reading a drive operation mode that is not supported by certain Motion Controllers). It is also possible that this functionality is only temporarily unavailable (e.g. because the drive device is in error state or a drive enable is missing).
4627	**17959**	Function	**DRIVEOPERATIONMODEBUSY** The activation of the drive operation mode failed, because another object with `OID`… is already using this interface.
4628	**17960**	Monitoring	**Drive operation mode changeover is not configured or the desired drive operation mode cannot be found** No drive operation mode changeover has been configured, and in this respect no reading or writing of a drive operation mode is possible. Or the desired drive operation mode has not been found in the list of predefined drive operation modes (e.g. for SoE/ SERCOS). Note for CoE Motion Controllers: Reading or writing the CoE Motion Controller operation mode is only possible if the CoE objects 0x6060 "Modes of operation" and 0x6061 "Modes of operation display" are in the cyclic process data (PDO list) and a valid default operation mode has been configured. Note for SoE Motion Controllers: Reading or writing the current SoE Motion Controller operation mode is only possible if this operation mode has been predefined in one of the SoE parameters S-0-0032 to S-0-0035.
4629	**17961**	Monitoring	**Feedback drive operation mode changeover** During drive operation mode changeover, the requested operation mode was not consistently reported back within the monitoring time of 8 cycles. **CoE-Motion Controllers:** Reading or writing the CoE Motion Controller operation mode is only possible if the CoE objects 0x6060 "Modes of operation" and 0x6061 "Modes of operation display" are in the cyclic process data (PDO list) and a valid default operation mode has been configured. **SoE-Motion Controllers:** Reading or writing the current SoE Motion Controller operation mode is only possible if this operation mode has been predefined in one of the SoE parameters S-0-0032 to S-0-0035.
0x4630 … 0x463F: Error codes are reserved for external drive errors (e.g. stepper motor terminal or MC_PowerStepper function block).			
4630	**17968**	Monitoring	**Overtemperature** Overtemperature was detected or reported in the drive or terminal.
4631	**17969**	Monitoring	**Undervoltage** Undervoltage was detected or reported in the drive or terminal.
4632	**17970**	Monitoring	**Wire break in phase A** A wire break in phase A was detected or reported in the drive or terminal.
4633	**17971**	Monitoring	**Wire break in phase B** A wire break in phase B was detected or reported in the drive or terminal.

Error(Hex)	Error(Dec)	Error type	Description
4634	**17972**	Monitoring	**Overcurrent in phase A** Overcurrent was detected or reported in phase A in the drive or terminal.
4635	**17973**	Monitoring	**Overcurrent in phase B** Overcurrent was detected or reported in phase B in the drive or terminal.
4636	**17974**	Monitoring	**Torque overload (stall)** A torque overload (stall) was detected or reported in the drive or terminal.
4640	**17984**	Initialization	**Drive initialization** Drive has not been initialized. Although the drive has been created, the rest of the initialization has not been performed: 1. Initialization of drive I/O, 2. Initialization of drive, 3. Reset of drive.
4641	**17985**	Address	**Axis address** The drive does not know its axis, or the axis address has not been initialized.
4642	**17986**	Address	**Address I/O input structure** The drive has no valid I/O input address in the process image.
4643	**17987**	Address	**Address I/O output structure** The drive has no valid I/O output address in the process image.
4650	**18000**	Monitoring	**Drive hardware not ready to operate** The drive hardware is not ready for operation. This can be caused by the following reasons: - the drive is in error state (hardware error) - the drive is in the start-up phase (e.g. after an axis reset preceded by a hardware error) - the drive lacks the controller enable (ENABLE) **The time required for the "start-up" of a drive after a hardware error can be in the range of several seconds.**
4651	**18001**	Monitoring	**Error in the cyclic communication of the drive (Life Counter)** Reasons for this could be an interrupted fieldbus or a drive that is in the error state.
4652	**18002**	Monitoring	**Changing the table ID with active controller enable not allowed** Changing (deselecting, selecting) the characteristic curve table ID is not permissible when the controller enable for the axis is active.
4655	**18005**	Monitoring	**Invalid I/O data for more than 'n' continuous NC cycles** The axis (encoder or drive) has detected invalid I/O data for more than 'n' continuous NC cycles (NC SEC task) (e.g. n=3). As a consequence it is possible that the encoder referencing flag is reset to FALSE (i.e. the encoder gets the state "unreferenced").EtherCAT fieldbus: "working counter error ('WCState')" Lightbus fieldbus: "CDL state error ('CdlState')"

7.8.8 Table Errors

Error(Hex)	Error(Dec)	Error type	Description	
4A00	**18944**	Parameter	**Table ID not allowed**	
			The value for the table ID is not allowed, e.g. because it has already been assigned, is less than or equal to zero, or is greater than 255.	
			Value range: [1 ... 255]	Unit: 1
4A01	**18945**	Parameter	**Table type not allowed**	
			The value for the table type is unacceptable because it is not defined.	
			Value range: [1]	Unit: 1
4A02	**18946**	Parameter	**Number of lines in the table not allowed**	
			The value of the number of lines in the table is not allowed, because, for example, it is smaller than two at linear interpolation and smaller than four at spline interpolation.	
			Value range: [2, 0xFFFF]	Unit: 1
4A03	**18947**	Parameter	**Number of columns in the table is not allowed**	
			The value of the number of columns in the table is not allowed, because, for example, it is less than or equal to zero (depends upon the type of table or slave).	
			Value range: [1, 0xFFFF]	Unit: 1
4A04	**18948**	Parameter	**Step size (position delta) not allowed**	
			The value for the step size between two lines (position delta) is not allowed, because, for example, it is less than or equal to zero.	
			Value range: [0.001, 1.0E+6]	Unit: e.g. mm
4A05	**18949**	Parameter	**Period not allowed**	
			The value for the period is not allowed, because, for example, it is less than or equal to zero.	
			Value range: [0.001, 1.0E+9]	Unit: e.g. mm
4A06	**18950**	Parameter	**Table is not monotonic**	
			The value for the step size is not allowed, because, for example, it is less than or equal to zero.	
4A07	**18951**	Initialization	**Table sub type not allowed**	
			The value for the table sub-type is unacceptable because it is not defined. Or the table sub-type and the table class (slave type) do not match. Table sub-types: (1) equidistant linear position table, (2) equidistant cyclic position table, (3) non-equidistant linear position table, (4) non-equidistant cyclic position table	
			Value range: [1, 4]	Unit: 1
4A08	**18952**	Initialization	**Table interpolation type not allowed**	
			The value for the table interpolation type is invalid because it is not defined. Table interpolation types: (0) linear interpolation, (1) 4-point interpolation, (2) spline interpolation	
			Value range: [0, 2]	Unit: 1

Error(Hex)	Error(Dec)	Error type	Description
4A09	**18953**	Initialization	**Incorrect table main type** The table main type is not allowed because it is not defined. Or the table main type and the table class (slave type) do not match. Table main types: (1) cam plate table (camming), (10) characteristic table, (16) 'motion function' table (MF)
4A10	**18960**	Initialization	**Table initialization** The table has not been initialized. Although the table has been created, the rest of the initialization has not been performed. For instance, the number of lines or columns may be less than or equal to zero.
4A11	**18961**	Initialization	**Not enough memory** The table could not be created, since there is not enough memory.
4A12	**18962**	Function	**Function not executed, function not available** The function has not been implemented, or cannot be executed, for the present type of table.
4A13	**18963**	Function	**Line index not allowed** The start line index or the stop line index to be used for read or write access to the table is not allowed. For instance, the line index may be greater than the total number of lines in the table.
4A14	**18964**	Function	**Column index not allowed** The start column index or the stop column index to be used for read or write access to the table in not allowed. For instance, the column index may be greater than the total number of columns in the table.
4A15	**18965**	Function	**Number of lines not allowed** The number of lines to be read from or written to the table is not allowed. The number of lines must be an integer multiple of the number of elements in a line (n * number of columns).
4A16	**18966**	Function	**Number of columns not allowed** The number of columns to be read from or written to the table is not allowed. The number of columns must be an integer multiple of the number of elements in a column (n * number of lines).
4A17	**18967**	Function	**Error in scaling or in range entry** The entries in the table header are inconsistent, e.g. the scope is empty. If the error is generated during the runtime it is a runtime error and stops the master/slave group.
4A18	**18968**	Function	**Multi table slave out of range** The slave master position is outside the table values for the master. The error is a runtime error, and stops the master/slave group.
4A19	**18969**	Function	**Solo table underflow** The slave master position is outside the table values for the master. The master value of the equidistant table, to be processed linearly, lies under the first table value. The error is a runtime error, and stops the master/slave group.

Error(Hex)	Error(Dec)	Error type	Description
4A1A	**18970**	Function	**Solo table overflow** The slave master position is outside the table values for the master. The master value of the equidistant table, to be processed linearly, lies above the first table value. The error is a runtime error, and stops the master/slave group.
4A1B	**18971**	Parameter	**Incorrect execution mode** The cyclic execution mode can only be "true" or "false".
4A1C	**18972**	Parameter	**Invalid parameter** The Fifo parameter is not allowed.
4A1D	**18973**	Parameter	**Fifo is empty** The Fifo of the external generator is empty. This can signify end of track or a runtime error.
4A1E	**18974**	Parameter	**Fifo is full** The Fifo of the external generator is full. It is the user's task to continue to attempt to fill the Fifo with the rejected values.
4A1F	**18975**	Parameter	**Point-Index of Motion Function invalid** The point index of a Motion Function Point of a Function Table is invalid. First, the point index must firstly be greater than zero and second, it must be numerically consecutive for a column of a Motion Function table (e.g. 1,2,3,... or 10,11,12,...). Note: The point index must not be changed online, but must be kept constant.
4A20	**18976**	Initialization	**No diagonalization of matrix** The spline cannot be calculated. The master positions are not correct.
4A21	**18977**	Initialization	**Number of spline points to small** The number of points of a cubic spline must be at least three. Therefore, the number of lines must be at least three.
4A22	**18978**	Initialization	**Fifo must not be overwritten** The Fifo of the external generator must not be overwritten, otherwise it would be written over the active processing line. It is up to the user to make sure that no changes or deletions are requested across the active line.
4A23	**18979**	Function	**Motion Function has too few points** The number of valid points defining a Motion Function is less than two. Either the total number is too low or the point type of many points is set to *Ignore Point*.
4A25	**18981**	Initialization	**Master start position of the table is invalid** For a periodic position table, it is mandatory that the master position of the table starts at zero. For a periodic motion function, the first master position must be greater than zero but not greater than the period of the table.

7.8.9 NC-PLC Errors

Error(Hex)	Error(dec)	Error type	Description
4B00	**19200**	Parameter	**Axis was stopped** The axis was stopped during travel to the target position. The axis may have been stopped with a PLC command via ADS, a call via AXFNC, or by the System Manager.
4B01	**19201**	Parameter	**The axis cannot be started**

Error(Hex)	Error(dec)	Error type	Description
			The axis cannot be started because: • the axis is in error state, • the axis is executing another command, • the axis is in protected mode, • the axis is not ready for operation.
4B02	**19202**	Parameter	**Control mode not permitted** No target position control, and no position area control.
4B03	**19203**	Parameter	**Axis is not moving** The position and velocity can only be restarted while the axis is physically in motion.
4B04	**19204**	Parameter	**Invalid mode** Examples: Invalid `Direction` at `MC_MoveModulo`. Inactive axis parameter *Position correction* with `MC_BacklashCompensation`.
4B05	**19205**	Parameter	**Command not permitted** • Continuous motion in an unspecified direction • Read/Write parameter: unsuitable type
4B06	**19206**	Parameter	**Parameter is not correct** • Incorrect override: > 100% or < 0% • Incorrect gear ratio: RatioDenominator = 0
4B07	**19207**	Parameter	**Timeout axis function block** After positioning all "MC_Move..."function blocks, check whether positioning was completed successfully. In the simplest case, the "AxisHasJob" flag of the NC axis is checked, which initially signifies that positioning was logically completed. Depending on the parameterization of the NC axis, further checks (quality criteria) are used: • "Position range monitoring" If position range monitoring is active, the system waits for feedback from the NC. After positioning, the axis must be within the specified positioning range window. If necessary, the position controller ensures that the axis is moved to the target position. If the position controller is switched off (Kv=0) or weak, the target may not be reached. • "Target position monitoring" If target position monitoring is active, the system waits for feedback from the NC. After positioning, the axis must be within the specified target position window for at least the specified time. If necessary, the position controller ensures that the axis is moved to the target position. If the position controller is switched off (Kv=0) or weak, the target may not be reached. Floating position control may lead to the axis oscillating around the window but not remaining inside the window. If the axis is logically at the target position (logical standstill) but the parameterized position window has not been reached, monitoring of the above-mentioned NC feedback is aborted with error 19207 (0x4B07) after a constant timeout of 6 seconds.
4B08	**19208**	Parameter	**Axis is in protected mode**

Error(Hex)	Error(dec)	Error type	Description
			The axis is in protected mode (e.g. coupled) and cannot be moved.
4B09	**19209**	Parameter	**Axis is not ready** The axis is not ready and cannot be moved.
4B0A	**19210**	Parameter	**Error during referencing** Referencing (homing) of the axis could not be started or was not successful.
4B0B	**19211**	Parameter	**Incorrect definition of the trigger input** The definition of the trigger signal for function block MC_TouchProbe is incorrect. The defined encoder-ID, the trigger signal or the trigger edge are invalid.
4B0C	**19212**	Function	**Position latch was disabled** The function block MC_TouchProbe has detected that a touch probe cycle it had started was disabled. The reason may be an axis reset, for example.
4B0D	**19213**	Function	**NC state feedback timeout** A function was successfully sent from the PLC to the NC. An expected feedback in the axis status word has not arrived.
4B0E	**19214**	Function	**Additional product not installed** The function is available as an supplement but is not installed on the system.
4B0F	**19215**	Function	**No NC Cycle Counter Update** The NcToPlc Interface or the NC Cycle Counter in the NcToPlc Interface was not updated.
Error numbers 0x4B10 .. 0x4B2F are used in ***TwinCAT NCI*** **context:**			
4B10	**19216**	Function	**M-function query missing** This error occurs if the M-function was confirmed, but the request bit was not set.
4B11	**19217**	Parameter	**Zero shift index is outside the range** The index of the zero offset shift is invalid.
4B12	**19218**	Parameter	**R parameter index or size is invalid** This error occurs if the R parameters are written or read but the index or size are outside the range.
4B13	**19219**	Parameter	**Index for tool description is invalid** The index for the tool description is invalid.
4B14	**19220**	Function	**Version of the cyclic channel interface does not match the requested function or the function block** This error occurs if an older TwinCAT version is used to call new functions of a later TcNci.lib version.
4B15	**19221**	Function	**Channel is not ready for the requested function** The requested function cannot be executed, because the channel is in the wrong state. This error occurs during reverse travel, for example, if the axis was not stopped with ItpEStop first.
4B16	**19222**	Function	**Requested function is not activated** The requested function requires explicit activation.
4B17	**19223**	Function	**Axis is already in another group** The axis has already been added to another group.
4B18	**19224**	Function	**Block search could not be executed successfully** The block search has failed.

Error(Hex)	Error(dec)	Error type	Description
			Possible causes: • Invalid block number
4B19	**19225**	Parameter	**Invalid block search parameter** This error occurs if the FB ItpBlocksearch is called with invalid parameters (e.g. E_ItpDryRunMode, E_ItpBlockSearchMode)
4B20	**19232**	Function	**Cannot add all axes** This error occurs if an auxiliary axis is to be added to an interpolation group, but the function fails. It is likely that a preceding instruction of an auxiliary axis was skipped.
Error numbers 0x4B30 .. 0x4B3F are used in the *TcMcCam-Lib* (MC_NC_TableErrorCodes):			
4B30	**19248**	Parameter	**Pointer is invalid** A pointer to a data structure is invalid, e.g. Null • Data structure MC_CAM_REF was not initialized
4B31	**19249**	Parameter	**Memory size invalid** The specification of the memory size (SIZE) for a data structure is invalid. • Memory size is 0 or smaller than an element of the addressed data structure. • Memory size is smaller than the requested amount of data. • Memory size does not match other parameters, such as number of points, number of rows or number of columns.
4B32	**19250**	Parameter	**Cam plate ID is invalid** The ID of a cam plate is not between 1 and 255.
4B33	**19251**	Parameter	**Point ID is invalid** The ID of a point (interpolation point) of a motion function is less than 1.
4B34	**19252**	Parameter	**Number of points is invalid** The number of points (interpolation points) of a cam plate to be read or written is less than 1.
4B35	**19253**	Parameter	**MC table type is invalid** The type of a cam plate does not match the definition *MC_TableType.*
4B36	**19254**	Parameter	**Number of rows invalid** The number of rows (interpolation points) of a cam plate is less than 1.
4B37	**19255**	Parameter	**Number of columns invalid** The number of columns of a cam plate is invalid. • The number of columns of a motion function is not equal 1 • The number of columns of a standard cam plate is not equal 2 • The number of columns does not match another parameter (ValueSelectMask)
4B38	**19256**	Parameter	**Step size invalid** The step size for the interpolation is invalid, e.g. less than or equal to zero.
Error numbers 0x4B0F, 0x4B40 .. 0x4B4F are used in several libraries (*TcNc-Lib / Tc2_MC2_XFC-Lib***):**			
4B40	**19264**	Monitoring	**Terminal type not supported**

Error(Hex)	Error(dec)	Error type	Description
			The terminal used is not supported by this function block.
4B41	**19265**	Monitoring	**Register read/write error** This error implies a validity error.
4B42	**19266**	Monitoring	**Axis is enabled** The axis is enabled but should not be enabled for this process.
4B43	**19267**	Parameter	**Incorrect size of the compensation table** The specified table size (in bytes) does not match the actual size
4B44	**19268**	Parameter	**Positional deviation** The minimum/maximum position in the compensation table does not match the position in the table description (ST_CompensationDesc)
4B45	**19269**	Parameter	**Not implemented** The requested function is not implemented in this combination
4B46	**19270**	Parameter	**Window not in the specified modulo range** The parameterized min or max position is not in the specified modulo range
4B47	**19271**	Monitoring	**Buffer overflow** The number of events has led to an overflow of the buffer and not all events could be acquired.
Error numbers 0x4B50 .. 0x4B5F are used in the *TcMcCam-Lib* **:**			
Error numbers 0x4B60 .. 0x4B6F are used in the *TcMc2-Lib* **in the** *buffered commands* **context:**			
4B60	**19296**	Monitoring	**Motion command did not become active** A motion command has been started and has been buffered and confirmed by the NC. Nevertheless, the motion command did not become active (possibly due to a termination condition or an internal NC error).
4B61	**19297**	Monitoring	**Motion command could not be monitored by the PLC** A motion command has been started and has been buffered and confirmed by the NC. The PLC has not been able to monitor the execution of this command and the execution status is unclear since the NC is already executing a more recent command. The execution state is unclear. This error may come up with very short buffered motion commands which are executed during one PLC cycle.
4B62	**19298**	Monitoring	**Buffered command was terminated with an error** A buffered command was terminated with an error. The error number is not available, because a new command is already being executed.
4B63	**19299**	Monitoring	**Buffered command was completed without feedback** A buffered command was completed but there was no feedback to indicate success or failure.
4B64	**19300**	Monitoring	**'BufferMode' is not supported by the command** The 'BufferMode' is not supported by this command.
4B65	**19301**	Monitoring	**Command number is zero** The command number for queued commands managed by the system unexpectedly has the value 0.
4B66	**19302**	Monitoring	**Function block was not called cyclically**

Error(Hex)	Error(dec)	Error type	Description
			The function block was not called cyclically. The command execution could not be monitored by the PLC, because the NC was already executing a subsequent command. The execution state is unclear.
Error numbers 0x4B70 .. 0x4B8F are used in the *TcPlcInterpolation-Lib* **:**			
4B71	**19313**	Parameter	**Invalid NCI entry type** The FB FB_NciFeedTablePreparation was called with an unknown nEntryType.
4B72	**19314**	Function	**NCI feed table full** The table is full and therefore the entry is not accepted. Remedy: Transfer the contents to the NC kernel with the FB FB_NciFeedTable. If bFeedingDone = TRUE, the table can be reset in FB_NciFeedTablePreparation with bResetTable and then filled with new entries.
4B73	**19315**	Function	**Internal error**
4B74	**19316**	Parameter	**ST_NciTangentialFollowingDesc: Tangential axis is not an auxiliary axis** The entry for tangential following contains a tangential axis that is not an auxiliary axis.
4B75	**19317**	Parameter	**ST_NciTangentialFollowingDesc** nPathAxis1 or nPathAxis2 is not a path axis. It is therefore not possible to determine the plane.
4B76	**19318**	Parameter	**ST_NciTangentialFollwoingDesc** nPathAxis1 and nPathAxis2 are the same. It is therefore not possible to determine the plane.
4B77	**19319**	Parameter	**ST_NciGeoCirclePlane** Circle incorrectly parameterized
4B78	**19320**	Function	**Internal error** An internal error has occurred in the calculation of the tangential following.
4B79	**19321**	Monitoring	**Tangential following** Monitoring of the deviation angle was activated during activation of tangential following (E_TfErrorOnCritical1), and an excessively large deviation angle was detected in the current segment.
4B7A	**19322**	Function	**Reserved** Reserved, currently not used.
4B7B	**19323**	Parameter	**Tangential following** the radius of the current arc is too small
4B7C	**19324**	Parameter	**FB_NciFeedTablePreparation** pEntry is NULL
4B7D	**19325**	Parameter	**FB_NciFeedTablePreparation** the specified nEntryType does not match the structure type
4B7E	**19326**	Parameter	**ST_NciMFuncFast and ST_NciMFuncHsk** The requested M-function is not between 0 and 159
4B7F	**19327**	Parameter	**ST_NciDynOvr** The requested value for the dynamic override is not between 0.01 and 1

Error(Hex)	Error(dec)	Error type	Description
4B80	**19328**	Parameter	**ST_NciVertexSmoothing** Invalid parameter. This error is generated if a negative smoothing radius or an unknown smoothing type is encountered.
4B81	**19329**	Parameter	**FB_NciFeedTablePrepartion** The requested velocity is not in the valid range
4B82	**19330**	Parameter	**ST_Nci*** invalid parameter
Error numbers 0x4B90 .. 0x4B9F are used in the *Tc3_MC2_AdvancedHoming-Lib* **(PLCopen Part 5: Homing Procedures):**			
4B90	**19344**	Parameter	**Drive type** The determined drive type is not supported.
4B91	**19345**	Parameter	**Direction** The direction is impermissible.
4B92	**19346**		**SwitchMode** The SwitchMode is impermissible.
4B93	**19347**		**Mode** The mode for the parameter handling is impermissible.
4B94	**19348**		**Torque limits** The parameterization of the torque limits is inconsistent.
4B95	**19349**		**Lag error limit** The parameterization of the position lag limit is impermissible (<=0).
4B96	**19350**		**Distance limit** The parameterization of the distance limit is impermissible (<0).
4B97	**19351**		**Saving parameters** An attempt was made to back up parameters again, although they have already been backed up.
4B98	**19352**		**Restoring parameters** An attempt was made to restore parameters, although none have been backed up.
4B9F	**19359**		**Cancellation of a homing** The abortion of a homing has failed.
Error numbers 0x4BA0 .. 0x4BAF are used in the *TcNcKinematicTransformation-Lib* **:**			
4BA0	**19360**	Function	**KinGroup error** The kinematic group is in an error state. This error may occur if the kinematic group is in an error state or an unexpected state when it is called (e.g., simultaneous call via several FB instances).
4BA1	**19361**	Function	**KinGroup timeout** Timeout during call of a kinematic block
Error numbers 0x4BB0 .. 0x4BBF are used in the *Tc2_MC2_Drive-Lib* **:**			
4BB0	**19376**	Function	**Invalid axis position** The current axis position or the axis position resulting from the new position offset exceeds the valid range of values.
4BB1	**19377**	Function	**Invalid position offset** The new position offset exceeds the valid range of values [AX5000: 2^31].

Error(Hex)	Error(dec)	Error type	Description
4BB2	**19378**	Function	**Invalid axis position** The current axis position or the axis position resulting from the new position offset falls below the valid range of values.
4BB3	**19379**	Function	**Invalid position offset** The new position offset falls below the valid range of values [AX5000: -2^31].
4BB4	**19380**	Function	**Deviation of the activated feedback and/or storage location** The activated feedback and/or storage location (AX5000: P-0-0275) differ from the parameterization on the function block.
4BB5	**19381**	Function	**Reinitialization of the NC actual position has failed** The reinitialization of the actual NC position has failed, e.g., reference system = "ABSOLUTE (with single overflow)" & software end position monitoring is disabled.
4BB6	**19382**	Function	**The setting or deletion of a position offset was rejected** The command to set or delete a position offset was rejected without feedback data, e.g., if the drive controller's firmware does not support the corresponding command.
4BB7	**19383**	Function	**The setting or deletion of a position offset was rejected** The command to set or delete a position offset was rejected with feedback data. The information in the feedback data may contain further clues to the cause, e.g. if the firmware of the drive controller does not support the corresponding command.
4BB8	**19384**	Function	**Invalid firmware version** A firmware version >= 19 is required for the servo terminal.
4BB9	**19385**	Function	**Deviating modulo settings** The modulo settings on the drive controller and in the NC are different.
4BBA	**19386**	Function	**Brake test failed** The brake test has failed.
4BBB	**19387**	Function	**Not supported drive type** The determined drive or axis type is not supported.
4BBC	**19388**	Function	**Command was aborted** The command was aborted by another command.
4BBD	**19389**	Function	**TimeOut** The command was aborted due to timeout.
Error numbers 0x4BC0 .. 0x4BCF are used in the *Tc3_DriveMotionControl-Lib* **:**			
4BC2	**19394**		**Invalid position offset** The new position offset exceeds the valid value range.
4BC3	**19395**		**Invalid I/O data** I/O data are invalid or the terminal is in an error state.
4BC4	**19396**		**ADS port not linked in the interface** The ADS port variable of the terminal was not linked to the axis interface of the PLC and parameters of the terminal are to be changed.

7.8.10 Kinematic Transformation

Error(Hex)	Error(Dec)	Error type	Description
4C00	**19456**		**Transformation failed** The calculation of the transformation failed
4C01	**19457**		**Ambiguous solution** The solution for the transformation is not unique.
4C02	**19458**		**Invalid axis position** The transformation cannot be calculated with the current position data. Possible causes: • The position is outside the working area of the kinematics
4C03	**19459**	Configuration	**Invalid dimension** The dimension of the parameterized input parameter does not match the dimension expected by the kinematic object. Possible causes: • Too many position values are supplied for this configuration. Check the number of parameterized axes.
4C04	**19460**		**Internal error** NCERR_KINTRAFO_REGISTRATION
4C05	**19461**	Internal	**Newton iteration failed** The Newton iteration does not converge.
4C06	**19462**	Internal	**Jacobi matrix cannot be inverted** The Jacobi matrix cannot be inverted.
4C07	**19463**	Configuration	**Invalid cascade** This kinematic configuration is not permitted.
4C08	**19464**	Programming	**Singularity** The machine configuration results in singular axis velocities.
4C0B	**19467**	Internal	**No metainfo** The metainfo pointer is null.
4C13	**19475**	Internal	**NCERR_RBTFRAME_INVALIDWCSTOMCS** The `WcsToMcs` component used leads to positions that the selected kinematics cannot assume. It is necessary to adjust the `WcsToMcs` parameters.
4C20	**19488**	Internal	**Transformation failed** Call of extended kinematic model failed.
4C30	**19504**	Programming	**Invalid input frame** Programmed Cartesian position cannot be reached in the ACS configuration.
4C50	**19536**	Internal	**Invalid offset** Access violation detected in the observer.

7.8.11 Bode Return Codes

The following bode plot specific error codes are used in the bode plot server:

Code Hex	Code Dec	Symbol	Description
0x8100	33024	INTERNAL	Internal error
0x8101	33025	NOTINITIALIZED	Not initialized (e.g. no nc axis)
0x8102	33026	INVALIDPARAM	Invalid parameter
0x8103	33027	INVALIDOFFSET	Invalid index offset
0x8104	33028	INVALIDSIZE	Invalid parameter size
0x8105	33029	INVALIDSTARTPARAM	Invalid start parameter (set point generator)
0x8106	33030	NOTSUPPORTED	Not supported
0x8107	33031	AXISNOTENABLED	Nc axis not enabled
0x8108	33032	AXISINERRORSTATE	Nc axis in error state
0x8109	33033	DRIVEINERRORSTATE	IO drive in error state
0x810A	33034	AXISANDDRIVEINERROR-STATE	Nc axis AND IO drive in error state
0x810B	33035	INVALIDDRIVEOPMODE	Invalid drive operation mode active or requested (no bode plot mode)
0x810C	33036	INVALIDCONTEXT	Invalid context for this command (mandatory task or windows context needed)
0x810D	33037	NOAXISINTERFACE	**Missing TCom axis interface (axis null pointer).** There is no connection to the NC axis. Either no axis (or axis ID) has been parameterized, or the parameterized axis does not exist.
0x810E	33038	INPUTCYCLECOUNTER	**Invalid input cycle counter from IO drive (e.g. frozen).** The cyclic drive data are backed up by an ‘InputCycleCounter’ during the bode plot recording. This allows firstly the detection of an unexpected communication loss (keyword: LifeCounter) and secondly a check for temporal data consistency to be performed. **Example 1:** This error can occur if the cycle time of the calling task is larger than the assumed drive cycle time (in this case, however, the error occurs right at the start of the recording). **Example 2:** This error can occur if the calling task has real-time errors (e.g. the "Exceed Counter" of the task increments or the task has a lower priority, as is often the case, for example, with the PLC). In this case the error can also occur at any time during the recording. **Example 3:** This error can occur more frequently if the real-time load on the computer is quite high (>50 %). **Refer also to the corresponding AX5000 drive error code F440.**
0x810F	33039	POSITION MONITORING (=> NC Runtime Error)	**Position monitoring: Axis position is outside of the maximum allowed moving range.** The axis has left the parameterized position range window, whereupon the recording was aborted and the NC axis was placed in the error state 0x810F (with standard NC error handling). The position range window acts symmetrically around the start position of the axis (see also parameter description *Position Monitoring Window*).

Code Hex	Code Dec	Symbol	Description
			Typical error message in the logger: *"BodePlot: 'Position Monitoring' error 0x%x because the actual position %f is above the maximum limit %f of the allowed position range (StartPos=%f, Window=%f)"*
0x8110	33040	DRIVELIMITATIONDETECTED	**Driver limitations detected (current or velocity limitations) which causes a nonlinear behavior and invalid results of the bode plot.** A bode plot recording requires an approximately linear transmission link. If the velocity or current is limited in the drive unit, however, this non-linear behavior is detected and the bode plot recording is aborted. Reasons for these limitations can be: choosing too large an amplitude for the position, velocity or torque interface, or an unsuitable choice of amplitude scaling mode (see also parameter description *Amplitude Scaling Mode, Base Amplitude, Signal Amplitude*). Typical error message in the logger: *"BodePlot: Sequence aborted with error 0x%x because the current limit of the drive has been exceeded (%d times) which causes a nonlinear behavior and invalid results of the bode plot"*
0x8111	33041	LIFECOUNTERMONITORING (=> NC Runtime Error)	**Life counter monitoring (heartbeat): Lost of communication to GUI detected after watchdog timeout is elapsed.** The graphical user interface from which the bode plot recording was started is no longer communicating with the bode plot driver in the expected rhythm (keyword: 'Life Counter'). Therefore the recording is terminated immediately and the NC axes are placed in the error state 0x8111 (with standard NC error handling). Possible reasons for this can be an user interface crash or a major malfunction of the Windows context. Typical error message in the logger: *"BodePlot: Sequence aborted with GUI Life Counter error 0x%x because the WatchDog timeout of %f s elapsed ('%s')"*
0x8112	33042	NCERR_BODEPLOT_WCSTATE	**WC state error (IO data working counter)** IO working counter error (WC state), for example due to real-time errors, EtherCAT CRC errors or telegram failures, EtherCAT device not communicating (OP state), etc.
0x8113-0x811F	33043-33055	RESERVED	**Reserved range**

7.8.12 Further Error Codes

Table 13:

Error(Hex)	Error(Dec)	ErrorType	Description
0x8120	**33056**	Environment	**Invalid configuration for Object (e.g. in System Manager).**
0x8121	**33057**	Environment	**Invalid environment for Object (e.g. TcCom-Object's Hierarchy or missing/faulty Objects).**
0x8122	**33058**	Environment	**Incompatible Driver or Object.**

Error(Hex)	Error(Dec)	ErrorType	Description
0x8124	33060	Function Block	**Command execution does not terminate (e. g. MC_Reset does not signal DONE).**
0x8130	33072	Communication	**Invalid ObjectID of Communication Target.**
0x8131	33073	Communication	**Communication Target expects Call in different Context.**
0x8132	33074	Communication	**Invalid State of Communication Target.**
0x8134	33076	Communication	**Communication with Communication Target cannot be established.**
0x813b	33083	Parameter	**Transition Mode is invalid.**
0x813c	33084	Parameter	**BufferMode is invalid.**
0x813d	33085	Function Block	**Only one active Instance of Function Block per Group is allowed.**
0x813e	33086	State	**Command is not allowed in current group state.**
0x813f	33087	Function Block	**Slave cannot synchronize.** The slave cannot reach the SlaveSyncPosition with the given dynamics.
0x8140	33088	Parameter	**Invalid value for one or more of the dynamic parameters (Acceleration, Deceleration, Jerk).**
0x8141	33089	Parameter	**IdentInGroup is invalid.**
0x8142	33090	Parameter	**The number of axes in the group is incompatible with the axes convention.**
0x8143	33091	Communication	**Function Block or respective Command is not supported by Target.**
0x8144	33092	State	**Command queue full.** Command queue is completely filled up and cannot accept additional commands until some commands are fully processed.
0x8145	33093	Function Block	**Mapping of Cyclic Interface between NC and PLC is missing (e.g. AXIS_REF, AXES_GROUP_REF, ...).**
0x8146	33094	Function Block	**Invalid Velocity Value.** The velocity was not set or the entered value is invalid.
0x8147	33095	Parameter	**Invalid Coordinate Dimension.** The dimension of the set coordinate interpretation does not meet the requirements.
0x8148	33096	Function Block	**Invalid Input Value.**
0x8149	33097	Parameter	**Unsupported Dynamics for selected Group Kernel.**
0x814a	33098	Parameter	**The programmed position dimension incompatible with the axes convention.**
0x814b	33099	Function Block	**Path buffer is invalid. E.g. because provided buffer has invalid address or is not big enough.**
0x814c	33100	Function Block	**Path does not contain any element.**
0x814d	33101	Function Block	**Provided Path buffer is too small to store more Path Elements.**
0x814e	33102	Parameter	**Dimension or at least one Value of Transition Parameters is invalid.**
0x814f	33103	Function Block	**Invalid or Incomplete Input Array.**
0x8150	33104	Function Block	**Path length is zero.**
0x8151	33105	State	**Command is not allowed in current axis state.**
0x8152	33106	State	**TwinCAT System is shutting down and cannot complete request.**

Error(Hex)	Error(Dec)	ErrorType	Description
0x8153	**33107**	Parameter	**Configured axes convention and configured axes do not match.**
0x8154	**33108**	Initialization	**Invalid Number of ACS Axes.** The number of ACS input axes does not match the number of ACS input axes expected by the kinematic transformation.
0x8155	**33109**	Initialization	**Invalid Number of MCS Data.** The number of MCS input data does not match the number expected by the kinematic transformation.
0x8156	**33110**	Initialization	**Invalid Value Set for Kinematic Parameters.** The numeric value set for the parameter does not reside within the respective definition range.
0x8158	**33112**	NC Programmin g	**The Given ACS Values Cannot be Reached.** The given ACS values result in an invalid machine configuration.
0x8159	**33113**	NC Programmin g	**The Set Target Positions Cannot be Reached.** The set target positions reside outside the admissible working space.
0x815d	**33117**	NC Programmin g	**Discontinuity in ACS axes detected.**
0x8160	**33120**	NC Programmin g	**Circle Specification in Path is invalid.** The specification of a circle segment in the programmed interpolated path (e.g. via MC_MovePath) has an invalid or ambiguous description. Probably its center cannot be determined reliably.
0x8161	**33121**	NC Programmin g	**Maximum stream lines reached.** The maximum number of stream lines is limited. Please refer to function block documentation for details.
0x8163	**33123**	Function Block	**Invalid First Segment.** The corresponding element can only be analyzed with a well-defined start point.
0x8164	**33124**	Function Block	**Invalid auxiliary point.** The auxiliary point is not well-defined.
0x8166	**33126**	Function Block	**Invalid parameter for GapControlMode.** Invalid parameter for GapControlMode, most likely in combination with the group parameter GapControlDirection.
0x8167	**33127**	External	**Group got unsupported Axis Event (e.g. State Change).** Group got unsupported Axis Event (e.g. State Change e.g. triggered by a Single Axis Reset).
0x8168	**33128**	Parameter	**Unsupported Compensation Type.** The compensation type was either not set or is not supported by the addressed object.
0x8169	**33129**	Function Block	**Master axis does not exist or cannot be used.**
0x816a	**33130**	External	**Invalid or Missing Tracking Transformation.** This error occurs at MC_TrackConveyorBelt if at the CoordTransform input an invalid object ID is used or the object ID points to an object that is not supported as coordinate transformation.
0x816b	**33131**	Function Block	**Position is not on Track.** Either Track cannot be activated because Actual Position is not on Track, or Target Position is not on Active Track or TrackPart.
0x816c	**33132**	Function Block	**Axis does not have an activated track.**
0x816d	**33133**	NC Programmin g	**Invalid Compensation ObjectId.** An Object with this ObjectId does not exist or it is not of the right type (has to be a compensation).

Error(Hex)	Error(Dec)	ErrorType	Description
0x816e	33134	Monitoring	**Axis is in error because axis was not in Target when InTargetAlarm Timer expired.**
0x816f	33135	State	**Coupling would cause a cyclic dependency of axis (e.g. via MC_GearInPos).**
0x8170	33136	Function Block	**Axis was not added to an axes group, the command is not valid.**
0x817f	33151	State	**Drive has invalid State.**
0x8181	33153	Function Block	**Parameter for gap control are invalid with the current configuration.** Function block with gap control was issued to an axis that is not in a CA group.
0x8182	33154	Monitoring	**Software position limit violation.** Software position limits of at least one axis have been or would have been violated by a command.
0x8183	33155	NC Programmin g	**Target position is not reachable.** There is no path available to the target position or target position is unreachable in general.
0x8185	33157	NC Programmin g	**The mover or one of its relevant coordinates is busy.** Either the whole mover or at least of its coordinates relevant to the command are busy.
0x8186	33158	NC Programmin g	**A collision has occurred or would occur.** Either a collision has occurred or would occur if the command was executed.
0x8187	33159	NC Programmin g	**Invalid Track Specification.** The geometric extension of this track is incompatible with the already existing geometry of this or the other tracks.
0x8188	33160	NC Programmin g	**Command not allowed in track state.**
0x8189	33161	Function Block	**Invalid Reference passed to Function Block.** An invalid reference (or pointer) was used in a function block call. This can happen if a reference type is used before it was initialized.
0x818a	33162	NC Programmin g	**Path is locked against modifications.** The path was locked to further changes. However, it might be resettable.
0x818c	33164	Parameter	**Position out of modulo range.** Position must be larger or equal to zero and less or equal to the modulo factor when using modulo positioning. When using modulo positioning, the target position is interpreted in consideration of the AdditionalTurns variable at the Options input.
0x818d	33165	Parameter	**The specified value AdditionalTurns at the Options input is not allowed.** The parameter AdditionalTurns must be zero for the specified value of the parameter Direction.
0x818e	33166	Function Block	**Master/Slave sync position incompatible with sync direction.** The given sync positions require the slave moving in a direction which is not allowed while in synchronization phase.
0x8f38 - 0x8f50	36664 - 36688	Internal	**Internal Error.**
0x8f56	36694	Internal	**Internal Error.**
0x8f59	36697	Internal	**Internal Error.**
0x8f5c - 0x8f62	36700 - 36706	Internal	**Internal Error.**
0x8f65	36709	Internal	**Internal Error.**

Error(Hex)	Error(Dec)	ErrorType	Description
0x8f68 - 0x8ffe	**36712 - 36862**	Internal	**Internal Error.**

More Information:
www.beckhoff.com/tx1260

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com